



CREATE

Canterbury Research and Theses Environment

Canterbury Christ Church University's repository of research outputs

<http://create.canterbury.ac.uk>

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g. Williams, J. J. (2018) An evaluation of the 'open source internet research tool': a user-centred and participatory design approach with UK law enforcement. Ph.D. thesis, Canterbury Christ Church University.

Contact: create.library@canterbury.ac.uk



**AN EVALUATION OF THE ‘OPEN SOURCE INTERNET
RESEARCH TOOL’: A USER-CENTRED AND
PARTICIPATORY DESIGN APPROACH WITH UK LAW
ENFORCEMENT**

By

Joseph James Williams

Canterbury Christ Church University

Thesis submitted for the degree of Doctor of Philosophy

2018

ABSTRACT

As part of their routine investigations, law enforcement conducts open source research; that is, investigating and researching using publicly available information online. Historically, the notion of collecting open sources of information is as ingrained as the concept of intelligence itself. However, utilising open source research in UK law enforcement is a relatively new concept not generally, or practically, considered until after the civil unrest seen in the UK's major cities in the summer of 2011.

While open source research focuses on the understanding of being 'publicly available', there are legal, ethical and procedural issues that law enforcement must consider. This asks the following main research question: *What constraints do law enforcement face when conducting open source research?* From a legal perspective, law enforcement officials must ensure their actions are necessary and proportionate, more so where an individual's privacy is concerned under human rights legislation and data protection laws such as the General Data Protection Regulation. Privacy issues appear, though, when considering the boom and usage of social media, where lines can be easily blurred as to what is public and private.

Guidance from Association of Chief Police Officers (ACPO) and, now, the National Police Chief's Council (NPCC) tends to be non-committal in tone, but nods towards obtaining legal authorisation under the Regulation of Investigatory Powers Act (RIPA) 2000 when conducting what may be 'directed surveillance'. RIPA, however, pre-dates the modern era of social media by several years, so its applicability as the de-facto piece of legislation for conducting higher levels of open source research is called into question. 22 semi-structured interviews with law enforcement officials were conducted and discovered a grey area surrounding legal authorities when conducting open source research.

From a technical and procedural aspect of conducting open source research, officers used a variety of software tools that would vary both in price and quality, with no standard toolset. This was evidenced from 20 questionnaire responses from 12 police forces within the UK. In an attempt to bring about standardisation, the College of Policing's Research, Identifying and Tracing the Electronic Suspect (RITES) course recommended several

capturing and productivity tools. Trainers on the RITES course, however, soon discovered the cognitive overload this had on the cohort, who would often spend more time learning to use the tools than learn about open source research techniques.

The problem highlighted above prompted the creation of Open Source Internet Research Tool (OSIRT); an all-in-one browser for conducting open source research. OSIRT's creation followed the user-centred design (UCD) method, with two phases of development using the software engineering methodologies 'throwaway prototyping', for the prototype version, and 'incremental and iterative development' for the release version.

OSIRT has since been integrated into the RITES course, which trains over 100 officers a year, and provides a feedback outlet for OSIRT. System Usability Scale questionnaires administered on RITES courses have shown OSIRT to be usable, with feedback being positive. Beyond the RITES course, surveys, interviews and observations also show OSIRT makes an impact on everyday policing and has reduced the burden officers faced when conducting opens source research.

OSIRT's impact now reaches beyond the UK and sees usage across the globe. OSIRT contributes to law enforcement output in countries such as the USA, Canada, Australia and even Israel, demonstrating OSIRT's usefulness and necessity are not only applicable to UK law enforcement.

This thesis makes several contributions both academically and from a practical perspective to law enforcement. The main contributions are:

- Discussion and analysis of the constraints law enforcement within the UK face when conducting open source research from a legal, ethical and procedural perspective.
- Discussion, analysis and reflective discourse surrounding the development of a software tool for law enforcement and the challenges faced in what is a unique development.
- An approach to collaborating with those who are in 'closed' environments, such as law enforcement, to create bespoke software. Additionally, this approach offers a method of measuring the value and usefulness of OSIRT with UK law enforcement.

- The creation and integration of OSIRT in to law enforcement and law enforcement training packages.

ACKNOWLEDGEMENTS

This thesis has been a substantial undertaking, and I would like to gratefully thank...

My supervisor, Dr Paul Stephens, who has been an invaluable source of positivity and creativity. Even during turbulent times, you were always capable of seeing the bright side and the bigger picture.

My wife, Jennifer, whose immeasurable lexicon and grasp of the English language provided me a great source of inspiration while writing. Thank you so much for tolerating me sitting in a room for months on end, only to be asked to proof read what came out at the end of it. Your love and support has been crucial in the completion of this thesis.

My mum, Anne, who is proud of me whatever I do or achieve.

My friend, colleague, and writing buddy Georgina who always remained positive when there seemed to be little left to be positive about... Plus an instinctive knowledge of the appropriate SmartArt to use always helps.

My colleagues David, Gerald, Abhaya and Robin who have provided unwavering support and an ear when things weren't quite going to plan.

Russell, lead high-tech crime trainer at the College of Policing, for his continued support. Without Russell, OSIRT would never exist.

Tim, who has been supportive of OSIRT from the very beginning and continues to champion OSIRT to this day.

Dr Cunningham and Dr Crellin for their insight, advice and putting the icing on the cake.

Finally, thank you to all those who use OSIRT, or have participated in the studies within this thesis, it is incredibly humbling and I am extremely grateful for all your support.

CONTENTS

1 INTRODUCTION.....	1
1.1 RESEARCH PROBLEM	1
1.2 RESEARCH QUESTIONS	4
1.2.1 <i>What constraints do law enforcement in the UK face when conducting open source research?</i>	4
1.2.2 <i>What do law enforcement need from a software tool when conducting opens source research?</i>	5
1.2.3 <i>What are the unique elements and challenges when engineering a software solution for law enforcement?</i>	5
1.2.4 <i>How can developers involve users in the design process in a ‘closed’ environment?</i>	6
1.2.5 <i>How can law enforcement be effectively trained to conduct open source research?</i>	6
1.3 RESEARCH METHODOLOGY OUTLINE	6
1.3.1 <i>Prototype stage</i>	6
1.3.2 <i>Release stage</i>	7
1.4 CONTRIBUTION TO KNOWLEDGE.....	7
1.5 CONTRIBUTIONS MADE BY THIS THESIS	8
1.5.1 <i>Peer-reviewed articles</i>	8
1.5.2 <i>Workshops</i>	9
1.5.3 <i>Posters</i>	9
1.6 THESIS STRUCTURE	9
2 A REVIEW OF PROCEDURAL, LEGAL AND ETHICAL ISSUES SURROUNDING OPEN SOURCE RESEARCH	11
2.1 OPEN SOURCES AND THEIR CONTRIBUTION	11
2.1.1 <i>The rise and rise of the Internet</i>	14
2.1.2 <i>Guidance and laws surrounding open source research</i>	15
2.1.3 <i>Open source research and Social Media</i>	18
2.1.4 <i>#thinkdigital</i>	21
2.1.5 <i>Levels of open source</i>	22

2.1.6 ISO/IEC 17025:2005	24
2.1.7 <i>College of Policing and the Researching, Identifying and Tracing the Electronic Suspect course</i>	25
2.2 SUMMARY OF LEGAL AND ETHICAL ISSUES	26
2.3 TOOLS AND PRACTICES SURROUNDING THE CAPTURE OF OPEN SOURCES IN UK LAW ENFORCEMENT	26
2.3.1 <i>Background to software usage by LEOs when conducting open source research</i>	27
2.3.2 <i>Current toolset</i>	28
2.4 TOOL SUMMARY	29
2.5 REQUIREMENTS FOR A BESPOKE OPEN SOURCE RESEARCH TOOL	30
2.6 OVERVIEW OF OSINT-STYLE BROWSERS	31
2.6.1 <i>Oryon OSINT Browser</i>	31
2.6.2 <i>Forensic Acquisition of Websites</i>	31
2.6.3 <i>Hunchly</i>	33
2.7 HUMAN ASPECTS AND CONSIDERATIONS WHEN SOFTWARE ENGINEERING	34
2.7.1 <i>Digital crime, policing and surrounding issues for officers</i>	34
2.8 CHAPTER SUMMARY	39
3 METHODOLOGY	40
3.1 RESEARCH DESIGN	40
3.1.1 <i>User-centred design</i>	40
3.1.2 <i>Methods overview</i>	42
3.2 OVERVIEW AND DISCUSSION OF SOFTWARE ENGINEERING METHODOLOGIES	43
3.3 PROTOTYPING METHOD	45
3.3.1 <i>Types of software prototyping</i>	46
3.3.2 <i>Prototype methodology summary</i>	47
3.4 RELEASE VERSION SOFTWARE ENGINEERING METHODOLOGY	48
3.4.1 <i>Iterative and incremental approach</i>	48
3.4.2 <i>Distribution and beta testing</i>	49
3.4.3 <i>Managing as a lone developer</i>	50
3.5 SAMPLE AND SAMPLING	51
3.5.2 <i>UTES course access and limitations</i>	53
3.5.3 <i>Data protection, confidentiality and consent</i>	53
3.6 DATA COLLECTION	54

3.6.1 <i>Data sources, methods and rationale</i>	54
3.6.2 <i>Measuring usability</i>	59
3.7 DATA ANALYSIS	61
3.7.1 <i>Qualitative data analysis</i>	61
3.7.2 <i>Quantitative analysis</i>	64
3.8 VALIDITY AND RELIABILITY	65
3.8.1 <i>Triangulation</i>	65
3.9 CHAPTER SUMMARY	66
4 A STUDY OF LEGAL, ETHICAL AND PROCEDURAL ISSUES FACED BY LAW ENFORCEMENT.....	67
4.1 INTERVIEW QUESTIONS	67
4.2 PARTICIPANTS	68
4.3 INTERVIEW RESULTS AND DISCUSSION SURROUNDING LEGAL AND ETHICAL ISSUES.....	68
4.3.1 <i>The ‘grey area’</i>	68
4.3.2 <i>Playing catch up</i>	70
4.3.3 <i>Ethics</i>	71
4.3.4 <i>Social Media companies</i>	72
4.4 LEGAL AND ETHICAL ISSUES SUMMARY	73
4.5 CURRENT TOOLS AND PRACTICES AND QUESTIONNAIRE	74
4.5.1 <i>Respondents</i>	74
4.5.2 <i>Results</i>	75
4.5.3 <i>Questionnaire summary</i>	76
4.6 CHAPTER SUMMARY	76
5 OSIRT PROTOTYPE DEVELOPMENT AND IMPLEMENTATION	77
5.1 JUSTIFICATION, DISCUSSION AND ISSUES SURROUNDING PROTOTYPE IMPLEMENTATION	77
5.1.1 <i>Focus of control</i>	77
5.1.2 <i>Artefact capture – to note or not to note</i>	78
5.1.3 <i>Date and time</i>	79
5.2 OSIRT PROTOTYPE CREATION AND IMPLEMENTATION	81
5.2.1 <i>Early design</i>	81
5.2.2 <i>Case management</i>	82
5.2.3 <i>Main browser</i>	84

5.2.4 Artefact management	88
5.2.5 Context menu handling	89
5.2.6 Download management.....	91
5.2.7 Logging websites.....	93
5.2.8 Static screen capturing	95
5.2.9 Image previewer.....	96
5.2.10 Video screen capturing	97
5.2.11 Attachments.....	98
5.2.12 Case notes	98
5.2.13 Image scraping.....	99
5.2.14 Audit log.....	101
5.2.15 Reporting.....	102
5.3 PROTOTYPE DEVELOPMENT SUMMARY	104
6 OSIRT PROTOTYPE RESULTS AND DISCUSSION	105
6.1 PARTICIPANTS	105
6.2 OBSERVATION RESULTS AND DISCUSSION	106
6.2.1 Issues surrounding OSIRT design.....	106
6.2.2 Issue surrounding technology choices	110
6.2.3 Positive feedback	111
6.3 SUS QUESTIONNAIRE RESULTS AND USER COMMENTS	112
6.3.1 SUS.....	112
6.3.2 User comments.....	113
6.3.3 SUS Results and feedback summary	114
6.4 PROTOTYPE INTERVIEWS	115
6.4.1 Interview guide.....	115
6.4.2 The tool ‘mish-mash’	116
6.4.3 The ‘dream’ tool	118
6.4.4 OSIRT’s integration.....	120
6.4.5 Useful features	121
6.4.6 Improvements and additions	123
6.5 SUMMARY OF CHANGES REQUIRED.....	124
6.6 CHAPTER SUMMARY	125
7 OSIRT RELEASE DEVELOPMENT AND IMPLEMENTATION	127
7.1 MAIN BROWSER.....	127

7.1.1 <i>GeckoFX</i>	128
7.1.2 <i>Awesomium</i>	128
7.1.3 <i>CefSharp</i>	128
7.1.4 <i>Browser control summary</i>	129
7.2 DOCUMENT OBJECT MODEL AND JAVASCRIPT.....	129
7.3 TABBED BROWSING	130
7.3.1 <i>Tab style</i>	130
7.3.2 <i>Tab implementation</i>	131
7.4 CONTEXT MENU HANDLING	133
7.5 CASE MANAGEMENT.....	134
7.5.1 <i>Case container</i>	134
7.5.2 <i>Complexities and considerations of using a custom file format</i>	135
7.6 ERROR HANDLING AND RECOVERY	136
7.6.2 <i>Manual case recovery</i>	138
7.7 STATIC SCREEN CAPTURING.....	138
7.7.1 <i>Full page screen capture</i>	138
7.7.2 <i>Timed Capture</i>	145
7.7.3 <i>Snippet and current view</i>	145
7.8 VIDEO CAPTURE.....	145
7.9 DOWNLOAD MANAGEMENT	146
7.10 PREVIEWERS.....	147
7.10.1 <i>Image previewer</i>	149
7.10.2 <i>Video Previewer</i>	150
7.10.3 <i>Text Previewer</i>	151
7.11 ATTACHMENTS	152
7.12 ACCESSING THE DARK WEB	153
7.13 LINK EXTRACTION	156
7.14 SOCIAL MEDIA ID EXTRACTION	156
7.15 ADVANCED BROWSER OPTIONS	157
7.15.1 <i>User agent spoofing</i>	157
7.15.2 <i>General browser settings</i>	158
7.16 WEBPAGE FILTERING.....	158
7.16.1 <i>Filtering and capturing responses</i>	159

7.17 AUTO-SCROLLING	161
7.18 AUDIT LOG	162
7.18.1 Artefact gridview.....	162
7.18.2 File previewer	163
7.18.3 Searching audit log	164
7.19 REPORTING.....	164
7.20 BOOKMARKING	166
7.21 PREVENTING MULTIPLE OSIRT INSTANCES.....	167
7.22 IMPLEMENTATION REFLECTION	168
7.22.1 WinForms.....	168
7.23 CHAPTER SUMMARY	168
8 RESULTS AND DISCUSSION OF OSIRT’S INTEGRATION, IMPACT AND CONTRIBUTION TO LAW ENFORCEMENT: PART ONE	170
8.1 COGNITIVE WALKTHROUGH OF OSIRT	170
8.1.1 Method	171
8.1.2 Cognitive walkthrough results	171
8.1.3 Recommended changes to OSIRT	175
8.1.4 Discussion of cognitive walkthrough	176
8.1.5 Summary of cognitive walkthrough.....	178
8.2 OSIRT SYSTEM USABILITY SCALE RESULTS.....	178
8.2.1 Discussion of mean SUS scores	180
8.2.2 SUS question score breakdown.....	181
8.2.3 Net Promoter Score.....	183
8.2.4 Additional comments.....	184
8.3 OBSERVATIONS	185
8.3.1 OSIRT’s design and look-and-feel	186
8.3.2 Bugs and user experience enhancements.....	186
8.3.3 Feature requests.....	187
8.3.4 Observation summary	188
8.4 OSIRT INTERVIEWS AND QUESTIONNAIRES	189
8.4.1 OSIRT’s usefulness during course	189
8.4.2 Recommending OSIRT to a colleague – Net Promoter Score	189
8.4.3 OSIRT integration into workflow.....	189
8.4.4 Automated logging and reporting	191

8.4.5 <i>Screen capturing</i>	192
8.5 ALTERNATIVE METHODS FOR USABILITY EVALUATIONS.....	193
8.5.1 <i>Qualitative usability measures</i>	193
8.5.2 <i>Quantitative usability measures</i>	194
8.6 CHAPTER SUMMARY	195
9 RESULTS AND DISCUSSION OF OSIRT’S INTEGRATION, IMPACT AND CONTRIBUTION TO LAW ENFORCEMENT: PART TWO	196
9.1 OSIRT’S INTEGRATION INTO THE RITES COURSE.....	196
9.1.1 <i>About the interviewee</i>	197
9.1.2 <i>The course before OSIRT</i>	197
9.1.3 <i>The course with OSIRT</i>	198
9.2 OSIRT’S INTEGRATION INTO PRIVATE OSINT TRAINING PACKAGES	199
9.3 OSIRT IN COMMERCIAL PRODUCTS.....	199
9.4 OSIRT USAGE QUESTIONNAIRE.....	200
9.4.1 <i>Demographic</i>	200
9.4.2 <i>OSIRT usage</i>	202
9.4.3 <i>In-house training packages</i>	204
9.4.4 <i>OSIRT discovery</i>	206
9.4.5 <i>Rate usefulness of OSIRT</i>	206
9.4.6 <i>Has OSIRT enhanced your capability at conducting open source research</i>	207
9.4.7 <i>Previous tool usage</i>	210
9.4.8 <i>Does OSIRT capture all relevant data for your open source investigation?</i>	211
9.4.9 <i>Recommend OSIRT</i>	211
9.4.10 <i>Tools usage within OSIRT</i>	212
9.4.11 <i>Does OSIRT being open source software impact the decision to use OSIRT?</i>	214
9.5 OSIRT AS FREE/LIBRE OPEN SOURCE SOFTWARE (FLOSS).....	214
9.5.1 <i>FLOSS integration into UK public services</i>	215
9.6 METHOD.....	216
9.6.1 <i>Interviews</i>	216
9.7 INTERVIEW RESULTS AND DISCUSSION	216
9.7.1 <i>Trust and security</i>	216

9.7.2 Maintenance.....	217
9.7.3 Technical Support	217
9.7.4 Cost	218
9.7.5 Training.....	219
9.7.6 Summary of interviews.....	219
9.8 SUMMARY	220
10 TRAINING OF LAW ENFORCEMENT OFFICIALS TO CONDUCT OPEN SOURCE RESEARCH WITH OSIRT	221
10.1 BACKGROUND	221
10.1.1 Designing Training Courses for Law Enforcement and Applying Learning Styles	221
10.1.2 Design of the RITES Course	222
10.1.3 Using Software for Investigative Work	223
10.1.4 Using Kirkpatrick's Training Evaluation Model.....	224
10.2 METHODOLOGY.....	224
10.3 RESULTS AND DISCUSSION	226
10.3.1 Pre-Course Questionnaire Results	226
10.3.2 Daily evaluations and observer comments	227
10.3.3 Participant course evaluation.....	230
10.4 DISCUSSION.....	234
10.5 LIMITATIONS AND FUTURE RESEARCH	235
10.6 CHAPTER SUMMARY	235
11 CONCLUSIONS AND FUTURE WORK	237
11.1 GOALS, FINDINGS AND ADDRESSING THE RESEARCH QUESTIONS	237
11.1.1 What constraints do law enforcement in the UK face when conducting open source research?.....	238
11.1.2 What do law enforcement need from a software tool when conducting opens source research?.....	239
11.1.3 What are the unique elements when engineering a software solution for law enforcement?.....	240
11.1.4 How can developers involve users in the design process in a 'closed' environment?.....	240
11.1.5 How can law enforcement be effectively trained to conduct open source research?.....	241

11.2 CRITICAL REVIEW OF THESIS, LIMITATIONS AND REFLECTION	241
11.2.1 <i>Sample</i>	241
11.2.2 <i>User-centred design</i>	242
11.2.3 <i>Prototype software engineering methodology</i>	242
11.2.4 <i>WinForms</i>	243
11.2.5 <i>FLOSS and licensing</i>	243
11.3 FUTURE WORK.....	244
11.3.1 <i>Further usability testing</i>	244
11.3.2 <i>A general framework for collaborative software engineering</i>	244
11.3.3 <i>ISO 17025</i>	244
11.3.4 <i>OSIRT</i>	245
11.4 CONCLUDING REMARKS	246
REFERENCES.....	247
APPENDICES	261

LIST OF TABLES

TABLE 3.1 UCD PRINCIPLE AND HOW IT WAS APPLIED WITHIN THIS THESIS	42
TABLE 3.2 UCD METHODS OF COLLECTION AND THEIR STAGE OF COLLECTION	43
TABLE 3.3 TYPES OF SAMPLES AND THEIR DESCRIPTION	52
TABLE 3.4 APPROACHES TO OBSERVATION (RUNESON <i>ET AL.</i> , 2012)	55
TABLE 3.5 EXAMPLE "OBSERVATIONS, QUOTES AND INFERENCES" SHEET.	56
TABLE 3.6 DATA TRIANGULATION METHODS AND THEIR APPLICATION.....	65
TABLE 5.1 A LIST OF AVAILABLE ACTIONS, AND THEIR DESCRIPTIONS, IN OSIRT	89
TABLE 6.1 PARTICIPANT DETAILS. DC – DETECTIVE CONSTABLE. DS – DETECTIVE SERGEANT.....	106
TABLE 6.2 BREAKDOWN OF SUS SCORES	113
TABLE 6.3 PARTICIPANT DETAILS	115
TABLE 6.4 INTERVIEW GUIDE FOR THE OSIRT PROTOTYPE.....	116
TABLE 7.1 PROPERTIES WITHIN RESOURCEWRAPPER.....	161
TABLE 7.2 TABS WITHIN THE AUDIT LOG	163
TABLE 8.1 ISSUES DISCOVERED IN COGNITIVE WALKTHROUGH, WITH DEVELOPER COMMENTS.....	176
TABLE 8.2 ADJECTIVE AND GRADE RANKINGS FOR OSIRT (OVERALL MEAN SUS SCORE EXCLUDING PROTOTYPE).....	179
TABLE 8.3 RAW MEAN SUS SCORES	180
TABLE 8.4 NPS SCORES BASED ON SUS SCORES	184
TABLE 8.5 FEATURE REQUESTS FROM OBSERVATIONS	188

TABLE 8.6 NPS SCORE	189
TABLE 8.7 RESULTS FOR THE QUESTION "OSIRT WILL SAVE ME TIME IN COMPARISON TO HOW I CONDUCT OPEN SOURCE RESEARCH NOW"	191
TABLE 9.1 JOBS ROLES AND HOURS USING OSIRT	204
TABLE 9.2 VERBATIM RESPONSES FOR "HAS OSIRT ENHANCED YOUR CAPABILITY AT CONDUCTING OPEN SOURCE RESEARCH?"	208
TABLE 9.3 USED TOOLS BREAKDOWN	210
TABLE 9.4 RAW RESPONSES TO QUESTION	211
TABLE 9.5 INDIVIDUAL TOOL USAGE WITHIN OSIRT (TOTAL USAGE AND TOTAL USAGE AS A PERCENTAGE).....	213

LIST OF FIGURES

FIGURE 2.1 MODEL REPRESENTING THE DIFFERENCE BETWEEN OSINF (WITH SOME EXAMPLES PROVIDED), OPEN SOURCE RESEARCH AND OSINT	13
FIGURE 2.2 JAPAN TEST FOR HANDLING PERSONAL INFORMATION	16
FIGURE 2.3 PROTECTION PRINCIPLES OF THE DATA PROTECTION ACT (2018) FOR LAW ENFORCEMENT PURPOSES	17
FIGURE 2.4 ADVERTISEMENT BY FACEBOOK AFTER THE CAMBRIDGE ANALYTICA SCANDAL	20
FIGURE 2.5 TREE MAP AND HEAT MAP OF DII CAPABILITIES (SCRIVEN AND HERDALE, 2015)	22
FIGURE 2.6 LEVELS OF OPEN SOURCE RESEARCH AND THEIR REQUIREMENT TO OBTAIN AUTHORITY UNDER RIP A.....	23
FIGURE 2.7 METHOD FOR CAPTURING OPEN SOURCE AS TRAINED BY THE COLLEGE OF POLICING IN 2014/2015	26
FIGURE 2.8 TYPICAL WORKFLOW FOR A LEO CONDUCTING OPEN SOURCE RESEARCH.....	27
FIGURE 2.9 FREE VERSION OF FAW BROWSER	32
FIGURE 2.10 FAW OBJECTS DIRECTORY	32
FIGURE 2.11 A MODEL OF THE ATTRIBUTES IF SYSTEM ACCEPTABILITY (NIELSEN, 1993)	36
FIGURE 3.1 WATERFALL MODEL ADAPTED FROM ROYCE (1970)	44
FIGURE 3.2 SPIRAL MODEL ADAPTED FROM BOEHM (1987)	45
FIGURE 3.3 SOFTWARE PROTOTYPING METHODOLOGY (PRESSMAN, 2014)	46
FIGURE 3.4 TYPES OF SOFTWARE PROTOTYPING	47

FIGURE 3.5 VISUALISATION OF HOW SOFTWARE GROWS USING AN INCREMENTAL AND ITERATIVE APPROACH	49
FIGURE 3.6 KANBAN BOARD SPREADSHEET WITH OSIRT-RELATED TASKS (ASPECTS REDACTED FOR CONFIDENTIALITY). KANBAN BOARD FROM VERTEX42.COM (2014).	51
FIGURE 3.7 DATA COLLECTION TECHNIQUES FOR SOFTWARE ENGINEERING, WITH EXAMPLES OF HOW TO COLLECT DATA (BASED ON LETHBRIDGE <i>ET AL.</i> (2005))	54
FIGURE 3.8 STEPS OF DATA ANALYSIS (ROBSON, 2002)	62
FIGURE 3.9 REPRESENTATION OF APPROACHES OF QUALITATIVE DATA ANALYSIS VISUALISED BY TABER (2013).....	63
FIGURE 3.10 STAGES OF THEMATIC ANALYSIS AS ADAPTED FROM BRAUN AND CLARKE (2006).....	64
FIGURE 4.1 TOOLS USED BY LAW ENFORCEMENT TO CONDUCT OPEN SOURCE RESEARCH	75
FIGURE 5.1 HIGH LEVEL VIEW OF THE SYSTEM WITH SUB-FUNCTIONALITY	81
FIGURE 5.2 CASE CONTAINER DIRECTORY STRUCTURE	82
FIGURE 5.3 ENTITY RELATIONSHIP DIAGRAM FOR CASE DATABASE	83
FIGURE 5.4 WIREFRAME FOR CASE CREATION.....	84
FIGURE 5.5 WIREFRAME OF MAIN BROWSER	86
FIGURE 5.6 THE MAIN OSIRT BROWSER.....	87
FIGURE 5.7 DEFAULT CONTEXT MENU AND ITS PLETHORA OF OPTIONS (LEFT) AND CUSTOM CONTEXT MENU (RIGHT).....	91
FIGURE 5.8 SIMPLIFIED CLASS DIAGRAM FOR OSIRT'S DOWNLOAD MANAGEMENT	93
FIGURE 5.9 ACTIVITY DIAGRAM FOR WEBPAGE LOGGING.....	94

FIGURE 5.10 SNIPPET TOOL	95
FIGURE 5.11 FULL DATE AND TIME OF A FACEBOOK POST WHEN CURSOR HOVERS OVER THE TIME.....	96
FIGURE 5.12 IMAGE PREVIEWER	97
FIGURE 5.13 ATTACHMENT AND NOTE TO BE ADDED	98
FIGURE 5.14 EXAMPLE CASE NOTES	99
FIGURE 5.15 WIREFRAME DESIGN OF AUDIT LOG	101
FIGURE 5.16 AUDIT LOG	101
FIGURE 5.17 REPORT OPTIONS	103
FIGURE 5.18 EXPORTED REPORTED AS HTML	104
FIGURE 6.1 VIDEO CAPTURE WITH DEFAULT UI	107
FIGURE 6.2 REPORT EXPORTING OPTIONS.....	108
FIGURE 6.3 ATTEMPTING TO LOAD AN OSIRT CASE DIRECTORY	109
FIGURE 6.4 SUS SCORES FOR OSIRT PROTOTYPE.....	112
FIGURE 7.1 EXAMPLE OF DOM FOR A TABLE IN HTML (LE HÉGARET, WOOD AND ROBIE, 2004)	130
FIGURE 7.2 OSIRT WITH TABBED BROWSING	131
FIGURE 7.3 OPENING A LINK IN A NEW TAB VIA THE CONTEXT MENU	132
FIGURE 7.4 CONTEXT MENU WHEN AN IMAGE IS RIGHT-CLICKED.....	134
FIGURE 7.5 CASE OPENING WITH CUSTOM FILE FORMAT	134
FIGURE 7.6 DIRECTORY STRUCTURE FOR THE CASE CONTAINER, THIS IS WRAPPED IN AN .OSR FILE WHEN NOT OPEN WITHIN OSIRT.....	135
FIGURE 7.7 ACTIVITY DIAGRAM FOR CASE RECOVERY	136

FIGURE 7.8 FATAL ERROR HANDLER	138
FIGURE 7.9 CASE RECOVERY PANEL.....	138
FIGURE 7.10 A REPRESENTATION OF AN ARBITRARY SIZED DOCUMENT.	139
FIGURE 7.11 DISK CACHE WITH PARTIAL SCREENSHOTS, AND TEMP.PNG AS THE COMPLETELY STITCHED IMAGE	142
FIGURE 7.12 NAVIGATION BAR WITH CSS PROPERTY POSITION:FIXED.....	143
FIGURE 7.13 SCREENSHOT WITH 150% DPI SCALING (LEFT) AND SCREENSHOT WITH 100% DPI SCALING (RIGHT). NOTE SCREENSHOT ON LEFT IS MISSING ELEMENTS.	144
FIGURE 7.14 TIMED SCREENSHOT DIALOG AND COUNTDOWN TIMER IS STATUS BAR	145
FIGURE 7.15 GENERAL PREVIEWER WIREFRAME	148
FIGURE 7.16 UML CLASS DIAGRAM REPRESENTING THE PREVIEWERS	148
FIGURE 7.17 IMAGE PREVIEWER	149
FIGURE 7.18 LARGE SCREENSHOT IN THE IMAGE PREVIEWER THAT HAS BEEN RESIZED.	150
FIGURE 7.19 VIDEO PREVIEWER	151
FIGURE 7.20 TEXT PREVIEWER WITH SOURCE CODE FROM A WEB PAGE.....	152
FIGURE 7.21 EXAMPLE ATTACHMENT TO CASE	152
FIGURE 7.22 A SUCCESSFULLY ATTACHED FILE	153
FIGURE 7.23 OSIRT IN TOR MODE, AS DENOTED BY THE PURPLE ADDRESS BAR.	155
FIGURE 7.24 STARTING TOR IN THE ADVANCED BROWSER OPTIONS.....	155
FIGURE 7.25 FACEBOOK ID FINDER	156
FIGURE 7.26 REQUEST/RESPONSE CYCLE BETWEEN A CLIENT AND SERVER.....	159
FIGURE 7.27 AUDIT LOG WITH EXAMPLE SCREENSHOT (BOTTOM LEFT)	164

FIGURE 7.28 EXAMPLE OF SEARCHING THE AUDIT LOG	164
FIGURE 7.29 REPORT EXPORTING.....	165
FIGURE 7.30 REPORT FRONT PAGE	166
FIGURE 7.31 EXAMPLE REPORT PAGE.....	166
FIGURE 7.32 BOOKMARK MENU	167
FIGURE 7.33 BOOKMARK MANAGER	167
FIGURE 8.1 MEAN OVERALL SUS SCORE FROM ALL SUS RESULTS (EXCLUDING PROTOTYPE) WITH CONFIDENCE INTERVAL.....	179
FIGURE 8.2 MEAN SUS SCORES WITH BENCHMARK SCORE (68) IN RED	180
FIGURE 8.3 AVERAGE SUS SCORE ACROSS EACH QUESTION FOR THE ALL RELEASE VERSIONS OF OSIRT.....	181
FIGURE 8.4 INDIVIDUAL MEAN SUS SCORES COMPARED TO LEARNABILITY SCORES. GREEN BAR REPRESENTS MEAN LEARNABILITY (82.8) AND THE BROWN BAR REPRESENTS MEAN SUS SCORE (68).	182
FIGURE 9.1 OSIRT TWEETED BY TODDINGTON INTERNATIONAL INC.....	199
FIGURE 9.2 JOB ROLE OF RESPONDENTS	201
FIGURE 9.3 TIME IN SERVICE ROUNDED UP TO NEAREST YEAR	201
FIGURE 9.4 HOW LONG RESPONDENTS HAVE BEEN USING OSIRT	202
FIGURE 9.5 AVERAGE WEEKLY OSIRT USAGES IN HOURS	203
FIGURE 9.6 HOW OFFICERS WERE TRAINED TO USE OSIRT	205
FIGURE 9.7 HOW OFFICERS DISCOVERED OSIRT	206
FIGURE 9.8 USEFULNESS RATING OF OSIRT	207
FIGURE 9.9 DOES OSIRT BEING FLOSS SOFTWARE IMPACT DECISION TO USE IT	214

FIGURE 10.1 RITES COURSE TOPICS BROKEN DOWN PER DAY	223
FIGURE 10.2 ROOM LAYOUT AT THE RITES COURSE	223
FIGURE 10.3 OVERALL DIFFICULTY OF THE COURSE	228
FIGURE 10.4 COHORTS' RATING OF OWN COMPUTER LITERACY	228
FIGURE 10.5 PERCEPTION OF DAILY SESSION PACE	228
FIGURE 10.6 OSIRT'S EFFECTIVENESS	229
FIGURE 10.7 IMPACTS FELT DUE TO APPLICATION OF LEARNING	234

LISTINGS

LISTING 5.1 META TAG REQUIRED TO GET PAGE TO RENDER USING MODERN WEB STANDARDS.....	87
LISTING 5.2 CREATING AND SETTING THE CUSTOM CONTEXT MENU	90
LISTING 5.3 CREATING AND SETTING THE CUSTOM CONTEXT MENU	90
LISTING 5.4 UNINTELLIGENTLY HANDLING FILE DOWNLOADS	92
LISTING 5.5 HANDLING MULTIPLE DOCUMENTCOMPLETED EVENTS.....	95
LISTING 5.6 SAVING IMAGES BY TRAVERSING THE IHTMLDOCUMENT2.....	100
LISTING 5.7 SAVING IMAGES BY PARSING THE DOCUMENT AND OBTAINING THE SRC.....	100
LISTING 7.1 IMPLEMENTATION OF ILIFESPANHANDLER AND ASSOCIATED CODE FOR OPENING A WINDOW IN A NEW TAB.....	132
LISTING 7.2 CREATING AND ADDING A TAB TO THE TAB CONTROL.....	132
LISTING 7.3 POPULATING THE CONTEXT MENU	133
LISTING 7.4 FATAL HANDLER IMPLEMENTATION.....	137
LISTING 7.5 PSEUDOCODE FOR SCROLLING SCREEN CAPTURE	140
LISTING 7.6 OBTAINING THE DOCUMENT'S HEIGHT USING JAVASCRIPT	141
LISTING 7.7 TRAVERSING THE DOM FOR POSITION:FIXED ELEMENTS	143
LISTING 7.8 DOWNLOAD HANDLER IMPLEMENTATION	147
LISTING 7.9 STARTING TOR PROCESS	154
LISTING 7.10 LINKS TOR PROCESS WITH TOR.NET	154
LISTING 7.11 EXTRACTING LINKS FOR THE PAGE USING HTMLAGILITYPACK	156
LISTING 7.12 FILTERING RESPONSES	160

LISTING 7.13 AUTO-SCROLLING A WEB PAGE USING JAVASCRIPT	162
---	-----

ABBREVIATIONS AND ACRONYMS

Acronym	Meaning
ACPO	Association of Chief Police Officers
API	Application Programming Interface
C#	C Sharp
CEF	Chromium Embedded Framework
CHIS	Covert Human Intelligence Source
CSS	Cascading Stylesheet
CSV	Comma Separated Value
DC	Detective Constable
DII	Digital Investigation and Intelligence
DMI	Digital Media Investigator
DOM	Document Object Model
DPI	Dots Per Inch
DS	Detective Sergeant
DPA	Data Protection Act
DSA	Directed Surveillance Authority
ECHR	European Convention on Human Rights
EXIF	Exchangeable Image File Format
FLOSS	Free/Libre Open Source Software
GDI	Graphical Device Interface
GDPR	General Data Protection Regulation
GSCP	Government Secure Classification Policy
GUI	Graphical User Interface
HMIC	Her Majesty's Inspectorate of Constabulary

HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IE	Internet Explorer
JPG	Joint Photographic Experts Group
JS	JavaScript
LEA	Law Enforcement Agency
LEO	Law Enforcement Official
NPCC	National Police Chiefs Council
NPS	Net Promoter Score
NTP	Network Time Protocol
OSINF	Open Source Information
OSINT	Open Source Intelligence
OSIRT	Open Source Internet Research Tool
OSR	Open Source Research
PNG	Portable Network Graphic
RIPA	Regulation of Investigatory Powers Act
UTES	Research, Identifying and Tracing the Electronic Suspect
SOCMINT	Social Media Intelligence
SUS	System Usability Scale
Tor	The Onion Router
UCD	User-Centred Design
UI	User Interface
WinAPI	Windows Application Programming Interface
XML	Extensible Markup Language

LIST OF APPENDICES

APPENDIX A –SUS: OSIRT PROTOTYPE.....	262
APPENDIX B – OBSERVATIONS: OSIRT PROTOTYPE.....	274
APPENDIX C – SUS: OSIRT RELEASE	287
APPENDIX D – OBSERVATIONS: OSIRT RELEASE.....	304
APPENDIX E – COGNITIVE WALKTHROUGH	318
APPENDIX F – INTERVIEWS	348
APPENDIX G – OBSERVATION TEMPLATE FOR RITES COURSE (CHAPTER 10)	372
APPENDIX H – SAMPLE OF DAILY QUESTIONNAIRES FOR RITES COURSE	375
APPENDIX I – IDownloadManager IMPLEMENTATION	400

1 INTRODUCTION

1.1 Research problem

Overt, open sources of information have been a fixture of civilisation for centuries. From reading the news posted around the market place in ancient Rome, to broadcasts over the television; these are open, freely available sources of information. The usage and exploitation of these open sources reaches as far back in history as the notion of ‘intelligence’ itself (Schaurer and Störger, 2013). It was not until the Second World War, though, did we see the collection, analysis and synthesising of these open sources of information (Andrew, Aldrich and Wark, 2009), and this is what became to be known as Open Source Intelligence (OSINT).

Official definitions of OSINT often vary. For example, the Central Intelligence Agency (CIA) considers OSINT to be “drawn from publicly available material” (CIA, 2010), while the UK’s Ministry of Defence (MOD) considers OSINT to be “publicly available [...] but has limited public distribution or access” (MOD, 2011). The key, and critical, aspect from both definitions is that OSINT is publicly available.

“Publicly available” information may not sound as though it would generate practical usage, but the power of OSINT as part of intelligence packages is not to be underestimated. Alan Dulles, former Head of the CIA, cited as much as 80 percent of their peacetime intelligence being gathered from open sources as early as 1947 (cited in Wells and Gibson, 2017). Even today, estimates maintain this figure or surpass it (Dover, Goodman and Hillebrand, 2013). While these approximations tend to focus around military or centralised intelligence units, law enforcement, other public services and

private companies can, and do, tap into OSINT resources daily. OSINTs impact is significant, but at what cost to personal privacy?

With Internet usages rising yearly in the UK (Office for National Statistics, 2018b), more people are leaving a publicly available online digital footprint; a rich source of open, freely available information and artefacts that could be trivially tapped into by law enforcement. However, the use of open sources by UK law enforcement has not always been well utilised; in fact, we only have to go back to 2011, a summer plagued with public disorder throughout the major cities within the UK. The spread of the unrest was largely blamed on the use of social media. Her Majesty's Inspectorate of Constabulary even went so far as to say that social media was "not well understood" by law enforcement, and even "less well managed" (cited in Hobbs, Moran and Salisbury, 2014). It was not really until after the summer of 2011 that UK law enforcement made a collective effort to integrate and use open, online sources as a means for intelligence and evidence gathering (Hobbs, Moran and Salisbury, 2014).

It is unsurprising, then, that since the summer of 2011 conducting investigations online is becoming ever more frequent for law enforcement within the UK. Part of their toolset for conducting these investigations is "open source research". Open source research is defined in a similar fashion to other OSINT definitions, and in 2013 The Association of Police Chief Officers (ACPO) (now The National Police Chiefs Council (NPCC)¹), released this definition for open source research:

"The collection, evaluation and analysis of materials from sources available to the public, whether on payment or otherwise to use as intelligence or evidence within investigation." (ACPO, 2013)

Given the similarity between previous definitions of OSINT, why UK law enforcement chose "open source research" over "OSINT" is not known, but speculatively it may stem from the use of the word "intelligence" as OSINT is considered to be a complete end-product. When compared to "research", this implies no such level of authority. This thesis

¹ ACPO will still be used throughout this thesis, where appropriate, as these are the originating guidelines for both open source research and digital evidence gathering.

will focus on and use the term “open source research” going forward, and the definition provided by ACPO as its meaning.

Issues extend beyond policing guidelines, though. While ACPO guidelines for open source research state there is unlikely a need for legal authorities due to the nature of open sources, legal frameworks such as the Regulation of Investigatory Powers Act (2000) (RIPA) are used as a means, in some instances, to lawfully conduct open source research. Yet, RIPA pre-dates the modern era of social media by a number of years, and came out at a time when only a quarter of the UK even had Internet access (Office for National Statistics, 2018a). New legislation, such as The Investigatory Powers Act (2016) (IPA), still makes no mention of open sources of information or social media; making law enforcements’ jobs both a legal and ethical minefield. With the introduction of the General Data Protection Regulation (GDPR) and subsequent Data Protection Act 2018, the need to ensure a person’s private data is managed correctly must be more than a glancing consideration. Not just by those in law enforcement, but private individuals conducting open source investigations and due-diligence checks.

Given the relative newness of open source research and its integration into UK law enforcement, the question then becomes how law enforcement collect and manage artefacts while conducting open source research. From a practical standpoint, the answer is simply “it depends”; it depends upon force policy or even on an individual. However, ACPO guidelines for the collection of digital evidence state that an audit trail must be maintained for the purposes of replicating results. Plainly, in a dynamic environment such as the web, replication of results is simply not feasible or even possible. The ability to maintain an audit trail and thought-processes, however, is achievable, and maintenance of the audit trail forms an integral part of conducting open source research for law enforcement.

While there are guidelines for the need of audit trail maintenance, there are no general standards or guidelines for how one, practically, obtains online artefacts while conducting open source research. Typically, open source research is performed using a web browser, software tools/browser add-ons to capture artefacts (for example, screenshot applications) and a means to maintain an audit log. The issue therefore lies in there being no standard as to what tools to use. For example, an officer may use Mozilla Firefox with a screenshot

add-on and a pen and paper to maintain an audit trail, while another may choose to use Google Chrome, Snagit (a screen capturing tool) with Microsoft Excel to maintain the audit trail. These varied tools are not only inconsistent, but can also be expensive and hard to train those who need them. Additionally, such inconsistency in information collection can breed confusion later down the line when facts are collated; officers may waste valuable time trying to decipher another's work. Repetition of work would be inevitable and may contravene the guidance set out by Chief Surveillance Commissioner Sir Christopher Rose about repeated viewings of open source material requiring legal authorisations under RIPA (Rose, 2014, pp. 20–21).

This lack of standardisation was perceived by the College of Policing, who provide the only accredited course for training open source research in the UK. In January 2015, the lead trainer for high-tech crime reached out with a broad specification for a software product that combined all the tools required for conducting open source research into one manageable tool. This tool ultimately became Open Source Internet Research Tool (OSIRT) and has become widely used in UK and international law enforcement, private training packages and by private individuals to diligently conduct open source research.

1.2 Research questions

The overarching contribution this thesis makes is the creation of OSIRT, which has been developed with and for law enforcement, along with the theoretical knowledge surrounding the design and development of OSIRT. There are several topics and areas surrounding the creation of OSIRT, and this thesis answers the following key research questions:

1.2.1 What constraints do law enforcement in the UK face when conducting open source research?

The aim of this question is to gain an understanding of the current issues law enforcement face when conducting open source research. Some issues include legal limitations, such as the use of laws which pre-date the modern era of social media. While other issues focus on ethical considerations such as; is it acceptable for law enforcement to obtain publicly available information, even if it contains personal and private data? Beyond ethics, the impact of social media companies' 'terms of service' may play a role in an officer's decision making.

This question also looks at how law enforcement, from a practical standpoint, conducts open source research. This sub-topic looks at the procedural difficulties surrounding the collection of open source materials, both in terms of software tools used and whether appropriate laws and guidelines are being followed.

Chapter 2, 4 and 6 focus on this question.

1.2.2 What do law enforcement need from a software tool when conducting open source research?

After establishing how law enforcement conducts open source research from a practical standpoint, this question looks to contribute and challenge the current status quo of tools law enforcement use by means of a loose specification based on a bespoke software tool designed to conduct open source research from the College of Policing.

This question looks beyond the technical aspects and considers the importance of the human aspects of software and the importance of users being involved in the creation of software.

Chapters 4, 6, 8 and 9 focus on this question.

1.2.3 What are the unique elements and challenges when engineering a software solution for law enforcement?

Engineering a product for law enforcement for the use of evidential capture is a substantial undertaking. Given a loose software specification and a broad concept from the College of Policing, this question looks at what needs to be done by a developer to ensure the product meets the requirements of law enforcement. Even at force-level, policies and internal guidelines makes this a challenge as each force may require a product to be different. Holistically, because it is a new tool and new facility, the hard-line specifics of what was needed and necessary had yet to be solidly defined. The fundamentals of a functional program and the parameters of basic use became more apparent after OSIRT's initial prototype.

Chapters 5, 6, 7, 8, and 9 focuses on the development aspects and discuss and reflect upon the unique elements surrounding this thesis.

1.2.4 How can developers involve users in the design process in a ‘closed’ environment?

In sensitive settings, such as those seen in law enforcement environments, how can a developer involve users in the design process of a software product with a possible heavy gatekeeping presence? Additionally, how does a developer obtain useful data to inform their design by maximising the time had with law enforcement officials without causing significant disruption to their role. This question aims to discuss and reflect upon a process for those choosing a user-centred design method for official and sensitive development projects.

Chapters 3,4,5,6,7,8 and 9 focus on this question.

1.2.5 How can law enforcement be effectively trained to conduct open source research?

For software to be integrated into law enforcement, those that are intended to use it require training. This question looks at how law enforcement are trained to use OSIRT and how to conduct open source research by means of a study that observed the College of Policing’s training package. Chapter 10 focuses on this question.

1.3 Research methodology outline

The goal of this research was to create a tool that law enforcement can use to conduct open source research. The overall approach followed a user-centred design method, discussed in Chapter 3, that was split into two key phases: a ‘prototype’ and a ‘release’ stage.

1.3.1 Prototype stage

The prototype stage focused on gaining a better understanding of the requirements for a software tool. Beyond the software specification, an understanding of the issues faced by law enforcement from a procedural, legal and ethical standpoint is equally as important. Initially, a review of the issues faced by law enforcement was conducted, this was to gain an understanding of the legal and procedural frameworks in place when performing open source research. To support the review, semi-structured interviews were conducted with serving law enforcement officials to capture their thoughts on procedural, legal and ethical issues.

A software engineering methodology was selected that was appropriate for the loose software specification provided by the College of Policing. For this, the throwaway prototype method was chosen as it offers a method to rapidly create a product that can be used that can be used in obtaining feedback. During the prototyping, observations were conducted of officers using the software at a College of Policing course with the outcome of this observational data shaping the future of the tool. Additionally, System Usability Scale questionnaires were distributed to those attending the course to gauge if OSIRT was usable. Finally, interviews were conducted with officers using the tool in their working environment. The data from these interviews along with the observational data and personal discussions via e-mail and phone calls resulted in a clear picture of what was required for an open source research tool, which fed into the 'release' phase of development.

1.3.2 Release stage

The second phase uses an iterative and incremental software engineering methodology, which acknowledges that a system cannot be fully completed from version 1. This method also has a key focus on communication and feedback which slots in with the user-centred design approach. Communication and feedback is an important aspect of the methodology as discovered from the prototype phase of the development.

To show the impact the tool has had on law enforcement, questionnaires, observations and semi-structured interviews were conducted with officers who are using or used the tool.

Additionally during this phase, an expert evaluation method, cognitive walkthrough, was conducted on OSIRT. Finally, an analysis and evaluation was conducted of the College of Policing course where OSIRT is used. This looks at teaching and learning models and the use of observations and questionnaires to get feedback from participants.

1.4 Contribution to knowledge

Given the unique aspects surrounding the building of a bespoke software product for law enforcement, this thesis contributes to knowledge in the following ways:

- The creation of a software tool for law enforcement based on a minimal software specification. This covers the entirety of the software development life-cycle, from conception to maintenance. It involves working closely with law enforcement and ensuring their requirements are met.
- An understanding of the issues faced by law enforcement officials from a legal, ethical and procedural perspective when they conduct open source research.
- A measure and method of showing the value of OSIRT and its effectiveness for UK law enforcement.
- An approach to working with law enforcement on the unique aspects of the development of the tool; this includes testing and training the software.
- Narrative and reflection on the approach taken to the development of the tool as a lone developer, and an analysis and discussion on the advantages and limitations of the approaches taken when developing a tool for law enforcement.
- Expectations from law enforcement officials, and any issues surrounding the integration of software into police and other law enforcement networks.
- The impact of whether a tool being free and open source software makes a difference to the integration and dissemination of a piece of software.

1.5 Contributions made by this thesis

The originality of this research has contributed to publication of several peer-reviewed articles and academic workshops, some of which have been included within the thesis.

1.5.1 Peer-reviewed articles

Williams, J. and Humphries, G. *Analysis of a Training Package for Law Enforcement to Conduct Open Source Research*. In: International Journal of Cyber Research and Education. August 2018. IGI Global. ISSN: 2577-4816. (Chapter 10)

Williams, J. *Creating and Integrating a FLOSS Product into UK Law Enforcement*. In Springer - IFIP Advances in Information and Communication Technology. Proceedings of the 14th International Conference on Open Source Systems, Harokopio University, Athens. 8-10 June, 2018. ISBN 9783319923741. (Chapter 4, 8 and 9)

Williams, J. *Legal and Ethical Issues Surrounding Open Source Research for Law Enforcement Purposes*. In: Skarzauskiene, A. and Gudeliene, N., eds. Proceedings of the

4th European Conference on Social Media, Mykolas Romeris University. Vilnius, Lithuania. 3-4 July, 2017. ACPIL. ISBN 9781911218463 (Chapter 2 and 4)

Williams, J. and Stephens, P. *Development of a Tool for Open Source Internet Research*, CFET 2015 Annual Conference, Canterbury Christ Church University, Canterbury, UK, September 3-4, 2015. (Accepted) (Chapters 5 and 6)

1.5.2 Workshops

Williams, J. and Humphries, G. *Effective Training of Investigators for Conducting Open Source Research*. In: 13th Annual Teaching Computer Forensics Workshop, 2nd November, 2017, Sunderland, UK.

Williams, J. *An Open Source “open source internet research tool”*. In: BCS Cybercrime Forensics and the Open Source SGs, BCS Hampshire Branch, June 9th 2016, Southampton Solent University, Southampton.

Williams, J. and Stephens, P. *OSIRT: a tool for law enforcement research and investigation*. In: 11th Annual Teaching Computer Forensics Workshop, 19th November, 2015, Sunderland, UK.

1.5.3 Posters

Williams, J. *Open Source Internet Research Tool (OSIRT): an investigative tool for law enforcement officials*. In: HEA National Conference for Learning and Teaching in Cyber Security, 5-6 April, 2017, Liverpool.

Williams, J. and Stephens, P. *Development of an Integrated Forensic Tool for Open Source Internet Research*, DFRWS EU 2015 Annual Conference, University College Dublin, Dublin, Ireland, March 23-26, 2015.

1.6 Thesis structure

Chapter 2 reviews the procedural, legal and ethical issues faced by law enforcement officials when conducting open source research. This includes a discussion and a review of the current tools used when conducting open source research by UK law enforcement.

Chapter 3 discusses the chosen methodology and data collection methods.

Chapter 4 discusses the results of the interviews with 22 law enforcement officials surrounding legal, ethical and procedural issues of open source research.

Chapter 5 discusses the technical implementation of the prototype version of OSIRT, which has been heavily influenced by the prototype feedback, and is closely linked to chapter 6.

Chapter 6 analyses, evaluates and discusses the feedback from the OSIRT prototype given by 11 participants of the RITES course, five semi-structured interviews, observation of a RITES course, and personal e-mails. The chapter critically focuses upon the feedback given, why the participants have provided the feedback and how the release version of OSIRT can integrate the feedback.

Chapter 7 discusses the technical implementation of the release version of OSIRT, which has been heavily influenced by the prototype feedback.

Chapter 8 discusses the integration, impact and contribution of OSIRT from a user experience perspective. Interviews, observations and SUS questionnaire results and comments are discussed and reflected upon in this chapter.

Chapter 9 discusses the integration, impact and contribution of OSIRT from a holistic perspective. Questionnaire results are discussed from 32 law enforcement officials, along with an interview from the lead high-tech crime trainer from the College of Policing. This chapter also reflects upon the development of OSIRT, particularly as a piece of free and open source software.

Chapter 10 critically analyses the RITES course and OSIRT's integration into it as an effective tool and training package for law enforcement to effectively conduct open source research.

Chapter 11 concludes the thesis and its findings, and provides a critique of the thesis itself. The thesis contributions are highlighted and discussed and provides a summary of further work and research.

2 A REVIEW OF PROCEDURAL, LEGAL AND ETHICAL ISSUES SURROUNDING OPEN SOURCE RESEARCH

INTRODUCTION

Open source research may give the impression that it is a simplistic capture method. For law enforcement in the UK, however, conducting online investigations is one fraught with subtle hazards and seemingly arbitrary stipulations. The tools and mindset LEOs need are not simply based around being able to use the Internet; a good legal and technical knowledge must back up a decision making and problem-solving skillset.

This chapter reviews the legal and ethical issues surrounding open source research for UK law enforcement and includes a review of open source research, how it contributes to intelligence packages, why it came to prominence in UK law enforcement, and the legal and ethical issues LEOs face when conducting open source research. The second section looks at the technical and procedural issues surrounding the capturing of artefacts as part of open source research, including a review of the software tools and practices being utilised in working environments and being trained.

2.1 Open sources and their contribution

When discussing open source research, there are three different terms to consider: Open Source Information (OSINF), Open Source Intelligence (OSINT) and open source research. As ‘open source research’ is, predominantly, a UK-centric term used in law enforcement this section looks to clarify the parlance used surrounding the notion of open source.

OSINF is data in its rawest form, and it is available to anyone. ACPO (2013) define OSINF as:

“Open source is defined as publicly available information (i.e., any member of the public could lawfully obtain the information by request or observation). It includes books, newspapers, journals, TV and radio broadcasts, newswires, Internet WWW, and newsgroups, mapping, imagery, photographs, commercial subscription databases and grey literature (conference proceedings and institute reports).” (ACPO, 2013)

OSINF is frequently coupled with OSINT, but there is a clear distinction. In order for information to be OSINT, there must be analysis of OSINF, with this analysis stage providing the ‘intelligence’ aspect. Best (2007) notes OSINF is the “Collection monitoring, selecting, retrieving, tagging, cataloguing, visualising and disseminating data” and OSINT “is the result of expert analysis of OSINF”. Best’s definition agrees with that of ACPO’s (2013):

“The collection, evaluation and analysis of materials from sources available to the public, whether on payment or otherwise to use as intelligence or evidence within investigation.”

The ambiguity arises as ACPO use the term ‘open source research’ instead of OSINT, and while there is a difference in terminology between open source research and OSINT, their definitions are similar. Why UK law enforcement chose open source research over OSINT, though, is unclear. Speculatively, “intelligence” implies a level of specialised filtering and analysis. While ACPO’s (2013) definition for open source research clearly states “evaluation and analysis of materials”, this may be from an individual officer’s perspective, and may fail to meet the “expert analysis” as defined by Best (2007) for OSINT. Going forward, this thesis uses the term open source research. Figure 2.1 proposes a model to distinguish the difference between the terms.

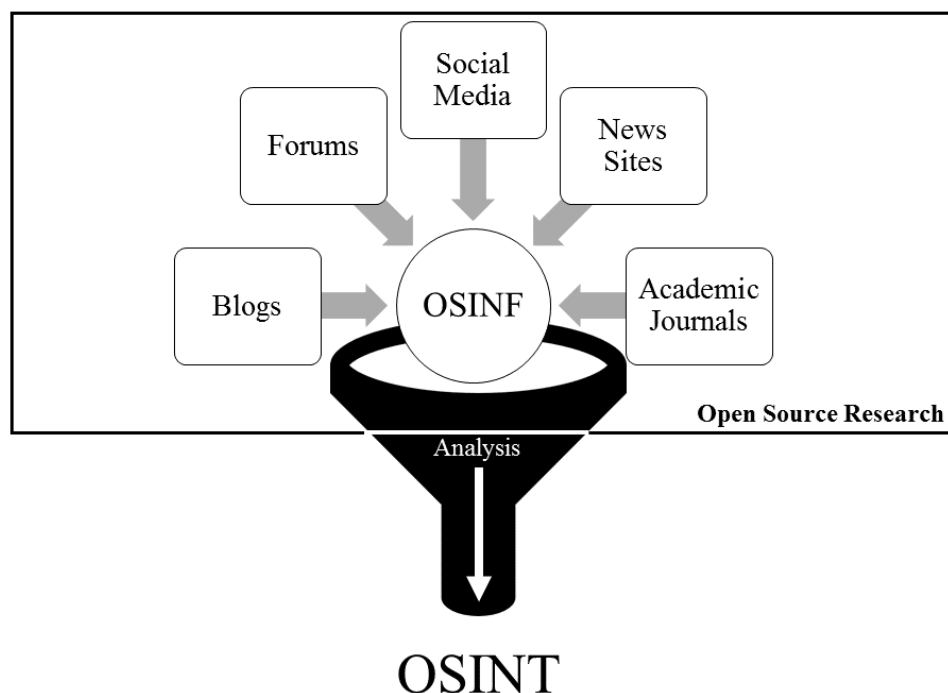


Figure 2.1 Model representing the difference between OSINF (with some examples provided), open source research and OSINT

Open source research's contribution plays a significant role, and forms a substantial element of intelligence packages. Allen Dulles, former director of the CIA in 1947 said:

“A proper analysis of the intelligence obtainable by these overt, normal and aboveboard means would supply us with over 80 percent, I should estimate, of the information required for the guidance of our national policy” (cited in Wells and Gibson, 2017).

The 80 percent stated by Dulles was in no way hyperbolic, with the utilisation of open sources still contributing to intelligence packages ranging from 80 percent (Thompson, 1998; Hulnick and Valcourt, 1999; Dover, Goodman and Hillebrand, 2013) to as much as 95 percent (Clarke *et al.*, 2015). While these contributions appear high, it is not too surprising given that open sources are considerably cheaper and simpler to obtain than their closed sourced counterparts.

Many constabularies around the UK now employ Digital Media Investigators (DMI) as part of their cybercrime units, with one of their duties being to conduct specialised open

source investigations. That said, open source research is not limited to DMIs; police officers from the Professional Standards Department to Anti-Terrorism Units also conduct open source research for their roles. Further afield, beyond policing, open source research is used by other UK agencies such as Trading Standards and the Food Standards Agency, and private companies performing due-diligence checks. With Internet usage growing ever larger, the need for open source research continues to be a necessity.

2.1.1 The rise and rise of the Internet

With the Internet being as freely accessible as it is, conducting open source research is, arguably, as easy as it has ever been. Internet use has seen yearly growth in the UK; the Office for National Statistics identified that in 2018, 90 percent of adults had recently used the Internet, an increase from the 89 percent seen in 2017 (Office for National Statistics, 2018a). Of those UK adults, 77 percent used “social networking sites” in 2017 (Ofcom, 2018)

Inspecting these figures of social media usage, it is not surprising to see the number of reported crimes involving two of the most used social networking sites, Facebook and Twitter, has risen to over 16,000 in 2014 (Evans, 2015). This upsurge affects the vast majority of police forces within the UK; Bartlett and Reynolds (2015, p. 22) notes “38 out of 43 UK police forces” have seen an increase in crime reports involving Facebook. Given these figures and the year-on-year increase of Internet usage within the UK, law enforcement is tasked with unique challenges that would have been unfathomable ten years ago.

The United Kingdom is not unique with these numbers, similar figures for Internet usage are seen across the 28 European Union member states, where an average of 85 percent of 16-74 year olds have used the Internet (*Internet World Stats*, 2017; *statista*, 2017). Likewise, North America also sees a large portion of its population as being Internet users, with an average Internet usage penetration of 87.9 percent (*Internet Usage and 2015 Population in North America*, 2015) and social media usage at 67 percent (‘Digital in 2016’, 2016)

2.1.2 Guidance and laws surrounding open source research

Regulation of Investigatory Powers Act (2000) (RIPA) is a key piece of legislation to look at when LEOs need to conduct open source research. However, given that RIPA pre-dates the modern era of social media platforms (e.g. Facebook was founded in 2004, Twitter in 2006) it largely covers covert interception of communications from technology available at the time. Communications like email, SMS messages and telephones all comfortably fall under RIPA's authority but, unsurprisingly, it does not mention anything about social media.

Despite no legislation like RIPA to provide concrete structure, and regardless of the 'open' nature of open source research, LEOs must still follow procedures and guidelines. The Association of Chief Police Officers (ACPO) in the *Online Research and Investigation* manual lays out one such set of procedures. The 'Guiding Principles' state that viewing open source information "does not amount to obtaining private information because that information is publicly available" (ACPO, 2013), and due to this it is "unlikely to require authorisation under RIPA" (ACPO, 2013). However, ACPO (2013) note that while the open sources may be collected, it must be "necessary and proportionate" and "does not necessarily mean a person has no expectation of privacy" (ACPO, 2013). Expectations of privacy are set out under Article 8, a right to respect for private and family life, under the European Convention on Human Rights (ECHR). Under the Human Rights Act (1998), decisions when handling personal information must be "necessary" and "proportionate". Kent Police use the JAPAN test when handling person information, seen in Figure 2.2 ('The JAPAN Test', no date; kent.gov, 2012). While the JAPAN test itself may not be followed by all police forces, its concepts will be. For example, authorisation, necessity and proportionality are the backbone of UK policing, and form part of statute laws such as RIPA. Additionally, auditability and justification is guided by ACPO and NPCC principles, along with data protection laws.

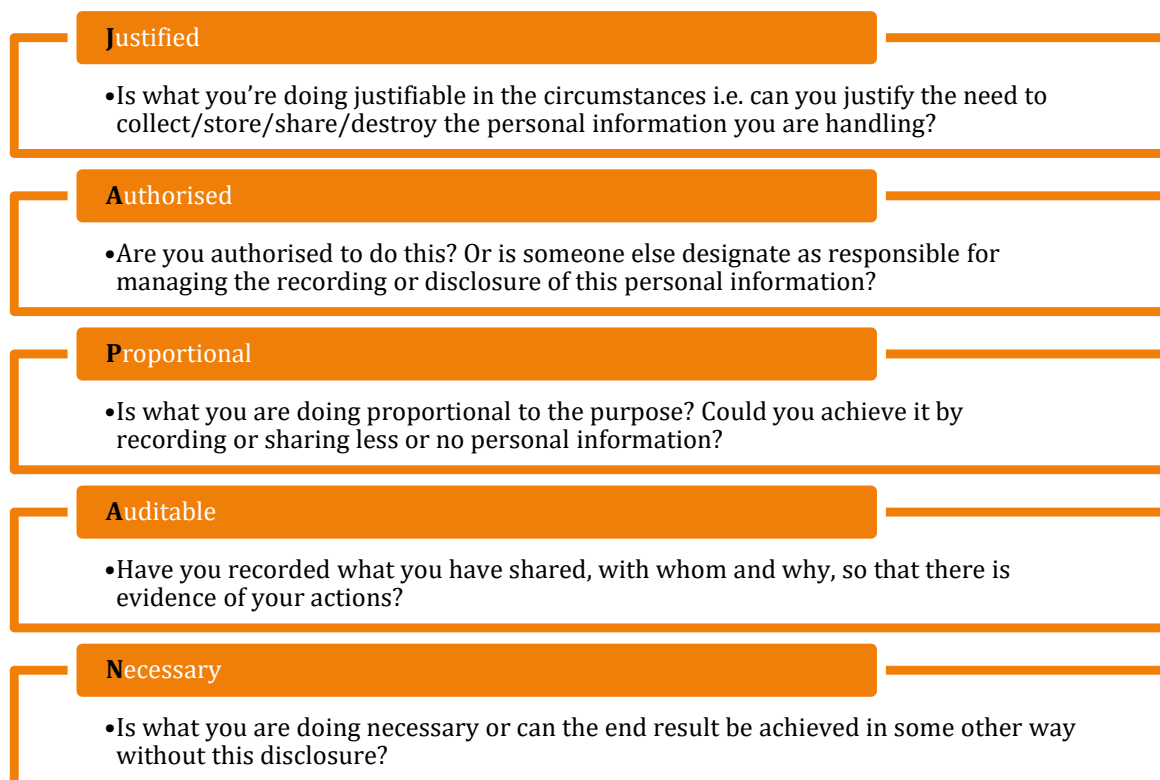


Figure 2.2 JAPAN test for handling personal information

The General Data Protection Regulation (GDPR) came in effect in May 2018 and has had an impact on how law enforcement² within the UK and the European Union (EU) manage personal data. The GDPR provides citizens (termed “data subjects”) with greater control over their personal data from “controllers” (i.e. those who control the data subject’s personal data). Data subjects now, trivially, can access and remove personal data upon request.

Previously, under the Data Protection Act 1998, law enforcement gathering and processing of personal data was largely exempt from data protection law under sections 28 (national security) and 29 (crime and taxation) (Data Protection Act, 1998). ACPO guidelines (2013) stated that data should still be managed and stored in adherence with other principles within the Data Protection Act (1998).

Now, GDPR provides member states of the EU provisions on how to apply GDPR, and in the UK this brought in the Data Protection Act 2018, superseding the 1998 Act of the

² Plus anyone else who handles personal data.

same name. The Data Protection Act (2018) covers aspects that “fall out of scope of EU law” (Information Commissioners Office, 2018), such as national security and how “intelligence services” (Data Protection Act, 2018) manage personal data; this is covered by Part 4 of the Act. However, Part 3 of the Data Protection Act (2018) covers “Law Enforcement Processing” and provides six “protection principles” in Chapter 2 of the Act for those managing personal data for law enforcement purposes. The principles are abridged in Figure 2.3.

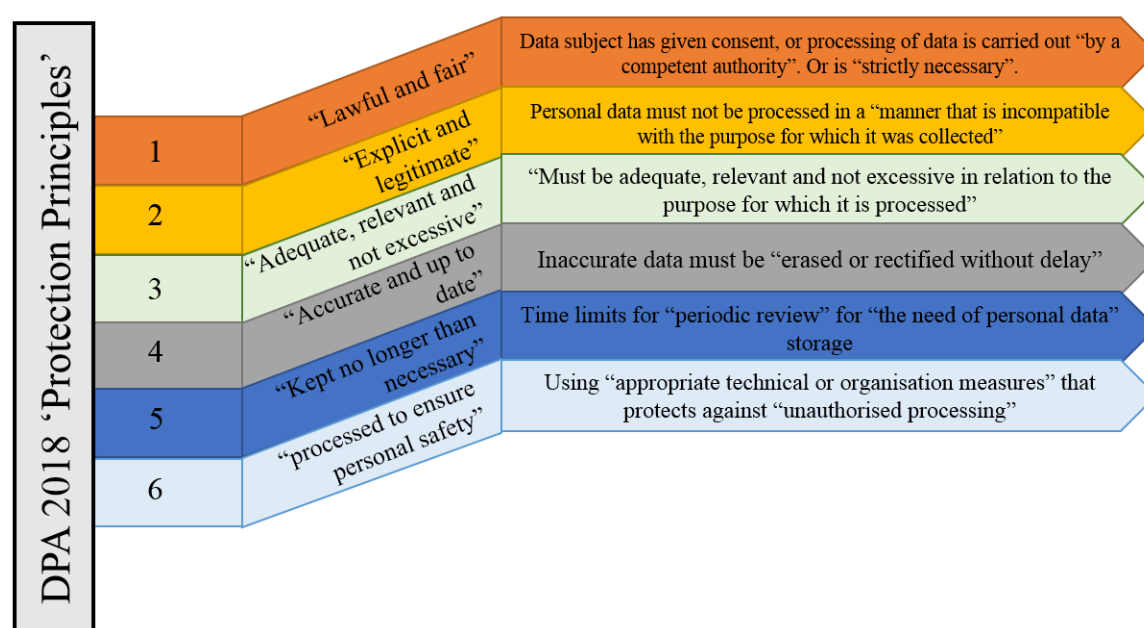


Figure 2.3 Protection principles of the Data Protection Act (2018) for law enforcement purposes

Law enforcement are afforded exemptions from the Data Protection Act (2018), but as seen from the ‘protection principles’ within the Act there are themes of necessity and proportionality when handling sensitive data.

In addition to statute laws, ACPO provide other procedures available under the Guidelines for Good Practice for Computer-Based Electronic Evidence handbook. In particular, principle 3 specifies that a “record of all processes applied to digital evidence should be created and preserved” (ACPO, 2012).

Case law itself provides few guidelines for the digital investigator when conducting open source research. A notable case is *Bucknor v R* [2010] EWCA Crim 1152, in which

Bucknor appealed against his conviction of murder. The judge ruled in the initial case that evidence presented from the social networking sites Bebo and YouTube were admissible. While the initial conviction was upheld, the judgement from the appeal means any evidence taken from the Internet must have full provenance. That is, when (the date and time) and where (the website) the evidential artefact was obtained should be audited.

2.1.3 Open source research and Social Media

With social media playing an important role in modern day law enforcement, SOCMINT (Social Media Intelligence), has been coined by Omand, Bartlett and Miller (2012) and focuses on the collection, analysis and use of social media data. While some intelligence analysts bundle SOCMINT with open source research, SOCMINT does not necessarily fall into the realms of open source, but neither does it fall into ‘secret intelligence’ according to Bartlett and Reynolds (2015). Bartlett *et al.* (2013) proposed an extensive “regulatory framework” for conducting SOCMINT; it provides reasonable steps law enforcement could take when investigating and gathering SOCMINT. It is important to note, though, that the notion of SOCMINT is merely a proposed framework, and not part of law enforcement parlance.

To compound confusion, the Chief Surveillance Commissioner Sir Christopher Rose, said in his annual report from June 2015 that “just because this material is out in the open, does not render it fair game” (Rose, 2015). Rose (2015) also notes that “authorisation under RIPA or RIP(S)A” is required for “repetitive viewing of what are deemed to be ‘open source’ sites”. Rose’s reasons stem from privacy concerns, and the ambiguity that has arisen from open information on social media websites, where its users may, or do, consider a reasonable expectation to privacy; especially from law enforcement or government agencies. An additional aspect for concern is the possibility of ‘collateral’. That is, a person who is innocent or irrelevant to an open source investigation being caught up in the research.

Nevertheless, RIPA is permissive legislation, so failing to obtain authorisation does not necessarily render surveillance unlawful; although given the advice above, that would be perhaps unwise. This guidance from the Chief Surveillance Commissioner is poignant. It adds to the debate of privacy versus security, showing law enforcement within the UK are concerned with protecting the right to privacy.

The juggling act of privacy versus protection can prove to be relatively challenging. In August 2011, London and other cities within the UK were plagued with civil unrest. Law enforcement was consequently criticised for their deficiencies; supposedly failing to capture social media and missing warning signs days before the incidents. Her Majesty's Chief Inspector of Constabulary, Sir Thomas Winsor, acknowledged these shortcomings by noting "[...] this kind of [social] media is not well understood and less well managed" (Cited in Hobbs, Moran and Salisbury, 2014). Winsor's comments appear to suggest that police were expected to trawl through hundreds of social media profiles to detect potential sources of trouble, compromising privacy, for the sake of potential protection. While at the time it can be argued that the role of social media as an open source of intelligence was underappreciated, the overall sentiment remains the same.

Open source research is not just controlled under legal and procedural frameworks as seen above; each social media platform has terms and conditions governing its usage. Even if law unambiguously permitted accessing and collating open source information, social media platforms may not. For example, obtaining user data from YouTube breaches its Terms of Service (ToS). Section 5.1I states that "You agree not to collect or harvest any personal data of any user of the Website" (*Terms of Service - YouTube*, 2019), additionally, section 5.1L states that videos must be viewed in "real-time" and they are "not intended to be downloaded (either permanently or temporarily), copied, stored, or redistributed by the user." (*Terms of Service - YouTube*, 2019). Given just these two terms of service, collating open source information from YouTube would be a breach of their ToS. Whether YouTube would ban or bring civil litigation against a LEO for contravening their ToS is yet unknown.

Facebook provides a Statement of Rights and Responsibilities (SRR) users must follow when using Facebook. The SRR makes it clear under Section 3.2 that if any personal information is collected, consent must be obtained from that user (Facebook, 2018). The inference being that law enforcement must tell a person when collecting their information; otherwise, it is a breach of Facebook's SRR. Interestingly, Facebook disallows the collection of users' content or information via an "automated means" (Section 5.7) (Facebook, 2018), however this implies a manual means is acceptable.

As part of their investigations, officers may use ‘false personas’, also known as ‘fake accounts’, to conduct research (discussed more in section 2.1.5). False personas have always been against Facebook’s terms and conditions, but Facebook came under intense scrutiny after the Cambridge Analytica scandal, with one element being the use of fake accounts. This prompted Facebook to create an advertising campaign in 2018 (Figure 2.4) making clear that fake accounts will be removed.



Figure 2.4 Advertisement by Facebook after the Cambridge Analytica scandal

In addition to the advertising campaign, Facebook updated its Community Standards to make clear that “Authenticity is the cornerstone of our community” (Facebook, 2018). Section 17 of the Community Standards discusses “misrepresentation” and what is against its terms and conditions that will cause immediate account deletion. The pertinent point is that creating “inauthentic profiles” or engaging in “inauthentic behaviour” are forbidden; this includes, explicitly, “fake accounts”.

2.1.3.1 National Police Chiefs Council – The hidden guidance?

In late 2016, a document marked “Restricted” with heavy redactions was uploaded on the Suffolk police website (Housego, 2015). The document, *NPCC Guidance on Open Source Investigation/Research*, is dated April 2015 and offers several minor extensions and enhancements to the previous guidance set out by ACPO in their *Online Research and Investigation* manual. While the guidelines still acknowledge that “a person may have reduced expectations of privacy” in public, NPCC use the example of two people having a conversation in public could have a “reasonable expectation of privacy”. This analogy is then extended and is “likely to apply” to social media conversations, regardless if

“friends control [...] has been activated” or not. The tenor of the language is, arguably, quite vague showing that legislation is allowing for ambiguity to arise.

Section 4 of the NPCC’s guidance is on ‘evidence’ and while this section is heavily redacted, it does provide two important guidelines surrounding open source capture. Point 4.1 stresses the need for processes to be in place for a full, auditable record of evidential capture that must be available to be examined later; as specified under *ACPO Good Practice Guide for Digital Evidence*. Point 4.2 focuses on veracity of evidence “for its life from capture to court” and while it notes that hashing is important for reasons of integrity, the key element from this point is that evidence “should always corroborated or attributed in some way”. In other words, law enforcement should justify their actions for any captures they have taken.

Further, points 4.4, 4.7 and 4.8 discuss the need for legal authorisations under RIPA, such as for surveillance (directed surveillance is discussed in ‘Levels of open source’ section 2.1.5). Point 4.7 state that if a profile’s privacy settings are such that they leave a profile open, then it is “unwise” to consider that to be open source, but guidance only explicitly states that the user’s reasonable expectation to privacy” is only applied to those where “[privacy] access controls are applied”. Point 4.7 concludes that RIPA authorisations should be considered on a “case by case” basis by officers.

2.1.4 #thinkdigital

The College of Policing, National Crime Agency and NPCC released a non-policy report titled “Digital Investigation and Intelligence” (DII) (Scriven and Herdale, 2015), with the aim of highlighting what is required from law enforcement in a digital age. A framework developed by the College of Policing, the “DII capability map”, displays four key capability areas: people, ways of working, digital exploitation and digital sources. Each of these areas have sub-capabilities, for example, digital sources contains “digital forensics” and “acquisition”. The report uses the results from a “comprehensive survey” that took place in 2014 to establish activity surrounding DII. While there are no quantitative figures, the report provides a heat and tree map (Figure 2.5). What is clear from the map is that “Open Source Assessment” in digital sources and “Research” in digital exploitation are in the red with the least progression and recorded activity, yet are

key capabilities highlighted in the DII and shows a clear need for enhancing these digital capabilities.



Figure 2.5 Tree map and heat map of DII capabilities (Scriven and Herdale, 2015)

2.1.5 Levels of open source

While definitions surrounding open source research frequently note source material as being “publicly available” that does not necessarily mean the research itself will be conducted in an overt, public manner. In UK policing, there is guidance surrounding the “levels” of open source research; seen in Figure 2.6. Of the five levels, only level 1 is considered to be “overt” with the rest being “covert” in nature. In fact, seldom are open

source investigations performed overtly (College of Policing, High-Tech Crime Trainer, personal communication [e-mail] June 2016).

The reason for using covert techniques, particularly at levels 2 and 3, is to minimise the ‘footprint’ of the investigating officer; i.e., the digital trace left behind when visiting a website. For example, an IP address, Internet Service Provider and location could show a law enforcement official from a police computer was visiting a website. “Covert” at these levels of open source capture focuses more on protection for the officer, and police network, as a counter-intelligence and surveillance measure. Plainly, the higher the level the more training is required. For example, level 1 usually only requires basic training around force policy of computer usage.

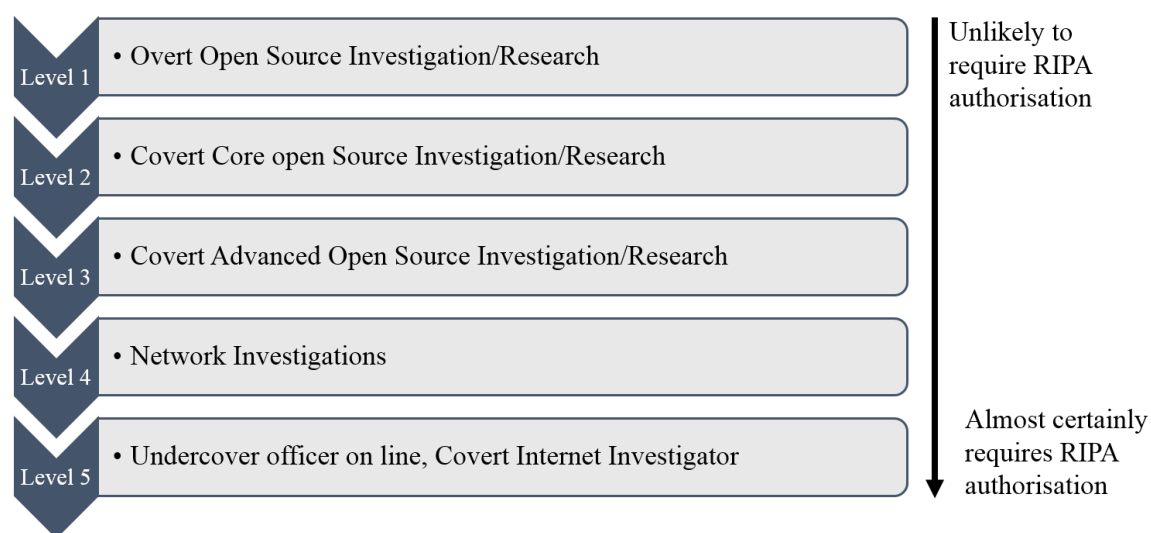


Figure 2.6 Levels of open source research and their requirement to obtain authority under RIPA

Level 5 raises a debate surrounding what is ‘open source’. An officer at this level will require extensive training to be an undercover officer or, formerly, a Covert Human Intelligence Source (CHIS). As a CHIS, an officer will act as a certain individual probably with the aid of a false persona. This may be to infiltrate a particular group, for example. Guidance for CHIS and covert activity is provided in a Home Office (2014) report:

“The use of the internet may be required to gather information prior to and/or during a CHIS operation, which may amount to directed surveillance. Alternatively the CHIS may need to communicate online, for example this may

involve contacting individuals using social media websites. Whenever a public authority intends to use the internet as part of an investigation, they must first consider whether the proposed activity is likely to interfere with a person's Article 8 rights, including the effect of any collateral intrusion. Any activity likely to interfere with an individual's Article 8 rights should only be used when necessary and proportionate to meet the objectives of a specific case. Where it is considered that private information is likely to be obtained, an authorisation (combined or separate) must be sought..." (Home Office, 2014)

The deliberate, covert capturing of an individual's personal information online as part of an investigation would, arguably, fall under "directed surveillance" and would require a Directed Surveillance Authority (DSA) under RIPA. However, any open source investigation at any level would, presumably, require a DSA under RIPA but the blurred lines of open source makes this a hard question to answer with certainty. Advice and guidance merely hint that officers probably should get a RIPA authorisation, as seen with ACPO and, now, NPCC guidance surrounding open source.

2.1.6 ISO/IEC 17025:2005

ISO/IEC 17025:2005 (herein ISO 17025) 'General requirements for the competence of testing and calibration laboratories' (ISO, 2005), is the standard that sets out and "specifies the general requirements for the competence to carry out tests and/or calibrations". ISO 17025 was adopted in October 2017 in the UK under the 'Forensic Science Regulator's Codes of Practice and Conduct'. While ISO 17025's focus from the Forensic Science Regulator surrounds the capturing and analysis of digital forensics outputs, such as imaging hard drives and data recovery, there was a push for the "Capture and analysis of social media and open source data" (Forensic Science Regulator, 2015) to also fall under ISO 17025 accreditation. In October 2015, the Forensic Science Regulator placed a date of "TBA" (to be announced) for the capture of open source materials, however, subsequent newsletters have made no mention of ISO 17025 for open sources. The Forensic Science Regulator's Codes of Practice and Conduct in 2017 are still stating that ISO 17025 accreditation for open source data as "TBA" (Forensic Science Regulator, 2017) as of writing in late 2018.

2.1.7 College of Policing and the Researching, Identifying and Tracing the Electronic Suspect course

The College of Policing runs a dedicated course to train officers in conducting open source research. The Researching, Identifying and Tracing the Electronic Suspect (RITES) course has been an aspect of the College's training portfolio for nearly two decades. Within the past six years, though, it has taken on the role of becoming the only accredited course for officers wishing to conduct open source research up to level 3, with the course's aims "To provide investigating officers with the skills necessary to obtain, evaluate and use online information." (College of Policing, 2017).

The College of Policing train around 100 officers a year to conduct open source research via the RITES course and has had considerable impact on how open source research is conducted by UK law enforcement.

2.1.7.1 RITES course open source capture and methodology

Figure 2.7 shows the College of Policing's methodology for the capture of open source materials. A theme that runs throughout this method is that there is a large amount of manual process involved, including the creation of folders to store artefacts, the manual maintenance of an audit trail and the capturing of artefacts.

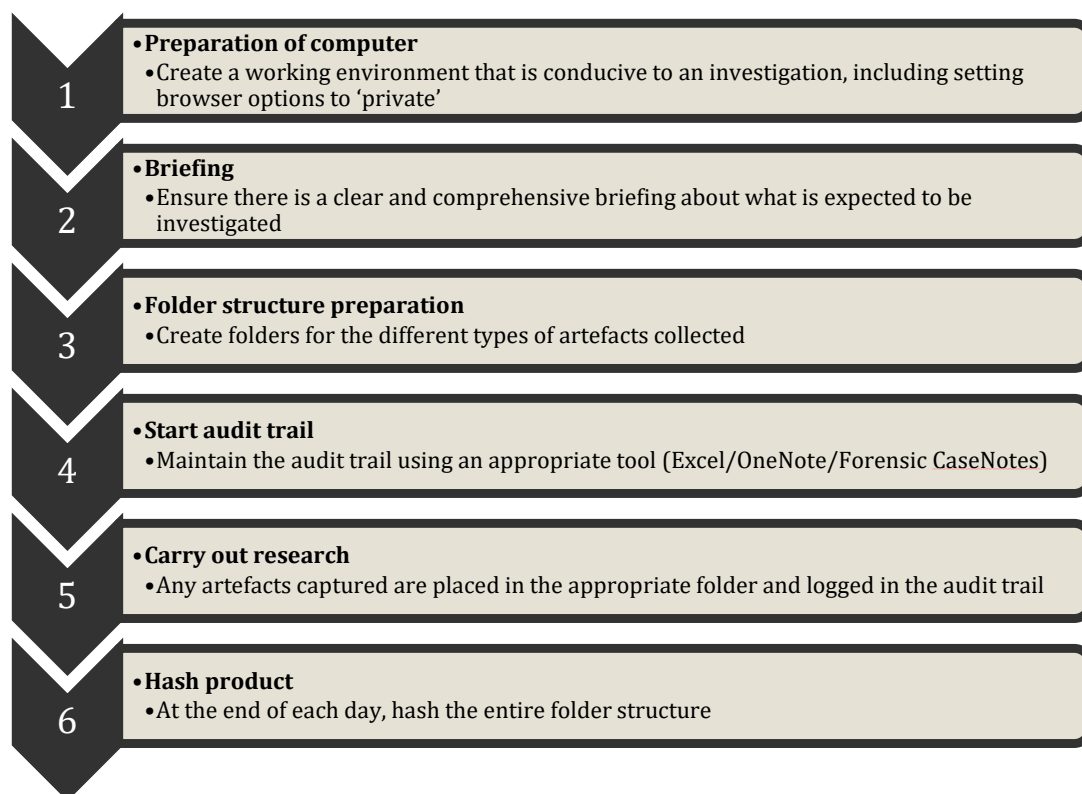


Figure 2.7 Method for capturing open source as trained by the College of Policing in 2014/2015

2.2 Summary of legal and ethical issues

Even with legislation and guidance there is confusion surrounding conducting open source research. This is further stymied by conflicting advice by ACPO/NPCC and the Chief Surveillance Commissioner, as well as the antiquated laws used to conduct open source research.

2.3 Tools and practices surrounding the capture of open sources in UK law enforcement

While the previous section looked at legal and ethical issues surrounding open source research, this section will focus on the technical and procedural aspects. This involves gaining an understanding of the software tools law enforcement use in order to conduct open source research and why they use those particular tools.

2.3.1 Background to software usage by LEOs when conducting open source research

Given the typical open source research workflow seen in Figure 2.8, LEOs must manually log any action they have taken. For example, every website visited must be logged with a date and time stamp. If anything tangible is obtained from that website, such as a screenshot or download, it must be hashed using a suitable hashing algorithm and logged with a date and time stamp in tandem with the originating URL. Any artefacts obtained (e.g. screenshots) are then placed into a suitable directory structure, or directly onto the note taking application of choice to complete the audit log. Any extra annotations the investigator wishes to make are also then added.

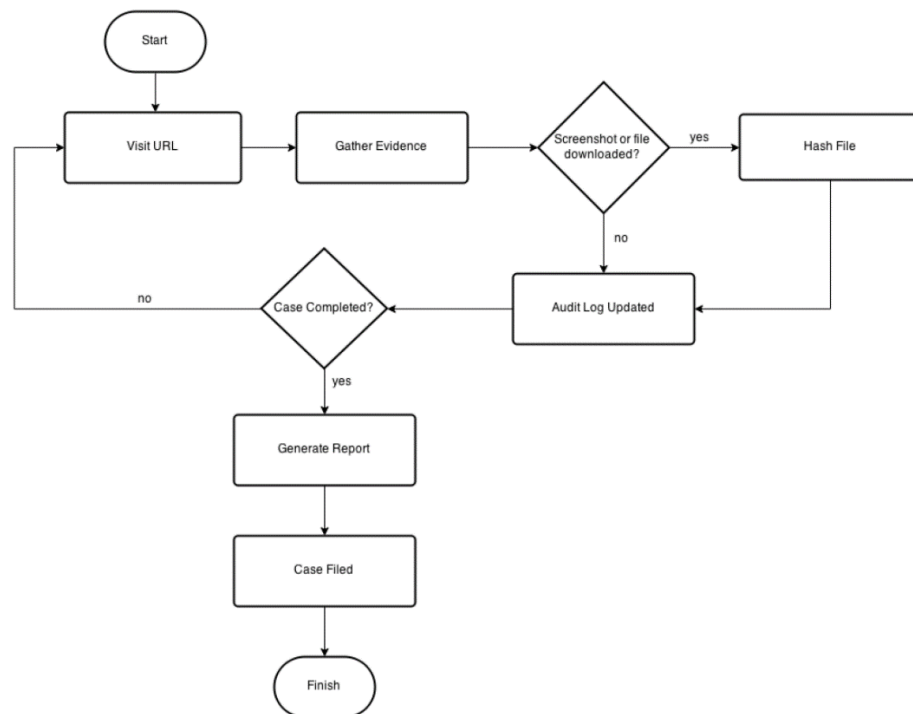


Figure 2.8 Typical workflow for a LEO conducting open source research

This only tells part of the story, however, as in order to obtain these artefacts, LEOs have to use an exhaustive variety of different tools. These tools differ in quality, usability, and price and will often vary from constabulary to constabulary. Largely, they amount to a web browser, static and dynamic screen capturing tools, a hashing tool and a note taking application for manually maintaining an audit log.

2.3.2 Current toolset

This section discusses the toolset used on the RITES course.

2.3.2.1 Fireshot

Fireshot (<https://getfireshot.com/>) is a browser add-on available to all major web-browsers. Fireshot provides the ability to capture static full-page screenshots in a multitude of file formats. In the ‘pro’ version, costing €39.95EUR, Fireshot also includes the ability to edit and annotate images with the addition of simple paint tools.

2.3.2.2 CamStudio

CamStudio is a free Windows application that provides the ability to record the screen with audio and outputs them as an AVI file. However, the latest version of the installer (v2.7) contains undesired applications bundled with it. The reviews on the Source Forge page (<https://sourceforge.net/projects/camstudio/>) also complain of “trojans” and “viruses” among other unwanted software additions. For obvious reasons, this is highly undesirable behaviour for an application that could be used for law enforcement purposes or, in fact, any purpose.

2.3.2.3 Ashampoo Snap

Ashampoo Snap (<https://ashampoo.com/Ashampoo-Snap-9>) can capture screenshots and video, whether partial or full. In addition, Ashampoo Snap also provides editing capabilities for images, such as screenshots, by offering paint commands to annotate the captured image, along with simple video editing functionality. The cost of Ashampoo Snap is \$49.99USD, but there are discounts available to authorities and bulk license purchases.

2.3.2.4 Camtasia

Camtasia (<https://www.techsmith.com/video-editor.html>) is at the top end of the screen recording market. Along with screen recording, Camtasia provides the ability to select parts of the screen to record and provides video editing capabilities. At a cost of £185GBP, it is at the top end of the screen capturing market, but does provide the most features in comparison to the other screen capturing tools. Discounts are available for bulk license purchases, along with further cost reductions for authorities.

2.3.2.5 Snagit

Snagit (<https://www.techsmith.com/screen-capture.html>) provides enhanced screenshot capabilities, such as full screen capturing and ‘snipping’ capabilities. Snagit provides the user with a scroll and drag option, to capture aspects of a window, along with a timed screenshot option to capture dynamic elements. Snagit costs £46, but if purchased in conjunction with Camtasia then discounts are available.

2.3.2.6 Karen’s Hasher

Karen’s Hasher (<https://www.karenware.com/powertools/pthasher>) generates a hash for text and files using a multitude of hashing algorithms ranging from MD5 to SHA512. Karen’s hasher can also be used to generate and verify checksums for files. The hashes, file name, date and time can be saved to a text file or to the clipboard. Karen’s Hasher is freeware, but is no longer maintained.

2.3.2.7 Forensic Case Notes

Forensic Case Notes (<https://www.forensicnotes.com/forensic-case-notes>) is a free utility that provides the investigating officer a means to maintain a manual audit trail and make notes about a case, such as their thought-processes. Forensic Case Notes automatically date and time stamps any entered notes, it is possible to drag and drop images into the notes section. The case notes file is encrypted when the application is closed. Forensic Case Notes is no longer maintained by the developer, and its latest version is 1.3.

2.3.2.8 Microsoft Excel

Excel (<https://products.office.com/en-gb/excel>) is, arguably, the most popular spreadsheet productivity tool on the market. Excel is used to manually maintain the audit log by the investigating officer. Excel is available for a monthly fee as part of the Microsoft Office package, and ranges from £8 to £10 a month per user.

2.4 Tool summary

There is not anything intrinsically wrong with most of the individual tools currently in use, but there is an overhead where officers must manually maintain their audit log. This is done by either using a spreadsheet, a note taking application, or by hand with a pen and paper. Adding new artefacts to the audit trail means obtaining the date and time, the URL

of the website, the name of any files downloaded, the hashes of those files as well as any associated notes an investigator wishes to make.

Trainers noted that the introduction of too many tools, and manual audit log entry, overloaded the delegates. Additionally, audit log maintenance was time consuming and prone to unintentional mistakes; such as a digital investigator forgetting to log when action was taken. Given the nature of what is being obtained, such oversight may compromise a case.

The trainers at the College of Policing identified these shortcomings, and issued a specification requesting a means to encapsulate the functionality required into a single tool; a specification was received in January 2015, sent to the thesis author's supervisor, from the College of Policing that provided the basis of a tool. The specification is provided verbatim in the section 2.5 below.

2.5 Requirements for a bespoke open source research tool

Essential requirements

Ability to set default homepage (e.g. www.google.co.uk).

Ability to enter username and password in protected sites.

Ability to create, save and load a case with any number of different cases to a location of user's choice.

Must record every URL visited in sequence with date and time URL is visited.

Ability to screen capture whole web pages, parts of web pages, videos and downloaded documents.

Ability to add notes when capturing screenshots/videos.

Must be able to automatically hash the screen captures (Still and moving) and documents.

Must be able to store screen captures, audit log in a case container/folder.

Must be able to produce a report showing audit log with screen capture file names and hash values.

Cheap licence (e.g. £30 a licence).

Desirable requirements

Ability to capture a video screenshot.

Ability to download a video.

Ability to attach Constabulary icon to reports as a default.

2.6 Overview of OSINT-style browsers

This section provides an overview and discussion of several popular OSINT-style browsers and applications and has been placed here because it is a review of similar software. It must be stressed that both the latest version of *Forensic Acquisition of Websites* and *Hunchly* came out after OSIRT was released.

2.6.1 Oryon OSINT Browser

NB: This product, according to SourceForge (the hosting service), contains malware.

Oryon OSINT Browser (Oryon) is a free browser built using Chromium, making its look and feel much like Google Chrome. The browser itself makes use of a plethora of add-ons and extensions, largely available via the Chrome web store, which makes Oryon extremely feature rich. While Oryon boasts “more than 60 pre-installed add-ons” (SourceForge, 2017), this leaves the interface brimming with icons to the point where it is bordering on overwhelming.

Oryon’s overall design leaves the impression it is for those who are advanced computer users who can happily make use of and understand the needs of the add-ons. Oryon does not offer hashing capabilities for files, or report exporting.

2.6.2 Forensic Acquisition of Websites

Forensic Acquisition of Websites (FAW - <https://en.fawproject.com/>) is not a browser designed for conducting open source research, but it is a browser designed for law enforcement purposes and reviewed for this reason. Initially, this review focused on the free, and only version, of FAW that was made available in November 2014 and not updated until early-2017.

The 2014 version of FAW was very much a simple, visual website saving application whereby a user visits the page to they wish to capture and clicks the “acquisition” button. FAW would then download the contents of the website and place it within a directory

structure. All items acquired were date and time stamped and logged in an XML file. The browser did not offer anything beyond this capturing ability in this version.

FAW lay dormant for several years, but came back with an updated version in 2017 that replaced the main browser with CefSharp. While FAW was initially a free product, a tiered pricing model was adopted from FAW version 5. This saw a free, professional (499EUR) and law enforcement (499EUR) licences added. The paid for versions unlock, amongst other features, Tor and user-agent spoofing.

Figure 2.9 shows the main FAW browser and Figure 2.10 shows the directory of artefacts FAW collects.

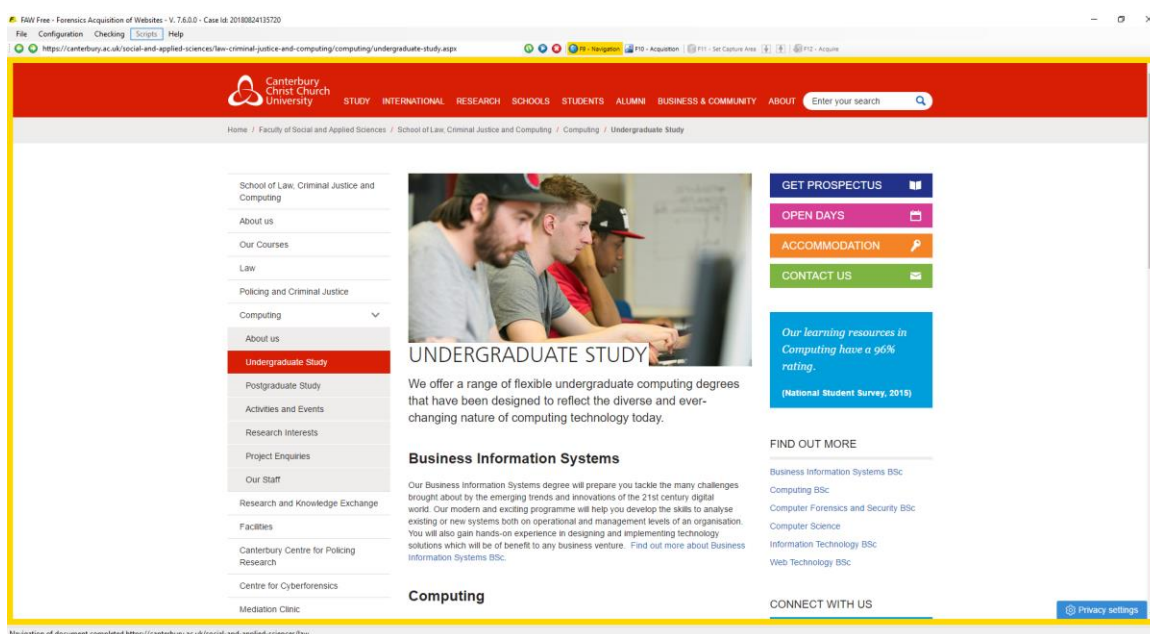


Figure 2.9 Free version of FAW browser

ImagesA4	24/08/2018 13:31	File folder	
Objects	24/08/2018 13:31	File folder	
Acquisition.log	24/08/2018 13:31	Text Document	1 KB
Acquisition.txt	24/08/2018 13:31	Text Document	3 KB
Acquisition.xml	24/08/2018 13:31	XML Document	7 KB
Acquisition_20180824132534_00002.pdf	24/08/2018 13:31	Adobe Acrobat D...	805 KB
Checking.faw	24/08/2018 13:31	FAW File	1 KB
Code.html	24/08/2018 13:31	Chrome HTML Do...	6 KB
Headers.txt	24/08/2018 13:31	Text Document	1 KB
hosts	18/03/2017 21:01	File	1 KB
Image.png	24/08/2018 13:31	PNG File	721 KB
SystemLogEvents.txt	24/08/2018 13:31	Text Document	0 KB

Figure 2.10 FAW objects directory

2.6.3 Hunchly

Hunchly (<https://www.hunch.ly/>) is a paid for extension for the Google Chrome browser. Hunchly costs \$129.99USD a year for a single license, or \$349.99USD per three licences with a 20% saving for more than three users. This review is based on the major update version of Hunchly released in April 2018.

Hunchly sits within the Google Chrome browser and automatically logs webpages when a user visits by placing them within a local case file; this is the big selling point of Hunchly. Case files can then be accessed by means of the “dashboard”, a separate application outside of the browser extension. Additionally, Hunchly contains features such as file attachments, automatic hashing, social media ID extraction and report exporting to both docx and PDF. Hunchly is a very capable addition to the OSINT browsing family, plus has the benefit of being cross-platform because it is a browser add-on.

However, there are several issues with using Hunchly that may impact its use, both from a legal and ethical perspective. In particular, the automated saving of every webpage visited creates an interesting dilemma. The immediate question: is it fair for law enforcement to make automated and automatic copies of webpages they visit without the need to make a conscious decision to do so? Previously, it was shown saving data using an automated means is a breach of Facebook’s terms and conditions, but there are ramifications further afield than just a website’s policy.

The process of “do first, ask questions later” is, in the opinion of the author, the wrong approach; particularly surrounding law enforcement’s collection of personal data. This chapter has shown that law enforcement need to take a careful and considered approach; one that focuses of necessity and proportionality. Is it then necessary and proportionate to automatically store carbon copies of all websites visited, without any interaction or acknowledgement from the investigating officer? The Data Protection Act (2018) explicitly states that personal data collection must be “Adequate, relevant and not excessive”, and debatably, visiting a webpage may be “relevant” to the investigation but arguably that maintaining a copy of every webpage is excessive, particularly with only having to optionally justify that capture with a note. Of course, users can simply delete these traces if not required, but then the audit trail is lost.

2.7 Human aspects and considerations when software engineering

Creating and integrating a tool in to law enforcement is a large undertaking and one filled with subtle nuances. As with any software, considering the human factor is important and spurred the adoption of the user-centred design method discussed in Chapter 3.

This section reviews and discusses the human aspects of development by reviewing the issues surrounding digital crime for UK law enforcement; this section then evolves into defining usability and reviewing usability studies surrounding law enforcement and how they can be applied to this thesis.

2.7.1 Digital crime, policing and surrounding issues for officers

Her Majesty's Inspectorate of Constabulary (HMIC) issued a report in 2015, last updated January 2018, outlining the importance of digital crime in policing. Data collection for the report took place over two months by visiting six police forces. The report uses examples of victim statements and how the police handled their reporting of the crime they had suffered.

The report stresses how integral technology is in modern society and how police must respond to the growing demand. The HMIC makes clear "[...] it is no longer appropriate, even if it ever were, for the police service to consider the investigation of digital crime to be the preserve of those with specialist knowledge." (HMIC, 2015b, p. 5) Further still, the report states:

"The public has the right to demand swift action and good quality advice about how best to deal with those who commit digital crime from every officer with whom they come into contact – from the first point of contact to an experienced detective." (HMIC, 2015b, p. 5)

Critically, HMIC is clear that all officers must have an understanding of handling and managing digital crime. The report from the offset sets out that regardless of job role, whether it is neighbourhood policing or anti-terrorism, officers must have the knowledge and skillset to police in the digital age, and that it is no longer a specialist's domain.

The HMIC does acknowledge, that to achieve the digital skillset required, those officers require to be trained in the technology they are meant to investigate. HMIC describe a

“mixed picture” (HMIC, 2015b, p. 12) of officers’ understanding surrounding digital crime, particularly highlighted by a response from an officer, “I am 46 years old. I do not have a computer; what do I know about Facebook?” (HMIC, 2015b, p. 30)

While officers recognised the need for being able to evidentially capture digital media, a “lack of confidence” of officers was found by HMIC. One officer acknowledged this by saying “[s]taff feel frustrated with their lack of ability to deal with digital investigations.” (HMIC, 2015b, p. 30)

The frustration and need for training was demonstrated by a neighbourhood policing officer who had received a complaint about a post on Facebook. The officer requested details from Facebook about the alleged offender, which was eventually declined by Facebook (HMIC, 2015b, p. 40). The officer then closed the case due to lack of evidence. HMIC uses this example as a “lack of awareness” surrounding the resources available to staff, with frontline managers noting “staff feel frustrated with their lack of ability to deal with investigations that involved social networking sites.” (HMIC, 2015b, p. 41) Another officer pointed out a “knowledge gap” whereby officers “do not know how to obtain the most basic of information [from social media].”

Some of the comments from officers surrounding investigations on social media are alarming, with one comment made showing a lack of willingness to understand social media “[w]hat do they [the victim] expect us to do about it? [...] I do not use social media; how am I supposed to investigate it?” (HMIC, 2015b, p. 42)

While the report does paint a seemingly sorry story, the report’s aim was to show the need for all officers to be digital media ready. In context, there were 123,142 police officers working in 43 forces as of 2017 (Home Office, 2017b), and training all these officers to have at least a minimal ability to obtain digital artefacts is a substantial undertaking; especially in times of austerity. Training new recruits the fundamentals of digital media investigations is, theoretically, straightforward as it can be introduced as part of the initial training they receive. For example, Kent Police now train all new recruits in open source research to at least level 1 as part of their 19 week training. However, that still leaves tens of thousands of experienced officers needing necessary training to achieve the goal of HMIC’s report.

The report shows there is a frustration among officers who do not have the skillset to conduct digital investigations on any level. What may appear as straightforward investigative lines of enquiry, such as obtaining a piece of information online, is plainly a struggle for some officers. These officers are likely to have extensive investigatory experience and have skills to offer in a digital world in that regard.

Ensuring officers have access to software tools they can use and training in how to conduct open source research is vital. From the perspective of the development of OSIRT, or indeed any software, usability is an aspect that requires more than simple consideration, it must be built-in with all officers in mind. The following sections defines and reviews ‘usability’ both generally and in the context of the needs of law enforcement.

2.7.1.1 What is usability?

There are several definitions of ‘usability’ all of which are based around similar concepts. The standard ISO 9241 definition specifies it as “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” (cited in Bevan, Carter and Harker, 2015). Bevan, Carter and Harker (2015) note this definition “has the benefit that is directly related to [...] user requirements”.

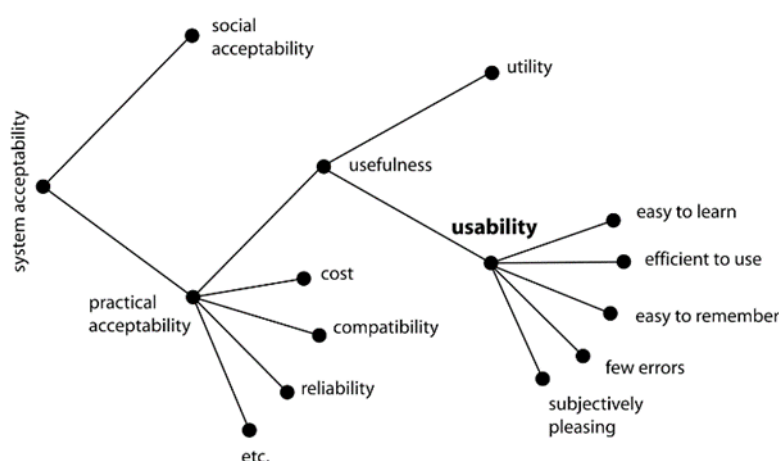


Figure 2.11 A model of the attributes of system acceptability (Nielsen, 1993)

Nielsen’s hierarchy of “System Acceptability” (Figure 2.11) is split into two categories, these are “social acceptability” and ‘practical acceptability’. Practical acceptability focuses on, arguably, the more obvious attributes a system should have and looks at the

cost, reliability and compatibility of a system. Ensuring a system is compatible with current needs of law enforcement and that the system is reliable are core attributes of a law enforcement-oriented system; or indeed any system. Additionally, Nielsen's (1993) system's acceptability hierarchy also sees 'utility', that is, does the system function and work as the user needs it to? This is seen as an extension of 'usefulness'. Without both utility and usability being considered, Nielsen (1993) argues, then a system is not useful to the user as either they cannot use it, or it does not do what is required of it.

Nielsen (1993) places usability more generally under a hierarchy of "System Acceptability" (Figure 2.11) and argues that usability is a characteristic of 'usefulness'. Nielsen splits usability beyond just efficiency and satisfaction and adds learnability (easy to learn), memorability (easy to remember) and error rate (few errors). Learnability, which Nielsen (1993) considers to be "the most fundamental usability attribute", measures how easy it is for someone new to start using the software. A user of a system that is easily learnable should spend less time completing a task once they become familiar with it. Efficiency measures the time a user takes to successfully complete a task. An efficient system means a task should be quick to successfully complete. Memorability measures how easy it is to use the system after a time gap between uses. Error rate measures the incorrect actions taken by a user that do not achieve the required goal; these errors may be unintended or intentional. Lastly, a system that is subjectively pleasing leaves the user feeling positive about using the system. Overall, the ideal system incorporates these five traits.

2.7.1.2 Review of usability studies surrounding law enforcement

HMIC's 2015 report show there is a diverse set of officers, some of which have never had or needed to conduct online investigations but are now in an environment where it is expected of them as part of their role as a police officer. These differing and new roles make finding literature which provides a robust and general-purpose method of creating software for law enforcement limited. Instead, a review of the literature surrounding usability of investigative tools was conducted.

Marcus and Gasperini (2006) conducted interviews with several frontline police officers in San Jose surrounding the use of 'mobile information display system' within police vehicles. Usability problems plagued the rollout and usage of the system, which prevented

officers from completing tasks and, frequently, being left not being able to communicate. The system was considered so dangerous that LEOs refused to use it as “it may endanger their lives”. The lack of end-user feedback gathered while designing, creating and testing the product ultimately lead to its demise, noted Marcus and Gasperini (2006), who further criticised the software by saying “usability did not seem to be a priority”. Marcus and Gasperini conclude that any frontline policing system should rely on a user-centred design approach.

This study shows the dangers of not considering the user when creating software. While the software within Marcus and Gasperini’s (2006) study is used in a moving vehicle, potentially under extreme stress, it still highlights the importance of including users throughout the development lifecycle. Had any of the software designers sat with officers, observed their routine and obtained feedback, this may have reduced the issues surrounding this system.

In extension to in-vehicle mobile displays, a study by Zahabi and Kaber (2018) discuss the number of police vehicles involved in crashes, and link this to cognitively demanding tasks officer’s conduct using their mobile display screens in the vehicles. While the authors do not claim the screen is the cause for these incidents, they do note the screens break several key usability principles of “using simple and natural dialog” and “minimising user memory load” (Zahabi and Kaber, 2018).

Bennett and Stephens (2009) reviewed the usability of the Autopsy Forensic Browser (Carrier, 2015), by using the cognitive walkthrough expert evaluation method. The authors identified “a number of usability issues” that violated basic usability principles and guidelines, one example was error prevention, which they found was “not considered as deeply as it should be”. The authors concluded that tools used by law enforcement are no different to any other mission critical piece of software, and if these tools were to fail it “could deny liberty to innocent persons”. While Autopsy Forensic Browser has advanced greatly since 2009, the sentiment of the authors’ work remains.

Nurse *et al.* (2011), looked at the usability of cybersecurity tools, and noted that not only is poor usability an issue in terms of “frustration and confusion”, but can lead to “inaccurate or inadequate configurations of tools”. A plethora of usability guidelines for cybersecurity tools were offered, many of which are good rules of thumb and follow key usability principles; the suggestions by Nurse *et al.* (2011) can be transposed to OSIRT.

Hibshi *et al.* (2011) also provided similar advice, which was based on survey results from 115 participants at the High Technology Crime Investigation Association along with eight interviews with law enforcement and industry digital forensic experts. Six user-interface issues were indicated by participants with pertinent issues highlighting “consistency”, an “intuitive interface” and reduced “information overload” (i.e. software presented information that is conceptually dense).

2.7.1.3 Summary of usability studies surrounding law enforcement

While a desktop application will not be operated under the pressures of a moving police vehicle, the advice and warnings offered by the literature shows the importance of following usability principles. A key theme throughout the literature shows that users should be a fundamental and central element to the development lifecycle and involved at every opportunity.

2.8 Chapter summary

This chapter provided a background and review of the legal, ethical and procedural issues law enforcement officials face when conducting open source research. In a background of uncertainties, where it is not plainly clear if law enforcement are allowed to look on Facebook without lawful authorisation, or even what software tools to use, one thing is clear: there is a need and requirement for a tool that aids LEOs both from a technical standpoint and a legal one.

The software used presently is both burdensome and costly, as noted by College of Policing trainers, and is prone to cognitive overload and human error. To abate this concern, the College of Policing requested a bespoke tool be created that combined all the abilities of the software range into one piece of software; this prompted the creation of Open Source Internet Research Tool (OSIRT).

The next chapter discusses the methodology and data collection methods.

3 METHODOLOGY

INTRODUCTION

Software engineering projects, particularly ones managed by an individual, require careful and considered planning. Given the nature of this research, and its output being used within law enforcement, there is a necessity for appropriate methods to gather and understand needs and requirements. This project goes beyond creating a tool, as it covers the entirety of the software development lifecycle, and establishes the ‘why’ and ‘how’ law enforcement conducts open source research.

This chapter covers the research design, software engineering methodologies for both prototype and release version and methods of data collection.

3.1 Research design

Chapter 2 showed the importance of involving users in the design and implementation of software. This section will focus on the research design, influenced by the guidance offered from the literature.

3.1.1 User-centred design

User-centred design (UCD) is a framework that focuses on the involvement of users in the design and implementation of software (Norman and Draper, 1986; Abras, Maloney-Krichmar and Preece, 2004). UCD’s overarching aim is for users to impact and shape the product, thereby ensuring it is created for their needs; it establishes the exact tasks and goals of the user symbiotically. UCD fulfils what the user actually needs, rather than attempting to shoe-horn or create a product they do not want (Abbras, Maloney-Krichmar and Preece, 2004).

ISO 9241-210 (Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems), the standard for UCD, does not specify any methods for data collection or assume any design process (ISO, 2010). Instead, ISO 9241-210 focuses

on “context of use” and provides six principles, seen below, to ensure the design is considered ‘user centred’:

- 1) The design is based upon an explicit understanding of users, tasks and environments.
- 2) Users are involved throughout design and development.
- 3) The design is driven and refined by user-centred evaluation.
- 4) The process is iterative.
- 5) The design addresses the whole user experience.
- 6) The design team includes multidisciplinary skills and perspectives.

(ISO, 2010)

Point 1 focuses on the context of use and centres on gathering the requirements, this is based on understanding the user, what they need and how the product is going to be used (Travis, 2011). Point 2 as it suggests, is ensuring that the users are involved in all aspects of development. This point is extended in the standard to note that the user must be “active”; i.e. ensuring they are engaged in the development process. Point 3 emphasises that users must evaluate the product at frequent intervals, and not merely at the end during the final acceptance stage of development. Point 4 makes explicit that to effectively utilise UCD, there will need to be changes within the system. These changes cannot be achieved using traditional software engineering methodologies such as the waterfall model. Point 5 the notion of ‘user experience’, will be discussed in section 3.1.1.1. While point 6 cannot holistically be applied to this study due to being managed by a lone developer, the experiences of the users and trainers are, in many regards, a part of the design team. Making use of their experiences and their understanding of what law enforcement need provide the different perspectives and multidisciplinary approach point 6 is looking to achieve. Table 3.1 shows how these principles were applied within this thesis.

UCD Principle	How it was implemented
1. Based upon explicit understanding of users	A prototype based on a specification from high-tech crime trainers at the College of Policing. Observations, System Usability Scale questionnaires, survey questionnaires and interviews were all used to obtain an explicit understanding.
2. Users are involved throughout design and development	Prototype and release versions are used and tested at the College of Policing. Users are actively encouraged to feedback. Observations also conducted at the College of Policing, along with interviews and SUS questionnaires.
3. Design is driven and refined by user-centred evaluation	Users complete a SUS questionnaire at the end of each RITES course and are encouraged to provide additional feedback which can be acted upon.
4. The process is iterative	OSIRT is in continuous development and follows an iterative and incremental methodology, making the process iterative in nature.
5. The design addresses the whole user experience	Actual users are at the forefront of development and can actively engage in shaping OSIRT. Users are encouraged to provide feedback through any channel necessary.
6. Design team includes multidisciplinary skills and perspectives	While development is undertaken by a lone developer, the skills of those law enforcement officials using OSIRT, by their nature, are multidisciplinary and provide differing perspectives. Additionally, the research also draws on academic knowledge by means of discussions with supervisors and the PhD panel.

Table 3.1 UCD principle and how it was applied within this thesis

3.1.1.1 User experience

User experience (UX) is summarised by Norman and Nielsen (no date) as something that “encompasses all aspects of the end-user's interaction with the company, its services, and its products”. Unlike traditional approaches seen in human-computer interaction, for example those that focused solely on task-oriented usability testing approaches (Hassenzahl and Tractinsky, 2006), UX encapsulates all aspects of the process: from usability to product support. UCD at its core is about the user’s experience, with ISO 9241-210 stating that UX:

“Includes all the users’ emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use.” (ISO, 2010)

3.1.2 Methods overview

While ISO 9241-210 does not specifically state methods for including users in UCD, Preece *et al.* (2002) (cited in Abras, Maloney-Krichmar and Preece, 2004) offer several data collection methods. These include interviews, questionnaires, observations and usability evaluations. Table 3.2 shows the methods used and their respective chapters

where their results are discussed. Section 3.6 discusses these methods, how they were applied and their advantages and limitations.

<i>Stage</i>	<i>Purpose</i>	<i>Description</i>		<i>Chapter</i>
Prototype	Requirement understanding and surrounding issues of open source research	Method	Analysis method	
		Interviews	Thematic analysis	4 & 6
		Observations	Descriptive analysis	6
		Questionnaires	Statistical analysis	4
		SUS	Statistical analysis	6
		Software engineering methodology		
		Throwaway prototype		5
Release	Establish OSIRT usage and for continual building	Method	Analysis method	
		Interviews	Thematic analysis	8
		Observations	Descriptive analysis	8 & 10
		Questionnaires	Statistical analysis	8 & 9
		SUS	Statistical analysis	8 & 9
		Expert usability evaluation method		
		Cognitive walkthrough		8
		Software engineering methodology		
		Iterative and incremental		7

Table 3.2 UCD methods of collection and their stage of collection

3.2 Overview and discussion of software engineering methodologies

Traditional development life-cycles followed a sequential model typically seen in the waterfall model (Royce, 1970). The waterfall model (Figure 3.1) follows a fixed, linear pattern where each phase of the life-cycle is completed before moving onto the next. The waterfall model is particularly well applied to systems whose complexity is low and the requirements well established (Laplante and Neill, 2004). In order for the waterfall model to be effective, the analysis team are required to be “nearly clairvoyant” argues Burbick (1998), as “there is no room for mistakes”. While large organisations may still opt to use the waterfall model (IEEE Software, 2018), its use in this development will not be productive given the need to take an iterative approach for UCD.

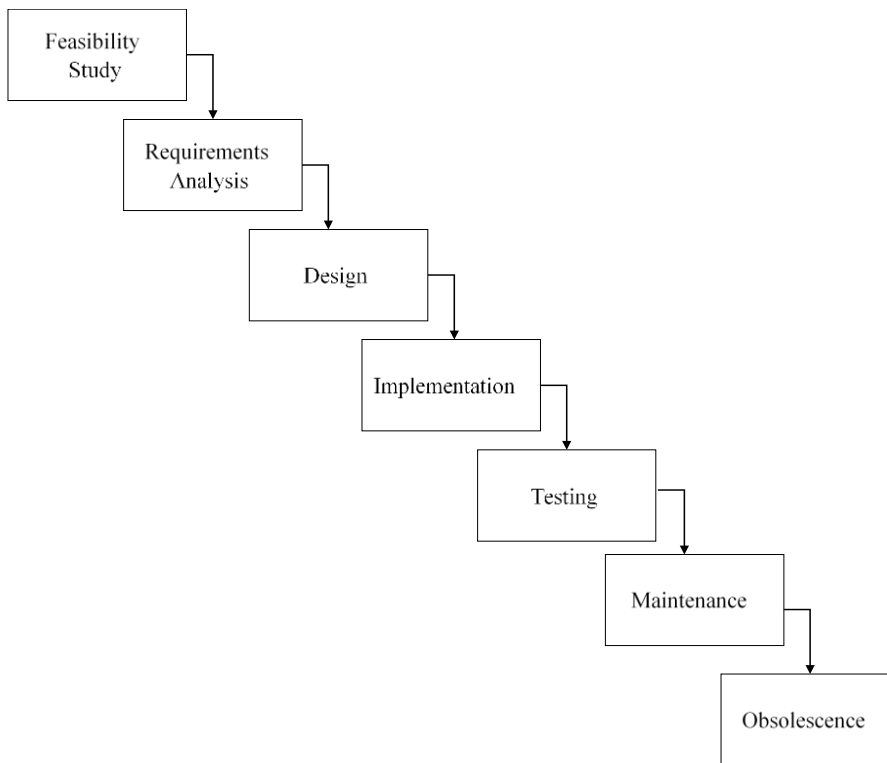


Figure 3.1 Waterfall model adapted from Royce (1970)

In contrast to sequential models are cyclical models, more commonly known as the spiral model (Boehm, 1987). Cyclical models (Figure 3.2) generally follow the same phases as seen in the waterfall model but each phase is not necessarily completed before moving onto the next stage, allowing for phases to be revised in an iterative fashion. This is useful when customer requirements change; providing flexibility for adjustments. After several cycles through the spiral, the final product is ready to be released. Ideally, each cycle is shipped to the customer in order to garner feedback and refine the requirements (Boehm, 1987). A major advantage of the spiral model over its sequential counterpart is its flexibility. Feedback can be obtained at any point in the process from users, providing the ideal software engineering methodology for integrating UCD. As new advances are made in technology and feedback gathered from users, these can be seamlessly implemented into the system (Burbick, 1998; Sommerville, 2006; Pressman and Maxim, 2014).

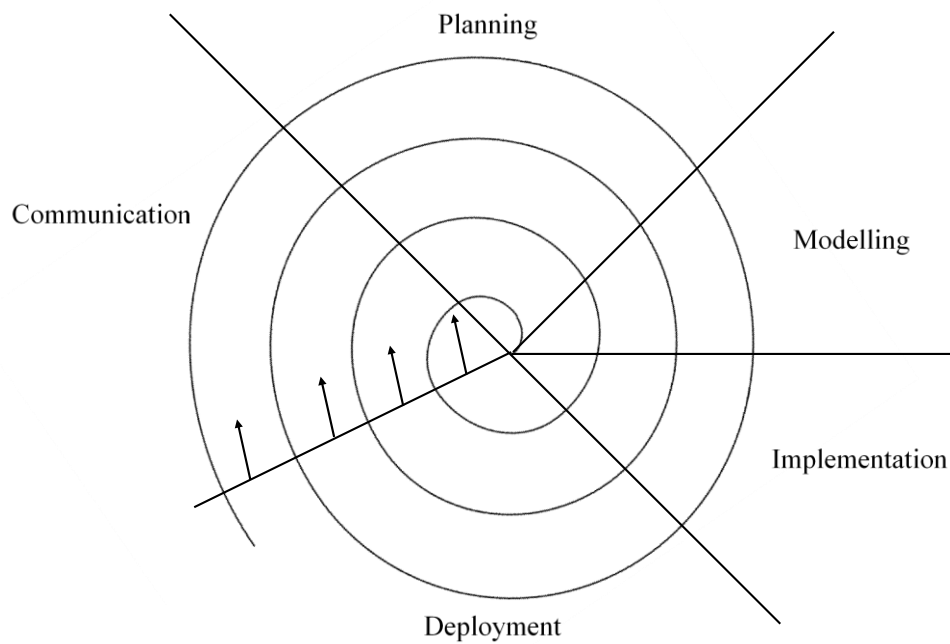


Figure 3.2 Spiral model adapted from Boehm (1987)

Between the sequential and cyclical methodologies, cyclical provides the opportunity to iteratively and incrementally build a system, obtain feedback from consumers and update it. For this reason, cyclical approaches are the better choice for the development of the OSIRT's use of UCD.

3.3 Prototyping method

Prototyping in software engineering is not, relatively speaking, a modern concept. Naumann and Jenkins (1982) provide a definition for prototyping in an 'information system' as early as 1982

“[...] a system that captures the essential features of a later system is the most appropriate definition [...] of a prototype. A prototype system, intentionally incomplete, is to be modified, expanded, supplemented, or supplanted” (Naumann and Jenkins, 1982)

Modern definitions, such as those by Pressman and Maxim (2014), also highlight that a prototype is to serve as “the first system” and is to be used “for identifying software requirements”. Figure 3.3 represents the development life-cycle of software prototyping.

Prototypes are not without their pitfalls, though. Pressman and Maxim (2014) warn that users may see the prototype working and assume it only requires “a few fixes” to produce a finished product, when the reality is likely to be different. Continuing to develop a throwaway prototype is problematic, as it has not been designed with maintenance or extensibility in mind (Pressman and Maxim, 2014). It is also important to remember that when developing a throwaway prototype, design decisions, such as algorithmic efficiency, are often omitted in order to generate the prototype (Sommerville, 2006; Pressman and Maxim, 2014). It is important to review those design decisions, as inefficiencies or maintenance issues will sneak into the main product if they are not carefully considered.

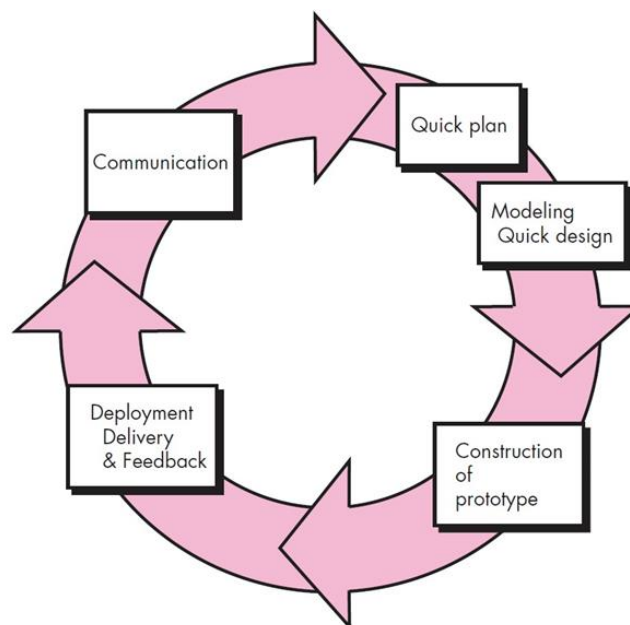


Figure 3.3 Software prototyping methodology (Pressman, 2014)

3.3.1 Types of software prototyping

Software prototyping falls under the umbrella of either ‘evolutionary prototyping’ or ‘throwaway prototyping’ (Figure 3.4). Evolutionary prototyping ends with a complete, functional and maintainable system (Crinnion, 1992; Vijayasarathy and Butler, 2016). When using evolutionary prototyping, the system is gradually developed allowing for the

product to be adapted if the requirements change (Floyd, 1984; Carr, 1997). Conversely, throwaway prototyping allows for ‘experimental’ systems to be created and tested with the end user, and is particularly useful when the requirements are not well known, or require further clarification is needed (Sommerville, 2006; Vijayasathy and Butler, 2016). Throwaway prototypes still produce a ‘working’ system (Sommerville, 2006), but the system will be missing functionality, or the functionality may be implemented in a less than efficient manner (Pressman and Maxim, 2014; Sommerville, 2006). Throwaway prototyping keeps in line with the notion of a ‘prototype’, and that is to test and gather requirements for a system. For those reasons, a ‘throwaway’ prototype was chosen for the early stages of OSIRT, particularly as the requirements from the College of Policing were broad in nature.

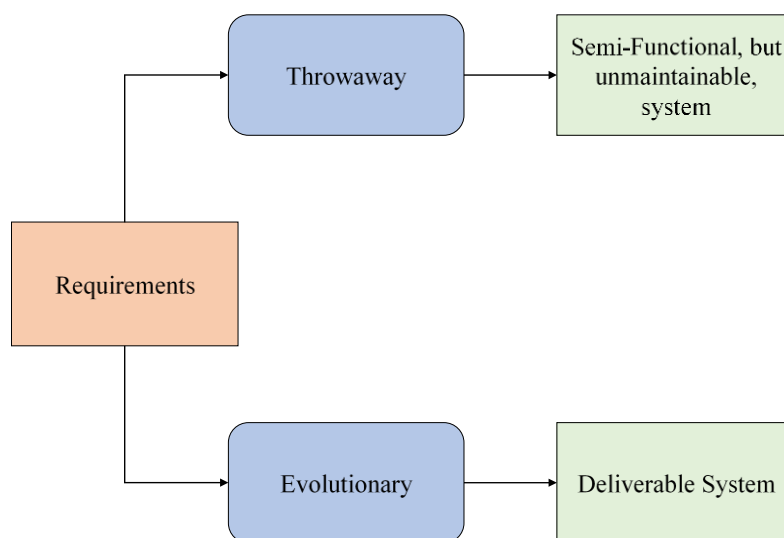


Figure 3.4 Types of software prototyping

3.3.2 Prototype methodology summary

This section looked at the prototyping method of software engineering, which provided a rapid way of constructing a software application in order to obtain feedback and to obtain a clearer understanding of the requirements; a critical aspect of the prototype.

The College of Policing provided an opportunity to attend a RITES course, allowing for data collection to occur.

3.4 Release version software engineering methodology

With the prototyping method providing a solid foundation and a better understanding of the overall requirements, the release version of OSIRT required a methodology that provided a system that can be maintained and enhanced.

The element from the prototyping method that was the most useful was communication and feedback from users; which is a fundamental aspect of UCD. Without feedback and communication, OSIRT could not go beyond the specification provided, which would have been based solely on the interpretation of the specification by the developer. Given how useful feedback was during the prototype, frequent communication continued to be fundamental to the chosen software engineering methodology for the release version of OSIRT.

Discussion of software engineering methodologies in section 3.2 showed that the traditional approach of the waterfall model (Royce, 1970) is too rigid and inflexible with no communication during critical development phases. Other methods, such as the incremental and iterative approach, were unfeasible during the requirements-gathering prototype stage due to the loose specification. Given that the requirements were better understood after the prototype's creation, an iterative and incremental approach suitably fitted the release version. Not only is a suitable method from a software engineering viewpoint, but also from the perspective of UCD and the core requirement of ensuring the process is iterative in nature.

3.4.1 Iterative and incremental approach

Iterative and incremental approaches have historically formed part of software development lifecycles, and can be traced far back as 1957, where “half-day” increments were used for NASA’s Mercury Project (Larman and Basili, 2003; Williams and Cockburn, 2003). This iterative and incremental approach provided a working system from the start, but future versions provided new features, enhanced features and bug fixes. Figure 3.5 visualises how a system grows, with each concentric circle representing an iteration and increment.

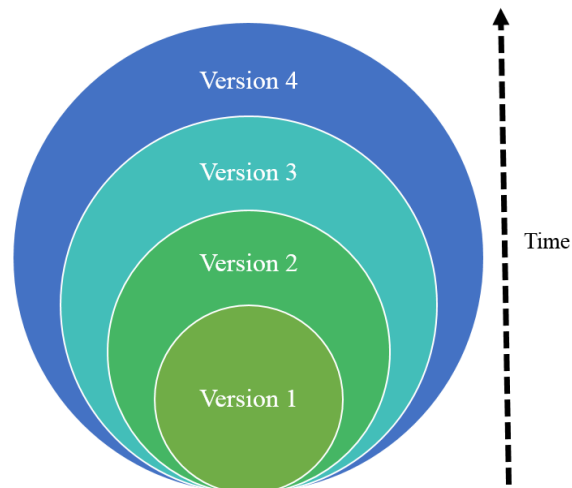


Figure 3.5 Visualisation of how software grows using an incremental and iterative approach

The incremental and iterative method is used by software development teams today to engineer large scale solutions where delivering a complete system to the client is not feasible or practical (Larman and Basili, 2003). Pressman (2014) uses the example of building a word-processing application incrementally. File management and document editing is the first increment, followed by spelling and grammar in the second increment and page layout in the third increment. From this example, the first version is a “core product” (Pressman, 2014) with only the fundamentals addressed, with later versions adding features. Later versions may add features not originally known, and enhancements may have grown or been provided by the users’ feedback and evaluation of the core product.

3.4.2 Distribution and beta testing

OSIRT required a distribution method for users, and this was fulfilled by <http://osirtbrowser.com>. The website contained the latest, live version of the software and a beta version if available. After the feedback received from the prototype, OSIRT was provided in a ‘portable’ and ‘installable’ format. The portable version does not require installation and can be extracted onto a pen drive to run OSIRT. The installable version, as the name suggests, needs to be installed onto the system. The installable version is useful to IT services to distribute OSIRT, and for those who want a simpler user experience when using OSIRT.

The release version is the most stable and is the recommended version for install. The beta version contains all the newest features, but is potentially unstable due to its beta status.

3.4.2.1 Beta versions and testing

Beta versions of OSIRT were used and tested on the RITES course, as well as being released under a 'beta' heading on osirtbrowser.com. This method, particularly the RITES course, provided an opportunity to obtain feedback and find any bugs from actual law enforcement officials, without the need to push OSIRT as a release version. Each increment was pushed out for beta testing, before becoming a release version.

3.4.3 Managing as a lone developer

While the development of OSIRT obtains frequent feedback, it is still only maintained and developed by a single developer. Single developer projects are inherently more at risk than projects with a team, as every aspect of the project is managed alone. Requirements gathering, development, testing and communication with users still need to be conducted for a project to be successful. The development of the OSIRT prototype was a process to gather requirements; where the development was pieced together with little need to consider future maintenance. The release version of OSIRT required more consideration around design decisions and time management. A system was put in place for managing feature requests and fixing bugs, and additionally as required, communication channels were open and available. To ensure the project was effectively managed, a Kanban (Ladas, 2008) approach was followed. That is: bugs and feature requests were all logged and prioritised using a template from Vertex42 (Vertex42, 2014) (Figure 3.6). The Kanban approach can aid in preventing feature-paralysis, where a single feature becomes the sole focus at the expense of the overall system and ultimate goal.

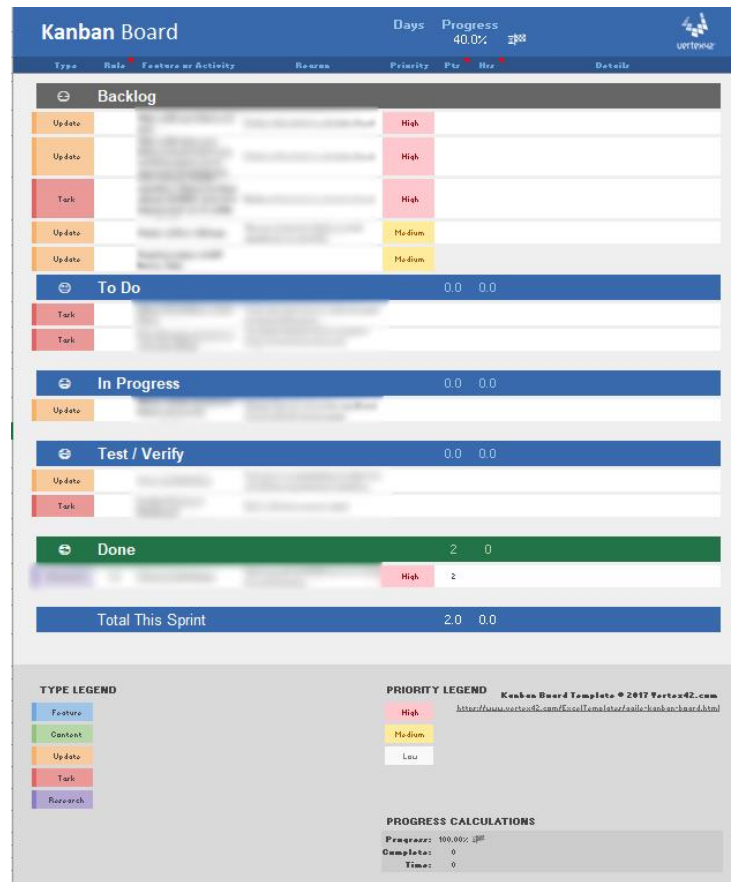


Figure 3.6 Kanban board spreadsheet with OSIRT-related tasks (aspects redacted for confidentiality). Kanban board from Vertex42.com (2014).

3.5 Sample and sampling

The target population for OSIRT is law enforcement officials who need to conduct open source research. The size of population is not known, and even if it was, obtaining data from every law enforcement official would not be possible. Instead, a sample of the target population was used. There are various types of sampling, of which a selection is provided in Table 3.3.

Opportunity	Sample is based on convenience of who is available at the time; it is quick and convenient. Limitations are that it can generate unrepresented samples, as sample only represents who was accessible at the time the research was undertaken.
Random	Everyone within the target population has an equal chance of being selected, this provides the best way of ensuring representative samples. A limitation is that selecting the population and ensuring everyone has an equal chance is impractical for large population sizes.
Stratified	Sample is divided into categories, and a representative sample is then proportionally selected. A limitation is that selecting the population and ensuring everyone has an equal chance is time consuming. Additionally, sub-categories are required to be proportionate given the same size.
Volunteer	Sample has chosen to be part of the study (self-selecting). Limitations are it can generate unrepresented samples in that those who volunteer to participate are likely to be known to the researcher, or those who are already familiar with what is being researched.

Table 3.3 Types of samples and their description

The bulk of data collection came from participants of the RITES course at the College of Policing, and this falls into the ‘opportunity sample’ category, as the participants were available at the time.

While the opportunity sampling method is one of convenience, in this instance it is also ideal. The target population is well represented at the RITES course; it is the purpose of the course, in fact. The course provided a diverse range of participants, with different skills-levels, different job roles and attendance from different constabularies. All these views and experiences allowed for making OSIRT a well-rounded tool for law enforcement.

3.5.1.1 Participant details

Demographic details of participants are placed in the appropriate sections of this thesis where the data is discussed. However, a general overview of the typical RITES course participant is provided here for clarity.

The RITES course is designed for those who are required to use the web as part of their investigations, with participants of the course having some experience in using a computer and other software productivity tools. Typically, participants are Detective Constables, but the course is open to all law enforcement agencies and officials, so often has Police Constables, Detective Sergeants, Inspectors, Analysts or even those from the Royal Air Force in attendance. It is possible that participants may already have good open source research knowledge but require the certification and accreditation the College of Policing provides. However, the majority of those in attendance are in new roles and are novices in conducting open source research. Additionally, they are likely novices in digital investigations altogether.

3.5.2 RITES course access and limitations

While the RITES course provides a rich source of data from experienced officers, participants are on the course to be taught how to conduct open source research. This meant that interaction and data collection with the participants became secondary to the overall course aims, so as to not hijack the course for the sake of data collection. This meant observations were passive, questionnaires were distributed at the end of the day/course and interviews conducted at the end of the day.

3.5.3 Data protection, confidentiality and consent

To protect participant's privacy, specific dates have been removed from this thesis as to not make participants identifiable. Interview and observational data is not used *ad-verbatim*; for example, pauses, inflections and complex tics (palilalia) have been removed as to not make the participant traceable. Unless consent has been given to do so, participant names are also removed and job roles may have been given generic titles such as "detective constable".

Ethical clearance for the ‘prototype’ research was granted on 28th June 2015, and ethical clearance for the ‘release’ research was given on 22nd June 2016. Consent was obtained from all participants before any research activity began, with all participants being fully briefed. All participants signed either a physical participation sheet or confirmed via a consent button/checkbox if data collection was conducted online. The participation sheet outlined their rights as a participant and the reasons for the study.

3.6 Data collection

As seen in Pressman’s (2014) software prototyping method (Figure 3.3), feedback and communication are integral elements of the prototyping method. This section discusses the methods of collection for the prototype, and why they were chosen.

3.6.1 Data sources, methods and rationale

Runeson and Höst (2009) stress the importance of collecting data from several sources “in order to limit the effects of one interpretation of one single data source” (Runeson and Höst, 2009). Runeson and Höst’s (2009) reasoning is that conclusions drawn from several sources are stronger than one based on a “single source”. Several methods of data collection will be utilised as it is an aspect of UCD.

Lethbridge, Sim and Singer (2005) (cited in Runeson and Höst (2009)), in their study on data collection techniques in software engineering, state that collection techniques can be split into three levels, as seen in Figure 3.7.

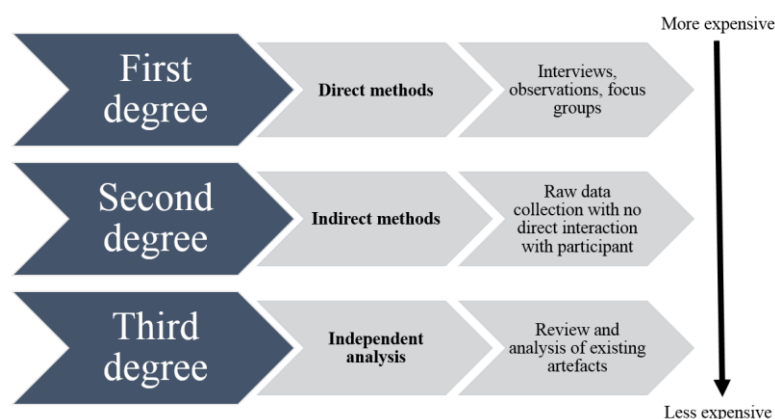


Figure 3.7 Data collection techniques for software engineering, with examples of how to collect data (based on Lethbridge *et al.* (2005))

While first and second degree methods are more expensive, they provide the advantage of the researcher being able to control the ‘what, where, why, which and how’ of the data collection. The quality of the data is under control and specific to the domain for which it is intended. Third degree methods do provide a low cost, but control of the data is lost and this may impact the overall quality (Runeson and Höst, 2009). Given the access to the cohort on the RITES course, first, second and third degree methods were integrated.

3.6.1.1 Observations

Observations were used to investigate how tasks were conducted when using OSIRT, for both prototype and release, and how OSIRT was interacted with. Observations are a first or second degree method, depending on implementation and the level of interaction the researcher has with the group (Runeson and Höst, 2009). Observations offer an advantage in that they are immersive and provide first-hand experience, thus providing a deeper understanding of the problem. A disadvantage of observations is that they can provide a substantial amount of qualitative data to analyse, so it is important to establish what data to collect.

Observational approaches fall into either ‘high interaction’ or ‘low interaction’ with the level of interaction having an impact on the participants (Runeson *et al.*, 2012).

	High awareness of being observed	Low awareness of being observed
High interaction by researcher	Researcher is seen as “observing participant”	Researcher is seen as “normal participant”
Low interaction by researcher	Researcher is seen “only as researcher”	Participants observed via video recording

Table 3.4 Approaches to observation (Runeson *et al.*, 2012)

Observations while at the College of Policing are expected to straddle the top two rows in Table 3.4, as interaction will be high due to the interactive nature of the RITES course. Covert observation was not possible, typified by video recordings, as not only were the College of Policing’s classrooms not designed for such an approach, but the College of Policing would not approve of covertly observing sessions.

For the prototype, an emphasis on OSIRT and how it was interacted with was the focus of the observations. Additionally, any comments made surrounding the conducting of open source research were also noted. As participants were on a course designed by College of Policing, observing pre-defined tasks was not feasible. Instead, observations of the participants using OSIRT were noted by means of pen and paper using the “observations, quotes and inferences” technique (Dumas and Redish, 1999, p. 292) during the prototype. Each participant was given a unique ID and an individual observation sheet (Table 3.5). Given the interactive nature of the RITES course, it was not possible to observe all participants at all times. Instead, a ‘best effort’ approach was taken when observing.

Participant ID: T1		
Participant Actions	Quotes	Inferences
Clicked on static screenshot icon	“I can’t find the fullpage screenshot option”	Fullpage screenshot option needs to be clearer

Table 3.5 Example "observations, quotes and inferences" sheet.

It was discovered during the ‘prototype’ observations that maintaining an observation sheet for individual participants was extremely difficult, time consuming, and often filled with repetition. Later observations made use of a daily observation sheet, where general observations were noted instead of individual sheets; this still followed the same ‘observations, quotes and inferences’ technique.

Observations were also used to collect data during a RITES course to study the effectiveness at training open source research and OSIRT, for a study in chapter 10. Appendix E shows the complete observation template for this study.

3.6.1.2 Interviews

Interviews³ provide an opportunity to explore views and experiences on a specific topic (Gill *et al.*, 2008.), and offer a better understanding of a particular phenomenon in comparison to a purely quantitative method, such as a survey (Gill *et al.*, 2008; Runeson and Höst, 2009). Interviews are best selected when there is a need for “detailed insights” from participants, as they are the best means to obtain the participants attitudes and motivations regarding a particular aspect (Oppenheim, 1998).

Interviews can be split into unstructured, semi-structured and structured (Robson, 2011). Semi-structured interviews were chosen, because they provide a more adaptable approach in this instance. In a structured interview, questions are rigid with little to no opportunity of deviation (Edwards, 2013). While unstructured interviews can take many forms (Jamshed, 2014), they generally follow the “flexibility is the key” rule (Edwards, 2013), as unstructured interviews are considered to be conversational in nature (Gray, 2009). The middle-ground is a semi-structured interview; it provides a fixed set of essential questions (Wildemuth, 2009) as seen in structured interviews, but also the ability to probe responses and carry on the conversational thread with additional questions as one would expect to see in unstructured interviews.

Interview guides were established beforehand with a set of topic areas and questions to ask the participants. Silverman and Marvasti (2008) note that the interview guide is there to aid the memory of the researcher and is not to read verbatim to the participant; this approach allows for additional discussion or questions to emerge.

As with using any survey-style method, bias can be problematic; in this case, interviewer bias. Interview bias is caused by the effect the interviewer had on the answers obtained, and this can be simply due to the presence of the interviewer themselves, or “social desirability bias”. Social desirability bias “is the systematic underreporting of undesirable

³ In law enforcement, an ‘interview’ has an explicit meaning and possible negative connotations to officers. It was made clear to those participating that these interviews were academic and they were free to withdraw at any time.

attitudes or behaviour” (Leeuw, Hox and Dillman, 2012). There are also limitations surrounding representability of interviews, as noted by Qu and Dumay (2011), but by using a combination of methods, such as questionnaires and observations, this can be minimised.

3.6.1.3 Questionnaires

Questionnaires are a second degree method and are a traditional, efficient method of data collection, as the researcher is not required to be present during their administration. Questionnaires can obtain both quantitative and qualitative data, depending upon the type of questions asked (i.e. open or closed). Questions can generate diverse opinions from respondents, which can then lead to generalisability of any conclusions derived from the responses. Responses are gathered in a more standardised way, particularly when compared to interviews (Milne, no date).

Limitations surrounding questionnaires are the potential for non-response, particularly for self-administered questionnaires, the consequence of a low/non-response rate may effective generalisability of the results. Additional limitations are that respondents may embellish their answers in order to provide a ‘socially acceptable’ response; this is known as social desirability bias.

Survey questionnaires were used early in the prototype phase to gather what tools law enforcement officials used to conduct open source research. Questionnaires were also distributed later after the release version of OSIRT had been available for 18 months.

3.6.1.4 Personal communication

With observations, interviews and questionnaires providing a consistent approach to data collection, personal communication via e-mail, telephone calls and face-to-face meetings were a rich source of feedback to enhance OSIRT. E-mails are naturally recorded and archived, but telephone calls and face-to-face meetings were only documented if there was an issue specifically involving OSIRT (e.g. a bug report) using the Kanban method. The information gained from these communication methods were used solely to enhance OSIRT as a software product, or to inform further ethically cleared research. Unless informed consent was given to use these communications, they are not discussed within this thesis.

3.6.2 Measuring usability

Two evaluation mechanisms in determining system usability can be seen in expert and user-based evaluations. Expert evaluations are conducted by usability experts, that is, those who have specialised knowledge in the field of usability. User-based evaluation methods focus on feedback from the target user group. The next section will briefly review how usability was measured based on these approaches in this study. The application of these methods in the broader sense must be put into the context of a project where a full set of usability evaluations was not feasible in terms of cost. In this case techniques are used that do not necessarily require physical access to users and do not use real-life, on-the-job scenarios. Access to users is challenging – on their training course there may be insufficient time and afterwards these busy users are spread across the country. The latter as real usage of the system is involved in law enforcement activity, which has significant issues relating to civilians observing the data collected.

3.6.2.1 Expert evaluation methods

While there are several methods for conducting expert usability evaluations, the two common methods are cognitive walkthrough (Wharton *et al.*, 1994) and heuristic evaluation (Nielsen, 1993). Heuristic evaluation for the best results requires the tester to be both a domain expert (i.e. someone who is skilled with the system) and knowledgeable in the field of usability. The expert assesses the user interface of the system against a set of rules-of-thumb: the heuristics. As an example, these include visibility of system status, error prevention and recovery, and help and documentation. In general, there are several approaches to the actual process of using the heuristics, e.g. system-oriented vs. task-oriented. A cognitive walkthrough follows a more guided approach and assumes the users will learn the system by exploring. To create a cognitive walkthrough, a set of tasks are created and performed by an evaluator. After each action is completed, the evaluator will ask themselves the following four questions for a suggested user profile:

1. Will the user be trying to achieve the right effect?
2. Will the user discover that the correct action is available?
3. Will the user associate the correct action with the desired effect?
4. If the correct action is performed, will the user see that progress is being made?

If any question is answered negatively, this may highlight an issue with usability and the evaluator provides a plausible ‘user story’ as to why the system failed, possibly followed by a suggestion as to how it may be improved. For example, for question 3 an answer may be “No, because the associated icon is ambiguous”.

3.6.2.2 User-based evaluation questionnaires and the System Usability Scale

A variety of questionnaires exist that measure usability. These can range from no-cost solutions, such as the System Usability Scale (SUS) questionnaire, to those requiring a licence, such as Software Usability Measurement Inventory (SUMI) (Kirakowski and Corbett, 1993). Tullis and Stetson (2004) compared five usability questionnaires and discovered that SUS and Computer System Usability Questionnaire (CSUQ) (Lewis, 1995) were the most reliable for small sample sizes of 8-12 users. SUS is a smaller questionnaire with only ten questions, compared to nineteen for CSUQ. SUS was selected for its size, reliability, validity and well-established usage, which will be explained in more detail below.

SUS is a system agnostic usability questionnaire, and despite describing itself as a “quick and dirty” usability scale, SUS has become a prevalent tool when assessing usability. In addition to working well with smaller participant numbers (Tullis and Stetson, 2004), Bangor, Kortum and Miller (2008) claims SUS is a “highly robust” tool that gives both valid and reliable results. Lewis and Sauro (2009) discovered that SUS, in addition to measuring perceived ease-of-use, also provides a means of measuring learnability and usability.

SUS is made up of ten questions, with odd numbered questions phrased positively and even numbered phrased negatively.

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.

7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Responses to the questions are on a five-point Likert scale that ranges from 1 ('strongly disagree') to 5 ('strongly agree'). To calculate the overall SUS score, odd numbered items have one subtracted from the participant's response score and even numbered have the participant's response score from five; all values are now scaled from 0-4. All converted responses are then summed and multiplied by 2.5 to give a score out of 100.

Scores range from 0 to 100, where a higher score is indicative of a usable system. Sauro and Lewis (2012, p. 202) studied SUS results from 446 studies, and found the average SUS score to be 68. Additionally, Bangor *et al.* (2009) introduced a SUS grading scale from A-F to make disseminating the SUS result simpler; where a grade 'A' signifies a highly usable system and a grade 'F' represents a system in need of immediate usability enhancements.

While SUS is traditionally administered immediately after usability tests, Sauro (2016) notes that SUS can be used as a stand-alone and standardised measure for collecting user's usability thoughts. This is how it is applied within this research.

3.7 Data analysis

This research adopted a mixed-methods approach that provided diverse data both qualitative and quantitative in nature. This section discusses the analysis methods used for both approaches.

3.7.1 Qualitative data analysis

Qualitative data analysis and collection in this study followed an iterative process, as analysis on existing data offered new insights. This means data collection did not follow a linear approach, but rather, collection and analysis fed into each other in an iterative fashion (Figure 3.8). The iterative approach to analysis also meant it could integrate into

both the UCD method and the iterative and incremental software engineering methodology.

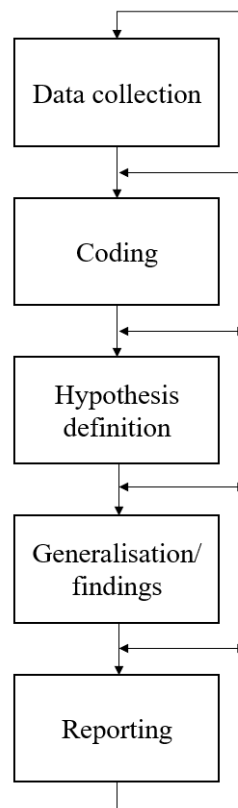


Figure 3.8 Steps of data analysis (Robson, 2002)

Runeson *et al.* (2012) stress the importance of a “structured approach” for qualitative analysis in software engineering projects. Robson (2011; cited in Runeson *et al.*, 2012) offers four approaches to structured analysis: Immersion, editing, template and quasi-statistical (described in Figure 3.9). Runeson *et al.* (2012) recommend either an ‘editing’ or ‘template’ approach for software engineering studies. An editing approach was adopted for this study, as this is “where the participants’ own words are used as text that can be edited into a form more suitable for reporting” (Robson, 2011).

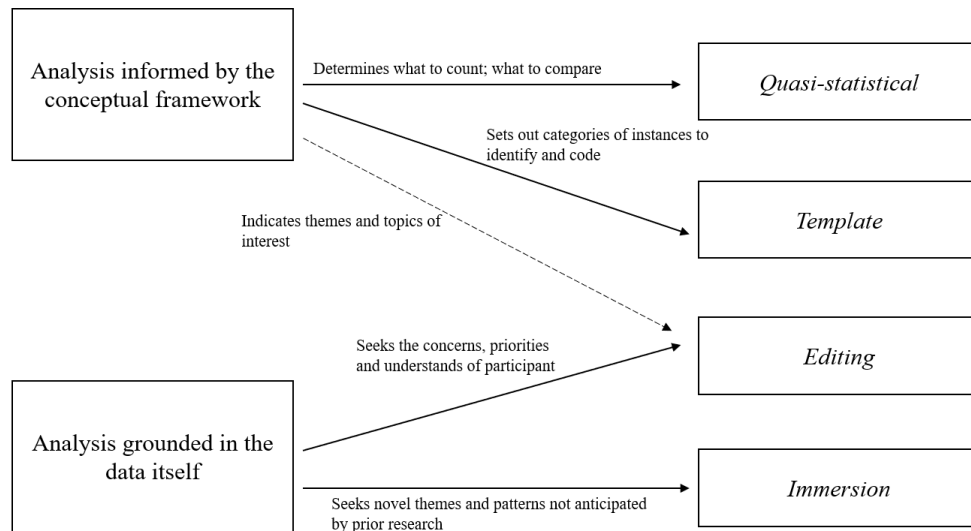


Figure 3.9 Representation of approaches of qualitative data analysis visualised by Taber (2013)

3.7.1.1 Thematic analysis

Thematic analysis as a method provides the function to organise, analyse, describe and report themes (Braun and Clarke, 2006; Nowell *et al.*, 2017). Thematic analysis has the advantage of being flexible and providing rich, complex data (Nowell *et al.*, 2017), while not being as complex to integrate as other qualitative analysis methods (Braun and Clarke, 2006). King (2004) also notes the benefits thematic analysis has for those who are unfamiliar with qualitative analysis techniques, as it can be learned and applied relatively quickly. Thematic analysis is not without its limitations, and while it is a flexible method, Halloway and Todres (2003) argue this can lead to inconsistency and incoherence within the analysis.

Braun and Clarke (2006) specify six phases of thematic analysis, seen in Figure 3.10. During the initial stages, codes (early themes) are generated manually and these are descriptive in nature. Coding was manually conducted using a pen and paper. Once initial codes are discovered, they are placed into groups, or categories, that are meaningful. Finally, a 'theme' is created based on the reviewing of the categories in the previous step.



Figure 3.10 Stages of thematic analysis as adapted from Braun and Clarke (2006)

3.7.1.2 Descriptive analysis

Observations were used to find issues surrounding user experience when using OSIRT and to obtain feedback and suggestions on OSIRT. This data feeds directly into OSIRT's development and is largely objective in nature. There is little need, or desire, for in-depth analysis of these observations as a descriptive summary provides what the issue was and why it was an issue.

3.7.2 Quantitative analysis

3.7.2.1 Statistical analysis

Quantitative data from the questionnaires was statistically analysed using SPSS to generate frequencies and obtain descriptive data. Additionally, SPSS was also used to analyse prototype SUS results. Further SUS results were calculated and analysed using Sauro's (2012) "SUS Guide & Calculator Package", a dedicated tool to analysing SUS results.

3.8 Validity and reliability

Validity is an important aspect of any research and is something that must be considered before any data is collected. A way to avoid the most serious threats to validity and mitigate the lesser ones, is to use a robust methodology. Furthermore, to improve validity and reliability, this research integrates the following approach, as recommended by Robson (2012).

3.8.1 Triangulation

Triangulation is a powerful tool in empirical research as it strengthens its validity (Runeson *et al.*, 2012). Triangulation comes in many forms, but it essentially means to take more than one approach towards what is being studied, ultimately generating a comprehensive picture (Runeson *et al.*, 2012; Heale and Forbes, 2013). Triangulation is especially important for qualitative data given its broad, rich, but less precise nature (Heale and Forbes, 2013).

Denzin (1973; cited in Runeson *et al.*, 2012), says there are four different types of triangulation seen in Table 3.6.

Triangulation Type	Definition	How it was applied
Data Triangulation	Using more than one data source or collecting the same data at different occasions	Data collected from various sources. RITES courses, e-mails, interviews and questionnaires.
Observer Triangulation	Using more than one observer.	RITES course trainers would observe and feedback throughout all courses.
Method Triangulation	Combing different types of data collection methods.	Several data collection methods are integrated as described above.
Theory Triangulation	Using alternative theories or viewpoints.	Continuous reflection and being open minded to new ideas.

Table 3.6 Data triangulation methods and their application

3.9 Chapter summary

This chapter discussed the research methods and the data collection methods. What should be taken from this chapter is that the choice of methodology and data collection methods were driven by the selected software engineering methodologies, which both require communication and feedback as part of their life-cycles. The use of both qualitative and quantitative methods of data collection provided an overview of the requirements for OSIRT, and why those requirements are in place; whether they are legal, ethical or procedural.

4 A STUDY OF LEGAL, ETHICAL AND PROCEDURAL ISSUES FACED BY LAW ENFORCEMENT

INTRODUCTION

Chapter 2 showed that law enforcement officials face a legal and ethical minefield when it comes to conducting open source research, and this is due to differing advice, guidelines and laws. Additionally, it shows that the College of Policing faces issues in the training of open source research based largely around the number of tools required to effectively conduct open source investigations. To assist with minimising tools and investigator overload, the College generated a specification for an open source research tool; which generated the Open Source Internet Research Tool (OSIRT) project.

This chapter looks at the results from 22 interviews conducted with law enforcement officials, and discusses the legal issues they face when conducting open source research. The second section of this thesis discusses the design and implementation of the OSIRT prototype. The chapter is split into the results of the legal, ethical and procedural issue interviews, then moves on to the results of an exploratory questionnaire surrounding the tools used when conducting open source research.

4.1 Interview questions

Interview questions focused around the effectiveness of legislation, policies and guidelines, along with social media website terms and conditions. The list of questions that were asked in relation to this chapter, along with the probing questions, are shown below.

- Does current legislation provide you with what you need when conducting open source research?
 - How does legislation, such as RIPA, integrate with open source research when it pre-dates modern social media?
 - How does current guidance aid you in conducting open source research?
- Discuss conflicting guidance with ACPO and Surveillance Commissioner
- Do you appreciate the concern of the public when it comes to collection of personal data?
- Is there an expectation to privacy when someone has a public profile on social media?
- Are you aware of policies, such as YouTube's, where you are either not allowed to collect personal information or you must tell people when you are doing so?
- How forthcoming are social media platforms when responding to data requests?

4.2 Participants

22 semi-structured interviews were conducted, ranging between 15 and 45 minutes. 18 participants were interviewed face-to-face during the College of Policing's five-day RITES courses. These interviews include the two trainers from the RITES course. Additionally, four participants from various constabularies gave interviews over the phone. Of the 22 participants, 21 are serving LEOs with one being retired after 30 years of law enforcement service and is now a trainer at the College of Policing. The serving LEOs have been in policing roles ranging between 6 and 22 years, with all of them having had some experience in conducting open source research.

4.3 Interview results and discussion surrounding legal and ethical issues

4.3.1 The 'grey area'

The participants were quick to highlight conflicting guidance when conducting open source research. Fourteen spoke about the guidelines from ACPO that state if the information is public, it is acceptable to collect and unlikely to require authorisation under RIPA, but this contrasted with what the Chief Surveillance Commissioner views that a RIPA authorisation is required for "repeated viewings". The "repeated viewings" element

of the Surveillance Commissioner's advice was a sticking point from thirteen participants, and it was not unusual to hear them ask for clarification, such as "What does that [repeated viewings] actually mean?". A participant said, and this was a sentiment carried by twelve other interviewees, "It may seem obvious that it means more than one, but what if I take a screenshot? What if I view their profile once, then again 6 months later? Does that constitute repeated viewings?" This contrasting advice often led the LEOs to be "cautious" or "careful" when conducting open source research on social media, ensuring to the best of their understanding that processes and guidelines are correctly followed.

This necessity to act cautiously not only stems from guidance, but also from the relative newness with conducting open source research. Eight participants mentioned that there have been few stated cases, that is, where someone has been prosecuted using evidence gathered from open sources, so the guidance provided to LEOs is "essentially someone's opinion" as "definitive answers" are scarce. One participant noted the processes in which social media artefacts are captured could change at any time, as "[...] one day a judge might say 'actually we're not happy with the way RIPA has been interpreted to get this data' and your entire process has to change!"

Participants were asked how current laws, such as RIPA, integrate into conducting open source research on social media. Seventeen participants explicitly mentioned the difficulty that surrounded conducting open source research using social media because current laws do not easily fit, with ten participants explicitly mentioning a "grey area", when it comes to current legislation. A Detective Constable with twenty years' service noted

"The difficulty I have with the current legislation is that it's a bit of a grey area with regards as to what's guidance and what's legislation. I think that's the crux of the matter. There's a necessity to clarify the situation so we know we can do and what we can't do."

Issues surrounding RIPA were frequently noted, with fifteen participants acknowledging that it is out of date and was made for a different era of communication, such as telephones, voicemails and letters. Participants would frequently comment about legislation being "outdated" and that it does not "fit around social media".

The ‘grey area’ is evidently a cause for concern to LEOs; laws that are meant to aid law enforcement are proving confusing, arguably a hindrance, in the modern era. Even new legislation does not provide the guidance required when conducting open source research on social media. While policies and guidelines are helpful, contrasting viewpoints from ACPO and the Chief Surveillance Commissioner compound the issues surrounding social media investigations. From those interviewed, there is a distinct need for clarity within the laws, whether that comes from new legislation or case law.

4.3.2 Playing catch up

Fourteen interviewees mentioned that they believe law enforcement was “playing catch up” with the criminals. “Sometimes it feels we are two to three years behind the bad guys and much of that boils down to having antiquated laws that do not meet our investigative needs” said one Detective Constable. A Detective Sergeant shed some light on these comments saying, “[...] it does really feel as though we are playing catch up, technology changes so rapidly, particularly these days, it makes keeping up difficult”. When the fourteen participants were asked if “playing catch up” was due to current legislation all of them replied affirmatively, “Yes, it certainly makes it challenging.”, one officer said. Even new powers available to law enforcement, such as those under the Investigatory Powers Act (IPA), are “already getting out of date” according to one Detective Sergeant because legislation “struggles to keep up”.

Legislation was not the only aspect that made participants feel they were lagging behind; seventeen participants also mentioned the technological standpoint in the fight against crime, with a Detective Sergeant saying:

“The bad guys have access to the latest tech, and they really do not care how they use it. Sometimes it can take us months to catch up to them, only for them [criminals] to change their methods and we’re back to square one. We have to follow an established process, not only legally, but as a moral duty to the public. They [criminals] do not care”.

Even with new legislation in place, the ever-evolving dynamic of social media and the Internet makes it feel outmoded and unconnected almost before it has time to be useful. With newer technologies being released on a daily basis, it stands to reason that laws surrounding technology will share the same fate and fall into the pit of obsolescence.

4.3.3 Ethics

Participants were asked if they appreciate the concern of the public when it comes to collection of data from social media. “There’s this misconception that we’re just sitting on Facebook all day harvesting reams of data...” said a Detective Sergeant, “...We don’t collect data willy-nilly, it has to be controlled, the correct authorities need to be in place.”. A Detective Sergeant made it clear that any officer who goes outside of their authority will be reprimanded, as it is taken “extremely seriously” with the lead trainer at the College of Policing noting that ignoring guidance is “at your peril”. A participant noted that these privacy concerns raised were a very “poignant question” and one that “still doesn’t really have an answer, and probably never will”. However, as one Detective Constable noted, “Under RIPA, everything is measured by proportionality and necessity. So, if it’s necessary for us to investigate someone who has been saying racist things on Facebook, a proportional response to that would be to, you know, look at their Facebook profile”. Fourteen other participants mentioned either proportionality and/or necessity when discussing ethical issues surrounding social media.

Four participants highlighted a disparity between law enforcement and privately-owned companies, and asked hypothetical questions whether private companies were acting ethically and in the interest of the public. One particular comment of note that summarises this notion came from a Detective Constable:

“If I went onto Google and searched you, I can find your home address and that’s being hosted by a private company. Now I can pop that [address] into Google maps and zoom right down to your house, I can have a good look in your garden from above, or a view of your paint-job on ‘street view’- and that’s open to everybody. Private companies are using this information to sell, to make money and it feels as though no-one questions it as much as we get questioned. We’re the police, we’re interested in saving lives, not making money”

Ten participants were asked if the social media users who have public profiles should have any expectation of privacy. This question in particular was a cause for thought by the participants, six participants answered “no”, but would qualify the “no” with the guidance set out by ACPO. Two participants answered “yes”, but noted this is why it is important to get RIPA authorisations in place.

Participants acknowledged members of the public could perceive what they were doing as unethical, but law enforcement is tasked with the responsibility to ensure public safety, and that can come with a compromise. The disparity between how a private company collects open source data versus law enforcement is an interesting point; one that requires further research.

4.3.4 Social Media companies

When participants were asked “Are you aware of policies, such as YouTube’s, where you are either not allowed to collect personal information or you must tell people when you are doing so?” Five participants said they were aware of these terms and conditions, but were permitted to ignore them due to their authority. “Yeah, I have read the ‘T and Cs’ [terms and conditions] of YouTube, but that’s really meant to apply to people looking to harvest data for nefarious purposes. We have a moral and legal duty as police to protect life and ensure safety and if that means breaking some website’s terms, then so be it.” said one officer. Nine interviewees pointed out the perils of breaking the terms and conditions of Facebook. One such term disallows using a false persona to create an account; one officer had first-hand experience where their account was deleted due to breaking these terms:

“I was part of this closed group on Facebook for [redacted – group name] using a [false persona] account. Then Facebook came along one day and deleted it. All those contacts, all that potential intelligence was gone. Obviously, you document as you go, so we had something, but it was a big loss”.

Depending on the social media platform, requests for further details can be a cause of frustrations to LEOs. Twelve officers said they have had to wait longer than “three months” for a response from some platforms, while other platforms never even replied “It can be hit or miss when it comes to making a request. Some [social media] sites are very helpful and as quick as they can be. Some take a long time, but sometimes it’s worth the wait. Others simply do not respond at which can be very frustrating.” a Detective Constable said. Seventeen participants said they had experienced, at least once, no response from a social media website when a request for further information was made. “We understand the pressure is on social media not to reveal private information to the police or agencies or whoever, and we do support that, but sometimes that denied or

ignored request could make or break an investigation” said one Detective Constable. When participants were asked if social media needed to do more for LEOs, fifteen responded affirmatively, usually with the caveat that it was only some platforms needing to be more approachable.

The social media companies themselves can play a large role in the outcome of an open source investigation, either through terms and conditions or data requests by LEOs. Clearly it can be frustrating to a LEO who has managed to gain access to a group on Facebook, only to have that link severed without any notice. However, social media websites have a duty to ensure privacy for all their users and that comes at a cost of inconvenience to LEOs.

4.4 Legal and ethical issues summary

With people living an ever more public life online, open source research gives law enforcement a needed tool in its arsenal. While judicious use of open source research has the potential to prevent or manage major atrocities, such as the London riots in 2011, care must be taken to ensure investigations stay within the confines of the law. Even if the information is considered to be open and freely accessible, it is not boundless. The debate of privacy versus security will inevitably be at the forefront of any discussion when collating information about individuals; it is, in the author’s opinion, a debate that can never be fully satisfied. Despite its irresolution, it is important such questions are asked, maintaining the checks and balances required in law.

Law enforcement within the UK currently directs open source investigations using legislation that pre-dates elements of the modern Internet, such as social media platforms. Permissive legislation, such as RIPA, is brandished for justification when conducting open source research that may be considered invasive but it begs the question, is that enough? How are LEOs to lawfully conduct open source research on social media using legislation written before it was a dominating influence in modern life? This perception of obsolescence is seemingly stirred as well when interviewed LEOs seem to feel they are behind criminals due to their industry’s inability to change as quickly as technology progresses.

Through no fault of their own they are asked to navigate a minefield of legislation, terms and conditions, case law, and to some extent, personal opinion, in order to gather evidence. Unsurprisingly, it breeds confusion and flags questions that have no precedence within guidance. Furthermore, it sparks the ever-looming negotiation between a person's right to privacy versus their capability to overtly broadcast their actions to the world, highlighting the debate of ethics and legality.

While apparent this chapter raises more questions than it can answer it also attempts to bring into the conversation the voice of those most affected by this deliberation. Open source research policies cannot be neglected or be expected to continue functioning in a grey area of interpretation. For the LEO to do their job and for the public to understand what parameters their right to privacy are defined by, the legal system must move forward and set out a structure. Allowing such ambiguity to persist breeds an uncertainty that should not exist within law.

4.5 Current tools and practices and questionnaire

To establish current practices and tool usage, a short questionnaire was distributed by Russell Taylor, lead trainer on the RITES course, to various LEOs via Google Forms within the UK.

4.5.1 Respondents

Respondents had a range of experience of specifically conducting open source research; the levels of experience ranged from less than one year, to over six years. The respondents ranked from Police Constables to Inspectors. 20 responses were received from 12 constabularies. In addition to establishing current tool usage, an exploratory question asked what LEOs would like to see from an all-in-one open source research tool.

4.5.2 Results

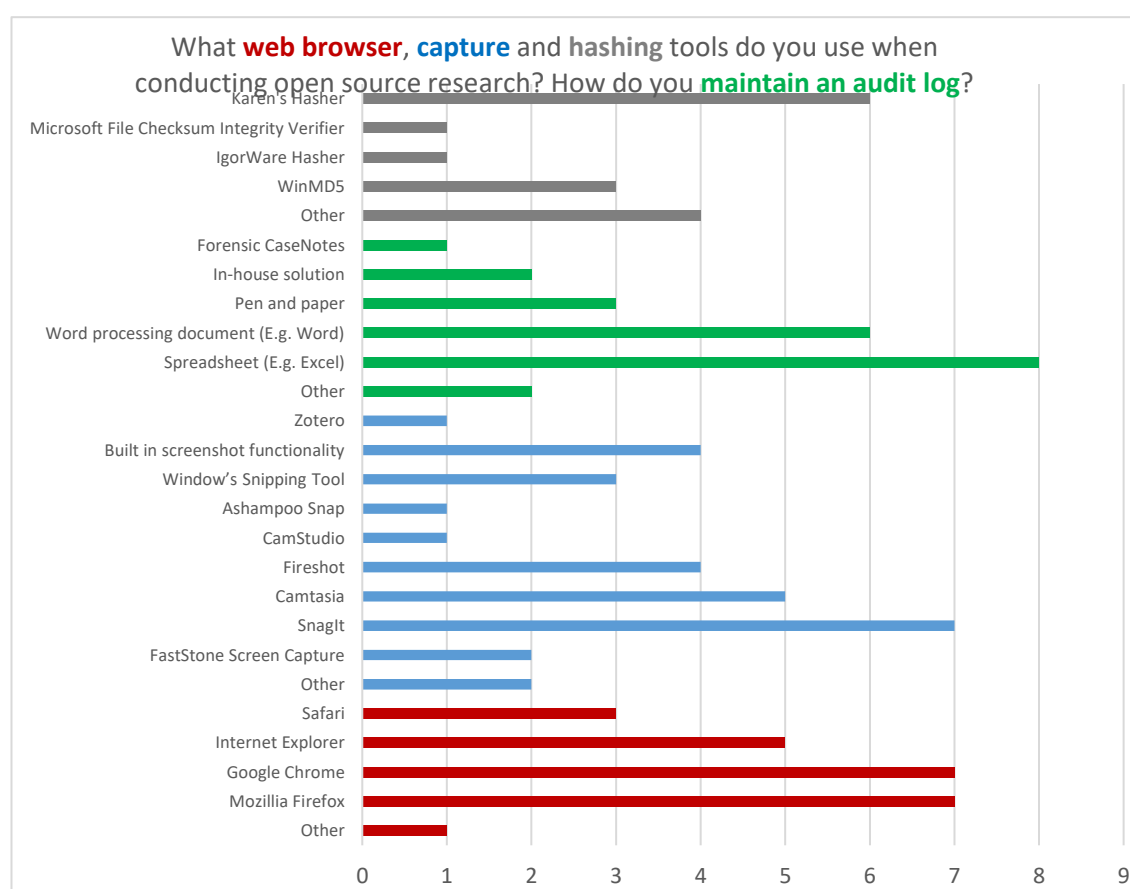


Figure 4.1 Tools used by law enforcement to conduct open source research

Tool usage, as seen in Figure 4.1, shows a variety of different tools used by law enforcement and is a theme that runs throughout this thesis.

Respondents were also asked “Does the cost of some tools prohibit you from being able to use them?” thirteen responded “yes”. A question then asked “I am more inclined to use a tool if it is free of charge.” twelve responded “yes”. Additionally, four respondents said they do not need to use a hashing tool.

While the majority of respondents said the cost of some tools are cost prohibitive (65%), out of twenty-eight capture tools highlighted in the questionnaire, excluding those marked ‘other’, free tools were used just thirteen times in comparison to the fifteen paid for tools. That said, the licencing cost of some of those paid tools are of a reasonable cost (such as SnagIt and Faststone Screen Capture) and may explain why they are used by respondents.

Cost restrictions are something that need to be considered; not only because the College of Policing's specification states that a software licence must be no more than £30. In times of austerity, it is typically public services that will see budget cuts, and the need to save money becomes imperative. Low cost software that combines and performs the function of several different tools designed specifically for law enforcement is beneficial for them. The prototype of OSIRT will be free, as it is designed to gather requirements, but a decision on whether OSIRT should remain free will need to be decided after the results of the prototype.

4.5.3 Questionnaire summary

The overwhelming result from the exploratory questionnaire was that there is no standardised toolset when conducting open source research. This is due to tool selection being the forces, or even the individuals, decision as to what application to use. It is clear from the respondents; however, a variety of tools are used to achieve their goal as there is an overlap in functionality.

4.6 Chapter summary

Evident throughout the review in chapter 2, and now backed-up from law enforcement officials with interviews and questionnaires from this chapter, there is a desire and a need for a tool like OSIRT to ensure officers attempt to stay within the ambiguous line of law and policy. What is not ambiguous, however, are the number of differing tools which vary in price and quality. A combination of these tools that aid law enforcement in streamlining open source research is the next step and discussed in chapter 5.

5 OSIRT PROTOTYPE DEVELOPMENT AND IMPLEMENTATION

INTRODUCTION

Chapters 2 and 4 showed that law enforcement officials face a legal and ethical minefield when it comes to conducting open source research, and this is due to differing advice, guidelines and laws. Chapter 2 showed that the College of Policing faces issues in the training of open source research based largely around the number of tools required to effectively conduct open source investigations. To assist with minimising tools and investigator overload, the College generated a specification for an open source research tool; which generated the Open Source Internet Research Tool (OSIRT) project.

This chapter discusses the technical implementation of the OSIRT prototype and is closely linked to chapter 6.

5.1 Justification, discussion and issues surrounding prototype implementation

This section will look through early thought processes, issues and proposed solutions to creating a bespoke system to be used by law enforcement.

5.1.1 Focus of control

An important consideration in the design of OSIRT was the notion of ‘control’. Typically, when users use a web browser, they are free to perform many actions and are offered a variety of options and tools to assist in conducting those actions. In a system like OSIRT, due consideration must be given as the system is designed to capture digital artefacts; this means OSIRT must control, or have a capacity to control, user actions. Principle 3 of the

ACPO guidelines for obtaining digital evidence state that an audit log must be maintained (ACPO, 2012); the implication of this is that all actions a user takes must be maintained for audit trail purposes. Web browsers, by their nature, are designed to aid users in surfing the web in whatever way they wish. Users can visit any number of sites, save an unlimited number of images, and download an arbitrary amount of content. For a regular web browser user this is not an issue, as they can save files where they want and have no need to maintain an audit trail. For OSIRT, however, every notable action a user takes must be logged; a right-click and ‘save image as’ becomes a larger problem. Now there is need to handle the entire process, from context menu handling, to download, logging and hashing. Preventing a user doing something they should not is part and parcel of being a software developer but overriding and changing the way the web browser works, without significantly impacting their workflow or challenging their normalised interactions with a browser, is a significant challenge.

Sauro (2013) notes that “familiarity breeds content”, with the implication “a user’s prior experiences can impact perceptions of usability”. OSIRT will fashion itself around the style of a traditional web browser, not dissimilar to the notion of a ‘metaphor’ as proposed by Carroll, Mack and Kellogg (1988). While metaphors themselves do not necessarily reduce complexity of a user interface, rather they provide the user with ‘familiarity’ as hinted at by Sauro (2013). While OSIRT requires to be more than just a web browser, using a browser metaphor as a starting point in its design will provide the user with ‘what they know’. For this reason, OSIRT will be considered successful if it can integrate itself into current police workflows and provide a seamless transition between what the user is familiar with.

5.1.2 Artefact capture – to note or not to note

As discussed in the review of legal, ethical and procedural issues in chapter 2 and the interviews with law enforcement officials in chapter 4, officers face a plethora of issues surrounding capture of open sources; one of these issues surround the justification of captured artefacts. ACPO guidelines, specifically principle 3, in regards to digital evidence state the need to maintain an audit trail, these guidelines also extend to ACPO/NPCC guidelines for conducting open source research. As previously discussed, Internet artefacts are inherently transient and what may be available one day for an officer may not be tomorrow, so the audit trail is one not of replication, but of process. The

overarching purpose of the audit trail for open source research is for disclosure (e.g. for the CPS) and/or examination purposes (e.g. by the OSC). There is no time frame for when an audit examination can occur, or when the captured artefacts may be used as evidence in court, so it is important for officers to be able to justify the decisions they have made and why they have made them. These justifications are not only necessary from a perspective that they may not remember six months later why they captured an artefact, but they may also need to justify that decision under scrutiny.

After much discussion with trainers at the College of Policing, OSIRT adopted an enforced ‘note’ rule. That is, captures are to have a note section and that note is not optional; an officer must justify their decision to capture. While this may appear inconvenient, it enforces several key guidelines and laws. While only partially enforced in this prototype, as the decision was made late-prototype, but enforced more robustly in the release version.

5.1.3 Date and time

When it comes to evidential collection, all the details matter; none more so than the date and time an artefact was collected. The issue is, on a globally connected network, what defines date and time? This section will discuss ‘date and time’ as a notion on the Internet, the usage of the investigating officer’s collection device as the source for the date and time, and time stamping methods using timing protocols.

It is not unrealistic that an officer’s machine could have the wrong date and time set. If an incorrect date and time were to be disputed, in theory, this should be simple enough to verify the error and make an adjustment accordingly. For example, if it is found an officer’s machine is 32 minutes and 12 seconds behind the ‘official time’ then all times in the log can be adapted. This is both inconvenient and avoidable. The question then becomes: how can we obtain a ‘good’ time?

One solution is to use an Internet service that provides this functionality. However, all that is happening is shifting of the responsibility away from the officer’s machine to obtain the date and time to, essentially, somebody else’s machine. Can we know, with certainty, that machine’s date and time is more reliable? Arguably, if it is a dedicated time

server, then it probably is more reliable, but unlike being able to physically check an officer's machine, using somebody else's server that is uncheckable may raise questions.

To abate those concerns, the use of the Network Time Protocol is recommended. NTP (<https://www.ietf.org/rfc/rfc5905.txt>) is an application layer protocol that has been part of the TCP/IP stack for decades, being one of the oldest protocols. NTP is used to synchronise computer clocks and is highly accurate, with its usage being from tested and reliable NTP servers. Once a reliable NTP server is found, it is just a matter of obtaining the date and time from one of them. There are a handful of reliable servers/services that provide the date and time as a service using NTP, with <http://www.pool.ntp.org/en/> being the popular choice. Given the nature of the Internet, there is then a potential problem of network latency. It takes less than a couple of microseconds to check the computer's time but it could take seconds to obtain the time from [pool.ntp.org](http://www.pool.ntp.org/), depending on which server was chosen and the complexity of the algorithm used to obtain the time. Although the scenario of a slow response is unlikely, as [ntp.org](http://www.pool.ntp.org/) has thousands of servers dotted all around the world, it still requires consideration.

Additionally, to use NTP, an officer's machine will require access to UDP port 123, and some network administrators could have this port blocked, meaning no access to a date and time service. In that instance, a fall back on to the computer's clock will have to suffice.

To summarise, there are three solutions to date and time:

1. Use the computer's date and time. It is fast and can be physically verified if needed. Conversely, the date and time could be incorrect.
2. Use an NTP server and get the date and time. It is reliable in the sense date and time is almost certainly always correct. Negatively, it could be slow to obtain the date and time. Plus, UDP port 123 could be blocked by network administrators. There is also no reasonable to verify that the time on the server is correct.
3. Use a mixture of both. Run a check when OSIRT starts to see if the computer's date and time falls within what the NTP server says (perhaps an arbitrary limit such as +/- 10 seconds). If it is out of kilter, warn the user and tell them OSIRT will be using NTP servers to get the date and time. If it is fine, then continue to

use the computer's date and time as there is no reason to make an external call if the user's machine is already synchronised with an NTP server.

5.2 OSIRT prototype creation and implementation

5.2.1 Early design

Given the relatively broad nature of the specification, a meeting was set-up with the College of Policing trainer from the RITES course to gain a better understanding of the requirements. During discussions, the trainer provided an example of how artefacts are collected, stored and managed during the RITES course, as seen in section 2.1.7.1. This methodology involved a web browser with screen capture add-ons, a spreadsheet for maintaining an audit trail and a hashing tool. Cases were managed by means of nested directories, where collected artefacts were placed into their respective directories (e.g. screenshots were placed in a 'screenshots' directory). Any websites visited were manually placed into a spreadsheet and with the date and time appended.

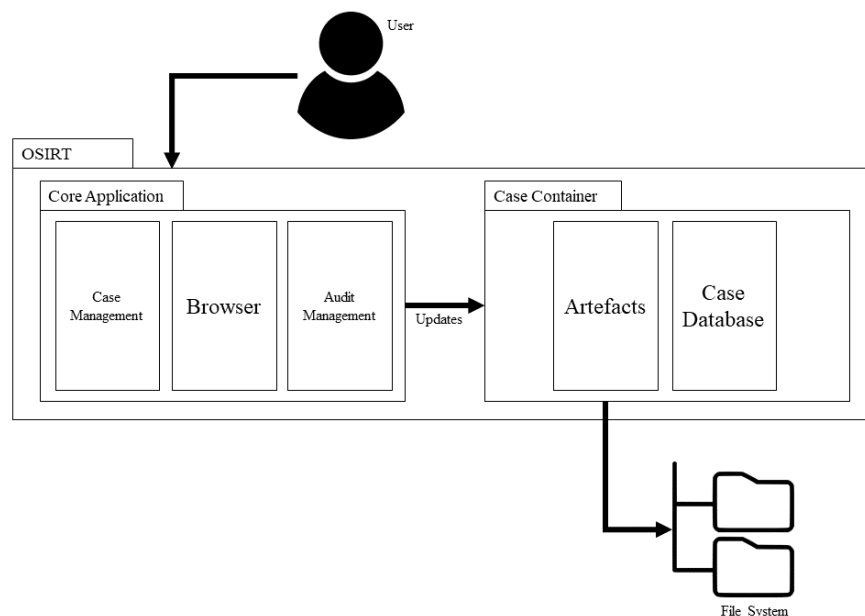


Figure 5.1 High level view of the system with sub-functionality

Early designs (Figure 5.1) focused around this methodology used by the College of Policing, with OSIRT being split into two core functional areas. The first area is the 'core application' and contains case management, the browser and audit log functionality. The

second area is the case container which houses artefacts (such as screenshots) and the case database; the case container interacts heavily with the operating system's file system.

The following sections will look over the technical aspect of the prototype, with each functional area discussed, and why it was implemented in such a way.

5.2.2 Case management

5.2.2.1 Case container

The case container (Figure 5.2) is a nested directory structure that contain downloads, images, videos, and attachments. A case database (`case.db`) within the container houses the audit log, as well as `settings.xml` that houses any user settings on a per-case basis. A brief description of each folder is provided below.

- *attachments* – Stores all external files uploaded to OSIRT.
- *downloads* – Stores all files downloaded via OSIRT. The `source_code` directory stores the saved source code from a website.
- *images* – Stores screenshots and snippets. 'Batchsnap' images are screenshots that have been obtained without physically visiting the page. The 'scraped' directory stores all images that have been downloaded from a website.
- *report* – Stores exported reports
- *settings* – Stores additional settings about the case

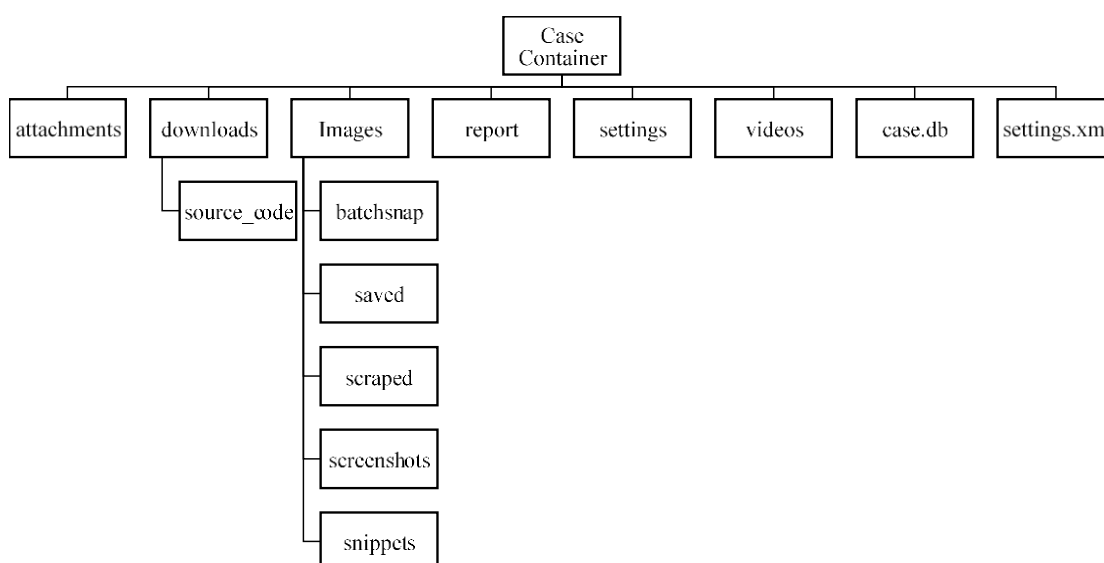


Figure 5.2 Case container directory structure

5.2.2.2 Case database

The case database is a SQLite database (<https://www.sqlite.org>) and was chosen as it is compact and lightweight. Additionally, the .NET Framework provides a fully-featured library to interact with the database. The prototype's case database had three tables: case details, evidence collection and case notes as represented in Figure 5.3.

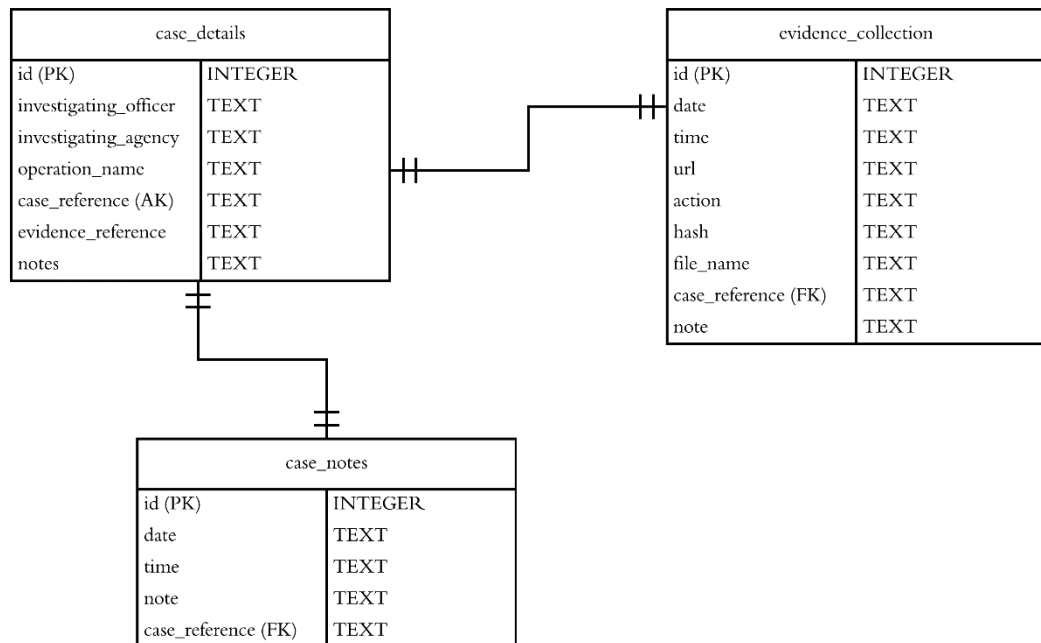
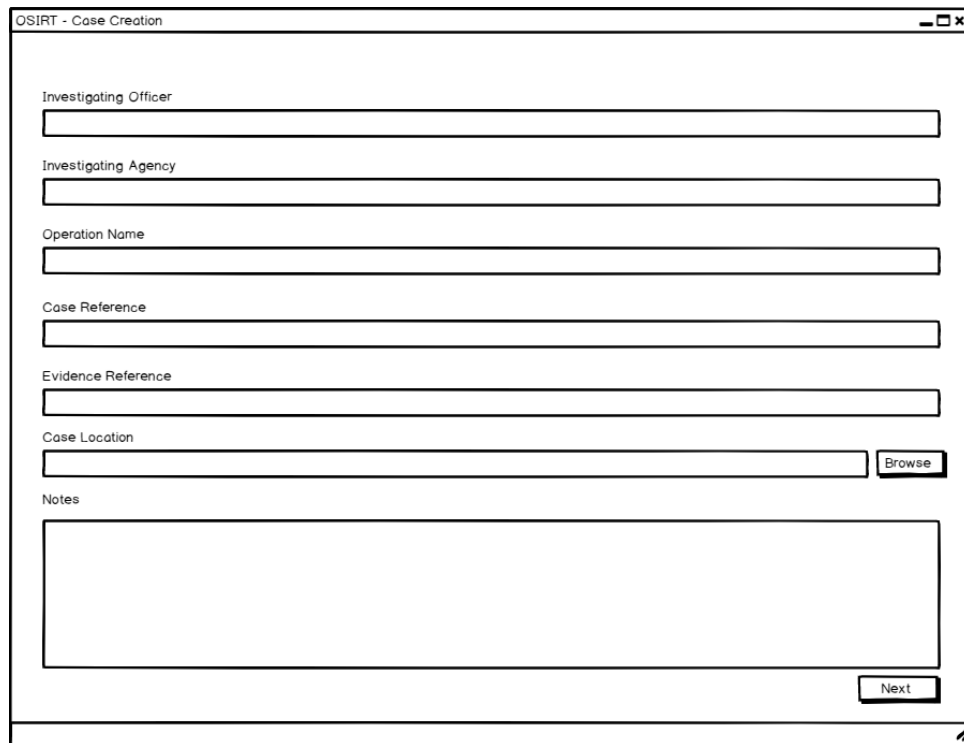


Figure 5.3 Entity relationship diagram for case database

Tables have been integrated with prototyping in mind, and are not normalised. The **case_details** table was designed based on early wireframe (Figure 5.4) designs shown to the College of Policing trainer.



The wireframe shows a window titled "OSIRT - Case Creation". It contains several input fields: "Investigating Officer", "Investigating Agency", "Operation Name", "Case Reference", "Evidence Reference", and "Case Location". The "Case Location" field has a "Browse" button next to it. Below these fields is a large "Notes" text area. At the bottom right of the form is a "Next" button.

Figure 5.4 Wireframe for case creation

5.2.3 Main browser

5.2.3.1 Custom browser compared to extensions

Early design phases looked at whether OSIRT should be a browser extension, or a bespoke application. This section will discuss the viability of both, along with any limitations.

Browser extensions, also known as add-ons, are typically created in client-side languages such as HTML and CSS for any UI elements and JavaScript to drive the add-on's functionality. Typically, extensions are a popular way of providing functionality that is not available in the browser. Modern extension usage by browser is not known, although figures of 85% of all Firefox users used an add-on, was noted from a blog post in 2011 by the Mozilla team (Mozilla, 2011). Chrome statistics are also minimal, with a figure of 33% provided by the Chromium team in 2010 (Chromium, 2010). Inspecting user downloads for popular add-ons in both Firefox and Google Chrome showed there are hundreds of millions of users for these popular browser extensions alone in 2018; one would expect the 33% quoted by the Chromium team to now be massively out-dated.

Arguably, extensions are popular because they can be simply integrated to enhance the user's existing browser experience. However, as section 5.1.1 discussed, in this specific domain there is a need to be able to control many elements of what a user can and cannot do, and what needs to be automated by the system (e.g. saving and hashing a downloaded image). While, largely, extension APIs have access to everything a browser can do, thus provide an opportunity of controlling certain actions, the changes required to achieve the necessary level of control will have to fundamentally change the way the user interacts with their standard browser. While any bespoke tool will also do the same, there is a clear distinction that one is a regular web browser with an extension, and a bespoke browser designed to conduct open source research.

A bespoke tool with an embedded browser provides substantially more control to the developer. With an add-on, the developer is limited to what the browser API provides. Using a desktop-class language such as C# provides a variety of libraries and frameworks, making additions to the application simpler. However, using an embedded browser relies on the developer to update and maintain it, or even rely on a third-party to do so. Given Google Chrome's market share, it is a safe assumption updates will be forthcoming.

After much consideration, the overarching thought was that the clear need for control, and to ensure that legal and ethical guidelines are adhered to when conducting open source research. After a discussion with the trainer surrounding these concerns, and a wireframe shown of the proposed main browser (Figure 5.5), the trainer agreed a bespoke application was the best choice for the prototype.

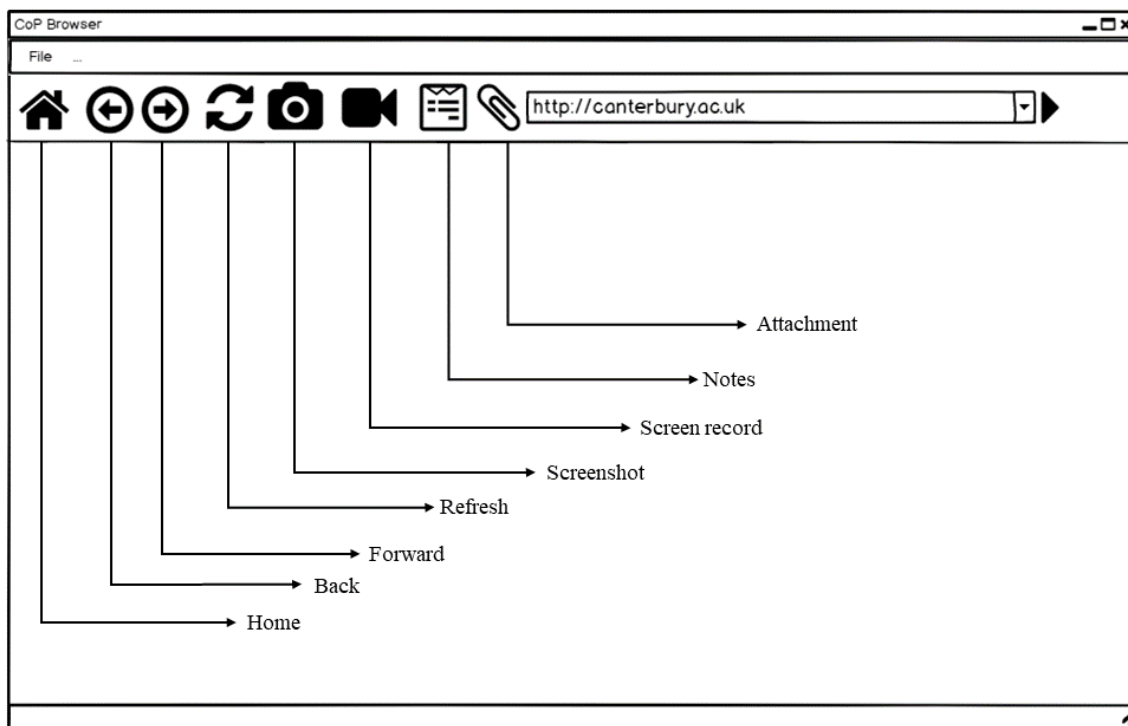


Figure 5.5 Wireframe of main browser

Early designs of the main OSIRT browser are modelled around a typical web browser and as such houses the standard options one would expect to see such as a home, forward and back, a ‘go’ and refresh buttons. In addition to this, common investigative tools are also placed in the toolbar, for example, screenshot options.

5.2.3.2 Embedded browser choice

There are several embedded browsers available to C# WinForm applications, these range from free and open source, to paid-for controls. For simplicity and ease of integration, OSIRT’s prototype made use of the built-in `WebBrowser` control that is available in the .NET Framework. The `WebBrowser` control is a wrapper for the unmanaged `WebBrowser` ActiveX control and works by creating an object instance of the control and adding it to an appropriate user interface control. In the prototype, the `WebBrowser` was implemented by creating a specialised `ExtendedBrowser` class that inherited from `WebBrowser`.

5.2.3.3 Browser implementation

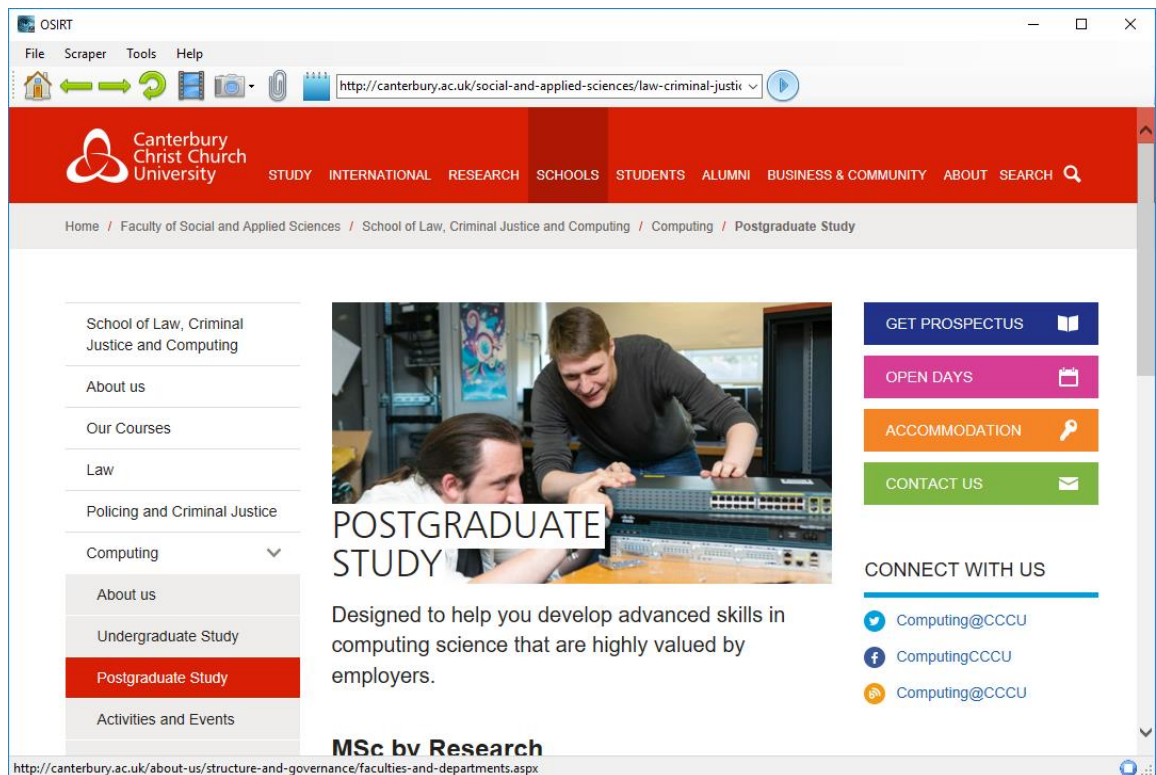


Figure 5.6 The main OSIRT browser

The default `WebBrowser` implementation uses rendering equivalent to that seen in Internet Explorer (IE) 7, and many modern websites no longer supporting IE 7. There are two ways around this. Firstly, it is possible to inject the underlying Document with an ‘X-UA-Compatible’ meta tag (Listing 5.1)

```
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

Listing 5.1 Meta tag required to get page to render using modern web standards

This is not a particularly robust solution as it involves intercepting the request, parsing, injecting the underlying Document with the meta tag, then displaying the page to the user. In addition, this method relies on the requested Document to be well-formed in order to parse and inject it. The X-UA-Compatible workaround is better suited if the Document is under the control of the developer. Most damningly for this method, setting the X-UA-Compatible does not change the way the `WebBrowser` will report itself

to the server. If a web-page uses the client-side user-agent string to display the correct content, which many websites do, this method will not work.

The second, better solution, is to use registry settings to force browser compatibility. Since IE 8, Microsoft allow developers to set the version of IE the `WebBrowser` control uses via a change to the `FEATURE_BROWSER_EMULATION` key in the registry. Setting the registry key provides better solution, as it does not require any code injection of the document to get the desired functionality. A downside to this method is that the target machine must have the correct version of IE installed, otherwise the `WebBrowser` reverts back to rendering mode compatible with that machine. For example, if IE 8 is installed then it will revert to that even though the target is set to IE 11. While the `WebBrowser` is lightweight as it can hook onto IE's rendering engine, the downside is having to download an additional browser (that is, the latest IE) to obtain the latest features. For the prototype, setting the `FEATURE_BROWSER_EMULATION` registry key was used⁴.

5.2.4 Artefact management

As part of the auditing process OSIRT splits artefact type, user and OSIRT generated activities into the notion of an 'Action'. Everything that can be logged within OSIRT has an appropriate action associated with it, as noted in Table 5.1 which lists all the Actions available in the OSIRT prototype, along with a description.

⁴ Chapter 5 will discuss the downsides of this choice in detail.

Action Type	Description
Opened	OSIRT case container has been loaded into OSIRT.
Closed	OSIRT has been closed and the case is no longer being worked on.
Loaded	A website has been loaded
Download	An artefact has been downloaded. This covers artefacts downloaded via the download manager, or items saved via the context menu (e.g. images and source code).
Screenshot	A partial of fullpage screen capture.
Snippet	A partial screen capture.
Scraped	Images that have been ‘scraped’ (i.e. parsed out from the current document) from the webpage.
Attachment	Files that have externally been added to this current OSIRT case.
Video	Video captures taken by the built-in screen recorder

Table 5.1 A list of available actions, and their descriptions, in OSIRT

Actions are logged when an appropriate event has finished executing. For example, when a `DownloadComplete` event has fired. When these actions have occurred, they are called ‘Page Events’ within OSIRT, and each Page Event stores the date and time, URL, action, file name, hash value and notes associated with the artefact. The Page Event is then passed onto the relevant case management class that handles the Page Events, and adds the details to the case database and places the appropriate files in the case container.

5.2.5 Context menu handling

The default context menu (that is, the menu that is displayed when a user right-clicks on a webpage (Figure 5.7) displayed within the `ExtendedBrowser` was the cause of several issues as it removed control away from OSIRT. For example, when a user saved an image using ‘Save target As...’ that saved image was not logged and hashed. Another example that if the user attempted to ‘Open link in new tab’, nothing would happen, as that functionality was not available. No trivial API is provided to change the context menu, or handle events within it, but the `WebBrowser` does expose a `boolean` property `IsWebBrowserContextMenuEnabled`. This property enables the context menu to be disabled, allowing for a custom context menu to replace it. In this instance,

creating an instance of a `ContextMenuStrip` and assigning it to the `WebBrowserContextMenuStrip` property (Listing 5.2).

```
//executed on load event
IsWebBrowserContextMenuEnabled = false;
contextMenuStrip = new ContextMenuStrip();
contextMenuStrip.Items.Add("Save Image As...", null, SaveImageAs_Click);
contextMenuStrip.Items.Add("Save Page Source", null, SaveSource_Click);
contextMenuStrip.Items.Add("View Page Source", null, ViewSource_Click);
contextMenuStrip.Opening += contextMenuStrip_Opening;
ContextMenuStrip = contextMenuStrip;
```

Listing 5.2 Creating and setting the custom context menu

The `contextMenuStrip_Opening` event allows for customisation of the menu based on what element within the Document has been right-clicked. Listing 5.3 illustrates the handling of an `img` element that has been right-clicked. In this instance, if the element's tag name is an image tag, then it displays the "Save Image As..." option in the context menu.

```
private void contextMenuStrip_Opening(object sender, CancelEventArgs e) {
    if (element == null) return;
    contextMenuStrip.Items[0].Enabled = element.TagName == "IMG";
}
```

Listing 5.3 Creating and setting the custom context menu

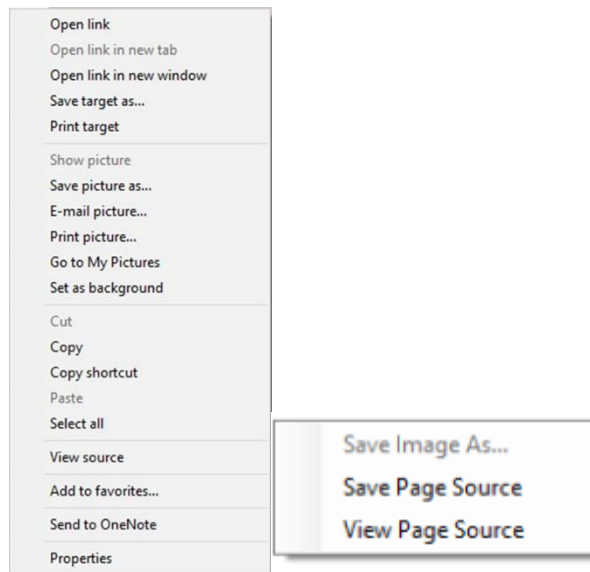


Figure 5.7 Default context menu and its plethora of options (left) and custom context menu (right)

5.2.6 Download management

As the `WebBrowser` is a hook onto IE, file downloads were handled by IE's download manager by default. This posed a problem as the default download manager had no methods that exposed where the file was downloaded to, as the user could select where to 'Save As...'. Part of OSIRT's usage could have been that the user always had to manually save to the specified case container, but this is too error prone and is the opposite of what OSIRT is meant to achieve with automation. To abate this weak user experience, initial attempts looked at potential files being intercepted via the `Navigating` event, which fires before the `WebBrowser` navigates to a newly loaded document. The `WebBrowserNavigatingEventArgs` houses a `Url` object as a property, which contains a `Segments` property, a string array of what constitutes the current URL.

```
private void ExtendedBrowser_Navigating(object sender,
WebBrowserNavigatingEventArgs e) {
    string fileName = e.Url.Segments[e.Url.Segments.Length - 1];
    if (fileName.EndsWith(".exe")) {
        e.Cancel = true;
        WebClient wc = new WebClient();
        wc.DownloadFileCompleted += wc_DownloadFileCompleted;
        wc.DownloadFileAsync(e.Url, /* save path */);
    }
}
```

Listing 5.4 Unintelligently handling file downloads

The example in Listing 5.4 is not a particularly robust solution, as it must check the file extension from the URL, the condition could be made easier to read by storing a list of downloadable file types in, say, a `HashSet<string>`, but this still does not solve its larger shortcomings. Some download links do not have the file name and extension within the URL, for example, it is common to see websites serve files using URLs like `http://www.example.com/download.php?=file`. This particular URL will not trigger the custom download of a file as it does not end with a white-listed extension.

While it is possible to implement a custom download manager that can handle file downloads, it is a convoluted and non-trivial process riddled with poor documentation and little guidance. To implement a custom download manager, the documentation implies it is as simple as implementing the `IDownloadManager` interface. Unfortunately, what the documentation does not make clear is the requirement of several other classes that heavily interact with the Component Object Model (COM) using `InteropServices` and other unmanaged libraries. Given the importance of being able to capture downloadable content, integrating this into the OSIRT prototype was considered unavoidable⁵.

⁵ It cannot be overstated how difficult it was to implement the download manager. When I finally achieved it, I posted my solution on StackOverflow (<https://stackoverflow.com/a/30617958/2215712>) in May 2015. On January 12, 2018 it attracted the following comment from user Valamas - AUS “*The best complete example in a sea of confusing documentation. Finally someone has something that works out of the box. Thanks for posting this*”. Out of all the feedback I have received for OSIRT, this is one of the comments that means the most to me.

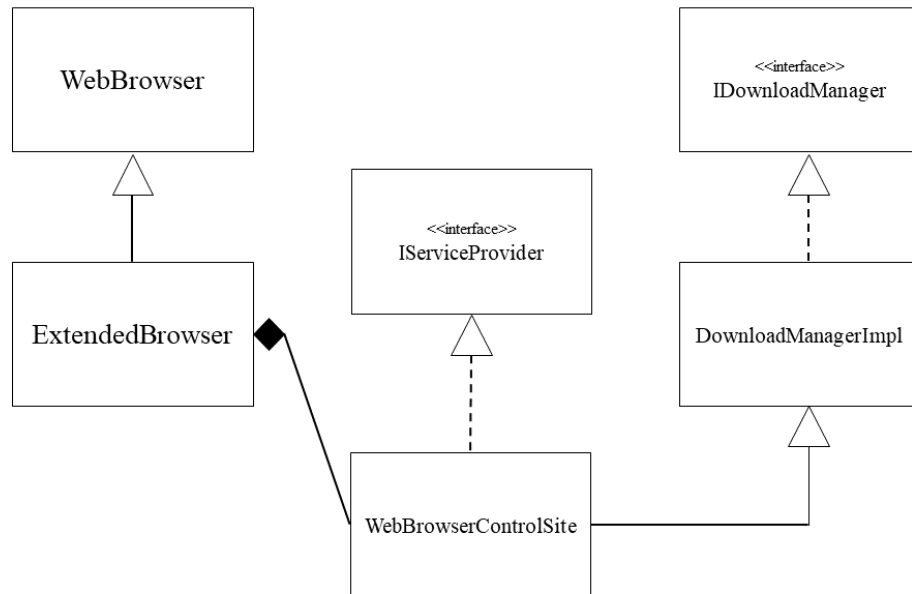


Figure 5.8 Simplified class diagram for OSIRT's download management

The complete code for the download manager will be placed into Appendix G, and Figure 5.8 shows a simplified class diagram for the `ExtendedBrowser`, however, a brief explanation will be provided here.

The `ExtendedBrowser` class is composed of a `WebBrowserControlSite` sealed class that inherits `WebBrowserSite` that provides an extensive API that allows for customisation of underlying controls, such as context menus and shortcut keys. `WebBrowserControlSite` also implements `IServiceProvider` that implements a single method `QueryService` where the implementation of the download manager is marshalled via the COM. For the prototype, an existing custom download manager created by Microsoft was used (Microsoft, 2011).

5.2.7 Logging websites

The `WebBrowser` provides several events that could capture when a new website is loaded and requires logging. These are the `DocumentCompleted` event, the `Navigate` event and the `Navigated` event. `DocumentCompleted` fires when the document has finished loading, `Navigate` fires when a new document is requested to

load, and Navigated fires when a new document has been navigated to and is now loading. The most obvious choice for capturing when a page has loaded is the DocumentCompleted event, however, this came with some caveats, the biggest of which is the DocumentCompleted event can fire numerous times. The reason for this is that external elements such as AJAX, frames and JavaScript will cause the event to fire triggering a visited web-page to be logged more than once. The same issue also occurred in the Navigate and Navigated events. To counteract this, a condition was placed within the DocumentCompleted event that checked if the document's absolute path matched that of the one loaded (Listing 5.5).

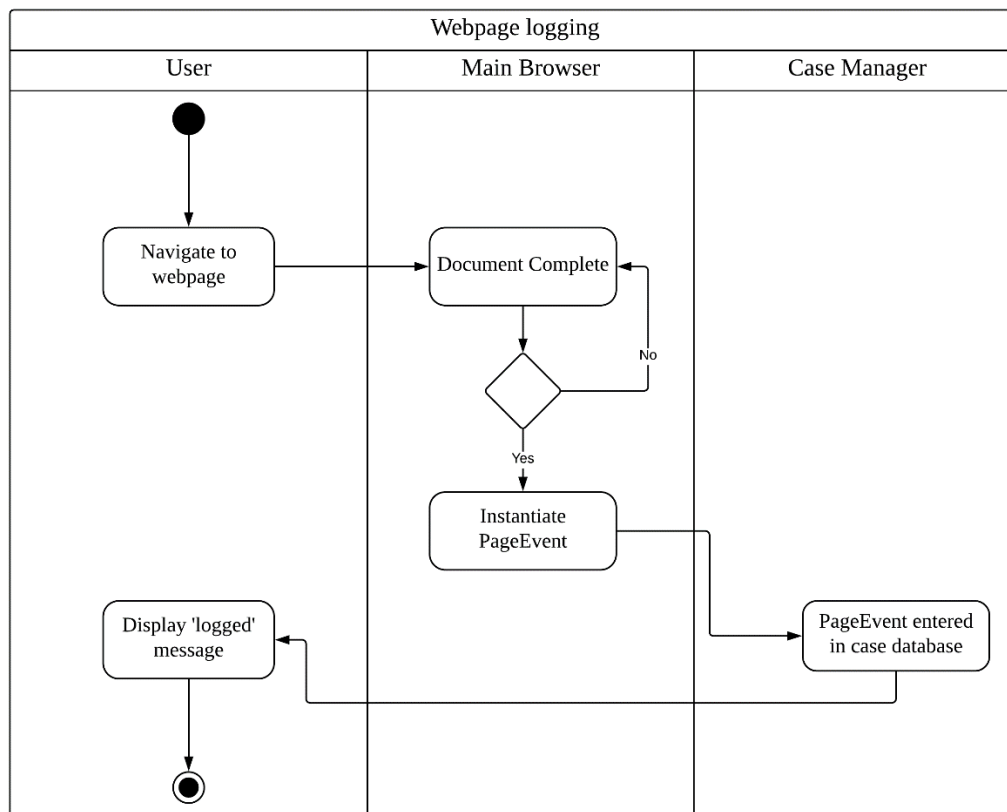


Figure 5.9 Activity diagram for webpage logging

```
private void ExtendedBrowser_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e) {
    string path = (sender as WebBrowser).Url.AbsolutePath;
    if (e.Url.AbsolutePath == path) {
        //log event
    }
}
```

Listing 5.5 Handling multiple DocumentCompleted events

5.2.8 Static screen capturing

5.2.8.1 Snippet tool

OSIRT provides several options to capture static images. Firstly, the ‘Snippet Tool’ allows the user to select an aspect of the screen to capture; this works similarly to the Snipping Tool seen in Windows. The snippet tool’s basis was taken from Hans Passant’s (Passant, 2009) answer on StackOverflow, but extended to allow snippets over multiple monitors, and fixed some minor issues surrounding graphics memory management. Snippet functionality is implemented by taking a screen capture of the displays, and the resulting screenshot is placed on a borderless form. A user can then click and drag a rectangle, which is handled using various mouse events, to snip a particular aspect of the image (Figure 5.10).

**Figure 5.10 Snippet Tool**

5.2.8.2 Current view

The current view screenshot takes a screen capture of the viewport within the browser, that is, what the user can currently see. The `Graphics.CopyFromScreen` method provides a simple means of achieving this. The current view screenshot can also be placed on a countdown timer. This is particularly useful if a website only displays content when the mouse is placed over an element, for example, Facebook shows the time something was posted as “9 hours ago” or “yesterday”, but in order to get the actual date and time the cursor has to be placed over this element (Figure 5.11).



Figure 5.11 Full date and time of a Facebook post when cursor hovers over the time

5.2.8.3 Full page

The full-page screenshot was the trickiest to implement, as in order to capture the entire page the `ExtendedBrowser` has to be temporarily resized by undocking the control. After undocking, the `ScrollRectangle`'s `Width` and `Height` must be obtained, these are properties available to the underlying `Document`. However, it is becoming common for websites to generate a dynamic layout based on the browser width and height, so resizing the browser for some websites would mean the image saved could be the mobile friendly version; the BBC website is a good example of this functionality. Worse still, some websites would rescale to a zero-pixel height, causing the application to break. To combat this until a better solution was found, the browser was resized to be at least 800 pixels wide and greater than 400 pixels high if the width was also a zero value. Pages that make use of `IFrames` also cause issues for the full-page screenshot function; although these problems also exist in other screen capturing products such as the browser extension *FireShot*.

5.2.9 Image previewer

On taking a static screenshot, a user can preview the capture by means of the ‘Image Previewer’ (Figure 5.12). The previewer provides details such as the originating URL, MD5 hash of the image file and the date and time. The image is placed within a standard `PictureBox` control for the investigator to view, in addition to a `TextBox` that

allows for notes to be entered in relation. The investigator is obliged to enter a note, to remind themselves why they have taken the screen capture. Images can be saved as png or pdf, which is provided by the PDFSharp library (*PDFsharp: A .NET library for processing PDF*, 2018).

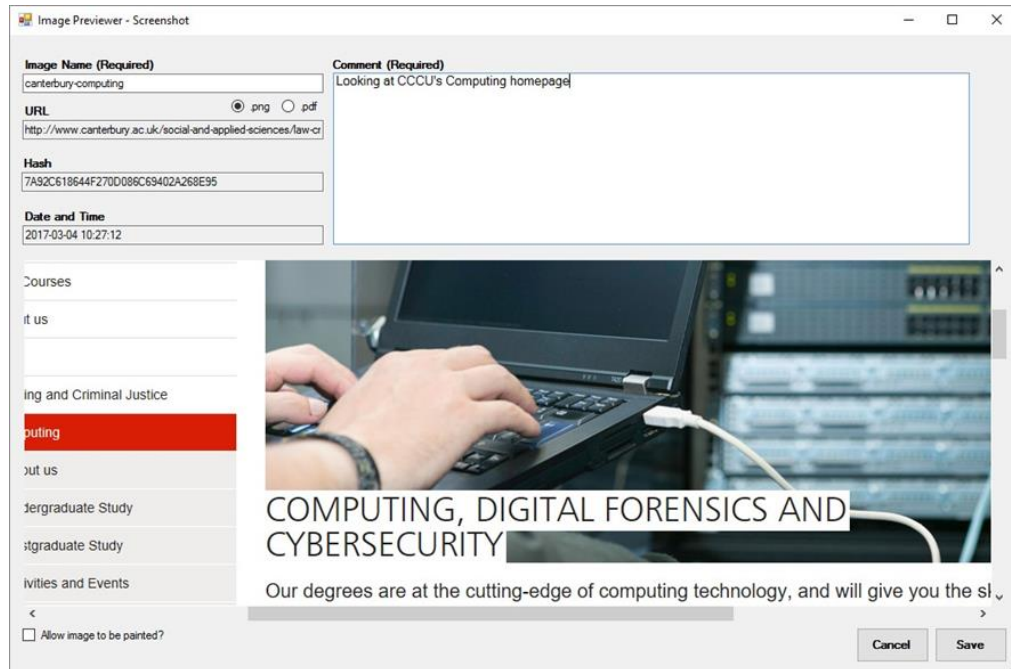


Figure 5.12 Image Previewer

The Image Previewer also provides a rudimentary ability to paint on the image; this is particularly useful to covert officers wishing to hide their details on, for example, a Facebook post.

5.2.10 Video screen capturing

With many websites making use of dynamic content generation, static screen captures could miss important evidence. For example, many websites use JQuery carousel sliders for images, making capturing this content time consuming. Another example is obtaining a video, some websites make obtaining their video content non-trivial to download (such as YouTube). To abate this, video capturing was implemented by making use of a library created by Matthew Fisher (2014). Fisher's library uses the H.264 codec, allowing for a recording resolution of up to 1920x1080, and includes the ability to only capture certain

sections of the screen using a marker window. Additionally, if the machine has stereo mix enabled, it is also possible to record system sound.

5.2.11 Attachments

Attachments (Figure 5.13) provide a way to add a file to an OSIRT case that were not obtained via OSIRT. This is useful when an investigator receives a file that relates to the case and requires it to be logged. The attached file is date stamped, hashed and placed in the attachment folder along with any associated notes. The attachments are implemented by prompting the user using a `OpenFileDialog` to select the file, then the file is copied over using the `File.Copy` method.

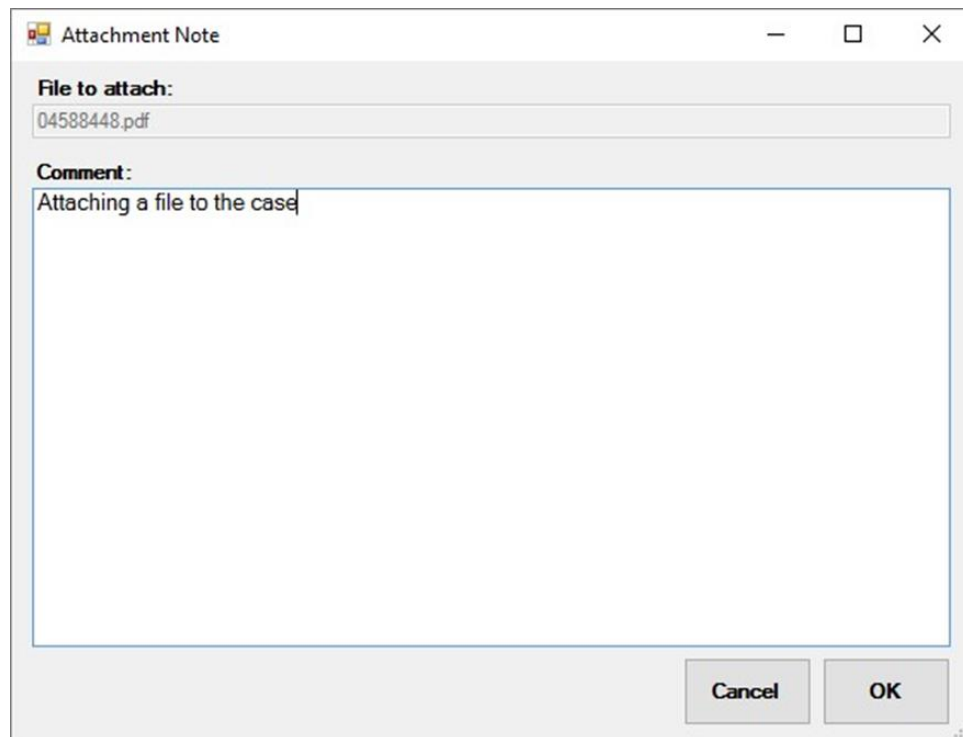


Figure 5.13 Attachment and note to be added

5.2.12 Case notes

Case notes (Figure 5.14) allow an investigator to take their own notes during the course of the investigation. This is in addition to any automated logging by OSIRT. Notes entered are automatically date and time stamped, stored within the case database within the `case_notes` table, and can later be exported separately as a plain text file or as part of the final report.

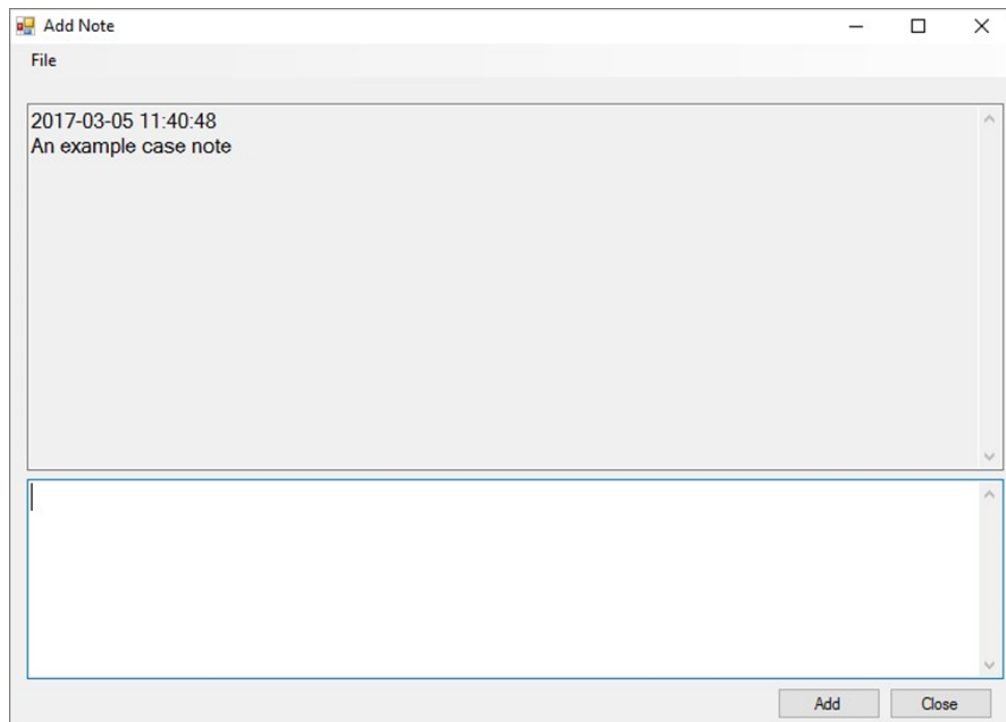


Figure 5.14 Example case notes

5.2.13 Image scraping

OSIRT provides two ways to obtain all the images on a website. The first option is under the “Static Screen Captures” icon in a menu item named “Download All Images”. This option queries the rendered document for all images tags, iterates over each image element, and attempts to download it (Listing 5.6).

```

IHTMLDocument2 domDoc = (IHTMLDocument2)browser.Document.DomDocument;
IHTMLControlRange imgRange =
(IHTMLControlRange) ((HTMLBody) domDoc.body).createControlRange();
foreach (var img in domDoc.images)
{
    imgRange.add((IHTMLControlElement)img);
    using (Bitmap bmp =
(Bitmap)Clipboard.GetDataObject().GetData(DataFormats.Bitmap))
    {
        //save and log image
    }
}

```

Listing 5.6 Saving images by traversing the IHTMLDocument2

The second option makes use of HTMLAgilityPack (Mourrier and Klawiter, 2012) which fetches a Document from a specified URL, parses the Document's content, obtains all the image URLs from the src attribute in the img tags and then places them in a list. For every node in the URL list, the image is fetched by calling the synchronous DownloadFile method available in the WebClient class (Listing 5.7).

```

var document = new HtmlWeb().Load(url);
var urls = document.DocumentNode.Descendants("img")
    .Select(evt => evt.GetAttributeValue("src", null))
    .Where(s => !String.IsNullOrEmpty(s));
WebClient wc = new WebClient();
foreach (var node in urls){
    //log file
    wc.DownloadFile(/*url*/, /*case container path*/);
}

```

Listing 5.7 Saving images by parsing the document and obtaining the src

5.2.14 Audit log

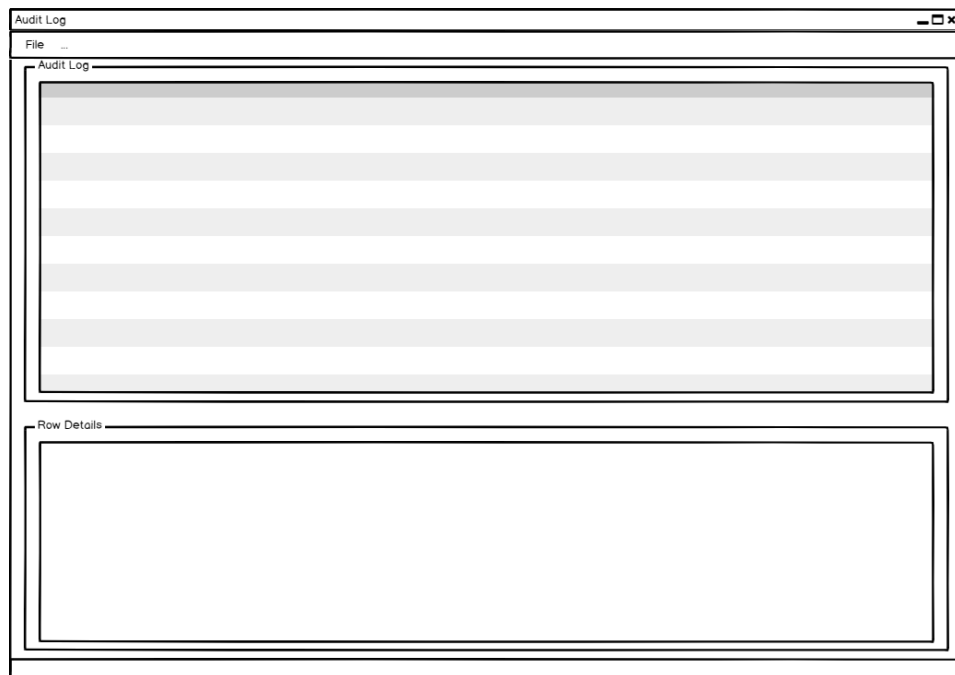


Figure 5.15 Wireframe design of audit log

The Audit Log (Figure 5.16) maintains a list of all activity for a case. Each action is logged with the current date and time, the URL, action taken (e.g. screenshot), any associated hash value, file name and notes.

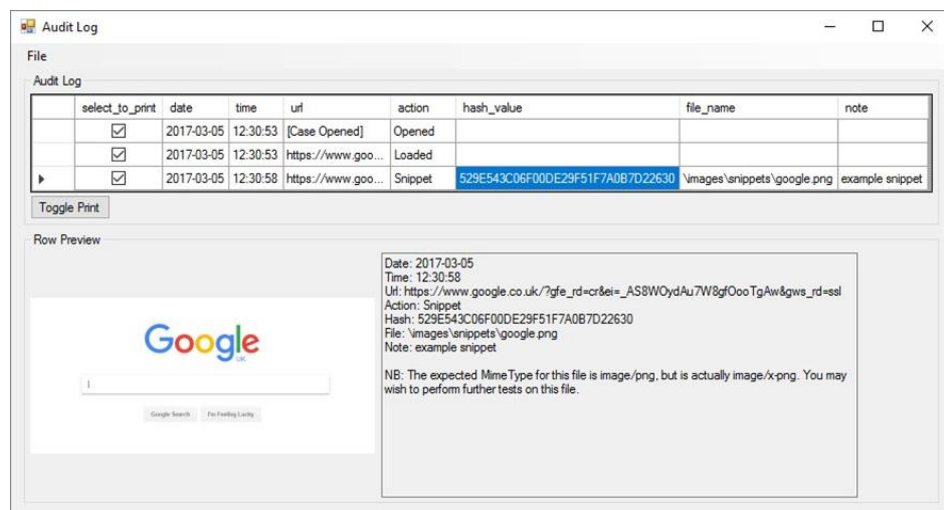


Figure 5.16 Audit Log

The Audit Log is made up of a `DataGridView` which has its `DataSource` set to a `DataTable` of all the artefacts from the `evidence_collection` table. As the `evidence_collection` table does not contain a boolean column for whether the investigator would like to print that row, one is injected into the `DataTable`.

The Audit Log also provides a row previewer, which is populated by means of the `RowEnter` event. If the row contains a file with an image extension, then the previewer will display the image within a `PictureBox`. The prototype also provided feedback as to whether a file's MIME (Multipurpose Internet Mail Extensions) type matched the extension. However, the way IE (and, by the extension, the `ExtendedBrowser`) handles MIME types for some files, for example `png`, the MIME checker would frequently report a mismatch when there was none; "The expected `MimeType` for this file is `image/png`, but is actually `image/x-png`. You may wish to perform further tests on this file."

Early iterations of the prototype stored the dates in `dd-mm-yyyy` format, a standard date format used by a multitude of countries across the globe and given this was a prototype for British law enforcement, an acceptable use-case. Better yet, the `DataGridView` control provides inbuilt sorting when clicking on a column header, meaning investigators are able to sort the columns how they wish. However, `SQLite` uses manifest typing which means data types are not strongly typed, but rather dynamically typed. For example, if a column was declared as `INTEGER` it would be possible to put `VARCHAR` data in there and vice versa. This is important, as that means there is not a notion of a "Date" type, so a date of 30-06-2015 would be 'greater than' 01-07-2015 which is plainly incorrect if sorting by most recent dates. To abate this, dates were changed to `yyyy-mm-dd` (ISO 8601), which means sorting by date is a trivial issue, as string comparisons will always put dates in the required order.

5.2.15 Reporting

Reporting provides several options to investigators (Figure 5.17). Firstly, a user can choose what they would like to print on an individual level within the Audit Log by unchecking the `select_to_print` column; this is useful if the investigator is unable to disclose certain aspects of a case. In addition, users can select to omit particular actions or images for the same reason, and place a GSCP (Government Secure Classification

Policy) stamp on the report. Once the report options have been selected, reports can be exported as HTML, which provides a richer experience, PDF, which provides a simpler experience but are easier to distribute, or as a CSV (comma separated value) file, which saves the complete audit log to a spreadsheet friendly format.

Reports (Figure 5.18) are exported to their respective 'report' directory in the case container and any artefacts are copied over to a similar file structure seen in the parent case container.

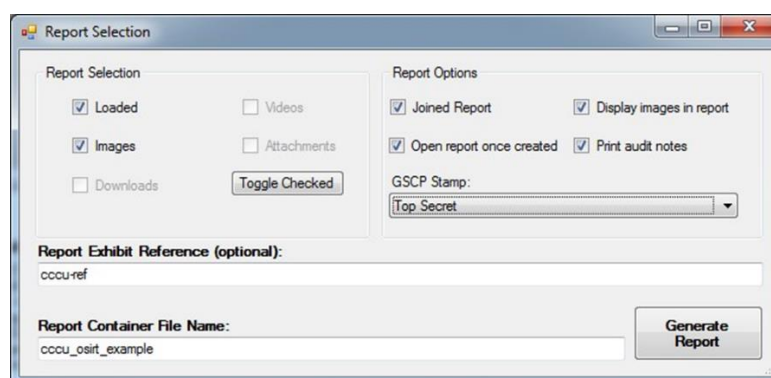


Figure 5.17 Report options

case details images downloads loaded complete	
<div>  <div> investigating officer: DC J Williams investigating agency: CCCU operation name: cccu-999 case reference: cccu-osirt evidence reference: 000-1029-1223 notes: Example Report Evidence Reference: cccu-ref </div> </div>	
date	21/07/2015
time	13:46
url	https://www.google.co.uk/webhp?hl=en&hl=en&q=cccu
action	Snippet
hash value	9D632F8578A3AE6C2505CC067F636328
file name	 google_cccu_result.png
note	looking at cccu results
date	21/07/2015
time	13:45
url	https://www.google.co.uk/webhp?hl=en
action	Snippet
hash value	FC697AE23CC587ECB927504669A20F14
file name	 google_logo.png
note	Looking at the google logo

Figure 5.18 Exported reported as HTML

5.3 Prototype development summary

This chapter looked at the design and development of the OSIRT prototype. The prototype provided a rough and ready, but complete, system that met all the essential requirements of the loose specification set out by the College of Policing. Chapter 6 will look at the results from the observations, interviews and SUS questionnaires from a RITES course that utilised OSIRT.

6 OSIRT PROTOTYPE RESULTS AND DISCUSSION

INTRODUCTION

Given there is a functional prototype of OSIRT, this chapter focuses on feedback received from officers based on observations and SUS questionnaires from a RITES course that utilised OSIRT for a week. Approximately six weeks after the RITES course, interviews were conducted with five LEOs who had taken the OSIRT prototype and used it as part of their investigations; this provided real world usage feedback.

The chapter is split into the system usability scale, observation then interview results.

6.1 Participants

Participants were eleven LEOs taking the RITES course at the College of Policing and came from a range of constabularies around the UK. All participants provided consent to be observed while using OSIRT in the classroom. At the start of the course, participants were asked to fill out a short online questionnaire that asked them to rate themselves on a scale of 1-10 on their perceived computer competency, their rank and years in service. Table 6.1 lists the participant's details.

Participant	Years' Service	Rank	Competency Rating out of 10 (self-rated)
1	18	DC	7
2	9	DC	5
3	22	DS	5
4	19	DC	6
5	14	DC	6
6	16	DC	8
7	25	DS	7
8	6	DC	6
9	15	Analyst	6
10	9	DC	6
11	5	DC	6

Table 6.1 Participant details. DC – Detective Constable. DS – Detective Sergeant

6.2 Observation results and discussion

This section is split into several key areas. Each sub-section provides results from observations and is injected with observer comments and reflections.

6.2.1 Issues surrounding OSIRT design

The complexity of the interface made performing certain tasks difficult for the users. There were several examples observed of the user interface (UI) stifling workflow and causing confusion. Examples that illustrated the weak UI design choices were seen in the video capture library, the download manager and the audit log. The video capture library already provided a UI (Figure 6.1) that had a “file name” field added to it. The interface contained advanced options such as choosing which application to screen-record, along setting the data and frame rate. The latter two fields in particular, are nonsensical to those with limited technical knowledge, and only marginally useful to those with technical knowledge. All participants had video capture issues at some stage, and all of them were a result of the complex interface, with participants having to ask what actions were required to start video capturing. While one participant did say “It’s handy to have these

[options] available”, it was evident offering complex options was slowing workflow down.

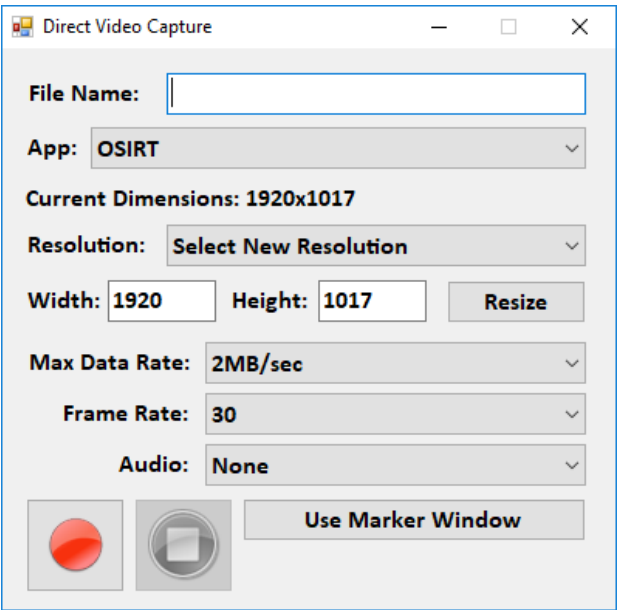


Figure 6.1 Video capture with default UI

Extraneous menu items also impacted on user experience; this was highlighted by the exporting of reports (Figure 6.2). To export a report as PDF, OSIRT would take the user through a barrage of menu items, taking six clicks to open the ‘report export’ options. All participants had issues at some point with report exporting, and it is perhaps understandable given the large number of clicks needed.

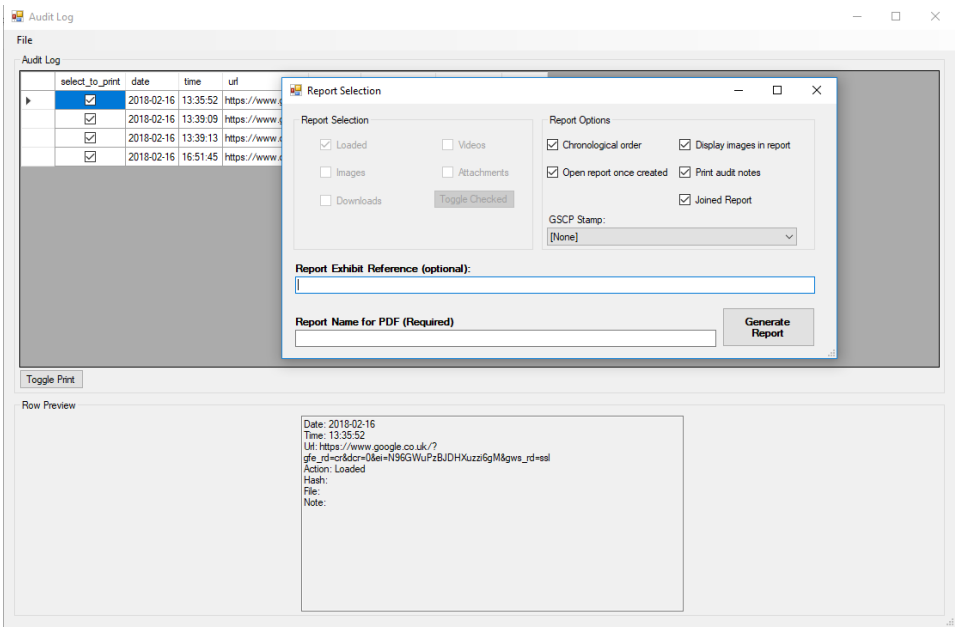


Figure 6.2 Report exporting options

Case management was also problematic for participants. OSIRT is designed so the entry within the 'Case Reference' field is used to name the case; this becomes the parent directory where all artefacts are stored. As it is a directory on the file system, the user must be prevented from entering certain characters. However, OSIRT provided no feedback to the user that only certain characters could be entered, rather, a generic error message was given if invalid data was entered. Six participants were observed inserting an invalid 'Case Reference', with comments noting the need for clearer error feedback. "I don't understand why I'm getting an error here" was a comment that emphasised the importance of useful feedback to users based on an incorrect action. It was observed that while some users knew they could only enter certain data, they could not recall what was needed exactly; this was noted by one participant "What do I have to put here again for it to work?". Four participants commented it would be useful to place a reminder as to what is acceptable.

Loading existing cases was also problematic, occurring on five occasions. Participants loaded the incorrect starting case directory because OSIRT did not provide a hint as to what constitutes a 'correct' OSIRT case directory. Instead, any directory could be loaded resulting in an unhandled exception being thrown and OSIRT crashing.

In the same vein, browsing for directories (Figure 6.3) also caused errors for users, OSIRT asked users to 'Browse For Folder' without any prompt as to what to browse for. One participant commented "I don't know what to load", and this was echoed throughout the week where participants would regularly need to be reminded or guided.

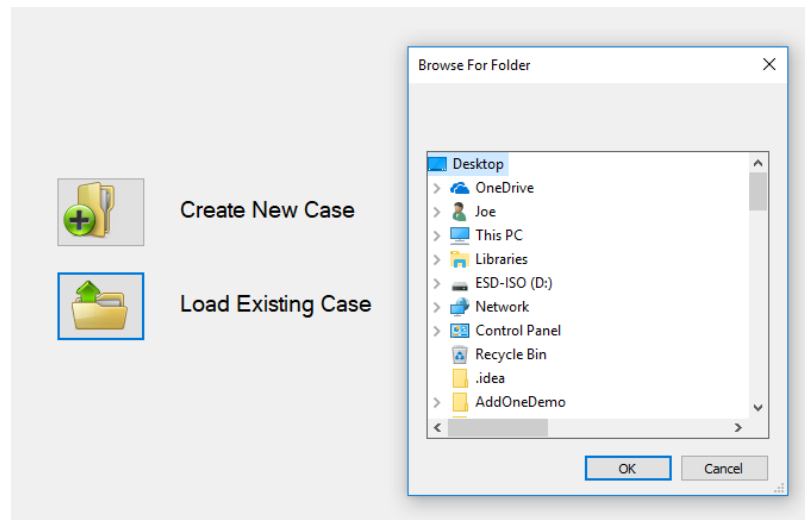


Figure 6.3 Attempting to load an OSIRT case directory

OSIRT suffered from considerable slow-downs which inevitably ended in the application crashing if left open long enough. This was noticeable when dealing with large screenshots, particularly those on Twitter and Facebook. Participants took this as an opportunity to highlight the importance of taking screenshots on social media websites, “I do a lot of work on Facebook” and “I must be able to capture social media websites” were noted by two participants, with several other participants in agreement. The Image Previewer was the culprit for some of these memory leaks, as Bitmaps were stored in memory with the streams not being appropriately disposed after use.

Three participants were observed to have an issue when right-clicking on an image element and attempting to save the image via the context menu. For two participants, this was due to the image being wrapped around an HTML anchor element (`<a>`), with one participant saying, “I’m pretty sure Firefox would be able to download this”. The final participant had found a bug in the way OSIRT calculated the cursor position relative to the image element.

Issues of tab key precedence were noted by two participants, who asked if it was possible that when the tab key was pressed it would go to the next logical text entry field where appropriate. For example, if a file name was entered in the Image Previewer then a tab press should move to the comment field.

6.2.1.1 Issues outside of OSIRT's control

Participants were concerned about the ability of being able to manipulate the case outside of OSIRT, with participants commenting “We could accidentally delete this” or “Things could be placed in this outside of OSIRT, right?” It not possible to prevent a user accidentally deleting a file, but it does highlight the need for ensuring data integrity in instances the case is manipulated.

6.2.2 Issue surrounding technology choices

OSIRT's use of the `WebBrowser` control was largely frowned upon by several participants. Once participants were told that the browser makes use of IE, there was a feeling within the room from the more knowledgeable users physically recoiling. Merely having the association with IE causes users to feel weary and cautious. While later versions of IE are vastly improved, the IE legacy clearly still lives deep within people's minds. Several participants spoke very negatively about the browser, asking “Why are we using IE?” or in some instances asking if it was possible to just use a different browser entirely, “Can it use Firefox instead?”

The issue surrounding the `WebBrowser` control were more than just negative perceptions. The training machines did not have the latest version of IE installed⁶, so Twitter refused to load for all participants, which cited an ‘outdated browser’. The machines had IE8 installed, and after this was upgraded to the latest version, OSIRT's browser engine was also updated. However, the damage was done from IE's negative legacy when participants could not load Twitter.

The `WebBrowser` also had problems with Adobe Flash content, in which some websites could not play it; one notable example being the BBC. The issue stems from a long-standing bug in Flash dating back as far as 2009 (JohanSt, 2009) caused by caching of the Flash object when loaded using external scripts. All participants experienced this when viewing a video on Facebook; they could view the video first but if they navigated away then back to the page again, the video would no longer play.

⁶ Firefox was the main browser

The problems with the `WebBrowser` culminated with a, unknown at the time, deep and serious flaw within the control itself. Essentially, the `WebBrowser` has a bug where unmanaged memory does not get reclaimed, this is exacerbated by the subtle way in which memory usage only marginally increases⁷ depending on circumstance. There were several times during the course where OSIRT ground to a halt and crashed, but this was initially attributed the Image Previewer not disposing correctly, which was a replicable cause of a crash. However, testing revealed memory usage would spike simply by navigating to a different page. Sometimes memory would increase after several hundred navigations, sometimes only one navigation would trigger the leak. The only replicable element was that OSIRT would ultimately crash and it was just a matter of when.

6.2.3 Positive feedback

OSIRT was spoken well of throughout the week by participants. The integration of tools encapsulated into one package resonated positively with the group, where they saw the value of a tool like OSIRT.

Functionality that made the capturing of artefacts simpler was popular with participants, who reacted positively to the screen capturing tools within OSIRT. The ability to take full and partial screen captures were often commented as being useful, “That’s [screenshot tool] really really useful” said one participant. The automated hashing in conjunction with the screen capture was also popular. While many of the participants were not familiar with the technical side of hashing, or in some cases not familiar at all with hashing, they appreciated the need for it once taught during the course.

The ability to export and print the report was well received, as the participants liked being able to have a physical copy of their work. One participant jokingly said, “I always make a physical copy, old habits die hard”, but this does show that hard-copies are important to some users.

⁷ Online programming community StackOverflow had a post discussing the issue that highlighted all the ‘fixes’ were no good. <https://stackoverflow.com/questions/8302933/how-to-get-around-the-memory-leak-in-the-net-webbrowser-control?noredirect=1&lq=1> (Last accessed: February 18th, 2018). Remarkably, this is not within any official documentation and no warnings are placed within the framework.

Automated logging of actions was overwhelmingly OSIRT's most popular feature. Participants frequently commented how useful and how much of a time saver it was that actions were automatically placed into the audit log. Participants remarked on existing methods of audit log management, with one highlighting noting, "I usually have to update an Excel spreadsheet. This is very useful."

6.3 SUS questionnaire results and user comments

6.3.1 SUS

OSIRT scored a mean SUS score of 85.22 (Table 6.2) with users on the RITES course, achieving a grade B on the Bangor *et al.*, (2008) grading scale. This placed the OSIRT prototype in the "excellent" category for usability. A breakdown of individual SUS results are seen in Figure 6.4.

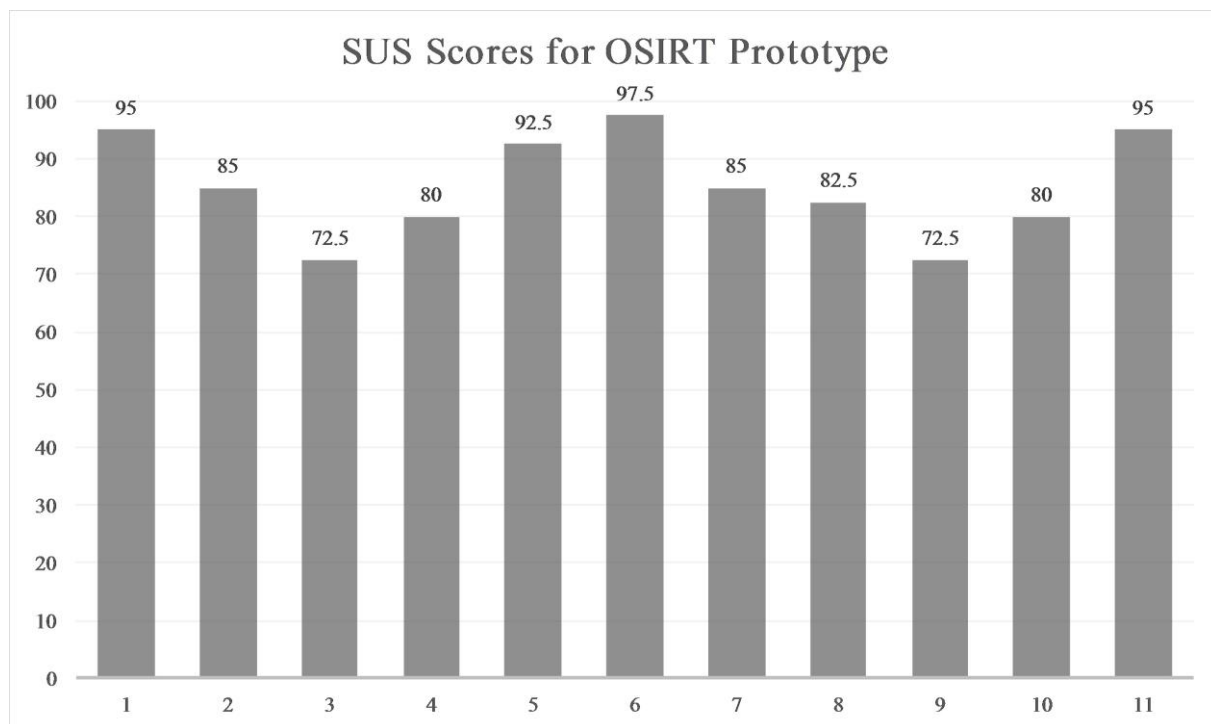


Figure 6.4 SUS scores for OSIRT prototype

Total Participants	11
Mean Score	85.22
Standard Deviation	8.84

Table 6.2 Breakdown of SUS scores

6.3.2 User comments

At the base of the SUS questionnaire, there was an opportunity to enter a free-form answer allowing participants to enter any comments they wished to make on OSIRT. Out of eleven participants, nine entered additional feedback, outlined verbatim below:

- Excellent product. Needs fine tuning. But with use by police and feedback I'm confident this will happen in a timely manner.
- Having not used alternate products I am unable to compare OSIRT against others. However, I found the principles of OSIRT and intended capabilities to be extremely efficient. Would just need back up technical advice if having problems. i.e telephone number.
- Very good product
- It's an easy to use system and it does everything I want it to do. Just keep it updated!!
- Future developments I would like to see. 1) Choice of browsers 2) History view - To allow a quicker return to a webpage. 3) Does OSIRT use private browsing? If not consider this or have option to clean cache.
- Personally, I think this is an excellent tool. Once all the little bugs are fixed I would feel 100% confident to use this tool to capture evidential product and produce evidential reports. This too is an essential tool for this type of work. Do believe that tech support required to fix bugs.
- A very useful tool that was easy to use. The bugs that were identified were ironed out very quickly and it is definitely something I would make use of.
- Good easy to use system.

- Very well thought out and user-friendly system. It's great that it's so adaptable and that the designer is happy to apply any recommendations to make the system the best it can be from suggestions he has been given.

While most of the comments were complimentary or suggested the need for technical support, one comment did propose an improvement about integrating private browsing or an ability to clear the browser's cache. The reason is that it is common for departments to have a single, non-attributable machine that is used to conduct open source research. When a new investigation is started on those machines, previous search results or cookies may provide undesired search results, or accidentally staying logged-in on websites. Depending upon the level of open source being conducted, this may have legal implications through not having the correct authorities in place, such as DSAs, to view that material.

6.3.3 SUS Results and feedback summary

On reflection, updating and fixing bugs within OSIRT throughout the duration of the course could be the reason the SUS score was high and feedback so generous; observations, or even participant comments, do not reflect a mean SUS score of 85. Fixing bugs within OSIRT over the duration of the course meant that the product participants started with was, albeit marginally, different from what they ended with. It is possible the scores reflected the author's ability to fix bugs, rather than the tool itself.

Conversely, it has now been well established that open source investigators must use a plethora of tools to conduct open source investigations. With no tool, at this stage, like OSIRT available, which combines all the tools into one product, it may emphasise OSIRT is a much-needed toolkit in the law enforcement arsenal.

The SUS and feedback also highlights the usefulness of observations as a method of gathering usability issues. Had the prototype been used during the RITES course without observations, the feedback and SUS results would suggest OSIRT was a highly usable system with minor bugs, when there are breaking flaws within the prototype. This drives home the need for using several differing data collection methods when investigating and creating software.

6.4 Prototype interviews

Semi-structured interviews were conducted via Skype with five participants, three were from the RITES course previously discussed and two who had been told about OSIRT from colleagues. The interviews were between September and October 2015 and were conducted to gather views on the OSIRT prototype. The participants had all been using OSIRT ‘in the wild’. Three participants were Detective Constables and two were Detective Sergeants, with policing experience ranging between seven and eighteen years. All five participants had previous experience of conducting open source research and had previous use of OSIRT. Table 6.3 breaks down participant details.

Participant	Rank	Years experience
1	DC	15
2	DS	18
3	DC	7
4	DC	10
5	DS	13

Table 6.3 Participant details

6.4.1 Interview guide

Questions focused on the OSIRT prototype and what they would like to see from an open source research tool. The interview was semi-structured in nature but contained several core questions that were asked to all participants, these are seen in Table 6.4 along with a reason these questions were asked.

Question/prompt	Reason for asking
Disregarding OSIRT, what tools do you use to conduct open source research?	To establish the type and style of tools used. Trying to find consistent tools used by officers.
Are you restricted by the cost of software?	Discuss this if officer brings it up when discussing the above.
What is your ideal open source research tool?	Looking for consistency amongst officers as to what they would like to see from an open source research tool.
How has OSIRT integrating into your current workflow?	Has the prototype been of any use to these officers and, importantly, can it be useful to them in real investigative scenarios.
What features of OSIRT have been useful to you?	Trying to find what features are most popular and need more focus.
What would you like to see added to OSIRT?	Discover other features the officers would like to see.

Table 6.4 Interview guide for the OSIRT prototype

6.4.2 The tool ‘mish-mash’

The previous tool usage questionnaire showed that LEOs use a varied range of tools to conduct open source research, this question aimed to not only further establish this tool usage, but to gather an understanding as to why these tools were used.

P1: “At the moment, I use a jumble of several different capture tools. I have the Fireshot add-on loaded in Firefox and that’s for screen capturing webpages. I also use a nifty little tool called SnagIt, that allows me to capture parts of a screen, like you see with that Snipping Tool in Windows 7, and it allows me to capture video”.

P1 was then asked how the maintenance of an audit trail was achieved and they responded, “It depends what I’m working on, some of the time it’s just pen and paper, other times I’ll use [Microsoft] Word”. Prompted if that was a typical workflow for the rest of their team, they replied “Fairly typical, yes”.

P2 highlights, in addition to their toolset, how ineffectual automated loggers have been and resulting in having to fall back onto the mundane task of manually logging URLs in a spreadsheet.

P2: “We use, probably, a half-a-dozen or so different bits of kit; it’s a bit of a mish-mash. This ranges from your web browsers, we tend to use Chrome, Firefox and Tor Browser and these will have some bolted-on add-ons; typically, a

screenshot capture tool, some way to change how we look on the web, you know. We've tried some loggers, but find they're not particularly effective, they all seem to miss logging something, so we just use an in-house spreadsheet and simply cut and paste links."

The use of a spreadsheet was criticised by participant 4 who noted that it was "Burdensome to keep track on visited [web] pages using Excel. It's easily the biggest chore I face when doing open source."

When asked if the cost of some tools were prohibitive, four participants answered affirmatively. While some participants can purchase tools if needed, such as participant 1

P1: "Our team is quite small and outputs a lot, so we are quite fortunate in that regard where we can buy licences for software such as SnagIt; but SnagIt is quite cheap in comparison to other tools."

Participant 5 highlighted how their choice of tool was influenced if it was free, "I tend to use free tools as it's not easy to go through the process of buying software. Money is tight. While the paid alternatives would be nice on some occasions, free works for me."

A theme that ran through the early development of OSIRT is the variety of tools LEOs must use to effectively conduct open source research. This was further reiterated in the interviews, with all five participants speaking about having to use a varied array of software to achieve their goal. Browsers aside, there was inconsistent tool usage between the five participants, highlighting a need for toolkit standardisation.

Additionally, there is no mention of IE for their choice of browser when conducting open source research. Given the observations and how negatively IE is perceived, this is not surprising and bolsters the necessity to shift away from IE as the main OSIRT browser.

The choice of language used by some of the participants when describing their current toolset showed how some officers felt about their workflow. Words that have negative connotations such as "mish-mash", "jumbled" and "burdensome" all evoke a sense of disorder and frustration, and arguably display an ineffective and onerous workflow for these participants.

6.4.3 The ‘dream’ tool

Each officer interviewed provided a unique perspective on their ideal tool, but the overarching theme throughout was this notion of an ‘all-in-one’ tool.

P2: “The dream for me, so to speak, is to have all the capture tools I need - which OSIRT provides now - but also functionality to hide my IP address, change who I am. You know, instead of it [the browser] saying “Hi, I’m Firefox on Windows 7” you can change it to say, “I’m on Linux using Chrome”.

Being ‘hidden’, or covert, is an important aspect when conducting online investigations, depending upon the ‘level’ of open source being conducted. Typically, level 2 and above open source investigations should provide functionality that makes using them ‘non-attributable’. Being non-attributable is important in a policing setting, as servers can maintain a log of visitors; this log will usually contain IP address and a user-agent string. These are counter-surveillance measures to minimise, as the above traces, if using an attributable machine, may then link the investigating officer to a police computer, potentially alerting a server owner who may then act by removing the website.

Beyond the ‘all-in-one’ capture tool, the ability to access and capture the dark web was mentioned by three participants, all of whom specifically mentioned the use of The Onion Router (Tor).

P3: “The ideal tool for me is, of course, to have an all-in-one capture tool that pops everything into one package. So that’s your screen capturing tools, your log, your case notes, your report and what have you. I’d like to see that combined, but with the ability to access Tor and when you’re browsing the dark web you have access to all these packaged capture tools just like you do when you’re in regular open source mode on the surface web.”

P1: “Being able to switch between Tor and regular web for a case with a button click would be a huge bonus. I’d also like it [the ideal tool] to be able to change who I am. There’s this add-on I use in Firefox where you can select what your browser looks like, ‘user agent switcher’ or something.”

Access to the dark web is an important element in online investigations. These investigations range across what one may expect from the dark web; crimes such as drug

trafficking, weapons trafficking, money laundering, fraud, and child sexual exploitation (Wells and Gibson, 2017). Within the past year, there has been a significant push in the UK to combat child sexual exploitation on the dark web, with Her Majesty's Inspectorate of Constabulary (HMIC) in 2015 acknowledging that the "The dark net provides abusers with a means of distributing indecent images of children" (HMIC, 2015a, p. 14). To continue the fight against child exploitation, the Home Office in 2017 provided a "£20 million boost to tackle" the exploitation of children online (Home Office, 2017a). The ability to be able to access the dark web is a necessity for those in law enforcement, and the drive from the UK government shows how it would like more focus on this area.

The dark web provides a level of anonymity to its users, purely by the nature of how it is accessed. This layer of anonymity means law enforcement are left without their traditional methods of obtaining personal data via legislation, such as RIPA. For example, lawfully obtaining an IP address and requesting user data from the Internet Service Provider is, for all intents and purposes, not feasible when the suspect is using Tor. Perhaps, then, a traditional, detective approach to policing is required whereby surveillance, artefact gathering and profile building are required. This 'traditional' approach to policing aided in the arrest and conviction of both Richard Huckle in 2016 (BBC News, 2016) and Mathew Falder in 2018 (BBC News, 2018; Davies, 2018). OSIRT can provide the means of accessing the dark web, and the tools to gather artefacts to build a criminal profile, all while remaining lawful.

6.4.3.1 Web scraping

P4 requested a "complete capture tool", the ability to download a webpage. They stated, "I'd love to be able to go on a website, click a button and scrape the entire website." While scrapers are useful tools, and do provide an investigative officer with an 'offline' copy, there are several issues surrounding the use of 'scraping' software. Scrapers can be tuned to work in various ways, but typically they involve establishing a connection to the server, downloading the index page, parsing then traversing the document for content. This content can be hypermedia (images and videos, for example) or hyperlinks to other documents on the server; although any element can be scraped given the appropriate parser commands. However, there are some concerns with this method ranging from technical to ethical. Firstly, scraping an entire website would leave an obvious footprint

on the server, as the number of concurrent connections and the speed of which the website was being traversed and downloaded would be, fairly obviously, from a non-human source, and go against trying to use covert, anti-surveillance techniques. Secondly, a user who has already visited a webpage has already downloaded it, to then download it again would be redundant, and leave an additional footprint. Finally, from an ethical standpoint, the question then becomes is it correct to have a ‘download all evidence’ button?

While OSIRT provides methods for capturing artefacts, these are manual actions that require a note and/or are logged within the audit trail, meaning consideration and justification should be given by the investigator. Integrating a scraper into OSIRT would be relatively trivial given the vast range of HTML parser libraries available, but a tool like OSIRT is to support an investigative open source researcher. Their audit trail, as specified by ACPO/NPCC, should illustrate thought processes and timelines. To then add a “download entire website” feature would be, in the author’s opinion, counterproductive in this regard.

6.4.4 OSIRT’s integration

All participants said that integrating OSIRT would be possible within their workflow, and often highlighted this move as a “simple” or “easy” one. Participants again noted that given the current variety of tools in use, the switch to one, combined tool should theoretically be a simple conversion.

P4: “Integrating a tool like OSIRT would be extremely simple, because it does exactly what we need from a tool. It’s basically then just a shift from using several tools to one, and I can’t imagine many would be put off by having to do that.”

P3: “As I said earlier, our workflow is quite simple, we screen capture something, we pop it on a Word document, write up about it and that’s that. So, the transition from that to something like OSIRT would actually be quite easy. But there are those who don’t like change [laughs].”

Participants 3 and 4 also both note there is a need to “shift” or “transition”, and this underscores that integration of software into an existing workflow is a hurdle that OSIRT must overcome if it is to be successful.

6.4.4.1 Resistance to change

Participant 3's comments about people not liking change is a complex issue, with resistance to change being a trait that is seen through all aspects of human interactions. These could be low-impact changes such as food products being moved around in the local supermarket, to high-impact changes seen in the change of a job or workflow. High-impact changes can be threatening and intimidating to users. Cohen and Sherman (2014) state changes that impact "self-integrity" can be factors in causing stress, and even "hamper performance and growth". However, aspects such as training, communication and being given a voice when change is being affected can aid in mitigating this anxiety (Cohen and Sherman, 2014). OSIRT is built around law enforcement feedback, where frontline officers are encouraged to comment and offer suggestions. In addition, OSIRT is a tool used to train at the College of Policing, providing training around the tool and an opportunity to use it in a safe environment. The effectiveness of this training is discussed in detail in Chapter 10.

A limitation of this data is that officers who participated in this interview were happy to take OSIRT and use it within their role. It is reasonable to assume that they could see OSIRT being integrated into their workflow as they are happy to adopt the change, but that does not necessarily represent the views of those who have not previously used OSIRT. Chapter 9 and 10 discusses the integration of the release version of OSIRT into real world job roles.

6.4.5 Useful features

All five participants acknowledged the usefulness of integrated capturing and the combination of several other features that reduce tool burden.

P4: "We really like the built-in capturing capability. It's not just that ability to capture, though, it's that ability to make notes, to have the time there, too, to also have it automatically provide a hash. The automated logging is a big deal for us, and at the end being able to produce a report is really good."

The DC with fifteen years' experience highlighted that while paid-for tools are of great use, they can be expensive or layered with so many features it can hinder those who are not computer savvy.

P1: "The biggest for me is the screenshot and video capturing. As I've said we use SnagIt and Camtasia, and these are great tools don't get me wrong, but they do cost a few quid. They have a ton of features, too, and while that is generally a good thing, too many buttons to click can cause issues."

The comments about expense were made by two other participants, with one noting that while individual licenses may seem cheap, buying in bulk can lead to increased costs:

P5: "The expense of these tools are not too much of an issue to us, mainly because we're a relatively small team. But open source [research] is gaining traction, and you think, really, we didn't do much open source at all five years ago and now it's forming a large part of our investigative toolset. £50 software sounds cheap individually, but when you're looking to roll that out to thousands [of users] and that's only one [said in an exclamatory manner] piece of kit, the cost is easily a six-figure sum. Six-figure sums for software that could be obsolete in three years. That's a lot of bobbies on the beat in comparison."

Automated logging is discussed by this participant, drawing attention to the maintenance of an audit log as not only a guideline, but also an important procedure when using directed surveillance.

P2: "I like the automated logging. One thing we have to be aware of is if the OSC [Office of Surveillance Commissioners] come in and say "Right, how have you obtained this? What processes do you have in place?" with a tool like OSIRT you can say "I did x,y and z here, here and here" and they can review those processes, because it's all audited for you. We are always extremely cautious of these things, so to have a computer system that does it for you is a bonus."

This officer reaffirms the issues surrounding open source capture as discussed in chapter 2. The necessity to ensure procedures are correctly followed can cause those conducting open source research to, arguably, show signs of unease; especially when there is a need to be "extremely cautious". Additionally, the spotlight is placed on the software system,

OSIRT in this instance, to ensure that the requisite processes are being met, and to a standard that satisfies those governing bodies such as the OSC.

6.4.6 Improvements and additions

The five participants all mentioned that the UI needed refining, describing it as “hard work” and “clunky”. These opinions echo what was seen during the observations, stressing the need to refine the UI to streamline tasks. For example, one participant mentioned that “video capturing [...] is quite confusing so would like to see that streamlined” and offered the suggestion of just having a “giant record button”. Three participants mentioned that OSIRT slowed down after continued usage, particularly on social media websites (websites that are media rich). This, again, aligns with what was observed on the RITES course with the need to shift away from the `WebBrowser` control.

Reporting was another issue raised by participants, with participant 2 expressing these views:

P2: “Reports are a bit of a pain to export. If it could just have a button that said ‘export report’ and it did it that’d be useful. The report itself can get a bit messy, particularly if you have lots of text and large images. If all that could fit neatly onto A4 paper, that’d be a huge bonus.”

Ensuring a report fits on an A4 piece of paper is not a trivial matter. As seen from the observations police like to have physical copies of their work. However, fitting arbitrary sized data on a piece of paper is not trivial. A Facebook profile can run into tens of thousands of pixels, representing this on a piece of paper is not feasible or practical. A compromise may be to allow the printing of images separately outside of the report, this would allow images to be scaled appropriately without causing problems with the rest of the report.

Unsurprisingly, given the ‘dream tool’ section, the biggest requests for OSIRT were Tor and tabbed browsing where all five participants mentioned being able to integrate these two features. “I suppose this goes back to the dream opens source tool, but realistically

and what I'd love to see in the future is the ability to use OSIRT with Tor to go on the dark web." noted one officer.

Tabbed browsing is standard in web browsers and users would expect to see that functionality. However, OSIRT struggles with one browser instance due to the issues with the `WebBrowser` control, and further pushes for a re-think of the browser OSIRT uses.

Two participants mentioned the ability to put 'add-ons' or extensions into OSIRT, highlighting that while these add-ons may appear trivial, they can have an impact on investigative prowess.

P3: "I'd like to see more add-ons. Things like image Exif viewers, IP addresses, domain registration viewers all small little tools that collectively make a big difference."

6.5 Summary of changes required

The feedback received via interviews and e-mails, plus the data gathered from observations provided an excellent basis to start creating the release version of OSIRT, this section discusses those necessary changes.

Foremost, there is a need to work on the rudimentary user interface as it was designed very much in the style for a throwaway prototype. Particular considerations around the user interface is that options are buried in confusing menus and/or dialogs, and to make frequently used features easier to access. Linked closely to the user interface issues are the lack of robustness within the system. When things go wrong in the prototype they are generally fatal errors, but with rigorous error handling these could easily be avoided; providing the user with a better experience.

Case management, and in particular loading a case, is clunky and confusing to users; which often leads to mistakes. This was not reported so much by the interviewees, but it was frequently witnessed in the observations. Given the integral nature of the case files, this process needs to be simplified.

A large stumbling block with some users was the use of the `WebBrowser` control and its reliance of Internet Explorer. Internet Explorer is, evidently, a very unpopular choice and leaves a negative perception causing users to question why OSIRT uses it. Beyond

the negative perceptions, the `WebBrowser` control is generally broken and suffers from some game-stopping bugs, such as memory leaks. The need for a new, robust browser control is required if the release version is to gain any traction. Another positive from integrating a new browser control will be the ability to add tabbed browsing, a highly requested feature from users.

6.6 Chapter summary

The overarching conclusion that can be drawn is that the prototype was a success and it showed that a tool like OSIRT was needed by the police. While the prototype was not streamlined and contained bugs, officers took OSIRT away and used it⁸ as part of their investigations, and passing it onto colleagues to use and test. These officers provided feedback based on real-world scenarios, which proved to be very useful.

During the prototype phase, over 100 e-mails were received from 40 different LEOs, mainly Detective Constables and Sergeants, along with trainers from the College of Policing. Many of these e-mails were introductory, and were usually followed the pattern “OSIRT is a fantastic tool, but...” where the suggestions made were issues previously observed or known about. These e-mails showed that OSIRT was gaining traction within UK law enforcement and there was interest, but OSIRT needed further work. Given this is a prototype that is expected feedback. In addition, there were also approximately two dozen phone calls involving OSIRT during the prototype period.

Observing OSIRT in action, being used by its intended audience was profoundly helpful, and presented a plethora of direct feedback. From a personal perspective, the observations provided crucial insight and made a significant impact in how the release version of OSIRT would be created.

A fallacy that followed the author throughout the creation of the prototype, before seeing OSIRT used first-hand, was that the police were, in some way, super-human. It never occurred that law enforcement are just regular people. Instead, assumptions were made

⁸ One officers, now retired, never used to the ‘release’ version of OSIRT, instead opting to use the prototype for 18 months. They just “preferred it”.

in that they all would be technological wizards able to use *any-old-software* without a problem, even if it was a bit buggy. The reality is, police officers come from all walks of life and are human beings. Some are technologically savvy but it is more likely they are not, or are even technophobes; just like any slice of society would be. As a programmer, it is easy to slip into the mentality of solving the problem and moving on. It is critical to pause and reflect on your design decisions, considering the potential effect they can make on both yourself in the future and, importantly, those using your product. The prototype had a profound effect on the author. No longer is software being made for fun, but rather to make a genuine impact with people relying on it to work.

7 OSIRT RELEASE DEVELOPMENT AND IMPLEMENTATION

INTRODUCTION

Chapters 5 and 6 looked at the creation of the OSIRT prototype, the feedback received from officers back on-the-job, and observation and SUS data from a RITES course. The prototype has already had a positive impact and contribution on policing, and comments received from those officers and trainers, in addition to the observations, is a crucial step in the continuation and creation of a brand new ‘release’ version of OSIRT. This chapter looks at the technical implementation of the release version of OSIRT.

The chapter dissects OSIRT’s features, design justifications and technical implementation of those features. The chapter then reflects upon the design decisions and summarises the impact these had upon the release version of OSIRT.

7.1 Main browser

Given the issues identified with the `WebBrowser` control in chapter 3 and 4, a new browser control is required for this version of OSIRT. To assist in the selection of a new control, several browser controls for the .NET Framework were reviewed and tested. Deciding factors for the browser controls’ integration should meet as much of the following criteria as possible: good documentation, actively maintained, represents good value for money if it is a paid-for control, although given this is an open source project, an open source browser would be preferable.

7.1.1 GeckoFX

GeckoFX (<https://bitbucket.org/geckofx/>) is a free and open source .NET control based on Gecko, the layout and rendering engine used by Mozilla Firefox. Integration of GeckoFX was relatively straight forward when the required libraries were added, but the JavaScript issues frequently caused the browser to freeze and display an unsightly error in a `MessageBox`. The documentation surrounding GeckoFX, was limited or non-existent, and made finding solutions to these types of problems difficult. Given that GeckoFX is an open source project that is to be expected, but it does make debugging and troubleshooting harder. Overall, GeckoFX gives the impression of being designed for kiosk applications; it provides a limited interface with a bespoke set of features. Fundamentally, it is not a control that is suitable as a web browser replacement. On a positive note, GeckoFX is actively maintained and receives frequent updates. However, given the limitations it is not suitable for this project.

7.1.2 Awesomium

Awesomium (<http://www.awesomium.com>) is a closed source and partially free browser control for .NET and C++ based frameworks. Awesomium uses the WebKit engine, as seen in Apple's Safari browser. There is reasonable documentation, and technical support if required, but to obtain access to all the features there is a \$2900 license fee; out of the scope for OSIRT at its stage of development. Additionally, Awesomium's latest update was November 2014 and has not seen an update since that time⁹. Given the lack of consistent updates, the expensive licence and closed source nature of Awesomium's licencing is not suitable for this project.

7.1.3 CefSharp

CefSharp (<https://github.com/cefsharp/CefSharp>) is a free and open source .NET control, available for WinForms and WPF, that is a managed wrapper for the Chromium Embedded Framework (CEF) (Greenblatt, 2018), which itself is a derivative of Chromium, the project behind Google Chrome (*The Chromium Projects*, 2018). CefSharp

⁹ As of May 2018, Awesomium's website is claiming that "Awesomium is getting an upgrade". Albeit, a little late for this project.

was tested early in the prototype for viability. While it was simple to integrate, it lacked core features such as viewing PDF documents, additionally Adobe Flash content was missing. However, during the development of the OSIRT prototype, the CefSharp project expanded significantly and provided a .NET wrapper for many of the features available in CEF, making it not only a viable alternative, but a better one compared to other browser controls. CefSharp is actively maintained, has excellent documentation and communication channels with the development team. CefSharp can also be associated with what users may consider a ‘good’ browser, in Google Chrome.

7.1.4 Browser control summary

While there is not a great amount of choice in regards to browser controls for the project, CefSharp provides all the functionality required of the browser, with additional useful features. CefSharp’s design is based around interfaces which means integrating new features, especially in comparison to the `WebBrowser`, is much easier. One example being the download manager, which only requires a single interface to implement, as opposed to several classes and interaction with the `WinAPI` using interop services. Additionally, the excellent documentation and community surrounding the CefSharp project is important to solve any potential issues. Ultimately, if there were a dozen other browser controls to choose from, it is hard to envision a better control than CefSharp.

7.2 Document Object Model and JavaScript

The Document Object Model (DOM) plays a role in how OSIRT can capture artefacts, and manipulate webpages if needed. This section briefly explains the purpose of the DOM, and how it can be utilised.

The DOM is defined by W3C as a “a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.” (W3C, 2009). The DOM is a hierarchical tree (Figure 7.1) structure, with each node in the tree being an object that can be manipulated with changes being reflected within the document (W3C, 2009).

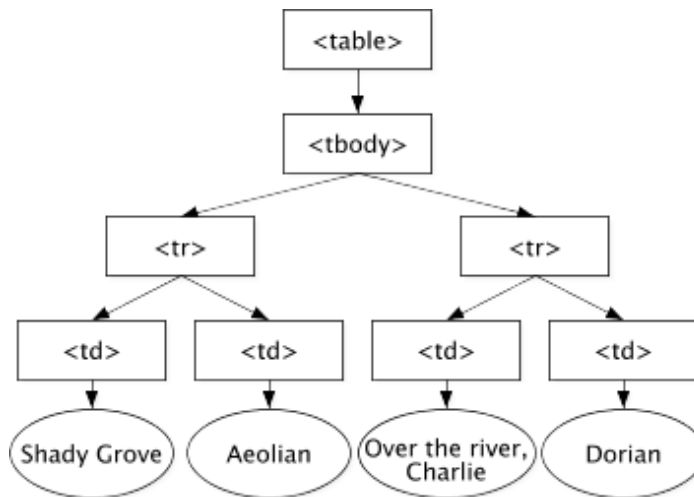


Figure 7.1 Example of DOM for a table in HTML (Le Hégarret, Wood and Robie, 2004)

DOM manipulation in CefSharp, and most browsers, is performed by using JavaScript, a client-side programming language that executes within the browser.

7.3 Tabbed browsing

7.3.1 Tab style

One of the most requested features from the prototype and subsequent feedback was the ability to have multiple tabs within OSIRT. Early iterations of this feature focused around using the .NET Framework's `TabControl`, which can be managed by adding and removing `TabPage`s. While a `TabPage` can be programmatically styled using the `DrawItem` event of the `TabControl`, this styling of the `TabPage`s proved to be cumbersome and produced tabs that were difficult to manage by the users. It generally provided an inconsistent user experience.

A third-party solution to these shortcomings was found in `DotNetChromeTabs` (Francis, 2016), a free and open source project, that offers Google Chrome-like tab management and look-and-feel (Figure 7.2). `DotNetChromeTabs` integration was not due to several minor bugs within the project, but fixing these was given priority. The control does provide the user with an experience they would be accustomed with in other web browsers. Several additions were made to the `DotNetChromeTabs` code to make it

compatible with OSIRT, and to provide a smoother experience in general¹⁰. A `SelectedIndexChanged` event was added, as when a user switches tabs the address bar needs to be updated with the URL of the selected tab. Additionally, functionality was integrated that ensured at least one tab was open at all times. This was due to a limitation in CEFSharp (specifically, CEF). Once `Cef.Initialize` has been called, it cannot be called again within an application's lifetime. Closing the final browser instance would cause `Cef.Initialize` to be called again if a new browser instance was created; this causes an exception to be thrown. To fix this issue, if there is only a single tab open the close button is removed from that tab.

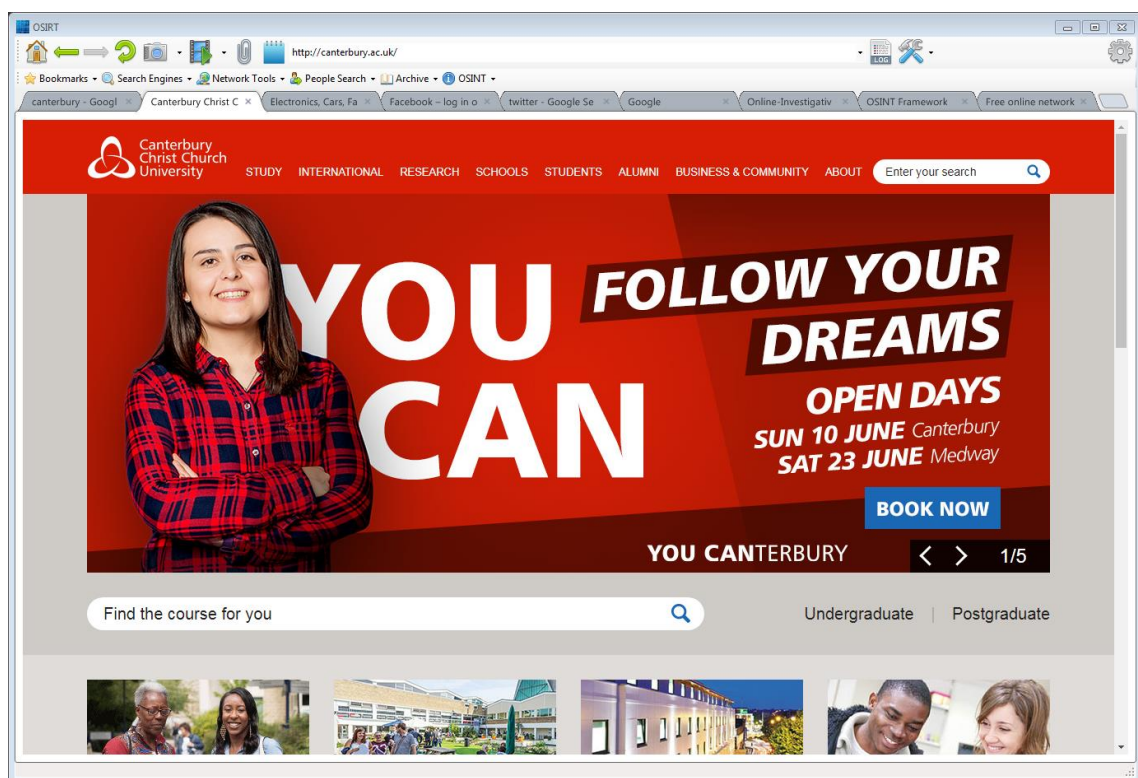


Figure 7.2 OSIRT with tabbed browsing

7.3.2 Tab implementation

Browsers implement tabbed browsing by either listening for `window.open` requests made by client-side scripting or from a user request (e.g. right-clicking on a link and selecting 'open in new tab'). Within a typical web browser, what occurs when `window.open` is called is a preference set by the user within the options for that browser; that is, whether to open in a new window (a pop-up) or to open a new tab¹¹. In

OSIRT, control must remain within the main application, so pop-up requests must always open within a new tab. To handle these requests, CefSharp provides an interface to manage the creation of new tabs by implementing the `ILifeSpanHandler`. The `ILifeSpanHandler` contains a method, `OnBeforePopUp`, which is called when a web page has made a `window.open` request. It is this method that calls the `OpenInNewTab` event with the requested target URL.

```
public class LifespanHandler : ILifeSpanHandler {

    public event EventHandler OpenInNewTab;
    public bool OnBeforePopup(...) {
        newBrowser = null;
        OpenInNewTab?.Invoke(this, new NewTabEventArgs(targetUrl));
        return true;
    }
    //omitted...
}
```

Listing 7.1 Implementation of `ILifeSpanHandler` and associated code for opening a window in a new tab

The `OpenInNewTab` event makes a call to the `CreateTab` method that adds a tab to the tab control.

```
public void CreateTab(string url) {
    BrowserTab tab = new BrowserTab(url, addressBar);
    uiBrowserTabControl.TabPages.Add(tab);
    AddBrowserEvents();
}
```

Listing 7.2 Creating and adding a tab to the tab control

Pages can also be opened in new tabs by right-clicking on any element with an anchor tag (Figure 7.3).

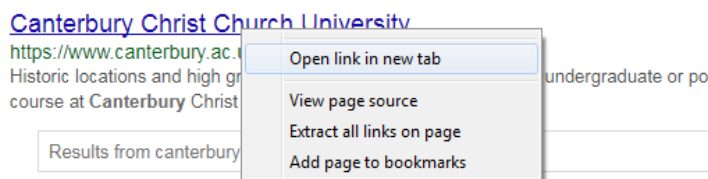


Figure 7.3 Opening a link in a new tab via the context menu

7.4 Context menu handling

CefSharp provides the `IContextMenuHandler` interface to customise the context menu. Upon implementation of the `IContextMenuHandler`, the `OnBeforeContextMenu` method is called to populate the context menu with appropriate options given the browser's current context. For example, if an image is right-clicked, additional options can be displayed (Figure 7.4)

```
void OnBeforeContextMenu(IWebBrowser browserControl, IBrowser browser, IFrame frame,
    IContextMenuParams parameters, IMenuModel model) {

    if (parameters.TypeFlags.HasFlag(ContextMenuType.Media) &&
        parameters.HasImageContents) {
        if (OsirtHelper.HasJpegExtension(parameters.SourceUrl)) {
            model.AddItem((CefMenuCommand)ViewImageExifData, "View image EXIF
data");
        }

        model.AddItem((CefMenuCommand)MenuSaveImage, "Save image");

        model.AddItem((CefMenuCommand)CopyImgLocation, "Copy image location to
clipboard");

        var sub = model.AddSubMenu((CefMenuCommand)ReverseImgSearchSubMenu, "Reverse
image search tools");
        sub.AddItem((CefMenuCommand)ReverseImageSearchTineye, "Reverse image search
using TinEye");
        sub.AddItem((CefMenuCommand)ReverseImageSearchGoogle, "Reverse image search
using Google");
        model.AddSeparator();
    }
    //code omitted...
}
```

Listing 7.3 Populating the context menu

As seen in Listing 7.3 CefSharp provides rich interfaces and context-aware parameters to query the current web page. In Listing 7.3, OSIRT uses the parameters available in the `OnBeforeContextMenu` method to query if the user has right-clicked an image and performs additional checks on the image to provide further functions to the user. For example, if the image is a JPG, then show EXIF related tools.

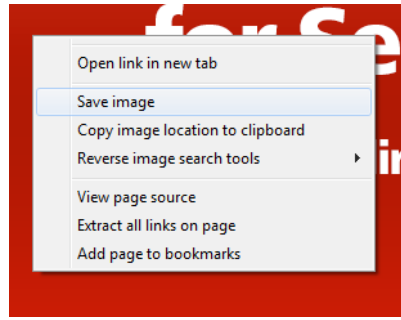


Figure 7.4 Context menu when an image is right-clicked

7.5 Case management

7.5.1 Case container

The difficulty found in loading cases within the prototype required a rethink in case handling and management. The most obvious choice was for cases to have their own file type with a custom file extension. This method makes loading cases simpler for the user, as the file chooser (Figure 7.5) can enforce the loading of this custom file type. Typically, custom files types are a compressed file format with some metadata associated with it; a simplified example being a Microsoft Word docx file. Docx files are, essentially, zip files that contain sub-directories with XML and other metadata. Using this approach for an OSIRT case file provides not only the opportunity for simple loading of cases but zip files can be password protected using strong encryption methods such as AES256. Given that password protection was suggested by several users in e-mail feedback, this is a useful feature.

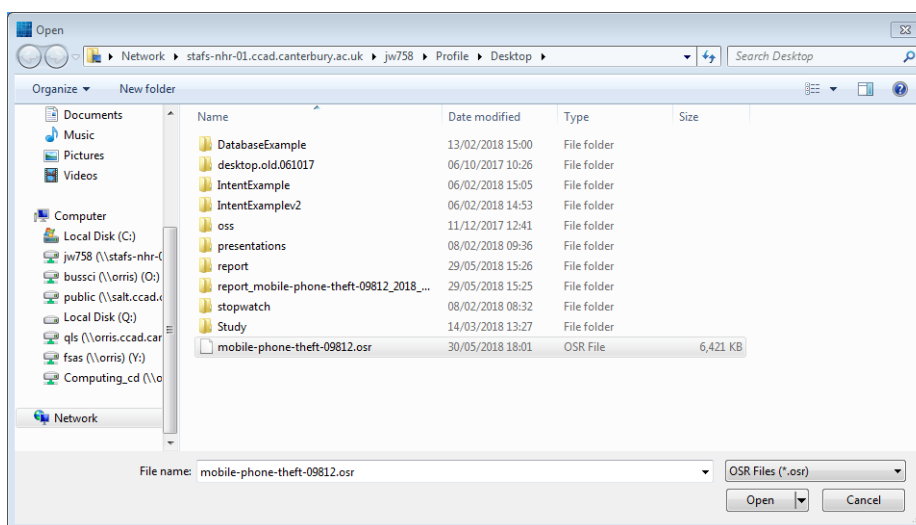


Figure 7.5 Case opening with custom file format

Case compression and extraction was implemented using `Ionic.Zip` library (Feldt, 2018). A third-party library was chosen over the .NET Framework's zip file management namespace, seen under `System.IO.Compression`, because it does not contain a method of adding passwords to compressed files. Figure 7.6 shows the directory structure for the case container.

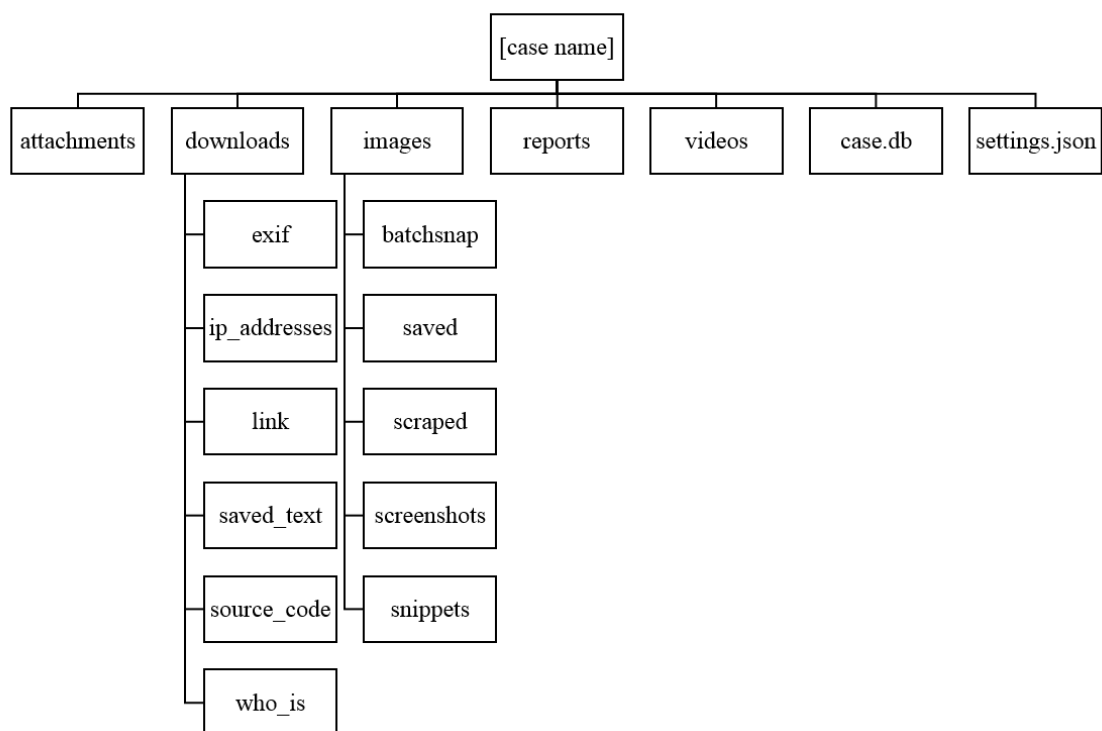


Figure 7.6 Directory structure for the case container, this is wrapped in an .osr file when not open within OSIRT.

7.5.2 Complexities and considerations of using a custom file format

Given that cases are now extracted after being loaded into OSIRT, then compressed when OSIRT closes, there is a possibility the case will remain in an open, uncompressed state if OSIRT closes ‘unsafely’. Unsafe closes can occur from hard-crashes or forced closure of the application (via control-alt-delete, for example). OSIRT must ensure that cases are

recoverable after this occurs. Figure 5.1 shows a UML activity diagram of the system handling case recovery on event of a crash.

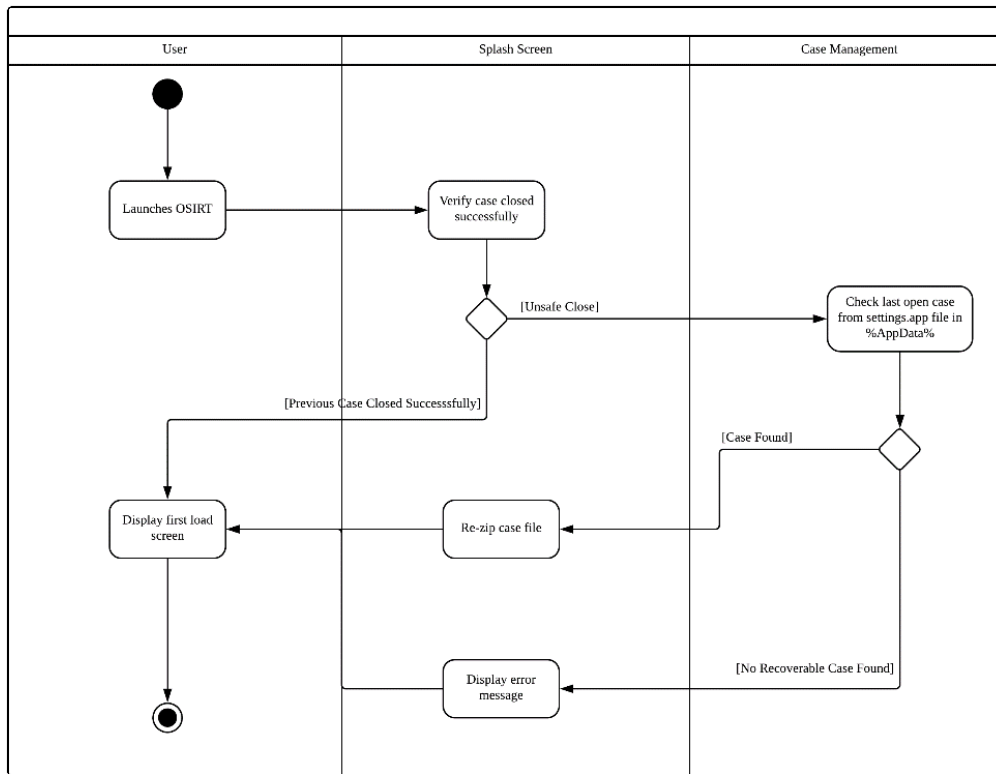


Figure 7.7 Activity diagram for case recovery

7.6 Error handling and recovery

An issue surrounding the prototype was its inability to appropriately handle exceptions and the subsequent crash. A web browser is an inherently complex application and the potential for something to go wrong is always a possibility. The users of this application will be obtaining evidential artefacts; error handling and error recovery is integral if OSIRT is to be a trusted tool. The release version of OSIRT is much more robust from this standpoint, and handles exceptions as they occur, if feasible. However, there may be occasions where an error will occur that OSIRT does not know how to handle and may need to restart. Rather than hard-crashing with no recovery, as seen in the prototype, OSIRT has integrated a 'Fatal Exception' handler. This handler has been designed to obtain a stack trace of what caused the exception (an output that can be sent to the developer to debug the application) and a safe way of closing and restarting OSIRT to ensure the integrity of the case.

7.6.1.1 Error handling implementation

The .NET Framework provides two event handlers to manage exceptions at an application level, these are `AppDomain.CurrentDomain.UnhandledException` (“occurs when an exception is not caught” (Microsoft, no date)) and `Application.ThreadException`. These handlers are attached before the main application is executed by placing them in the application’s entry point, the main method, in `Program.cs`. Listing 7.4, derived from the answer provided by peSHir (2009), shows how this was achieved.

```
static class Program
{
    [STAThread]
    static void Main(string[] argv)
    {
        try
        {
            AppDomain.CurrentDomain.UnhandledException += (sender,e)
            => FatalExceptionObject(e.ExceptionObject);

            Application.ThreadException += (sender,e)
            => FatalExceptionHandler.Handle(e.Exception);

            Application.Run(new MainForm());
        }
        catch (Exception ex)
        {
            FatalExceptionHandler.Handle(ex);
        }
    }

    static void FatalExceptionObject(object exceptionObject) {
        var ex = exceptionObject as Exception;
        if (ex == null) {
            ex = new NotSupportedException(
                "Unhandled exception doesn't derive from System.Exception: "
                + exceptionObject.ToString()
            );
        }
        FatalExceptionHandler.Handle(ex);
    }
}
```

Listing 7.4 Fatal handler implementation

`FatalExceptionHandler` is a form (Figure 7.8) that displays the stack trace in plain text format and can be copied to the clipboard for ease of sending via e-mail.

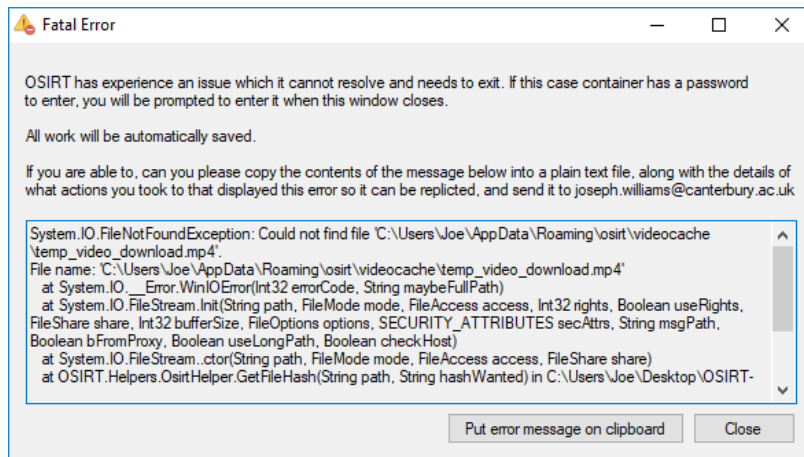


Figure 7.8 Fatal error handler

7.6.2 Manual case recovery

While OSIRT provides automatic recovery of cases, there may be instances where automatic recovery is not possible. For those instances, manual case recovery is provided on the first load screen. The user selects the parent directory of the case, and clicks ‘Recover’ (Figure 7.9)

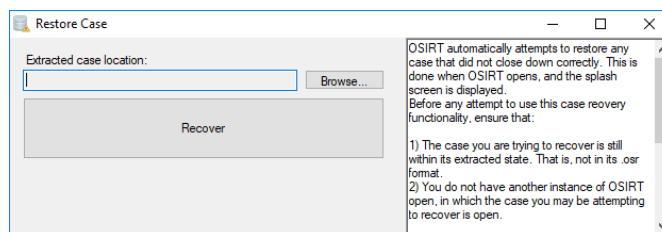


Figure 7.9 Case recovery panel

7.7 Static screen capturing

Static screen captures were a popular feature within the prototype, with all of them being integrated into the release version of OSIRT. Some of the captures needed redesigning after feedback from the prototype users, and this section discusses the issues surrounding that.

7.7.1 Full page screen capture

As seen from the prototype, capturing full page screenshots caused OSIRT to run out of memory and fail in some instances. Given screenshots can be of an arbitrary size, it is not

reasonable to expect OSIRT to be able to capture pages of an ‘infinite’ length. That said, OSIRT should be robust enough to provide the perception that it can. For example, Facebook and Twitter feeds offer seemingly ‘infinite’, lazily-loaded content when scrolling the page and, as seen from feedback from the prototype, these social media platforms offer important artefacts to LEOs. The issue then becomes how OSIRT can manage the capture of large screenshots, given a document of an arbitrary height (Figure 7.10) and content is out of view.

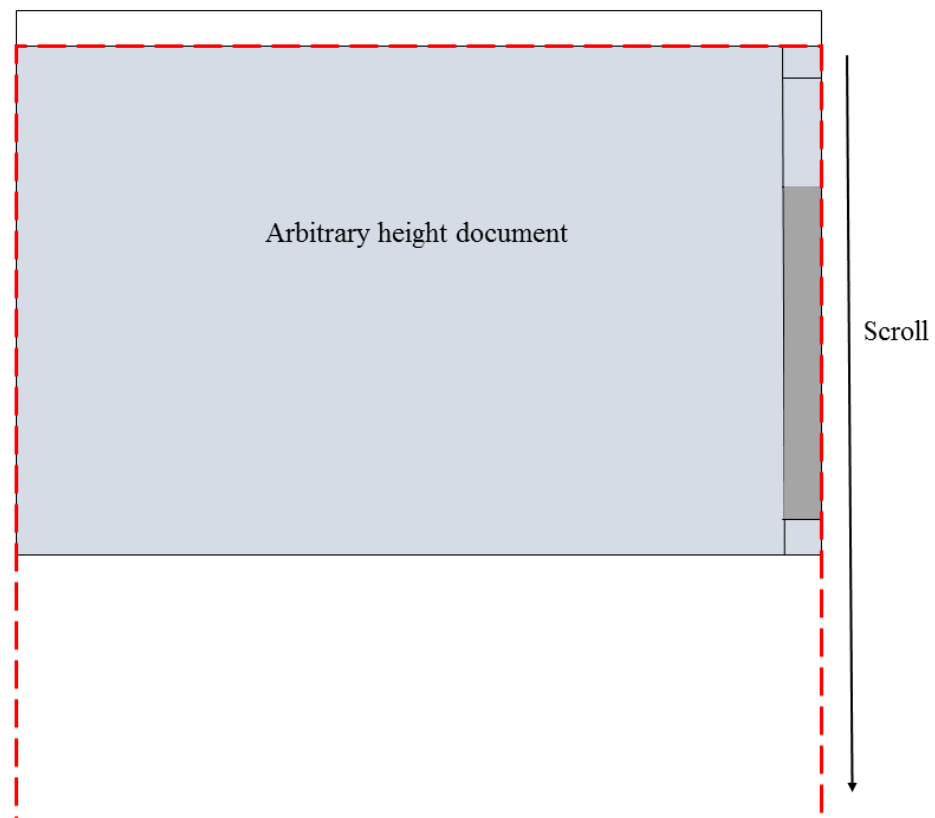


Figure 7.10 A representation of an arbitrary sized document.

Early attempts focused around the native GDI API of Windows, and transparently capturing the entire page, only this time using the hard disk to store the image rather than store it in memory. However, the nature of using this API would cause memory usage to spike for large images and would inevitably lead to `OutOfMemory` exceptions, leading to this idea being dropped entirely.

Instead, full page capturing was re-designed and focused around the use of scrolling capture. The general concept is to ‘divide and conquer’ by taking current viewport

screenshots, scrolling the webpage, take another screenshot and to continue to do so until the page reaches the bottom. After completion of the scroll-and-capture, the chunks are vertically stitched back together again, thus forming a complete screenshot. The general algorithm is seen in Listing 7.5, with discussion surrounding its integration after.

Procedure ScrollingFullPageScreenshot

```

SET scrollHeight TO document scroll height
SET pageLeft TO scrollHeight
SET viewportHeight TO browser viewport height
SET count TO 0

While not at end of document
  IF(pageLeft > viewportHeight)
    Scroll document by count * viewportHeight
    Get current viewport image
    Put current viewport image in temp cache
    INCREMENT count
  ELSE
    Get current viewport image
    Crop remaining viewport image
    Put cropped image in temp cache
    BREAK LOOP
  END IF
  SET pageLeft TO pageLeft - viewportHeight;
End While
Build screenshot

```

End Procedure

Listing 7.5 Pseudocode for scrolling screen capture¹²

7.7.1.1 Implementation of scrolling capture

Unlike the `WebBrowser` control, `CefSharp` does not provide wrapper interfaces to the underlying `Document`. Rather, all interaction with the `Document` is performed via `JavaScript`, which is executed using the `EvaluateScriptAsync` method of the browser.

While the `Document` does expose a `scrollHeight` property, it was discovered during testing that it did not always return the actual scrollable height of the `Document`, often returning either a height of zero, or an incorrect `Document` height. A look at the `JavaScript`

¹² The algorithm OSIRT uses to create a full page screenshot has been integrated into the `CefSharp` documentation and examples.

documentation showed several properties that can be used obtain the Document's height. The algorithm OSIRT implemented, Listing 7.6, uses an approach that queries all the appropriate properties relating to the Document's height and returns the highest value using `Math.max`. Testing has shown this to be very effective for obtaining the correct Document height.

```
(function() {
    var body = document.body;
    var html = document.documentElement;

    return Math.max(
        body.scrollHeight,
        body.offsetHeight,
        html.clientHeight,
        html.scrollHeight,
        html.offsetHeight );
})();
```

Listing 7.6 Obtaining the Document's height using JavaScript

Pages like Facebook and Twitter can ‘infinitely’ scroll, giving the impression that the document height is unknown or exceptionally large. However, the height of the document is always resolved when the page has finished loading. If there is *potentially* more content that gets loaded as the page scrolls (i.e. lazily-loaded content) it is not included in the document height calculation. Once the user manually scrolls the page to load the content, it will reset the document height. For example, a Facebook user profile loads the Document height is set to 3000 pixels. A user scrolls and loads more content, changing the Document height to 6000 pixels.

To store the temporary screenshots that get joined back together, an image cache directory is placed in the user's application data folder (`%AppData%`) (Figure 7.11), with screenshots immediately deleted from this cache after the whole screenshot is saved into the case container.

While joining the screenshot back together is a trivial process, inbuilt image APIs of the .NET Framework, from previous experience, will not be able to handle the potential size of the joined screenshot. A third-party library, Magick.NET (dlemstra, 2018), was introduced and was capable of handling large sized images. Magick.NET is a wrapper for the command line image processing tool ImageMagick (ImageMagick 7, 2018).

Magick.NET can deal with very large images, tens of thousands of pixels in size, which is ideal for large screenshots.

Full page screenshots are now a default option within OSIRT. When a user clicks the camera icon, a full page screenshot is taken. This was added after a meeting with two officers from the Metropolitan Police Service, who recommended full page capturing should be the default option.

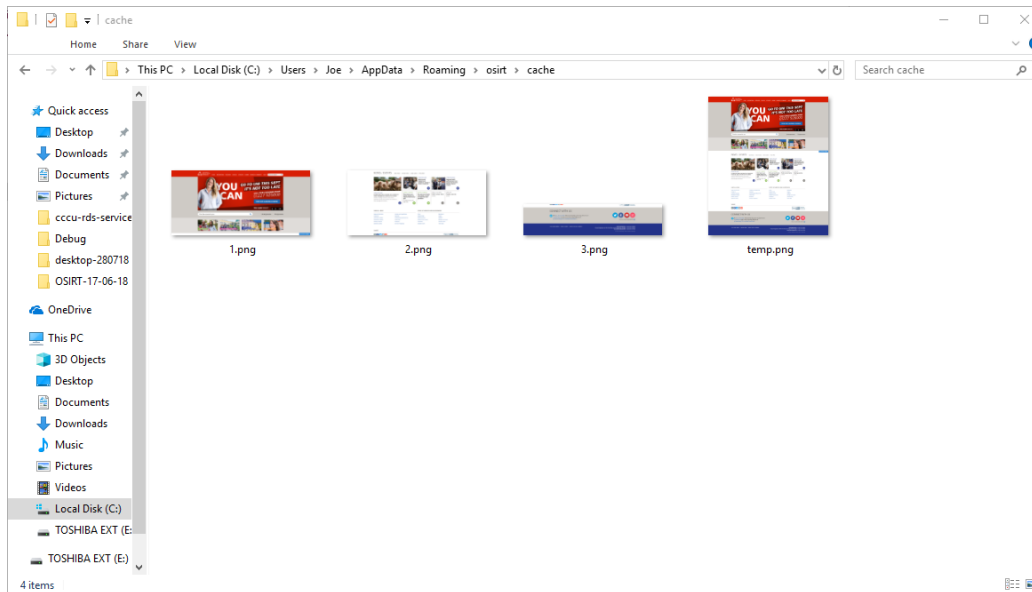


Figure 7.11 Disk cache with partial screenshots, and temp.png as the completely stitched image

7.7.1.2 Issues surrounding full page capture

Capturing of full page content is now vastly improved since the prototype, but there are still several barriers given the inevitable complexities of web pages. Firstly, web pages contain content that may be fixed in place using CSS position property `fixed`. A fixed element “is removed from the normal document flow” (MDN web docs, no date). One example of this is when a page is scrolled, the element will continue to scroll with the page as it is fixed in place. Fixed elements are typically implemented for navigation bars (Figure 7.12).

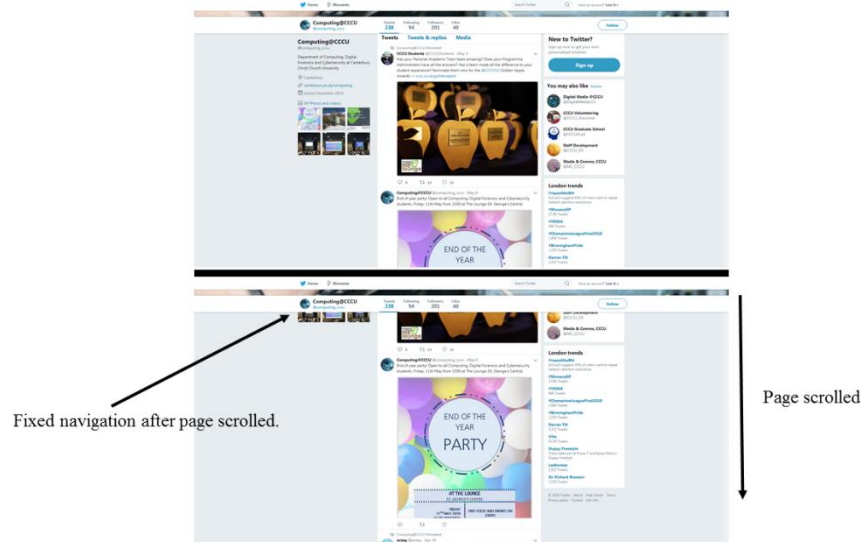


Figure 7.12 Navigation bar with CSS property position:fixed

The issue surrounding `position:fixed` elements is that they remain in place when OSIRT uses the scrolling capture method and therefore capture these elements with every scroll. To combat this, documents were scrubbed of `position:fixed` elements after the first scroll. Simply, every element within the Document Object Model (DOM) is traversed, and if an element is fixed, its `visibility` property is set to `hidden` (Listing 7.7).

```
(function() {
    var elements = document.querySelectorAll('*');
    for (var i = 0; i < elements.length; i++) {
        var position = window.getComputedStyle(elements[i]).position;
        if (position === 'fixed') {
            elements[i].style.visibility = 'hidden';
        }
    }
})();
```

Listing 7.7 Traversing the DOM for `position:fixed` elements

After capture, the document is traversed again in the same manner, only this time the `visibility` property is set to `visible`.

Users of devices that had high DPIs (Dots Per Inch) flagged another issue with screen capturing. Windows scales applications automatically for devices with high density displays, and these are typically in ranges of percentages; e.g. “scale an application to

appear 150% larger on high DPI displays”. On high DPI displays, CefSharp would be automatically scaled and resized. However, this rescaling would break the calculations used to capture the scrolling screenshots, resulting in screenshots losing and missing information (Figure 7.13). After researching this issue and having a discussion with the maintainer of CefSharp, it was established that it was too complex to precisely know how much scaling had been performed thereby making it impossible to apply an offset to any calculations used to create a screenshot. Instead, OSIRT forces the device to never scale the browser by setting the "force-device-scale-factor" to "1" in the CefSettings. The downside to this is that OSIRT's browser does not scale on high DPI devices.

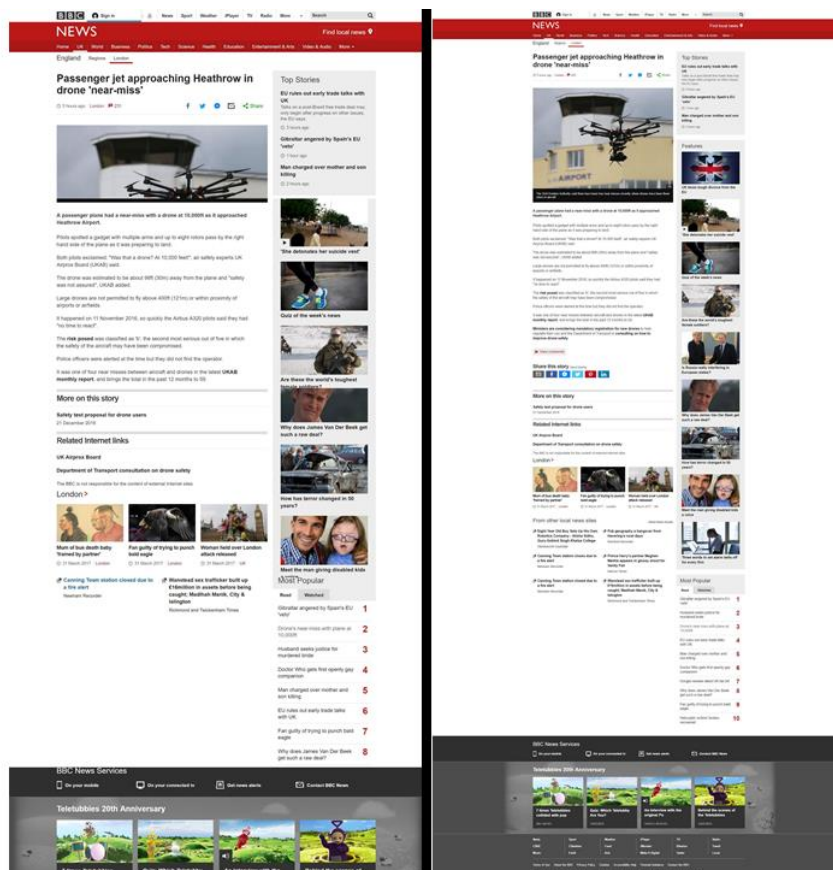


Figure 7.13 Screenshot with 150% DPI scaling (left) and screenshot with 100% DPI scaling (right). Note screenshot on left is missing elements.

Issues surrounding DPI go beyond screenshots and are discussed in more detail in the section ‘Implementation reflection’ (section 7.22).

7.7.2 Timed Capture

The time screenshot feature has been updated to provide used with a countdown timer within the status bar.

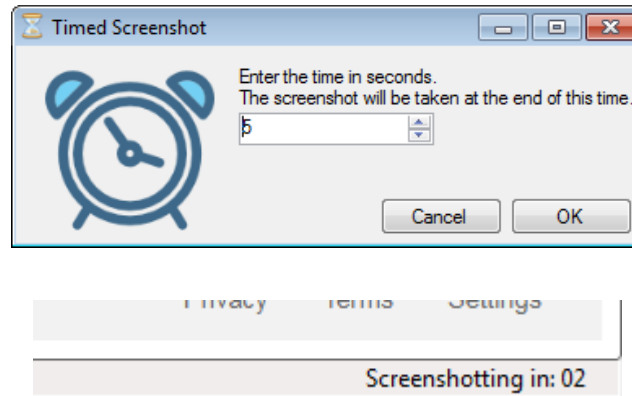


Figure 7.14 Timed screenshot dialog and countdown timer in status bar

7.7.3 Snippet and current view

These two options remained the same and were brought over from the prototype.

7.8 Video Capture

As requested from the prototype users, the complexity of video capturing has been removed, and video capture is integrated directly into the UI. Video capture can be accessed by clicking the video capture icon to start recording, and pressing again to stop recording. Once the video capture has completed, the Video Previewer (Section 7.10.2) is displayed for the user to log their capture.

Video capture still provides the 'marker window', whereby a user can record a certain section of their screen within the marker window, but this option has been placed in a drop-down next to the video capture button. Complex video capture options, such as setting the frames-per-second (FPS) and bitrate have been placed into the options menu.

The video capture is not without its flaws, however. There have been reports from users that videos can playback ‘sped-up’ or ‘blank’, while there are known issues¹³ surrounding the blank screen problem, there is still no solution as to why playback is sped-up. Currently, tests are being conducted to integrate a new video capture library into OSIRT.

7.9 Download management

Complexities from the previous download manager were completely removed as CefSharp provides an `IDownloadHandler` where two methods, `OnBeforeDownload` and `OnDownloadUpdate`, need to be implemented. The `OnBeforeDownload` method provides options to show default, Windows-styled ‘save as’ dialogs; an interface the users should be familiar with. The `OnDownloadUpdate` method provides events to update the interface with progress percentage and to check if the file has completed downloading. As the user can save the file anywhere on the file system, a copy of the file is placed within the case container as a master copy.

¹³ Issue for video capture recording a black screen: <https://github.com/joe-williams-cccu/OSIRTV2/issues/1>
(Last accessed: 03 August 2018)

```

public class DownloadHandler : IDownloadHandler
{
    public event EventHandler DownloadUpdated;
    public event EventHandler DownloadCompleted;

    public void OnBeforeDownload(IBrowser browser, DownloadItem downloadItem, IBeforeDownloadCallback callback) {
        if (!callback.IsDisposed) {
            using (callback)
            {
                callback.
                    Continue(downloadItem.SuggestedFileName, showDialog: true);
            }
        }
    }

    public void OnDownloadUpdated(IBrowser browser, DownloadItem downloadItem, IDownloadItemCallback callback) {
        DownloadUpdated?.Invoke(this, new DownloadEventArgs(downloadItem));

        if (downloadItem.IsComplete) {
            DownloadCompleted?.
                Invoke(this, new DownloadEventArgs(downloadItem));
        }
    }
}

```

Listing 7.8 Download handler implementation

7.10 Previewers

This version of OSIRT has made the collection and logging consistent across artefact types. The general UI design for the previewer (Figure 7.15) made better use of vertical space than the prototype, offering more emphasis on the artefact collected.

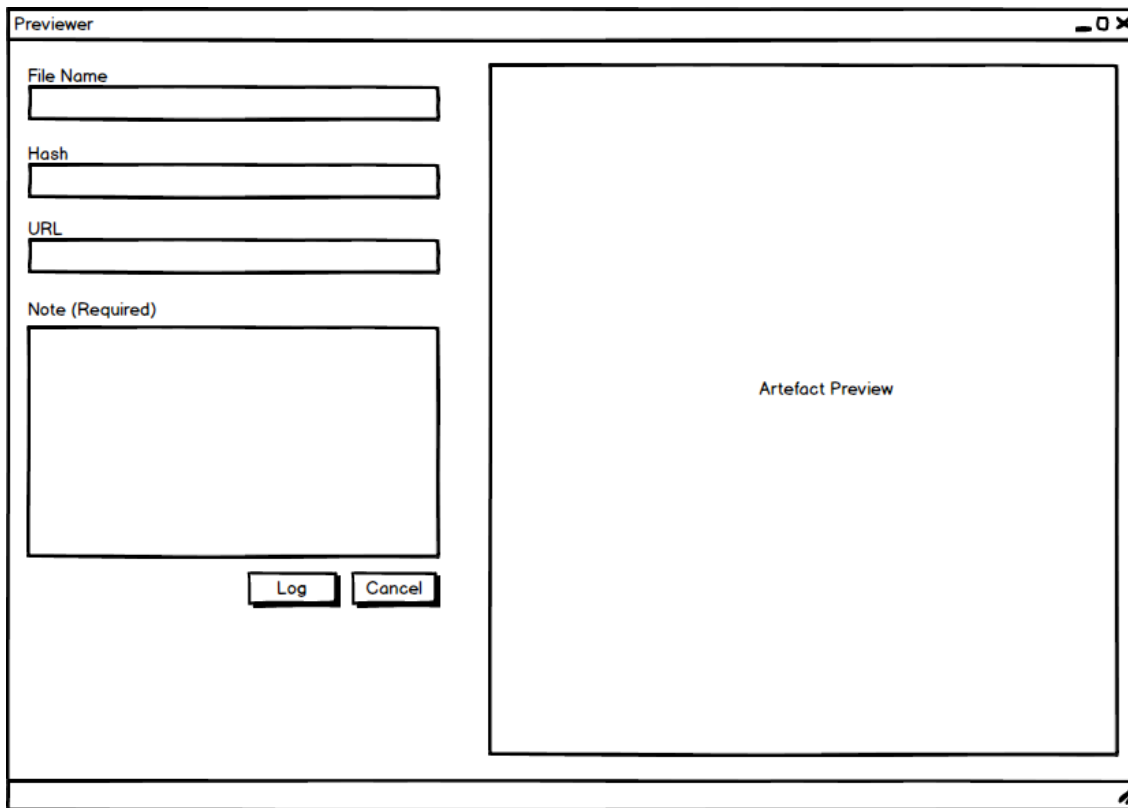


Figure 7.15 General previewer wireframe

The design of the previewers focused around the use of inheritance, whereby all previewers inherited from a Previewer base class (Figure 7.16).

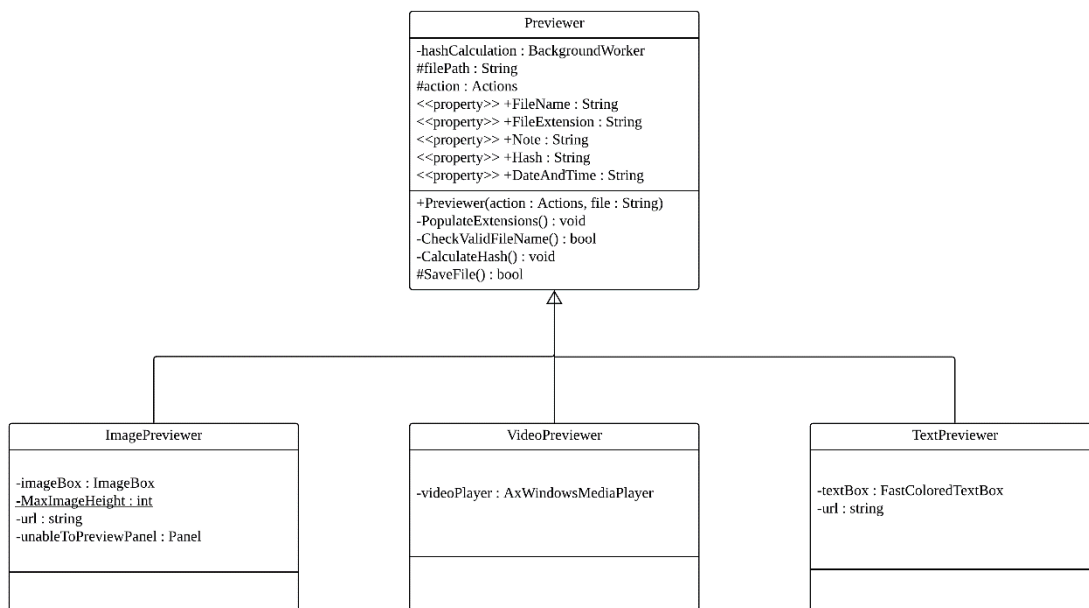


Figure 7.16 UML class diagram representing the previewers

7.10.1 Image previewer

From the three previewers, the Image Previewer (Figure 7.17) proved to be the most complicated to implement due to the complex nature of image sizes. It was evident from the prototype that the default `PictureBox` within the .NET Framework was inadequate at displaying large images so a suitable replacement was needed. `ImageBox` (Moss, 2018) was selected as it met the core criteria of being free, open source and capable of displaying images larger than the default `PictureBox`. `ImageBox` also provides other options such as zooming and dragging.

The image previewer can save as PNG, PDF and JPG. By default, PNG is selected as it is a lossless format and, additionally, web pages generally have a repetitive pallet choice, meaning screenshots should be relatively small in file size due to PNGs compression algorithm. The saving as PDF feature was brought over from the prototype based on the feedback when at the College of Policing. Saving as JPG was a request made by an officer from Dorset police whose force had a policy of saving as JPG. The `Magick.NET` library is used to convert the images from PNG to either JPG or PDF. The images are rehashed before being logged as the hash displayed in the Image Previewer is for the PNG within the image cache.

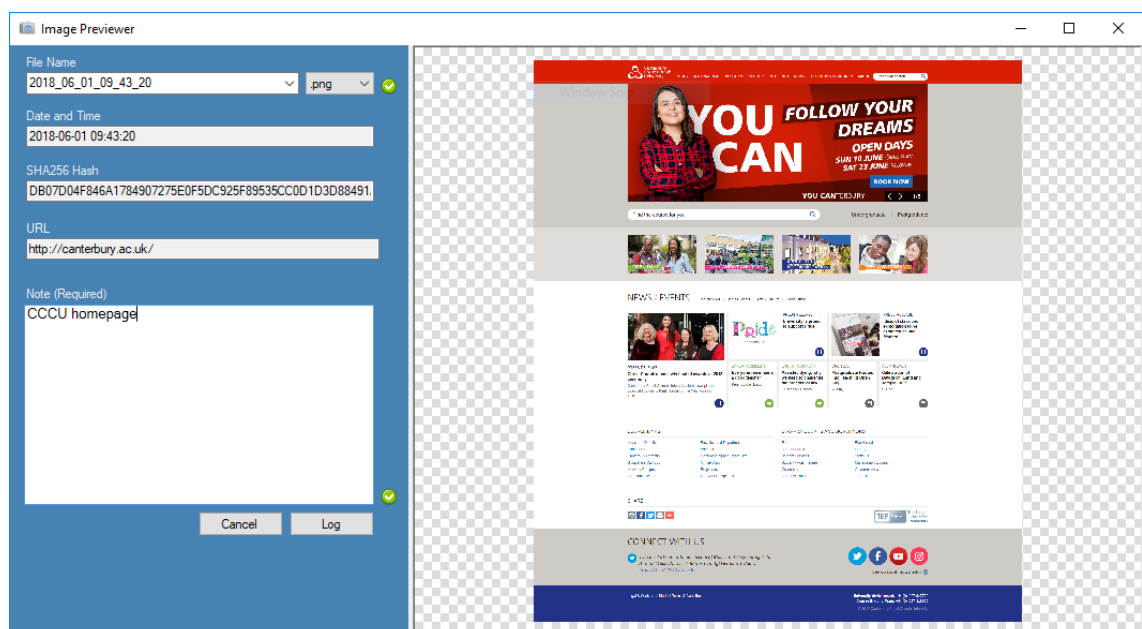


Figure 7.17 Image Previewer

7.10.1.1 Issues surrounding the display of images

There are potentially several outcomes when attempting to load an image. Firstly, the image loads correctly and no further action is required. Secondly, the image is too big to load within the ImageBox (i.e. consumes too much memory due to its size) and requires resizing for display purposes (Figure 7.18). While the displayed screenshot is resized, the original image remains the same and that is the one logged. Thirdly, a failure to display the image at all due to an exception being thrown in the Magick.NET library or ImageBox library. In this instance, a message is displayed to the user informing them that OSIRT is unable to display the image, but can view it by clicking a link.

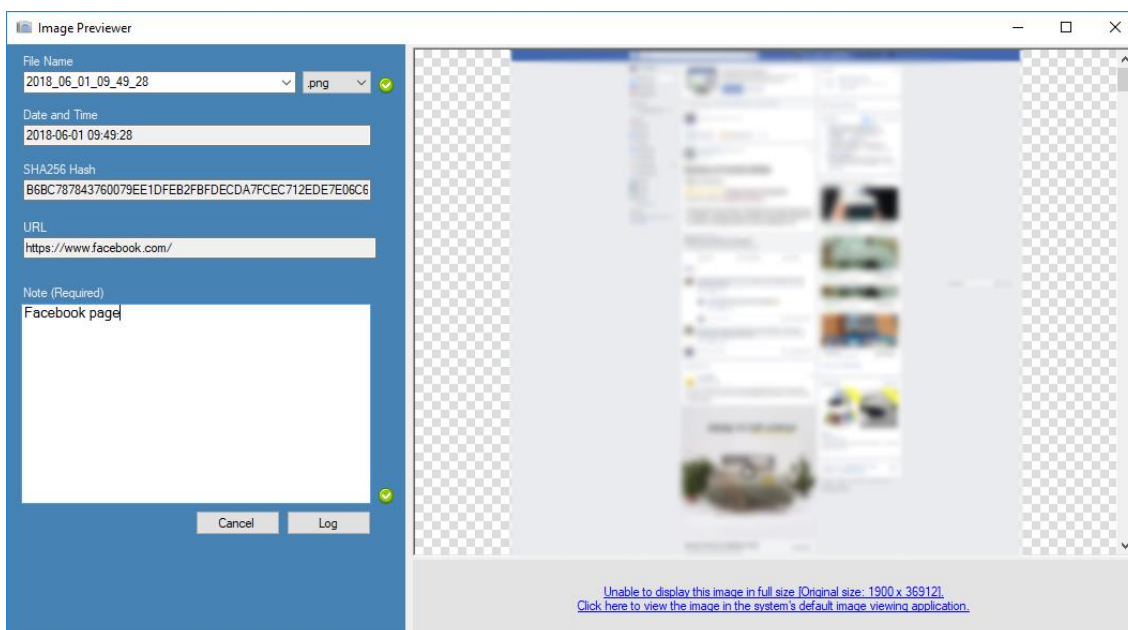


Figure 7.18 Large screenshot in the Image Previewer that has been resized

7.10.2 Video Previewer

The Video Previewer (Figure 7.19) uses the `AxWindowsMediaPlayer` control available in the .NET Framework for displaying video content. The control can play a range of formats, including MP4, which is OSIRT's video format.

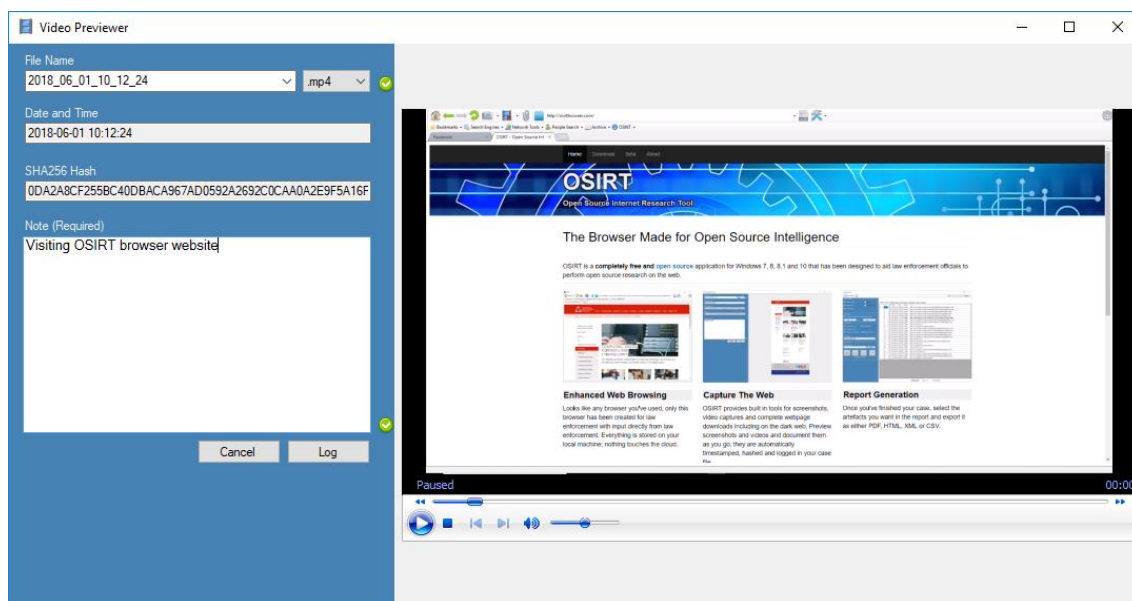


Figure 7.19 Video Previewer

7.10.3 Text Previewer

The Text Previewer (Figure 7.20) was integrated after a cognitive walkthrough in Chapter 8 was conducted and discovered an inconsistency with how artefacts were displayed. The Text Previewer implements the FastColoredTextBox (Torgashov, 2018) library, as it provides syntax highlighting.

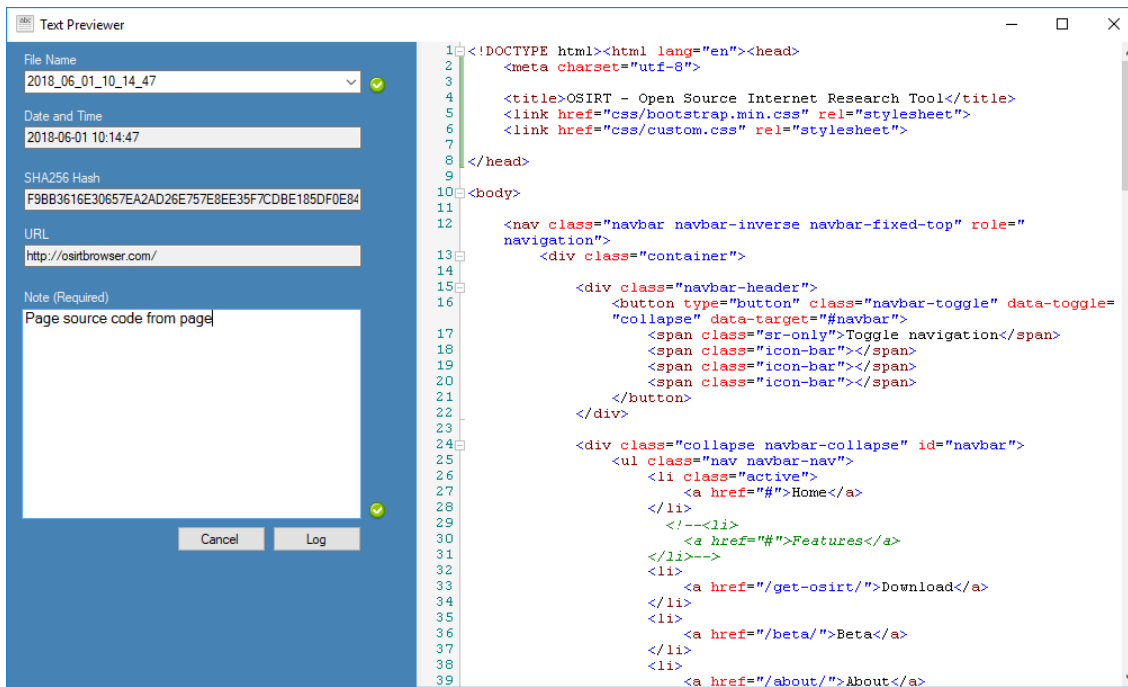


Figure 7.20 Text Previewer with source code from a web page

7.11 Attachments

The attachment form was redesigned from the prototype as it did not provide users with feedback on their actions. OSIRT shows the file to be uploaded, its hash value and size (Figure 7.21).

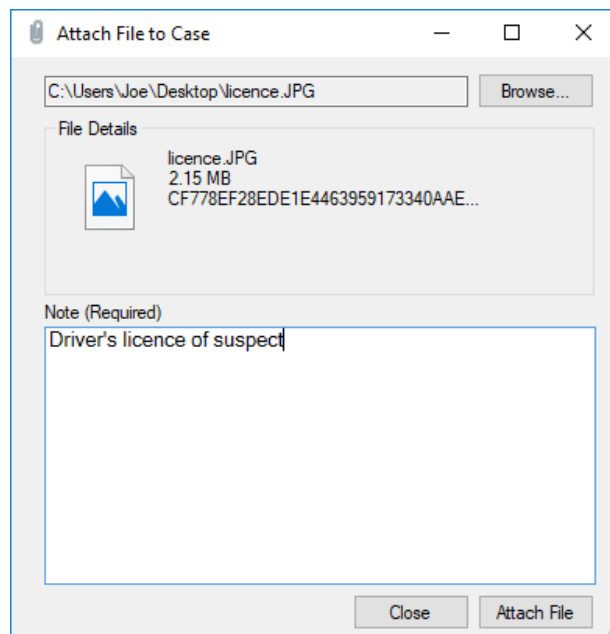


Figure 7.21 Example attachment to case

Upon attachment, users are provided with positive feedback on successful upload and given the opportunity to attach another file (Figure 7.22).

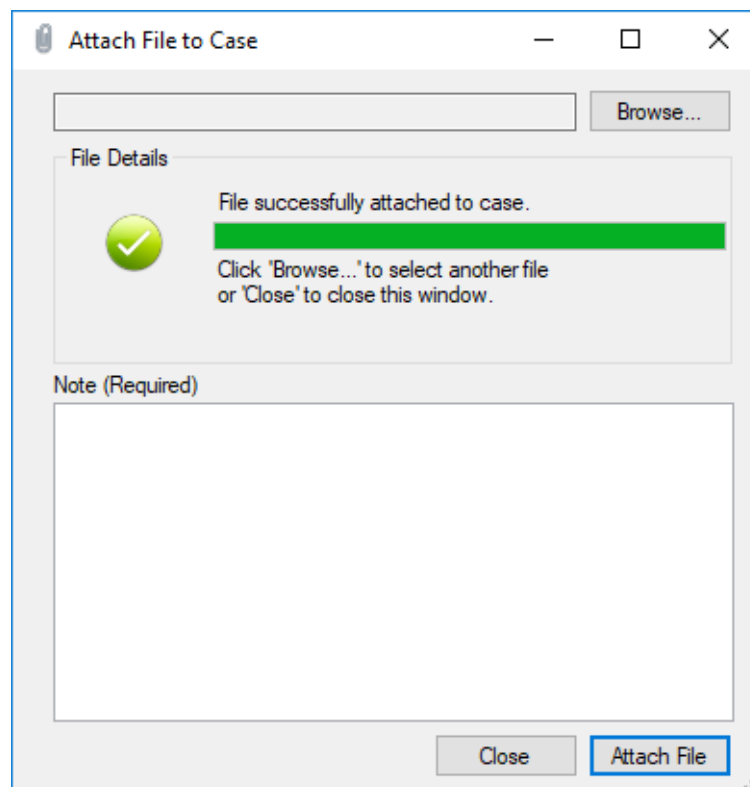


Figure 7.22 A successfully attached file

7.12 Accessing the Dark Web

As seen from feedback in Chapter 6 the ability to access the dark web was a feature officers wanted to see added to OSIRT. This section discusses the implementation of Tor into OSIRT.

Tor provides an 'expert bundle' (Tor, 2018) that houses the core executables and libraries of Tor, which allows for integration into custom applications. Integration of Tor into OSIRT involved two core components: Tor's expert bundle and the use of Tor.NET, a library for integrating Tor into .NET applications. Tor.NET handles much of the complexity of dealing with the networking side of Tor, with the difficulties falling into integrating Tor into CefSharp itself. Firstly, the proxy server within CefSharp must be set via `CefCommandLineArgs` using the proxy-server key/value pair. By default, Tor uses localhost via socks5 on port 9050 and this address and port number was kept for ease

of implementation. Secondly, the Tor process (Tor.exe) must be started by creating a new Process (Listing 7.9). As part of the Processes construction, the Tor executable's window is hidden as it is a command prompt window.

```
settings.CefCommandLineArgs.Add("proxy-server", "socks5://127.0.0.1:9050");
Process[] previous = Process.GetProcessesByName("tor");
if (previous != null && previous.Length > 0)
{
    foreach (Process p in previous) p.Kill();
}

var process = new Process
{
    StartInfo = new ProcessStartInfo
    {
        FileName = @"Tor\Tor\tor.exe",
        CreateNoWindow = true,
        WindowStyle = ProcessWindowStyle.Hidden
    }
};

process.Start();
```

Listing 7.9 Starting Tor process

Once the Tor process was started and the proxy server was set, control was then handed over to the Tor.NET library by creating a ClientRemoteParams (Listing 7.10) instance and setting the address (localhost) and port number (9050).

```
ClientRemoteParams remoteParams = new ClientRemoteParams();
remoteParams.Address = "127.0.0.1";
remoteParams.ControlPassword = "";
remoteParams.ControlPort = 9050;

Client.CreateForRemote(remoteParams);
```

Listing 7.10 Links Tor process with Tor.NET

When in Tor mode, as denoted by the purple address bar, OSIRT loads and takes the user the *.onion* version of DuckDuckGo (Figure 7.23). Users can use OSIRT's features and log websites as they would normally.

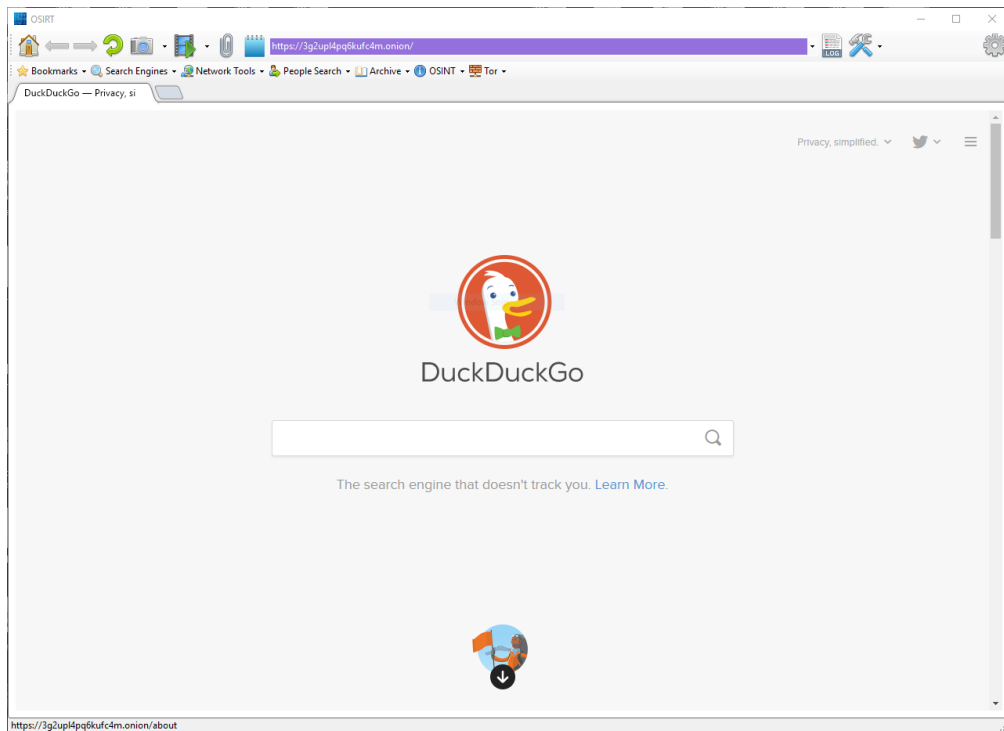


Figure 7.23 OSIRT in Tor mode, as denoted by the purple address bar.

Users can start Tor by checking the 'Load Tor' button in the Advanced Browser Options menu (Figure 7.24) before loading or creating a case.

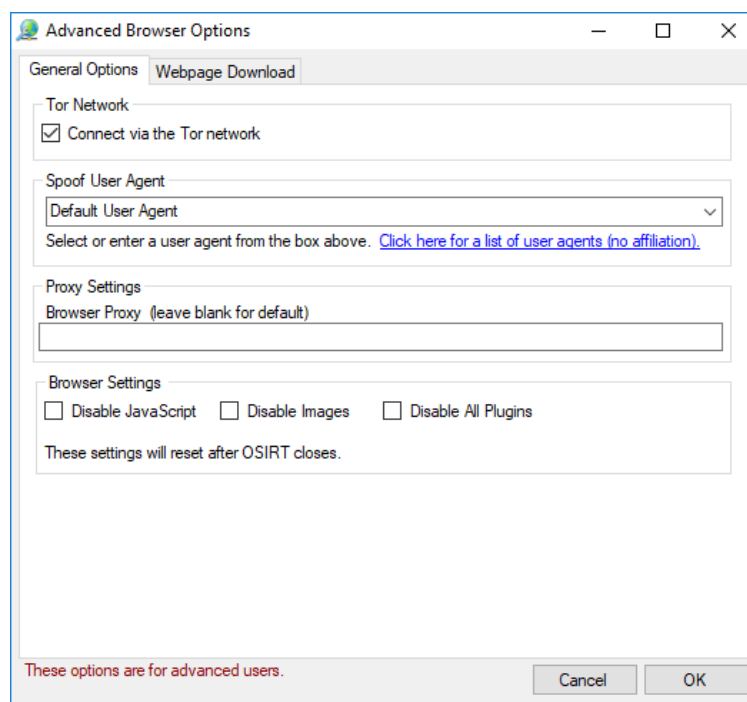


Figure 7.24 Starting Tor in the Advanced Browser Options

7.13 Link extraction

On request of an officer, extraction of all links from a web page was added. A user can execute this tool by right-clicking on a page and selecting ‘Extract all links from page’. This tool works by obtaining a copy of the page’s source code and parsing it for links using HtmlAgilityPack (Mourrier and Klawiter, 2012) to query the document for all anchor tags (Listing 7.11).

```
string source = await GetBrowser().MainFrame.GetSourceAsync();
HtmlAgilityPack.HtmlDocument doc = new HtmlAgilityPack.HtmlDocument();
doc.LoadHtml(source);
string links = "";
foreach (HtmlNode link in doc.DocumentNode.SelectNodes("//a[@href]")) {
    string value = link.Attributes["href"].Value;
    if (value == "#") continue;
    links += value + "\n";
}
```

Listing 7.11 Extracting links for the page using HtmlAgilityPack

7.14 Social Media ID extraction

Social media websites allow the creation of aliases or vanity tags, that is, a friendly identifier for the user to distribute to friends. For example, Facebook provide vanity URLs such as facebook.com/joe.bloggs.99. These vanity tags, though, make it difficult for investigators as the user can change them at will. Every user on Facebook, and other social media websites, has a unique identifier that does not change, and this is what investigators look for when conducting social media investigations. OSIRT provides ID finders for social media websites: YouTube, Facebook, Twitter and Instagram. Other social media platforms are continually added.

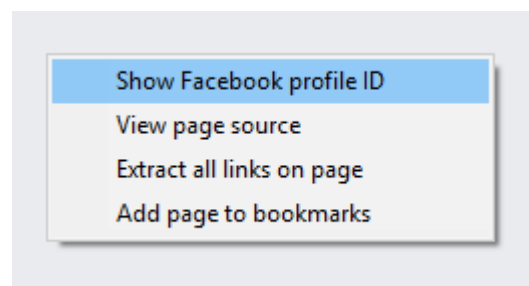


Figure 7.25 Facebook ID finder

Users right-click on a profile and select ‘Show [website] profile ID’ (Figure 7.25), this then displays the ID to be copied to clipboard.

7.15 Advanced browser options

7.15.1 User agent spoofing

As requested by users of the prototype, OSIRT provides a method to spoof the user agent string. User agents for web browser requests typically contain the browser type, version and operating system. These details are sent to the server when the web browser makes a request. An example user agent is seen below:

```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36
```

The pertinent elements of this user agent are that the user is using Windows NT 6.1 (Windows 7) and Google Chrome version 66. This user agent string is a trace left in a server log. If an officer visits a website repeatedly, they may wish to change their user agent footprint. OSIRT provides a pre-defined list of user agents or an option for a user to enter their own user agent. Setting of the user agent can only be done before `Cef.Initialize` is called, so it cannot be changed once a single browser instance has been created.

Setting of user agents goes beyond hiding footprints, they can also be changed to return a different style website. For example, a server may return a mobile version of a website if the user agent is from a mobile browser¹⁴. An application of this was seen where a member of the public reported a crime¹⁵ surrounding something occurring on a website. When the investigating officer had a look on the offending website, everything was normal and the officer was not seeing what the reporter was. It transpired that this officer had viewed the website on their desktop PC, and the person reporting the crime on their

¹⁴ While it is considered bad practice to inspect the user agent string to determine a user’s device, many web developers still use it for precisely that purpose.

¹⁵ Details of this are deliberately vague for reasons of disclosure.

mobile phone. When the officer then spoofed their user agent via OSIRT to a mobile user agent, they were then able to capture the offending material.

7.15.1.1 User agent spoofing – useful?

From personal communications with officers, there is a split between whether the user agent string is useful and worth spoofing. Taking the above example user agent as a trace being left on a server, a Windows 7 machine using Google Chrome is not particularly remarkable: it is entirely normal, in fact. The issue comes when users may get creative with their choice of user agent, and that may cause suspicion from a server owner. For example, using Iceweasel on a Gentoo Linux is a lot less common than Google Chrome on a Windows 7 machine. When it comes to open source research, it is the author's opinion that it is better to blend in than be different and stick out.

7.15.2 General browser settings

Users can opt to disable JavaScript, images and plugins, such as Adobe Flash. If browsing via Tor, these options are recommended to be disabled.

7.16 Webpage filtering

Webpage filtering is one of the larger and more complex features within OSIRT and requires an explanation of the HTTP request/response cycle to understand what this feature is and is not.

At the request/response cycle's simplest, the client makes a request for a resource (HTML, for example) that is stored on a remote server on the Internet (Figure 7.26). The server checks if this resource exists and returns the requested resource via a response. The client making the request in this instance is CefSharp.

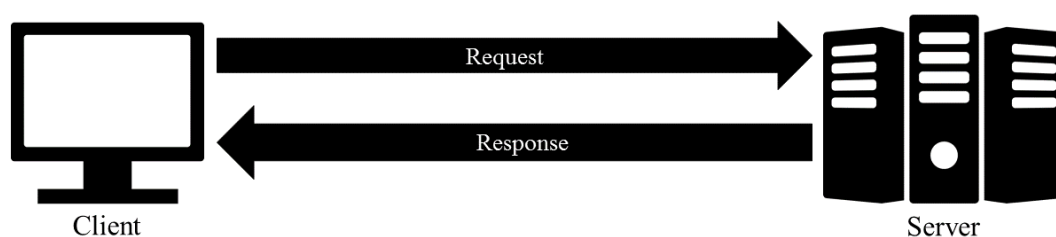


Figure 7.26 Request/response cycle between a client and server

For every resource required by the client, a new request is made to the server and the server will send a response to the client. The impact of this request/response cycle is important for those officers who do not wish to leave a heavy footprint but want to download webpages. As discussed in section 6.4.3, web scrapers will fetch a document, parse it, establish HTTP connections for each request and the server will respond with the appropriate content. In a web browser, a user will have already visited the web page and the document would have loaded, triggering the request/response cycle. If the user were to then ask, ‘scrape this webpage’, the content would be requested again; leaving an unnecessary footprint and a potential for a scraper to obtain data that was not requested. A solution to this is to filter and capture responses as they are returned by the server.

7.16.1 Filtering and capturing responses

OSIRT can capture these responses by implementing two classes: `IRequestHandler` and `IResponseFilter`. `IRequestHandler` handles all actions associated with a request, for example what action to perform before a resource loads, before the browser navigates to a new document and the handling of responses via the `GetResourceResponseFilter` method. `GetResourceResponseFilter` returns an instance of an `IResponseFilter`, which is implemented by the `MemoryStreamResponseFilter` class.

`MemoryStreamResponseFilter` implements two methods: a method to initialise the filter, `InitFilter`, and the `Filter` method (Listing 7.12), which handles the response stream and performs the bulk of the filtering.


```

FilterStatus Filter(Stream dataIn, out long dataInRead, Stream dataOut, out
long dataOutWritten) {

    if (dataIn == null) {
        dataInRead = 0;
        dataOutWritten = 0;
        return FilterStatus.Done;
    }

    dataInRead = dataIn.Length;

    if (dataIn.Length > dataOut.Length) {
        var data = new byte[dataOut.Length];
        dataIn.Seek(0, SeekOrigin.Begin);
        dataIn.Read(data, 0, data.Length);
        dataOut.Write(data, 0, data.Length);

        dataInRead = dataOut.Length;
        dataOutWritten = dataOut.Length;
        return FilterStatus.NeedMoreData;
    }

    dataOutWritten = Math.Min(dataInRead, dataOut.Length);
    dataIn.CopyTo(dataOut);

    dataIn.Position = 0; //reset
    dataIn.CopyTo(memoryStream);

    return FilterStatus.Done;
}

```

Listing 7.12 Filtering responses

Responses are managed by Stream instances, Streams are essentially a wrapper mechanism for raw bytes. While the request response is streaming into the rendering engine, the Filter method checks that the data received is complete, otherwise the FilterStatus is set to NeedMoreData. If the response has completed, the data is copied to the member memoryStream, and the filtering is complete.

The MemoryStreamFilter class contains a public property Data, which returns a byte[] of the completed response. The .NET Framework provides the File.WriteAllBytes method that saves the file to a specified path.

Once the resource has completed loading, the OnResourceLoadComplete is called within the RequestHandler class. The loaded resource is placed within a HashSet<ResourceWrapper>, where ResourceWrapper holds details about this particular resource. The ResourceWrapper and its contents are seen in Table 7.1.

Resource Item	Resource Description
Request URL	The location of the resource. For example, <code>http://canterbury.ac.uk/images/computing/example.png</code>
Resource Type	The type of resource. For example, stylesheet, image, script.
Mime Type	The resources mime type. For example, image/png.
Data	The raw data of the resource as a byte[]
Request Identifier	Each request is given an ID, starting with 1 and incrementing for each new request.

Table 7.1 Properties within ResourceWrapper

Pages are saved when the user presses the web page download icon. The entire contents of the page are zipped up and placed in the audit log, and another working copy is placed in the user's documents.

7.17 Auto-scrolling

An officer requested if page auto-scrolling could be implemented, as pages like Facebook can take a long time to manually scroll. While the implementation of auto-scrolling works in a similar way to how the scrolling screen capture works, there is the added complexity of having to wait for a page to lazily-load its content, then continue scrolling.

The feature was relatively simple to conceptualise: execute a JavaScript function that scrolls the page on a timer event and call that function every n milliseconds. The `setInterval` function in JavaScript is an ideal candidate as it takes two arguments: the JavaScript function to execute and a time in milliseconds that sets the interval to execute the function (Listing 7.13).

```

function getDocHeight() {
    return Math.max(
        body.scrollHeight,
        body.offsetHeight,
        html.clientHeight,
        html.scrollHeight,
        html.offsetHeight );
}

var scroll = (function() {
    var body = document.body;
    var html = document.documentElement;
    var docHeight = getDocHeight();

    window.scrollTo(0, docHeight);

    if (prevDocHeight == docHeight){
        clearInterval(pidScrollToEnd);
        return true;
    }
    prevDocHeight = docHeight;
})();

var pidScrollToEnd;
(function() {
    prevDocHeight = 0;
    window.scrollTo(0, getDocHeight());
    pidScrollToEnd = setInterval(scroll, [scrollTime]);
})();

```

Listing 7.13 Auto-scrolling a web page using JavaScript

Some Internet connections are slow, and lazily-loaded content may take longer to display before the Document's height is correctly recalculated, which results in prematurely stopping the auto-scroll function. To help the user with a slow connection, the [scrollTime] argument can be set by the user within OSIRT's options. Slower Internet connections are advised to use a higher time, such as five seconds, before the scroll function is executed again to ensure the Document has loaded its content.

7.18 Audit log

The Audit Log was redesigned based on the prototype's feedback. The audit log now splits actions into their own tab, provides a search feature of logged actions and a streamline ability to preview collected artefacts.

7.18.1 Artefact gridview

From previous observations and feedback, placing all artefacts in one complete chronological GridView was too much for users to absorb and find relevant artefacts. The main audit log is now tab-separated into meaningful areas and splits out common

actions, but still maintains the ‘complete’ chronological audit for those who would like it. Table 7.2 describes these areas.

Tab	Purpose
Websites Loaded	Lists all visited websites in chronological order.
Website Actions	Lists all possible actions taken on a website. For example, screenshots and downloads.
OSIRT Actions	All actions OSIRT has taken. For example, opening and closing cases.
Attachments	Artefacts attached to this OSIRT case
Videos	All videos captured using the screen recorder, or videos downloaded.
Complete	A complete set of all actions in chronological order.

Table 7.2 Tabs within the audit log

7.18.2 File previewer

The audit log provides a file previewer (Figure 7.27) that rescales images for easier viewing when looking through the audit log. If the file is not able to be previewed, then the previewer displays the file icon associated with the extension, if available.

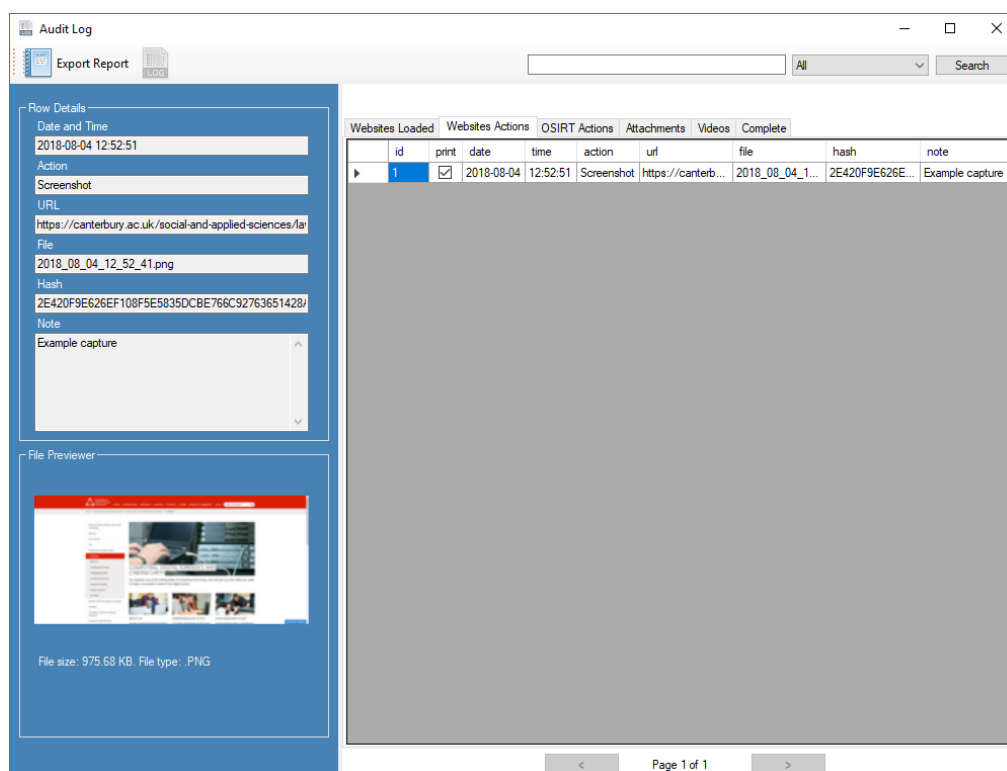


Figure 7.27 Audit log with example screenshot (bottom left)

7.18.3 Searching audit log

As investigations grow larger, the ability to be able to search through the audit log to find artefacts makes the investigators job easier. OSIRT provides full searching across all artefact types, along with individual artefact type searches (Figure 7.28).

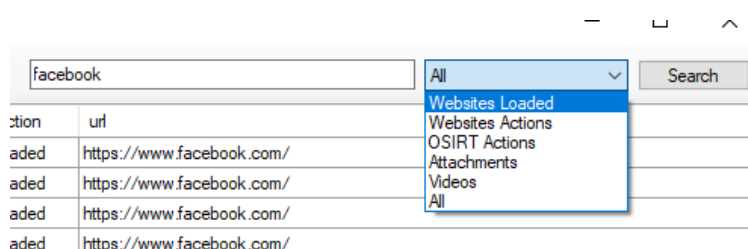


Figure 7.28 Example of searching the audit log

7.19 Reporting

Report exporting has been streamlined from several pop-up forms seen in the prototype, to one Panel within the Audit Log. Users select the 'Export Report' option and it swaps the 'row details' out for the reporting options. Users can select what elements of the report they would like to export on an individual or table level. For example, all OSIRT actions

can be omitted for dissemination purposes. Reports can contain links to videos and images or removed for reasons of disclosure.

Report Selection

☒ Websites Loaded ☒ Videos

☒ Website Actions ☒ Attachments

☒ OSIRT Actions

Select Dates

☒ Print everything between these dates only (inclusive)

01 May 2018 and 29 May 2018

Report Options

☒ Link images in report ☒ Print audit notes

☒ Link videos in report

☒ Open report once created ☒ Merge case notes

GSCP Stamp:

Not Protectively Marked

Export Report

Export Location

\\stafs-nhr-01.ccad.canterbury.ac.uk\jw758\ Browse...

HTML PDF CSV XML


Export report as HTML

Figure 7.29 Report exporting

Added to reporting from the prototype is the ability to merge the case notes with the report in chronological order and to export everything within selected dates. An officer requested this date selection option because several days can pass at the start of conducting an open source investigation with nothing of value being discovered, this option omits large chunks irrelevant data if needed.

Reports can be exported as HTML, PDF, CSV and XML.

Not Protectively Marked



Audit Log

Investigating Agency: CCCU
 Operation Name: ccu-999
 Case Reference: mobile-phone-theft-09812
 Evidence Reference: CCCU
 Notes: Inspecting ebay website for suspected stolen mobile phones.

Created: 2019-05-29 15:25:09 (GMT Daylight Time)

Not Protectively Marked

1

Figure 7.30 Report front page

Not Protectively Marked

Date	Time	Action	URL	File	Hash	Note
2019-05-29	15:20:11	Case Loaded		mobile-phone-theft-09812	[No Hash - Case Created]	
2019-05-29	15:20:25	Loaded	https://www.google.com/imgres?id=...			
2019-05-29	15:20:41	Loaded	https://www.ebay.co.uk/			
2019-05-29	15:21:05	Snippet	https://www.ebay.co.uk/	2019_05_29_15_20_52.png	621A355D6A4E4485F0K07F8E 7E9A8A55C7017C718859A5 suspected stolen phone 4C205F13	
2019-05-29	15:21:32	Screenshot	https://www.ebay.co.uk/	2019_05_29_15_21_16.png	9CDB8C9F9AC9A12256A4C790 8E2CF8E3962A8C33D9F581C8 A73635246	list of mobile phones for sale on ebay
2019-05-29	15:21:55	Attachment		IMG_0474.JPG	8943B8E1C3F1C3780C8A0F 4B130A47C2CF1EEF189A43E41F 76838FD	Photo of IMEI number from CC Stephans
2019-05-29	15:24:11	Loaded	https://www.ebay.co.uk/itm/7... from:W888,head:2380002,info: 7013,TRC2,TRC2,AD10,imobileph one:TRC2,view=mobile:iphone,s can=0			

Not Protectively Marked

2

Figure 7.31 Example report page

7.20 Bookmarking

OSIRT has integrated bookmarking after several requests from users. Bookmarking is performed on an application level, so bookmarks are available through all cases.

Bookmarks are saved as delimited key/value pairs in a file, `favourites.config`, within the user's application data folder.

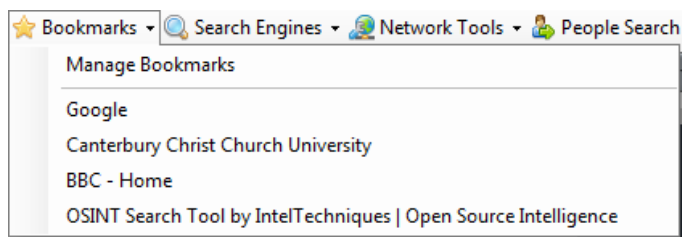


Figure 7.32 Bookmark menu

Bookmarks are placed into the toolbar (Figure 7.32) and will open a new tab when clicked. Users can also manage bookmarks by selecting the 'Manage Bookmarks' menu item (Figure 7.33).

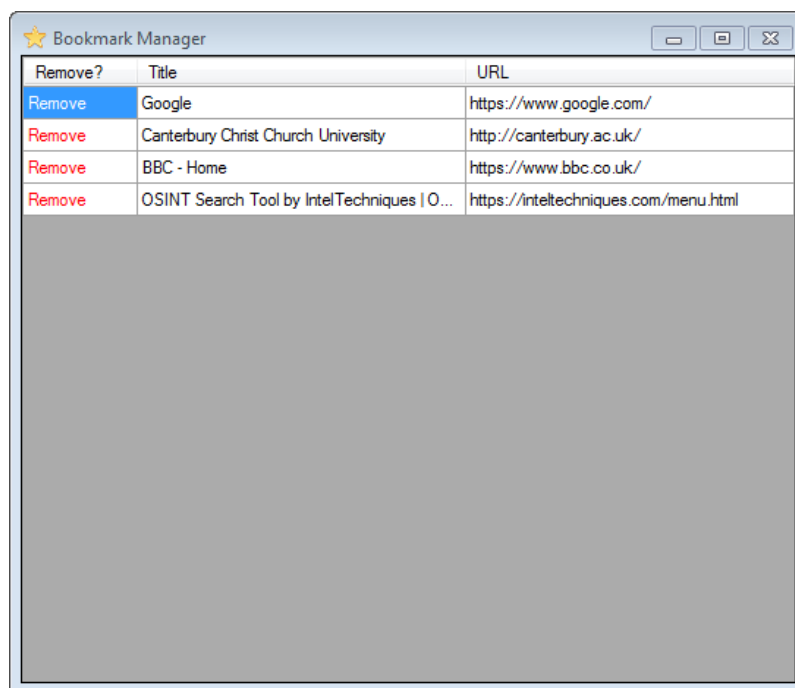


Figure 7.33 Bookmark manager

7.21 Preventing multiple OSIRT instances

An issue seen in the observations was that users attempted to launch OSIRT when it was already open and not realising they already had their case open. This often caused confusion in that users thought they had lost their case, when in reality it was already

loaded in another OSIRT instance. The solution is to create a singleton application by means of implementing a `Mutex` (Allen, 2004), this prevents multiple instances of OSIRT and brings the application to the user's focus if they attempt to relaunch.

7.22 Implementation reflection

This section will reflect upon the implementation of OSIRT and some of the design decisions made.

7.22.1 WinForms

The use of WinForms as the basis of the user interface has caused issues with high DPI (dots per inch) displays. Typically, monitors have density of 96 DPI but as technology has advanced, this density has increased resulting in 'smaller' user interfaces. The implication being user interfaces need to scale to be visible on high DPI monitors. User interfaces designed for monitors with a DPI of 96 that get scaled are typically blurry. This is due to Windows taking charge of the scaling when the DPI goes past 100%, and renders the app as a bitmap and scales the bitmap when it then draws the application to the screen.

GUI frameworks available in the .NET Framework, such as WPF (Windows Presentation Foundation), are capable of scaling automatically as controls are described using a mark-up language, XAML (Extensible Application Mark-up Language), to generate the user interface. Conversely, WinForms uses either a WYSIWG approach or dynamic control creation at runtime. These two approaches in WinForms are, essentially, absolutely positioning controls to the interface (e.g. 'place this button 100x50 pixel button at this position on the Form'). While it is possible to scale WinForms applications, these are largely 'hacks' that attempt to solve a fundamental problem with WinForms applications and high DPI displays.

7.23 Chapter summary

This chapter looked at the creation of the release version of OSIRT along with the issues and design decisions surrounding its implementation. While this may be titled as a 'release' version, it is not to imply there will be a final, completed version. OSIRT is in continual development.

There have been several business opportunities for OSIRT surrounding its integration into OSINT training packages. OSIRT has also been integrated into a covert Internet intelligence¹⁶ package that has been taken up by several police forces into the UK, making OSIRT the de-facto tool for conducting open source research in this constabularies. The following chapters evidence this impact and contribution.

As of May 2018, OSIRT has had over 25,000 downloads from osirtbrowser.com. That does not include IT services downloading and integrating OSIRT into working environments, or its integration into commercial platforms; meaning the overall user-base is most likely much higher than the download count. OSIRT has been trained, and now used in, countries across the globe, from Barbados to Israel, and has been utilised in several high-profile investigations to capture artefacts from the web.

¹⁶ Due to reasons of disclosure, details surrounding this must be kept limited.

8 RESULTS AND DISCUSSION OF OSIRT'S INTEGRATION, IMPACT AND CONTRIBUTION TO LAW ENFORCEMENT: PART ONE

INTRODUCTION

The following two chapters focus on OSIRT and its integration, impact and contribution into law enforcement. This chapter has a heavier focus on usability and the user experience, and discusses the results of a cognitive walkthrough conducted on OSIRT, 72 SUS results, observations of three RITES courses, and interviews from 22 law enforcement officials along with 42 questionnaires results.

8.1 Cognitive walkthrough of OSIRT

The need for the cognitive walkthrough became apparent as while OSIRT was used heavily during the RITES course and provides SUS results and feedback, participants are not there to be dedicated usability testers. This prompted the cognitive walkthrough expert evaluation method to be conducted on OSIRT.

Usability issues that arose during the walkthrough have been split into several themes and are discussed for a better understanding of why they may affect usability. While a complete transcript of the walkthrough is not available in this chapter, it is available in Appendix E.

8.1.1 Method

A persona and scenario were created that would represent a typical user and use-case for OSIRT. The persona was a LEO who has experience in using computers and a web browser, and has been previously tasked with conducting open source research as part of an investigation. The open source investigation was conducted on a colleague who had provided consent to do so. 22 representative tasks were created that a typical LEO would perform to conduct open source research within OSIRT. These ranged from creating a new case, viewing and logging social media pages of the research target, to exporting the final report.

Tasks were performed slowly by the usability expert, using the cognitive walkthrough method, with the above persona in mind. If any issues were found, these were noted under the appropriate question related to the task. The developer and usability expert then discussed these areas for improvement, if necessary. If possible, the usability expert also noted potential solutions to these issues as part of the evaluation.

Penultimately, the developer took the suggestions for improvements; generating a new version of OSIRT. Any issues found via the cognitive walkthrough that were not implemented were noted with a justification.

8.1.1.1 About the evaluator and developer

The usability expert was a senior lecturer in Computing and had seventeen year's experience in teaching usability and human-computer interaction at undergraduate and postgraduate level. In addition to this, he had previously researched usability of software tools used by LEOs. The developer of OSIRT was a PhD researcher and had six year's experience of writing software. He was the sole developer of OSIRT, which he had been working on for two years.

8.1.2 Cognitive walkthrough results

8.1.2.1 Terminology inconsistencies

Part of OSIRT's domain vocabulary is the notion of "logging". For example, websites visited by the investigator are automatically 'logged' (i.e. placed into the case database).

To place a taken screenshot into the case container, a user must press a 'log' button. However, there are occasions, notably when a user views something text-based, for example, a webpage's source code the domain language changes from 'log' to 'save'. It is recommended that 'log' is the preferred term to be used, as it is consistently applied in other areas of the system. Further suggestions surrounding the 'save vs log' are noted under 'General Interface Issues'. This language mismatch was also noted when the user exports a report, checkboxes are available to omit particular logs, but the checkboxes do not match the names given with the tabs within the Audit Log.

OSIRT, inevitably given the nature of an investigative tool, uses technical language a user with only average computer experience may not understand. This is best seen when a user creates a new case and is presented with a drop-down list of hashing functions, with the only available prompt being a label above the list saying "Hash Function". A button is placed next to the drop-down list that may signify help, however, clicking this button did nothing so no help was provided to the user. It was suggested that due to its technical nature a sensible default should be offered.

OSIRT's naming for features are generally good, but there are times where ambiguity arises. Two examples highlighted were the names given to two functions. 'Snippet Tool', a feature that allows a user to select a section of the screen to save, like the 'Snipping Tool' seen in Windows, and 'Marker Window', a pop-up form used by the video capture library that allows the user to select a region of what is recorded. It is unlikely given these names that a user would recognise them as the action they wish to take. Changing the title for the 'Marker Window' provides more detail as to what it does, for example, a title of "Marker Window for Video Capture" is more descriptive.

8.1.2.2 General interface issues

OSIRT has a heavy reliance on icons. As OSIRT is a self-proclaimed 'browser on steroids', its usual look-and-feel is that of a web browser. Modern web browsers (e.g. Google's Chrome, Mozilla Firefox, and Microsoft Edge) use icons to represent their tasks and hide additional tools within a 'burger' menu. While the first few icons in OSIRT follow the browser 'metaphor', the extra icons may cause confusion for new users to the system. OSIRT's icons are, generally, suitable choices with one example being a camera to represent taking a screenshot. There are occasions, though, where icons do not

represent their action. One example of this can be seen within the Audit Log, where in order to export a report, the user has to click on an icon of a book. This is not immediately obvious and is recommended that the icon either be changed or to simply place text under the icon that says "Export Report".

There were also times where OSIRT could have provided enhanced user experience. This was seen when an error occurred, usually from invalid user entry, and focus was not brought to the text entry field (TEF) where the error occurred, or in the instance of multiple invalid entries, the first TEF containing the error. The suggestion was made to implement TEF focus on error.

OSIRT makes use of a status bar and label at the bottom right of the interface, with a default label style. This status bar is used to note when an action has been logged, or to display when an action will occur, such as the countdown timer for a 'Timed Screenshot'. However, the status bar is not particularly visible or obvious, and is not used consistently. An example of this inconsistency is seen when a full-page screen capture is in progress, while the page automatically scrolls and menu items are disabled, no status message is given to the user. The status label would be better improved if it was consistently used, as users would know looking there would provide feedback.

Better usage of cursor type was also noted. Firstly, the 'click and drag' cursor for the snippet tool uses the default pointer, a crosshair cursor would be better as it is not immediately clear that a click-drag operation is required.

OSIRT uses downward arrow buttons to denote additional options, these are placed next to certain buttons, such as the screenshot and video capture, on the menu bar. The extra options are used for both screenshot and video capturing functionality. The arrow placed next to these icons is too small, making it not immediately obvious as to what this arrow represents, or that it even houses extra features. A larger arrow that is closer to its 'parent' button icon was suggested to make these options clearer.

8.1.2.3 Interface mismatch

OSIRT's primary way of displaying user captured content was via the means of a 'Previewer'. For example, an 'Image Previewer' was displayed when a user captures an

image (such as a screenshot) and a 'Video Previewer' is displayed when a user captures a video. Details about these captures are placed in a panel on the left and a preview of the file is placed on the right. A user can either click 'Log', to move the capture into the evidential container, or 'Cancel'. However, when a user captured anything text related, e.g. the webpage's source code, this was displayed in a standard Form with a text control and a menu bar which displayed a floppy disk; with a ToolTip informing the user to 'Save'. The save button displayed a message box to confirm that the text had saved, this process was inconsistent with how the other Previewers worked, in that they close and display a confirmation in the status bar. It was recommended that a 'Text Previewer' was created to introduce consistency between captures.

8.1.2.4 Error reporting

Error reporting and prevention could be improved in places within the system. OSIRT uses built-in error providers from the programming framework, but these are only triggered when the user selects the 'Finish' action. One example is on the case creation screen; a user can freely enter invalid data, but is only informed when they click the 'Next' button that it is not valid. These errors can be triggered when a user is typing, or leaves a particularly TEF, allowing them to immediately rectify any invalid data.

8.1.2.5 Use of defaults

OSIRT could make better use of sensible default values. One such example is the 'Timed Screenshot' function, where a user is prompted to enter a time in seconds, the TEF uses a default value of zero. Nielsen (2005), notes that by using a "representative value" it can serve as an instruction to the user as to what they are expected to enter. OSIRT's default of zero seconds is not representative, because taking a 'Timed Screenshot' after zero seconds is just a regular screenshot. Instead, a suggested default of five seconds was recommended. Other examples where defaults were perhaps not considered as carefully as they should be, were within the search feature of the audit log, where the default search option only searches the "Websites Loaded". A better, and expected, search option would be to search all the contents of the Audit Log by default; yet this option was at the bottom of the search choice drop down menu.

Technical defaults, such as setting the hash function to MD5 when the case is created, need review. It was recommended OSIRT defaults to SHA variant, such as SHA256 or above.

8.1.3 Recommended changes to OSIRT

The walkthrough highlighted several usability issues, the subsequent enhancements are summarised below with the suggestions being implemented into an updated version of OSIRT, these are summarised in the bulleted list below with the full list of issues and developer's comments in Table 8.1.

- Use consistent language throughout the system.
- Provide accessible help for technical options.
- Use text with icons for non-obvious icons.
- When errors occur in TEFs, place focus where the error happens and place focus there immediately.
- Status bar requires consistent use and prominence.
- Provide default values that are representative where possible.

Usability Issues Discovered via Cognitive Walkthrough	Fixed - Developer Comments
Case Location name is unclear	Yes - Label text clarified
TEFs do not provide error messages/support for invalid entry until user clicks 'Next'	Yes - Errors now highlighted when user leaves/types in TEF
Hash help '[?] label' did not work when clicked on	Yes - Bug fix.
Hash default unsuitable	Yes - Defaults to SHA256
When capturing screenshot, no "capturing" message is displayed.	Yes - Displayed in status bar
Additional screenshot options are "hidden" in a small menu denoted by black arrow	No - From observations, this hasn't been an issue.
Timed screenshot default is not useful at 0 seconds.	Yes - Defaults to 5 seconds
Timed screenshot dialog text could be larger/spread out	No - From observations, this hasn't been an issue.
Timed screenshot countdown timer is not overly/immediately visible	Yes - Status is now used more consistently
When "snippet" selected not obvious that a click and drag is necessary to select area. Change of cursor recommended.	Yes - Cursor changed to +.
Language mismatch when saving text-based artefacts vs other artefacts ('save' vs 'log'). Use 'log'.	Yes - Consistent language and interfaces now used.
Source code previewer is different from other previewers (video/image) inconsistent. Implement a text previewer.	Yes - See directly above.
Start video recording icon ambiguous.	No - Recording icon not been an issue from observations.
Marker Window naming ambiguous	Yes - Marker window title text changed to clarify its functionality
Marker Window hard to resize, use handles	No - Unable to add handles to forms.
Double click row to open file in previewer	No - Felt it wasn't necessary.
Default field to search audit log should change to 'complete' by default	Yes - Default changed to complete
Unclear clicking the Log button means "go back to audit log"	Yes - text added next to icon
Unclear icon for exporting the report	Yes - text added next to exporting icon
Report formatting isn't ideal, but hard to improve upon	No - Not a trivial problem to solve. How do you represent arbitrary length data on an A4 sheet of paper?
Language mismatch between the checkboxes for tables to export and the tabs	Yes - Language now consistent
Print checkboxes in audit log not immediately obvious as to functionality	No - Not sure what to do about it.

Table 8.1 Issues discovered in cognitive walkthrough, with developer comments.

8.1.4 Discussion of cognitive walkthrough

8.1.4.1 Usefulness of the cognitive walkthrough

The cognitive walkthrough was a useful method for finding usability issues, discovering 22 issues in a relatively short period of time; with approximately 24 person-hours spent on the creating, conducting and analysing the cognitive walkthrough. Out of the 22 issues and suggestions identified, 15 were implemented by the developer. Out of the seven remaining issues not implemented, four were deemed unnecessary or have been previously observed to not be a problem by the developer and the remaining three were

either too complex to implement for the developer (two) or not possible to implement (one).

The cognitive walkthrough proved an efficient and useful form of evaluation and complemented the user-based evaluation mechanisms seen with beta testing and the SUS questionnaires. Additionally, while a RITES course runs for five days, the cognitive walkthrough found a good number of issues, more than typically received from beta testers from the RITES course. This highlights the importance of integrating other evaluation methods, such as the expert-based cognitive walkthrough, as part of the development cycle of software.

8.1.4.2 The 'co-operative cognitive walkthrough'

An aspect from this expert evaluation that was not initially considered before the cognitive walkthrough was conducted is the notion of a "co-operative walkthrough" where both the developer and usability expert are sitting side-by-side.

Spencer (2000) noted the concerns of having an expert evaluator evaluating user interfaces in teams causing tension and prolonged discussion. Spencer (2000) instead offers a different approach to the cognitive walkthrough called the 'streamlined cognitive walkthrough' (SCW). The SCW has been designed so it can be used to "defuse defensiveness" by setting out ground rules and save time reducing the number of steps in the walkthrough from four to two. The need of the SCW, however, was for large projects at Microsoft where there are, according to Spencer (2000), time pressures and often drawn-out, defensive discussions when using the regular cognitive walkthrough method.

For this study, there was no large team to appease: only a lone developer and the evaluator. From the perspective of the evaluator and the developer, the cognitive walkthrough certainly benefitted from having both present when the evaluation was being conducted. This 'co-operative cognitive walkthrough' provided a higher level of insight into the system for the usability expert. Additionally, the developer being there throughout enabled them to see the issues first hand. This is important, as for an evaluation technique to be valuable it must not only find usability problems, but it must enable them to be reported in a manner that makes sense. Ideally, this feedback is outlined in such a way as to influence change, John and Marks (1997) called this "persuasive

power". With the developer in the room witnessing the walkthrough, it was certainly persuasive.

This area of a 'co-operative cognitive walkthrough' could be considered further, as usability experts are unlikely to be domain experts, and vice-versa. It may be an effective method for lone developers, or for those individuals writing software within academia who will likely have access to a usability expert capable of conducting the walkthrough.

8.1.5 Summary of cognitive walkthrough

The cognitive walkthrough was a useful exercise which found a good number of potential usability issues. In fact, an expert evaluation method such as the cognitive walkthrough may have been well placed during the prototype phase. One limitation of this walkthrough was that no measure was made of the severity of the issues found, which while not a significant issue for the developer, it would have been useful to grade the severity as this may help prioritise larger issues. In this walkthrough, issues found were discussed between the expert evaluator and the developer at the time.

8.2 OSIRT System Usability Scale results

SUS questionnaires were distributed to 72 participants at the end of seven RITES courses. From the seven RITES courses, OSIRT scored an overall mean SUS score of 87.9 (Figure 8.1), with a standard deviation of 7.3, and a Cronbach's alpha of 0.715; which is acceptable internal reliability. The confidence interval is +/- 1.71. Given this, we can be 95.0% confident the population SUS score is between 86.21 and 89.63. Additionally, we can be 97.5% confident that the mean SUS score is above 86.21.

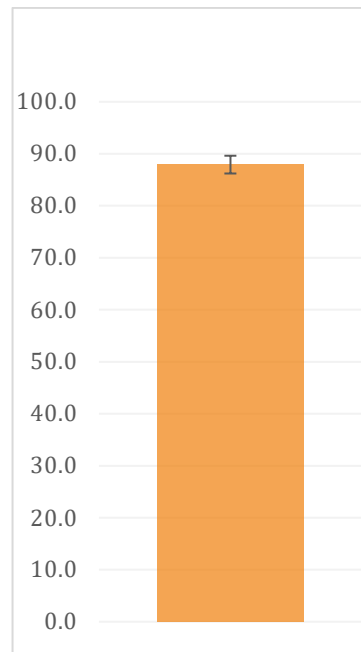


Figure 8.1 Mean overall SUS score from all SUS results (excluding prototype) with confidence interval

Table 8.2 shows the adjective, grade and acceptability of OSIRT. These scores are calculated by taking an overall average score of all SUS means across the release versions of OSIRT and mapping them to the Bangor *et al.* (2009) grading, the adjective rating and the Sauro and Lewis (2011) grading.

Adjective (Bangor <i>et al.</i> , 2009)	Excellent
Grade (Bangor <i>et al.</i> , 2009)	B
Grade (Sauro and Lewis, 2011)	A+
Acceptability	Acceptable

Table 8.2 Adjective and grade rankings for OSIRT (overall mean SUS score excluding prototype)

OSIRT throughout its lifespan has consistently scored very well with RITES course users, and has always scored well above the average benchmark of 68 (Figure 8.2 and Table 8.3).

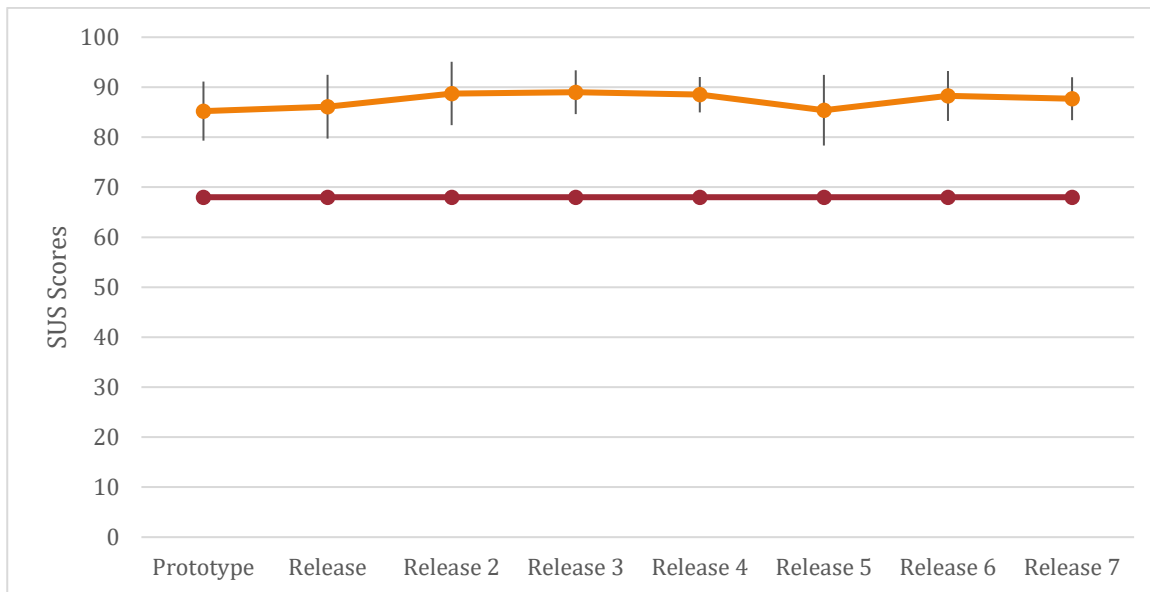


Figure 8.2 Mean SUS scores with benchmark score (68) in red

	Prototype	Release	Release 2	Release 3	Release 4	Release 5	Release 6	Release 7
Mean	85	86	89	89	89	85	88	88
Standard Deviation	8.8	8.6	9.0	6.8	5.3	8.4	7.4	5.8
Sample Size	11	9	10	12	11	7	11	9

Table 8.3 Raw mean SUS scores

8.2.1 Discussion of mean SUS scores

There is no significant difference in SUS scores between the prototype version of OSIRT and the latest release SUS questionnaire (Release 7) ($t=0.77$; $p=0.44$; $df=17$; 95% CI ± 9.78). This is an interesting result given the complete re-write of OSIRT from the prototype, which included major stability fixes and new features. One would reasonably expect to see an increase in the SUS score from the prototype to the latest version.

One reason for this may be the ‘Pareto principle’, commonly known as the 80:20 rule. The rule when applied to software development is that 80% of users only use 20% of the features. While the ‘80:20 rule’ is commonly used as anecdotal adage in the modern era, research by Standish Group (Duong, 2009) has shown that 64% of features are “never” or “rarely” used, while only 20% are “always” and “often” used. Since the prototype OSIRT has provided the key features to ensure open source research is conducted following appropriate guidelines. That is, OSIRT has always had audit log maintenance and functionality to capture artefacts at some level. Perhaps, for most users, this is all they

need and any features beyond that are simply 'nice to have' or are rarely, if at all, used. Chapter 9's usage questionnaire would certainly lend credence to this theory.

8.2.1.1 Mean SUS score summary

While we cannot be certain as to why the SUS scores have not significantly increased or decreased, the '80:20 rule' provides an interesting, perhaps even likely, explanation. Chapter 9 discusses the use of SUS in more detail as part of a study, and pontificates on its use as part of OSIRT's feedback strategy.

8.2.2 SUS question score breakdown

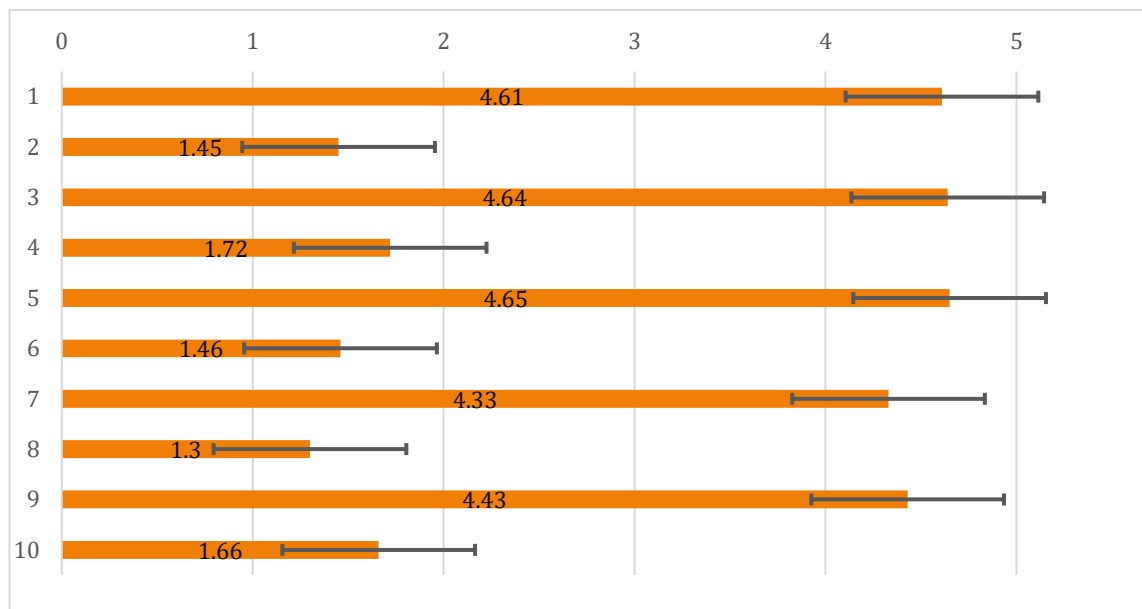


Figure 8.3 Average SUS score across each question for the all release versions of OSIRT.

8.2.2.1 Learnability

Figure 8.3 breaks down the average SUS score across individual questions. Brooke (1996) advised that “scores for individual items are not meaningful on their own.” However, Lewis and Sauro (2009) discovered via “factor analysis of two independent SUS data sets” that there are subscales within SUS questionnaires that measure usability and what they coin as “learnability” (that is, how simple it is to learn a system).

Items 4 (“I think that I would need the support of a technical person to be able to use this system”) and item 10 (“I needed to learn a lot of things before I could get going with this

system”) can be used to measure learnability. However, unlike a benchmark of 68 as a mean SUS score, there is no mean learnability benchmark. From the study by Lewis and Sauro (2009), learnability deviated approximately 10% from the mean, making a mean learnability benchmark of 75 ($68 * 1.1, 2sf$). The mean learnability score was 82.8.

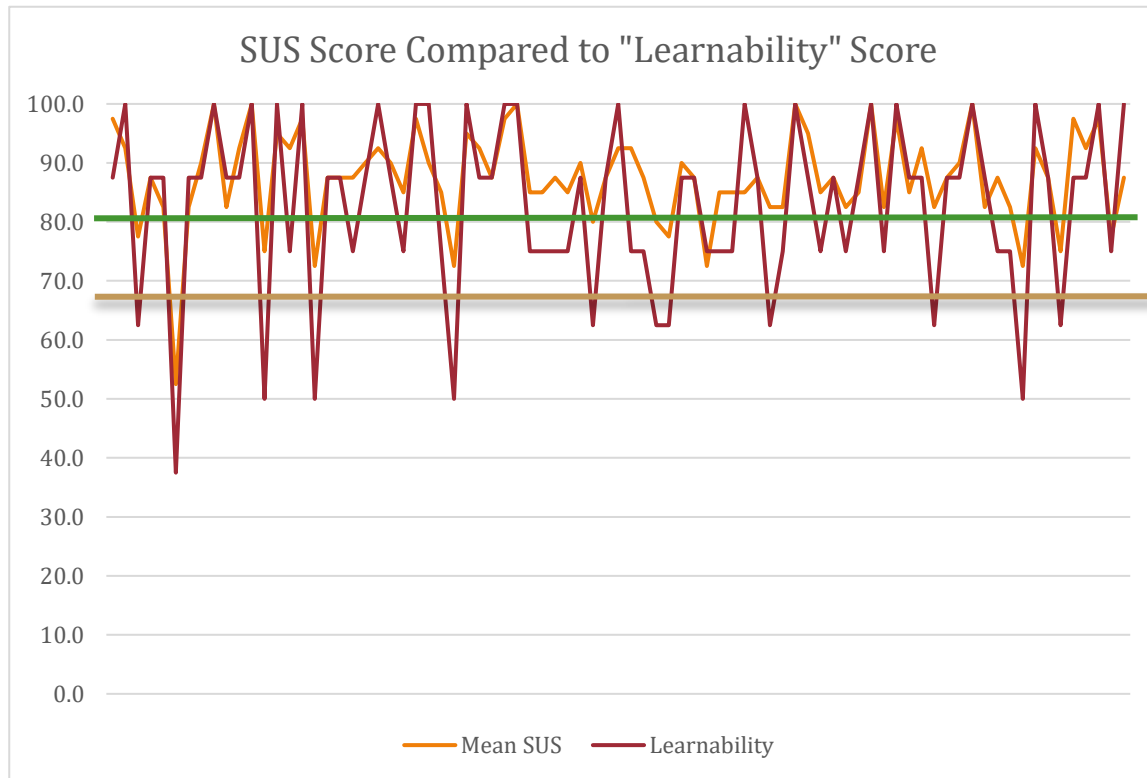


Figure 8.4 Individual Mean SUS Scores Compared to Learnability Scores. Green bar represents mean learnability (82.8) and the brown bar represents mean SUS score (68).

As seen in Figure 8.4, which maps all participant SUS scores to their learnability scores, shows some participants struggled to learn OSIRT, with the lowest learnability score of 37.5. A reason for this may be that participants use OSIRT during the course on top of the course objectives, and this may impact how participants learn to use OSIRT. OSIRT also contains many tools, and these are all showcased and utilised throughout the week. While many officers may not need to use all these tools when back on the job, they are all made use of during the RITES course. This means participants need to remember both what these tools are called (e.g. “WhoIs?”) and where it would be located within OSIRT. Certainly, if some participants are struggling to remember where these tools are, then more can be done within OSIRT to enhance its learnability. Arguably, there is an

extensive user manual available for OSIRT that breaks down individual features with a numbered list on how to use those features.

8.2.3 Net Promoter Score

The Net Promoter Score (NPS) was designed by Reicheld (2001) and is a popular metric when discovering product loyalty. Reicheld (2001) postulated that “customers can be divided into three categories”: promoters, passives and detractors. Asking the following question on an 11-point (0 – 10) Likert scale determines which group the respondent falls in: “How likely is it that you would recommend this [company/product] to a friend or colleague?” Promoter’s rate 9 or 10, passives rate 7 or 8, and detractors’ rate 0-6. To calculate the overall score, subtract the percentage of detractors from the percentage of promoters: $NPS = Promoters - Detractors$ (Mattox, 2013). The NPS provides an additional metric to OSIRT’s development by providing insight in to how user’s may be talking about OSIRT to colleagues. If users are promoters, they are more likely to recommend OSIRT to someone else.

Sauro (2010) notes that there is a positive correlation (.61) between SUS scores and those who are ‘detractors’ and ‘promoters’. “Promoters have an average SUS score of 82 while detractors have an average score of 67 ($p < .01$).” (Sauro, 2010). To calculate the NPS, Sauro (2010) proposes the following formula:

$$NPS = 0.52 + 0.09(SUS)$$

Sauro (2010) then recommends a SUS score “above 80” will put the product in the “promoter range”. OSIRT has a mean SUS score of 87.9, comfortably putting it at promoter status.

Table 8.4 shows an individual breakdown of ‘promoters’, ‘passives’ and ‘detractors’.

NPS name	n
Promoters	61
Passives	11
Detractors	0

Table 8.4 NPS scores based on SUS scores

8.2.4 Additional comments

The SUS questionnaire also provided the participants with an opportunity to provide free-form comments about OSIRT. This section discusses the feedback received from this channel.

8.2.4.1 The open source research novices

Several participants commented on how they were not “particularly technical” or “a complete novice”, but found OSIRT to be useful to them. One of the self-identified novice respondents said, “I know if I was using it on a day to day basis it would be an easy tool to get to grips with.” With another noting that “OSIRT [is] simple to use and effective”. One respondent went as far to say, “It’s rare a system used by police is so straight forward + ‘idiot proof’”.

A large part of OSIRT’s goal is to capture those who are not advanced computer users, or even technically minded. The report by HMIC in 2015 showed there is a very real knowledge gap in some officers’ ability to capture even those most rudimentary digital evidence. A tool like OSIRT aids in bridging the skills gap as the above comments suggest.

8.2.4.2 Taking OSIRT back into the workplace

Several officers took the opportunity to comment on how they will be taking OSIRT back on-the-job. One officer commented “very good system which I will be looking to introduce to the work place” with another noting it will be “very helpful” back on-the-job. Chapter 9 and 10 both discuss OSIRT’s integration into policing job roles in more

detail, but this early insight shows the impact OSIRT has on the participants at the RITES course.

8.2.4.3 Open source research and beyond

One participant left the comment “Could have other uses, not just open source research”. OSIRT is a web browser and is capable of capturing any content that can be rendered by a web browser. One example where OSIRT is used outside of its intended market is network router log investigations; which fall under level 4 of the open source research levels. Settings in routers for home users tend to be HTML based, where a user logs in to the system and router statistics provide investigators with useful artefacts. OSIRT is used by certain police forces to conduct and capture router logs.

8.2.4.4 Technical support

One issue that was written about by participants was the need for technical support. One participant used the comments to justify why they rated “agree” to the SUS question “I think that I would need the support of a technical person to be able to use this system”. The participant said: “Tech support + advice is always a necessity no matter how apparently simple the system is (e-mail/telephone helpline would suffice).” OSIRT being a free and open source product is a negative trait in this regard. The main reason OSIRT can be free is because there is no obligation from the developer to create and generate updates or support it. One participant also pondered about support in the long term by saying “would there be some support in place in [the] coming years?” Chapter 9 discusses, from an open source software perspective, the impact the limited technical support had on the decisions of a police force when integrating OSIRT into their workflow.

8.3 Observations

Three five-day observations of 28 participants on RITES courses were conducted over a nine-month period for the OSIRT release versions. Observations were conducted using Dumas and Redish's (1999, p. 292) “observations, quotes and inferences” technique. During the courses, participants are tasked with conducting open source research within OSIRT and it is this usage that was observed.

Immediately observable was the vast improvement in terms of user interaction with the release version in comparison with the prototype. In fact, the third observation yielded no bugs at all, but the cohort did still offer suggestions and features which they would like to see. This section discusses the observations and the impact on OSIRT with the observation sheets provided in Appendix D.

8.3.1 OSIRT's design and look-and-feel

An issue seen with the prototype was the clunky and confusing design, which was seen with the complex video capturing and audit log report exporting. From the first interactive demo session on the RITES course, which lasts approximately one hour on the first day, users were making positive and complimentary comments about OSIRT. Small adjustments, such as having the screenshot button default to fullpage screenshot rather than open a menu to select it, and video capture recording no longer having an unsightly and hard-to-use dialog, made a big difference. There was a plethora of comments surrounding OSIRT's usefulness, with just one example being "That's so much easier than what I use now."

8.3.2 Bugs and user experience enhancements

Observations provide a good opportunity to see first-hand any bugs within OSIRT. There were several bugs noted during the observations that will be discussed later, but one issue in particular shows the importance of selecting observations as a data collection method. When a user was attempting to save a screenshot in the Image Previewer, they would occasionally receive an error saying the image with that name has been previously saved. Observing this participant showed that the way they held the mouse meant they would occasionally depress the scroll wheel on the mouse, causing the 'combo box' to scroll to a previously saved image name. The fix was a simple case of disabling the mouse scroll event in the 'combo box', but had it not been for the observations there would have been little chance of replicating this bug as the user had not realised they were pressing the scroll wheel.

Several times during the first observation, users would attempt to reload an already opened case in OSIRT prompting the user to ask where their case was, or to think their

case had disappeared: "I've lost my case", said noting one participant. The solution to this, as discussed in section 7.21, was to create a singleton instance of OSIRT.

Two separate issues surrounding tabs were discovered by two users. The first being if a user clicked the 'close' button on the only remaining tab this would throw an exception; this was fixed by removing the close button from the last remaining tab. The second was an issue around memory management with several dozen tabs open causing OSIRT to freeze under some instances; an update to how tabs were disposed of and opened fixed this problem.

While downloading is much improved in comparison to the awkward prototype download manager, the way OSIRT managed file downloads meant users were occasionally forgetting how to find what they had just downloaded. To reduce the confusion, files were downloaded to where the user had selected them to be saved and a copy of the download was then placed in the case container.

Context menu items were confusing for some users and felt arbitrarily ordered; these were put into a sensible order for the final observations and appeared to offer a much-improved experience to users.

8.3.3 Feature requests

Tor was not immediately built into the OSIRT release, but was, again, heavily requested by users and the trainers during the first observations. By the second observation, Tor had been integrated into OSIRT and was well received by the cohort although Tor's integration did require some tweaking as there were observable slow-downs for some users. This was fixed by waiting several seconds for the Tor process to start before OSIRT fully loaded.

Many of the observations were users requesting small features or additions. For brevity, these are listed below in Table 8.5.

Feature request	Comment
Log extracted EXIF data.	Added.
Integrate case notes into report export.	Case notes can be exported separately, but the ability to integrate them within the main report may be useful for disclosure purposes and ACPO guidelines for audit trail maintenance. Added.
Add preview of screenshot in audit log.	Searching through audit log can be time consuming, so the ability to see a preview of the screenshot in the audit log would be helpful. Added.
Add find on page (Ctrl+F).	The old browser control had this by default. Added in CefSharp.
Add the ability to uncheck items in the audit log that have been searched for to exclude from report.	Added.
Use the address bar as a search bar.	Common feature in other browsers. Added.
Export audit log between certain dates.	User may start an investigation and discover nothing of note for several days, the ability to exclude bulk items based on date is a suggestion received via e-mail, too. Added.
Export report as XML.	An analyst asked for this on the course and said "XML is very handy". Added.

Table 8.5 Feature requests from observations

8.3.4 Observation summary

Observations provided an extremely useful insight and many bugs and feature requests would not have been known about if it was not from being able to observe them first-hand. By the third, and ultimately final OSIRT-specific observation, OSIRT was in an excellent position and had shown to make a positive impact on the course and to those using it.

8.4 OSIRT interviews and questionnaires

This section looks at the 22 interviews and 42 questionnaires conducted with various LEOs and trainers around their usage of OSIRT. The topics covered OSIRT and how, if applicable, the participants conduct open source research.

8.4.1 OSIRT's usefulness during course

From the 42 questionnaire respondents, 40 found OSIRT to be useful during the course. This is a positive result, showing OSIRT is being well received during the course. Of the two who responded "no", a further question asked them why they did not find it useful. One answered stating "they did not get on with it" with another leaving no answer.

8.4.2 Recommending OSIRT to a colleague – Net Promoter Score

The question "How likely is it that you would recommend OSIRT to a colleague?" was asked to calculate the Net Promoter Score of OSIRT. Only one response was from a "detractor" and a further four to be "passives" with the rest of the respondents being promoters. Table 8.6 shows percentage breakdown of respondents and the NPS score.

Types	
Detractors	2.38%
Passives	9.52%
Promoters	88.1%
NPS score	86

Table 8.6 NPS score

8.4.3 OSIRT integration into workflow

Respondents were asked in the general questionnaire "Can you see OSIRT being integrated into your current role?" 37 out of 42 responded "Yes". Of the five who could not see OSIRT being integrated, four cited IT-related issues, and one did not want to integrate OSIRT.

During the interviews, participants from the RITES course were asked about how they could see OSIRT's integration into their roles, with 13 participants making positive comments such as that it would be "simple" or "easy" to do so. A response from a

Detective Sergeant noted “It’s quite a simple sort of transition to move away from our current system, which is to use pen and paper to record things, and straight into using OSIRT”, another noted that their procedure involved a spreadsheet and a notebook, and while they would not stop hand writing notes, OSIRT’s automated logging of actions was “a God send”. Those who responded via interview that could not see OSIRT being integrated either said their current IT infrastructure makes it unfeasible (two), or that OSIRT could not integrate into their role at all (one).

The IT “policy” issue is typically the most cited reason as to why an officer may not be able to integrate OSIRT. Different forces will have different policies, and these policies are often restricted for public viewing for obvious reasons. However, one officer commented:

“If they’re [managers] looking to roll out [a product], then usually they’re looking for a service contract with some level of support and guarantee It’s a balancing act, really, and some would, understandably, prefer to pay a sum of money to ensure that support was in place. It also depends upon the team who’s using it [the software], are they technical people?”

This response further reiterates what was written in the questionnaire comments (section 8.2.4.4), but it does provide a bit more context surrounding the importance of support for those who are less technically able.

8.4.3.1 The time saver

Of those respondents who were looking to integrate OSIRT into their workflow, a follow up question asked to consider the phrase “OSIRT will save me time in comparison to how I conduct open source research now”. Table 8.7 shows out of the 37 respondents, 30 either strongly agreed or agreed to the statement, with one remaining neutral. The remaining six were unable to compare as they have not previously conducted open source research.

Strongly agree	24
Agree	6
Neither agree or disagree	1
Disagree	0
Strongly disagree	0
I have not previously conducted open source research as part of my role to compare	6

Table 8.7 Results for the question "OSIRT will save me time in comparison to how I conduct open source research now"

The four officers interviewed who have been using OSIRT as part of the roles all remarked how it has saved them time; "Its [OSIRT] at least halved, probably more actually, how long it takes me to conduct [open source] research", noted one interviewee. Another participant reflected upon how they had previously conducted open source research:

"I look back upon to how we conducted open source [research] in the past, and it makes me laugh to think how we use to do it with spreadsheets, Firefox add-ons and what-have-you. I truthfully cannot imagine what we'd do now if we didn't have OSIRT. For me, it has changed how I do open source."

A pleasing aspect from this development is the consistent response from users of how much time OSIRT saves them.

One interview participant, an Inspector, spoke about wanting their officers to be proficient in open source research but acknowledged a skills-gap for some:

"I want everyone trained in open source, it's so important. I'd have everyone on this [RITES] course but I know that's not possible so it's about what software can we use to achieve that? What OSIRT does for us is that it makes open source accessible; it puts it all in a neat package. Which is perfect for those who need to bring up their skills with technology."

8.4.4 Automated logging and reporting

The end product after an investigation is crucial for LEOs with all respondents noting the report output by OSIRT was in their "top three" features. An interviewee noted that

reporting “[...] can be a complete pain, so anything that can do it for me is fantastic”, a sentiment echoed by other interviewees.

OSIRT's automated logging and report generation were very popular amongst interviewees and questionnaire respondents. By now it is common to hear an officer criticise the monotony of having to manually maintain an audit log. Questionnaire results in section 9.4.10 focus upon tool usage within OSIRT in more detail.

8.4.4.1 Legal compliance and guidelines

Several officers commented on how OSIRT ensures they comply with laws and procedural guidance. Two officers discussed the impending (now integrated) EU data protection laws; the GDPR (General Data Protection Regulation) and the revised Data Protection Act:

“Obviously as police we get concessions in the law so we can actually do our jobs, but the new data protection act is looking to have, I think, six principles that we must follow when obtaining personal information. Looking through that [list of principles], you see how OSIRT not only helps us meet them but enforces them, if you like, because you have to write notes, it automatically keeps an audit, and it puts all the data in one package that we can easily encrypt and store.”

The automated logging of actions was another popular choice, with thirteen LEOs acknowledging during the interviews the difficulty and complexity of logging every action; particularly in reference to ACPO/NPCC guidelines of audit trail maintenance. An interviewee noted that “Seeing my audit in OSIRT surprised me, [...] I performed a lot of actions that I wouldn't really think twice about. Opening Google, performing a search and clicking a link are actually three [actions], but I've always considered [it] just one”. The majority of interviewees all explicitly mentioned how the automated log was a time saver. Beyond time saving, OSIRT also improves the auditing trail process.

8.4.5 Screen capturing

The ability to capture screenshots and screen recordings was also favourable among respondents. Interviewees frequently commented that having this functionality for free is good, as they do not necessarily have the budget to afford the licenses for some tools.

“Screen recording tools can be very expensive, or have an upper limit of how much you can record if they are free. The inbuilt video capture in OSIRT does not impose limits, plus it's free.” One interviewee said, when asked about what screen capturing tools they use, “Anything I can find and is free. I used to use FastStone Capture but the free trial run out, and I cannot obtain a license.”

Ten interviewees commented that being able to take full-page screenshots of large pages, such as Facebook, was beneficial to them. An interviewee noted “We have to take small screenshots, then stitch them back together. So OSIRT is going to be extremely useful.”

8.5 Alternative methods for usability evaluations

This chapter has analysed and discussed the results from a cognitive walkthrough, observations, interviews and SUS questionnaires. However, there are many other methods for obtaining usability data. This section will briefly discuss several alternative qualitative and quantitative methods, and the appropriateness of applying them during the RITES course where the bulk of testing occurred.

8.5.1 Qualitative usability measures

8.5.1.1 Task Analysis

Task Analysis, at its core, observes users to understand how they perform tasks. Common techniques for Task Analysis are seen in Cognitive Task Analysis (CTA) and Hierarchal Task Analysis (HTA). CTA focuses on decision-making and are “typically applied to understand work processes” (Tofel-Grehl and Feldon, 2013) and is a Task Analysis based heavily on cognitive processes. HTA is based on decomposition of high-level tasks into sub-tasks and is useful when tasks have a clear-cut, consistent structure (Mills, 2007; Felipe *et al.*, 2010). HTA can be represented by using tree structures that show high-level tasks reduced to their component sub-tasks.

Task Analysis in the forms of HTA and CTA are useful methods of gaining insight into users' needs. In some regards, early discussions during the prototype with the RITES course trainers provided this level of understanding but could have been enhanced using Task Analysis approaches. However, as an exploratory metric it still requires users to

follow a scenario that the researcher has control over. During a RITES course, this is not possible. That said, it may still have been possible to conduct Task Analysis on a different group of users (e.g. non-LEOs) and this could have provided additional, useful feedback.

8.5.1.2 Think aloud

Think aloud testing, as the name suggests, asks users to verbalise their thought processes as they are using the system. This is a cost-effective method of testing, as it does not require any specialised equipment to conduct. However, integrating this into the RITES course would be disruptive to the core course objectives. One possibility would be to run small think aloud tests at the end of the training day, but the course is intensive and extending the day for those participants would be an unreasonable expectation. Much like other methods, this would be better executed in a controlled environment where flexibility is possible.

8.5.2 Quantitative usability measures

8.5.2.1 Error rate

Error rate looks at unintended actions a user takes on the system. While it is normal for a user to accidentally click a button, measuring how many errors a user makes during tasks, and their resulting severity, can provide insight into whether there is an issue with the system itself. Norman (2013) notes there are two types of errors: “slips” and “mistakes”. A slip occurs when a user has the correct goal in mind but has conducted an unintended action; an example being a mistyped password. A mistake is seen when the goal is wrong; an example being entering wrong data into a field.

Sauro (2012) notes that unlike task completion rates, error rates can occur several times per task and as such are better treated as binary data. While this may lose granularity and details, it can still offer useful statistics. Additionally, combining the number of errors as part of a Single Usability Metric (SUM) (Sauro and Kindlund, 2005) to provide an overall single measure that includes task completion and satisfaction rates is an effective way of summarising usability issues.

8.5.2.2 Task-completion rates and task-completion time

Task-completion rates are a measure whether users have successfully completed a task and are used as a measure of effectiveness (Tullis and Albert, 2013). This can be represented

as binary data of whether the task was successfully completed, so if 8 out of 10 users successfully complete a task, then that task has a completion rate of 80%. Sauro (2011) collected data from 1189 tasks from 115 usability tests and discovered that 78% was the average completion rate for a task but notes a limitation of these “inflated completion rates” may be down to the Hawthorne effect. Additionally, the context of what is a ‘good’ task-completion rate is system dependent. For a mission critical system or there is a threat to life, then task-completion rates should be close to, if not, 100%.

As a measure on its own task-completion is already useful, but when coupled with task-completion times, that is, how long the task took to be completed provides additional levels of insight into a system. As with previous usability metrics discussed in this section, they are not something that can be trivially integrated while on a RITES course; for reasons already specified previously. However, it is still a method to be considered for participants in a controlled setting outside of the RITES course.

8.6 Chapter summary

This chapter looked at OSIRT from a user experience perspective and the impact OSIRT had. Observations, SUS questionnaires, interviews and general questionnaires have consistently shown OSIRT has made a positive contribution and impact to those who have used it both on the RITES course and when taking it back on-the-job. The next chapter continues with OSIRT's contributions and impact, and a reflection and study on OSIRT as a piece of free and open source software.

9 RESULTS AND DISCUSSION OF OSIRT'S INTEGRATION, IMPACT AND CONTRIBUTION TO LAW ENFORCEMENT: PART TWO

INTRODUCTION

As OSIRT's usage marches forward, this chapter looks upon its impact and contribution made to law enforcement. This chapter discusses and reflects upon how OSIRT has been fully integrated into the College of Policing's RITES course; this is supported by means of an interview with the lead high-tech crime trainer. The chapter then discusses results from a general OSIRT usage questionnaire distributed to those using OSIRT with law enforcement. Finally, a discussion surrounding OSIRT as free/libre open source software is debated and supported by means of a case-study of a police force who has integrated OSIRT into their workflow.

9.1 OSIRT's integration into the RITES course

To recap, OSIRT has been an aspect of the RITES course since its prototype, and became a central element of the course after the release version. The RITES course provides sessions throughout the year, typically 10 or 11 sessions, to a cohort of around 10 delegates per session. While chapter 10 discusses the influence OSIRT and the RITES course has on those attending in more detail, it is worth stressing that OSIRT is trained to over 100 officers across a range of policing a year due to the RITES course. This section looks at OSIRT's integration into the RITES course and its importance on the course by

means of an interview with the lead course trainer, Russell Taylor, and the contribution OSIRT has made to the training package on the RITES course.

9.1.1 About the interviewee

Russell Taylor is the High-Tech Crime Course Manager at the College of Policing and trains investigative techniques involving the Internet and mobile phones. Russell has been a trainer with the College of Policing (and formerly National Police Improvement Agency) for over eleven years and is a retired police officer from the Metropolitan Police Service with 30 years' experience. Collectively, Russell has spent over 40 years in policing.

9.1.2 The course before OSIRT

Before OSIRT, it was the usual story of using several tools. Russell noted the complexity of adhering to the ACPO principles and the maintenance of the audit log in particular:

“Pre-OSIRT, the ACPO principles have always required that you maintain an audit log to demonstrate what you're doing. So, we ended up with the students on the course creating an Excel spreadsheet with the websites they went to, and then doing still captures. So, you had all those processes working through that, and then they had to have another column with any notes, comments or information they gathered. And that was just so time consuming, you've noticed on the course that we have people with different skill bases, and then if you add in complexity of trying to operate an Excel spreadsheet, and trying to get an image to fit inside a box, a cell.”

The “different skill bases” Russell notes is one that surrounds the use of technology and computer literacy. Officers that attend the course are, generally, experienced police officers with solid understandings of investigative techniques. Those attending the course are not necessarily experienced in open source research, but still bring with them a range of experiences that play a critical role in OSIRT's development; regardless of computer literacy. Arguably, these are the officers that OSIRT is designed for.

When asked about the manual maintenance of the audit log, Russell said it was an “administrative nightmare” due to its complexity. Russell also noted the non-standardisation of tools on the course, as “everybody had their own particular preference in tool and if they brought that with them to the course that added additional complexity to the course with them using something different to the rest of the class.”

At the end of the week, the cohort are tasked with conducting an open source investigation. Russell said, before OSIRT, that it was “extremely rare” for anyone to completely finish this task due to the complexities highlighted above. These complexities noted by Russell could easily spill over back into the working environment, too. While officers may not be under exam pressure as seen on the course while back on-the-job, there were certainly larger issues afoot with the use of multiple software tools.

The struggles of the course when it came to discussing the software tools used were evident in Russell's tone and previous, anecdotal discussions with Russell showed the difficulty in managing the course using a variety of tools. In many respects, Russell painted a picture of a course that spent less time teaching investigative techniques and more time teaching students how to use software productivity tools.

9.1.3 The course with OSIRT

The first aspect Russell pointed out was the increase in completion of the open source investigative task. “The very first time we used OSIRT, we had 3 students complete the [redacted] investigation. That's when it was obvious to me OSIRT would transform the course and, looking back, transform how we generally conduct open source”. When asked what it was about OSIRT that made this transformation, Russell noted the combination of several different tools into one is what made the difference.

“Simplicity. It [OSIRT] just made things so much simpler. OSIRT pulled it all together and because we discussed what we needed and you came and saw what we needed, you managed to pull together a product that compensated for all those different tools that you were trying to bring together to work with”.

When asked how important OSIRT is to the RITES course, Russell described it as a “pivotal linchpin”.

9.2 OSIRT's integration into private OSINT training packages

Beyond in-house training and the RITES course, OSIRT is known to be trained on several, private OSINT training packages. The founder of Toddington International Inc. (<https://www.toddington.com/>) reached out and has said OSIRT is demonstrated during their course. Toddington's OSINT training is internationally recognised and, arguably, one of the largest OSINT training packages available. They were even kind enough to Tweet about OSIRT (Figure 9.1).

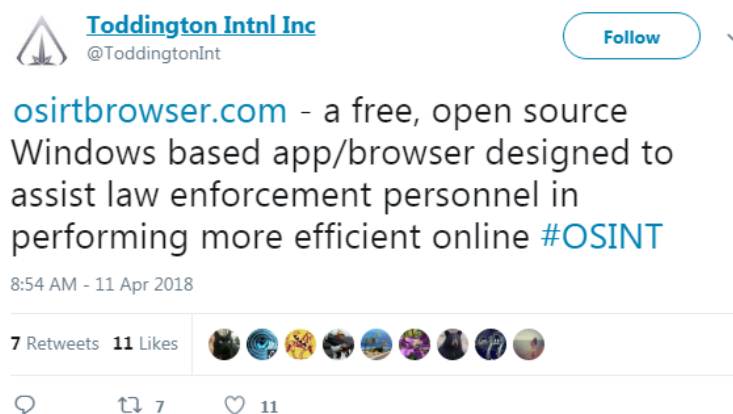


Figure 9.1 OSIRT Tweeted by Toddington International Inc.

Jane's OSINT training by IHS Markit (<https://ihsmarkit.com/products/consulting-open-source-intelligence-training-osint.html>) is a multi-national OSINT training company that uses OSIRT on its training courses.

Cyber Intelligence Solutions, an OSINT training provider that training in USA, UAE, Australia (where the company is based) and Fiji requested permission to use OSIRT in their training package. OSIRT is now used as their case-building tool for the course.

9.3 OSIRT in commercial products

OSIRT has been integrated into Internet Investigation Solutions Limited product LongArm. LongArm "is a secure, real-time open source investigation platform that enables Law Enforcement Agencies to investigate and research online material in a completely discreet and non-attributable manner." (GOV.UK Digital Marketplace, no

date) While statistics surrounding OSIRT's use within LongArm are confidential for both business and privacy reasons, OSIRT is a very popular tool within this platform.

OSIRT has not received any remuneration from being integrated into LongArm.

9.4 OSIRT usage questionnaire

While the SUS questionnaire provides immediate feedback post-RITES course, it does not offer an insight into how OSIRT has integrated into working roles of law enforcement. To gather a broader understanding of how OSIRT is used, a questionnaire covering key areas such as weekly OSIRT usage time, previous tools used and whether OSIRT has impacted upon their working role was distributed to LEOs. This section discusses the results from the OSIRT usage questionnaire.

The questionnaire link was given to Russell Taylor, lead trainer of the RITES course, who placed the questionnaire up on the OSIRT section of POLKA (Police OnLine Knowledge Area).

9.4.1 Demographic

This section details the officers (n=32) who participated in the questionnaire and provided details about themselves. These responses were optional, so results may not always add-up to 32.

As expected, there is a mix of job roles and experience (Figure 9.2 and Figure 9.3). There is a high proportion of analysts and detective constables, which is not surprising given that OSIRT is a hands-on tool designed specifically for investigators.

9.4.1.1 Role

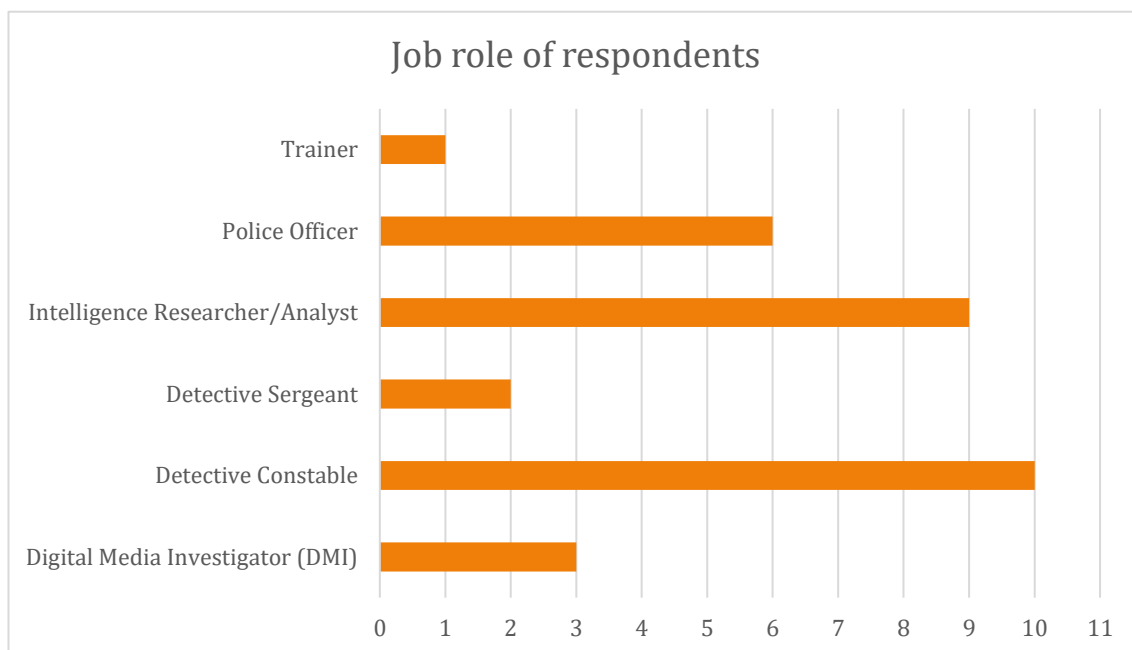


Figure 9.2 Job role of respondents

9.4.1.2 Years active in job



Figure 9.3 Time in service rounded up to nearest year

9.4.2 OSIRT usage

Figure 9.4 breaks down the how long the participants have been using OSIRT. Of the respondents, 63% have been using OSIRT for a year or more. Most respondents, 80%, have been using OSIRT for at least 10 months. Those users who have been using OSIRT for two or more years are likely to be users of the prototype and have been using OSIRT around its initial release.

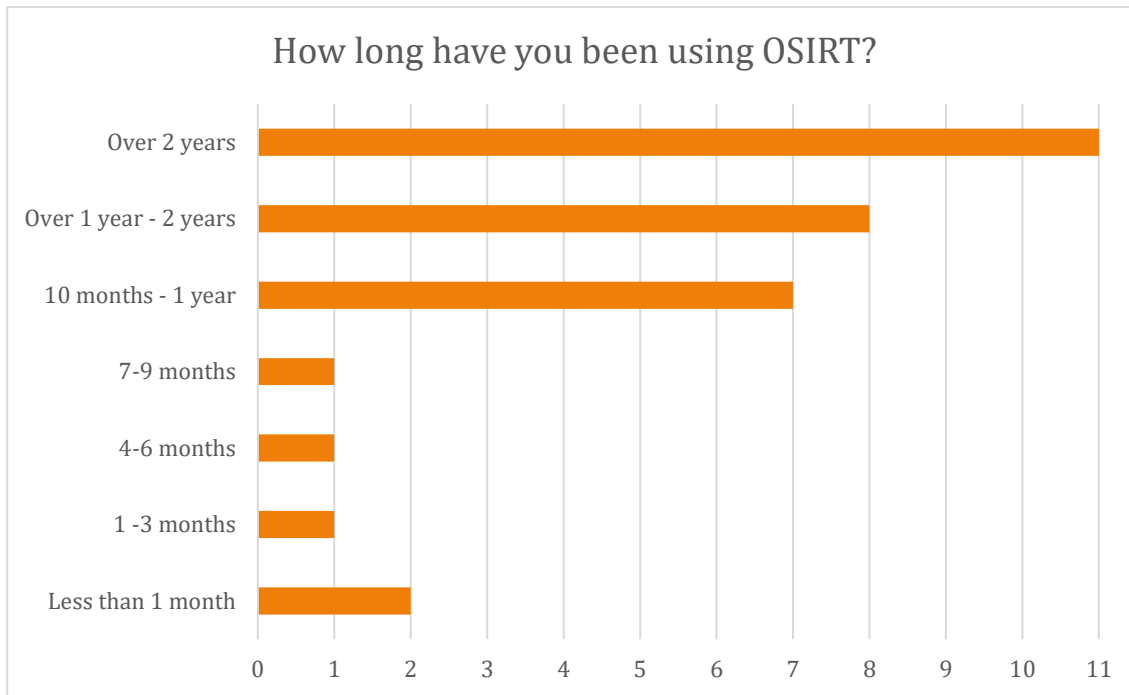


Figure 9.4 How long respondents have been using OSIRT

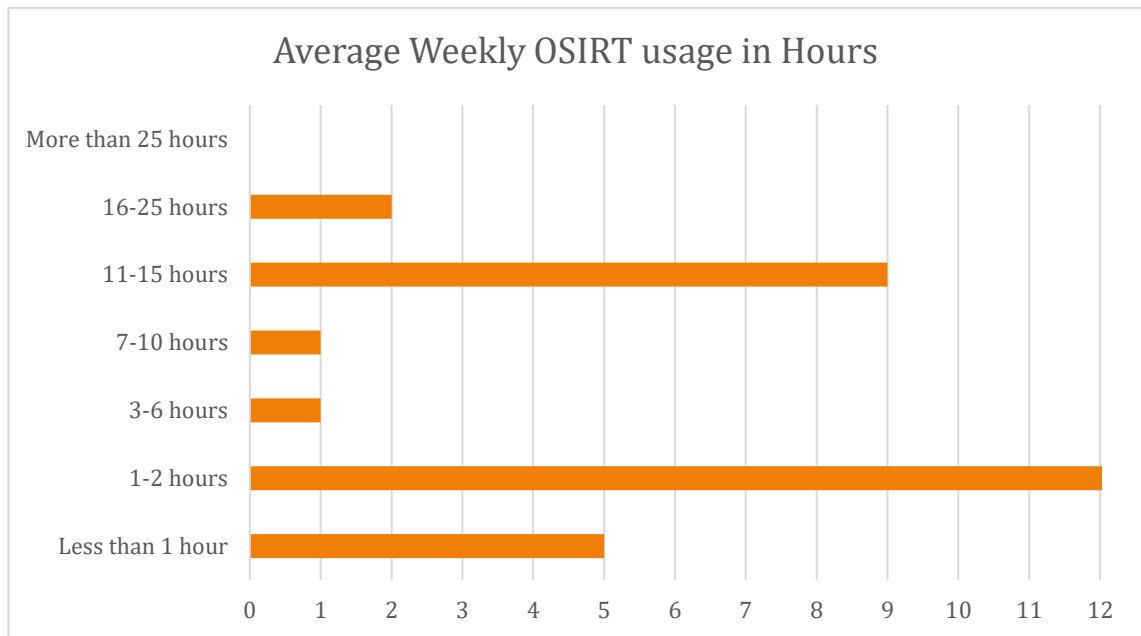


Figure 9.5 Average weekly OSIRT usages in hours

What is striking about the results of average weekly usage in Figure 9.5 is there are two groups of users. One that uses OSIRT for a fair amount of their work, 11 use OSIRT between 11 and 25 hours a week, and those that use it in a more casual manner; 17 use it for two hours or less a week on average. These are not particularly striking results, as not all officers will be tasked with conducting open source research all the time. Some respondents are likely to be “satellite” open source researchers, in that they may start an open source investigation for the dedicated team to start later. For example, starting a case during the night for a Digital Media Investigator to pick up in the morning.

One respondent was part of the “NPT” (Neighbourhood Policing Team), this is a uniformed officer who is visibly present in a community. Plainly, their OSIRT usage is going to be less than that of a DMI, but the fact they are using OSIRT to begin with shows its reach, impact and importance across all aspects of policing.

Table 9.1 shows OSIRT’s usage time per week in hours for those respondents.

Job Role	Hours	Number of Participants
DMI	11 - 15 hours	3
Detective Constable	1 - 2 hours	4
	3 - 6 hours	1
	11 - 15 hours	4
	16 - 25 hours	2
Detective Sergeant	Less than 1 hour	1
	11 - 15 hours	1
Police Officer	Less than 1 hour	2
	1 - 2 hours	4
Trainer	11 - 15 hours	1
Intelligence Researcher/Analyst	Less than 1 hour	2
	1 - 2 hours	3
	7 - 10 hours	1
	11 - 15 hours	3

Table 9.1 Jobs roles and hours using OSIRT

9.4.3 In-house training packages

While OSIRT is utilised during the RITES course, it is often trained as part of in-house training packages, as seen in Figure 9.6. 25 respondents were either trained directly as part of an internal training package, or by a colleague. Unsurprisingly, internal training is popular as it is cheaper than sending officers to training sessions. Sending officers away will mean losing a resource for a week on top of the cost of the training itself. Additionally, keeping training in-house means officers can be trained to that force's operating procedures and standards. While the RITES course teaches open source research techniques, it can only discuss methods and procedures in a generic manner for the diverse cohort; ultimately this will boil down to force policy. As seen in chapter 10,

some cohorts attend the RITES course in order to feedback and train in-house; this is very cost efficient.

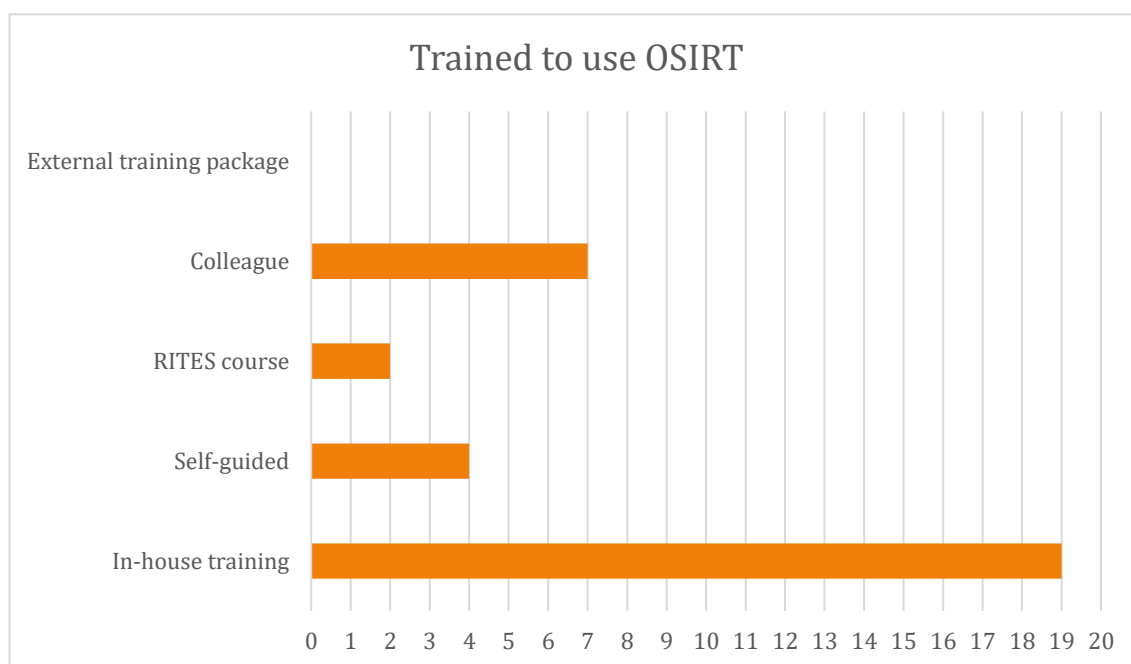


Figure 9.6 How Officers were trained to use OSIRT

This is what occurred with Dorset police. An officer was at the RITES course during OSIRT's prototype, and took OSIRT back with them to the force and disseminated it to their colleagues. While this officer is now retired, OSIRT is still extensively used and extensively trained in-house within Dorset.

Hampshire Police is another force that delivers its own training, and train OSIRT as part of their open source investigations. An article from Policing Insight, which interviewed a Hampshire trainer, said:

“Incorporated into this [open source] training was the use of OSIRT and IBM's i2 software. Under development by Canterbury University, OSIRT is an internet browser specifically designed for Policing to enable online research. By consolidating these otherwise separate training requirements, it has become possible to deliver what would otherwise be 13 days training into five. This approach saves valuable police time by reducing abstraction from duty for training.” (Munro, 2017)

9.4.4 OSIRT discovery

Given the high percentage of those trained in-house, this is where most respondents (22) discovered OSIRT (Figure 9.7).

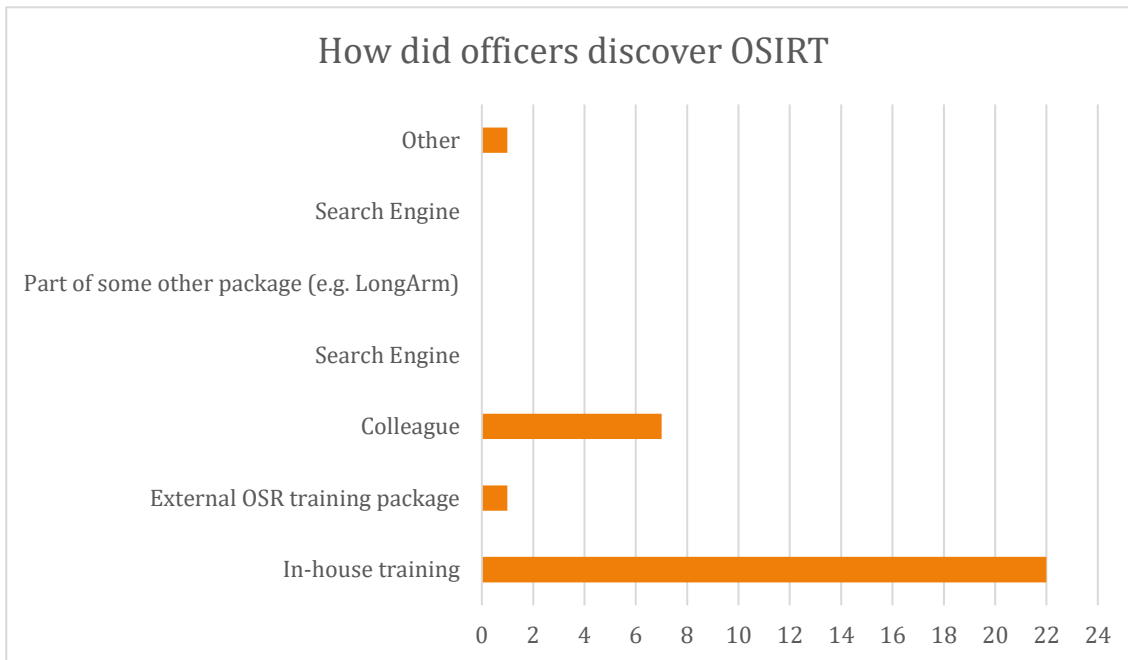


Figure 9.7 How officers discovered OSIRT

9.4.5 Rate usefulness of OSIRT

Respondents found OSIRT to be useful, with all rating OSIRT a seven on the scale provided. 19 of the participants rated it the maximum score of 10.

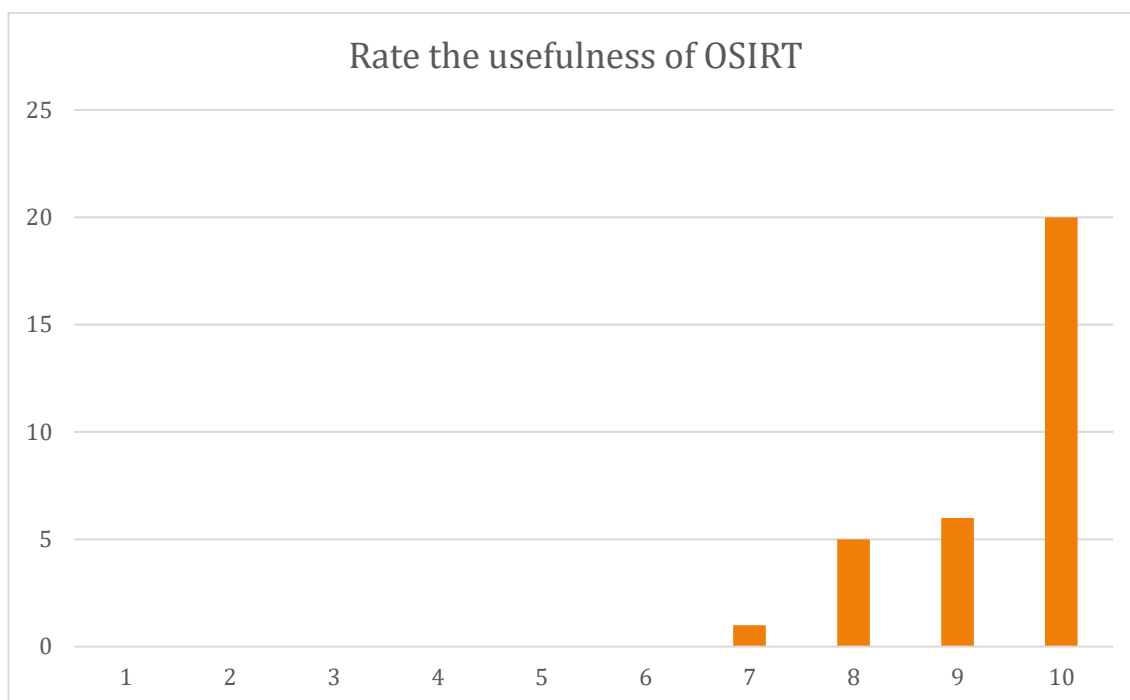


Figure 9.8 Usefulness rating of OSIRT

9.4.6 Has OSIRT enhanced your capability at conducting open source research

Comments are provided verbatim, omitting the quote below, in Table 9.2.

Responses

Yes, big improvement from our old manual system with spreadsheets.

Yes

Yes, especially when looking at Whois in relation to IP

Yes

Yes

Yes, the creation of a report and the ability store everything in an easily accessible folder structure.

Yes, it has made it easier to set up and record an open source session and also it is very easy to produce an evidential report

I am unable to comment as I have only ever used OSIRT

yes, no completed in a much more professional, presentable manner
the ability to revisit research and have a saved version of the url is great
Yes - the automatic logging and recording makes the whole process a lot more efficient.
It has not enhanced - may be user error but it is great at tracking my movements evidentially.
Yes, it has professionalised the product that I can provide to use in evidence and provides a robust report that can withstand professional scrutiny.
Yes, made it easier definitely.
yes
yes, evidential capture of research is vital and this product tick all of the boxes
It has added professionalism to our Open Source Research
Research with confidence that the 'trail' is being captured
Yes - It ensures that I can evidentially capture what I need
no but it has enhanced our methods of recording our research and auditing process.
Yes, made it a lot easier.
Yes. Use it for all my open source investigations
Yes it has made my job much easier
Yes, made OSIR easier
Yes completely changed my work
Yes - automated logging and reporting!
Yes as much easier to record what I have done and saves a lot of time
Yes, it has made open source research a lot more accessible.

Table 9.2 Verbatim responses for "Has OSIRT enhanced your capability at conducting open source research?"

This free-form optional question generated 29 responses. Of the responses, 22 started their sentence with “yes” and a further 4 responses were positive in nature. One comment from an officer who has used OSIRT for over two years notes OSIRT’s integrated tools and the fact it was designed specifically for law enforcement as a reason for why it has enhanced their capability:

“It has [enhanced my capability], but, it's the fact that this tools places all the relevant functionality of other tools all in one place that is specifically designed for Law Enforcement and the challenges that we face around continuity of evidence.

It also gives peace of mind as we know that all data is locally held and OSIRT is not reporting back to any servers, meaning we can trust it for security around our information.

OSIRT above all saves loads of time and gets to the information that we need fast. Its exports are also used to help find common denominators across multiple social media accounts and provide a vast amount of intelligence about criminal groups and their associates.”

Of the negative comments, those who said OSIRT has not enhanced their capability, still provided positive feedback “it has enhanced our methods of recording our research and auditing process” and “It has not enhanced - may be user error but it is great at tracking my movements evidentially”.

Word frequency analysis showed “easier” was mentioned 6 times. In context, these comments all noted that OSIRT had made conducting open source research easier, with one comment even mentioning it “made my job much easier”.

The notion of professionalism OSIRT brings to respondents was also emphasised via word frequency analysis with three participants mentioning how OSIRT provided an output that is “more professional”, with another respondent saying, “It has added professionalism to our Open Source Research”.

9.4.6.1 It's all I ever knew...

For one respondent, they had “only ever used OSIRT” to conduct their open source research. While this is only one respondent, it perhaps shows that for many incoming officers who are required to conduct research, OSIRT will be the de-facto piece of software they use. This will, speculatively, only increase as OSIRT has only been available for several years, so some of those officers who joined the force in 2016 will

now be coming off of probation into different roles, and perhaps require using OSIRT. This is also highlighted in the next section (previous tool usage), where several respondents did not list tools as they had only used OSIRT.

9.4.7 Previous tool usage

Table 9.3 shows a list of previous tools used by respondents. These results are fairly typical from what has been previously discussed. Popular tools such as Microsoft Excel and Word would be used to maintain the audit log, and various other tools and add-ons to capture. In this questionnaire, the browser extension Fireshot was the most popular screenshot tool. Even with a pool of 32, it clearly showed the disparate use of different tools that OSIRT has largely replaced.

Tools	Mentions
Excel/Spreadsheet	9
Unspecific add-ons/extensions for browsers	6
Word	5
Fireshot	5
Karen's Hasher	4
Notepad(++)	3
Camtasia/Screen recording	3
WhoIs? Add-ons	3
Snagit	2
None	2
Ashampoo	1
One Note	1
Windows Screenshot	1
HTTRACK	1
Tor	1

Table 9.3 Used tools breakdown

Two respondents have never used other software to conduct open source research, with one stating they have “only ever known OSIRT”.

9.4.8 Does OSIRT capture all relevant data for your open source investigation?

This free-form question, with responses in Table 9.5, offered the respondents a chance to provide feedback on whether OSIRT captures relevant data as part of their open source investigation. Word frequency analysis of the text showed there were 29 occurrences of the word ‘yes’. Two respondents noted an issue surrounding video capture.

Responses	N
"Yes" or "yes"	21
Yes, although the ability to download videos from more websites would be great.	1
Yes, the tool is particularly useful for audit and reporting.	1
Only current issue is video capture.	1
Yes - I always video capture my screen and produce this in evidence.	1
for me it does yes	1
Yes. I particularly like the screen recording options and the automatic page logging.	1
I struggle capturing video and sound	1
yes - extremely easy to use and professional means of recording what we do on open source	1
Yes - and more!	1
Yes and then some	1

Table 9.4 Raw responses to question

9.4.9 Recommend OSIRT

100% of respondents said they would recommend OSIRT to others.

9.4.10 Tools usage within OSIRT

Table 9.5 lists individual tool usage within OSIRT. The usage figures lend credence to the previous discussion during the analysis of SUS results surrounding the 80:20 rule. All tools within OSIRT are used, but of the 20 tools listed seven are used half of the time with only four used at least two-thirds of the time. No individual tool is listed as 100% usage.

These figures certainly lend credence to Pareto's '80:20' principle as discussed previously. If we consider a tool to be 'popular' that is used by at least two-thirds of respondents, we see a ratio close to 70:30. Given the modest sample size, that is remarkably close to the original principle. These results certainly aid in understanding the SUS results, too, as the top four tools have been in OSIRT since the prototype.

	Usage	
	Total	%
Tools		
Video screen capture	22	70.97
Audit log	22	70.97
Full screenshot capture	21	67.74
Snippet capture	21	67.74
Case notes	17	54.84
Report exporting	17	54.84
Tabbed Browsing	16	51.61
Full webpage downloading	13	41.94
Timed screenshot	12	38.71
Saving page source code	12	38.71
Attachments	11	35.48
Video downloader	11	35.48
WhoIs? finder	11	35.48
IP address saver	11	35.48
Facebook and Twitter ID finder	11	35.48
Extracting links on webpage	9	29.03
Tor (dark web browsing)	6	19.35
Exif viewer	6	19.35
Reverse image searching	6	19.35
History viewer	5	16.13

Table 9.5 Individual tool usage within OSIRT (total usage and total usage as a percentage)

9.4.11 Does OSIRT being open source software impact the decision to use OSIRT?

OSIRT is both free and open source software (FLOSS), and this question looked to see if that impacted the user’s decision to use OSIRT. As seen in Figure 9.9, 27 respondents (87.5%) answered “no”, meaning overwhelmingly that OSIRT being FLOSS does not impact usage for these participants.

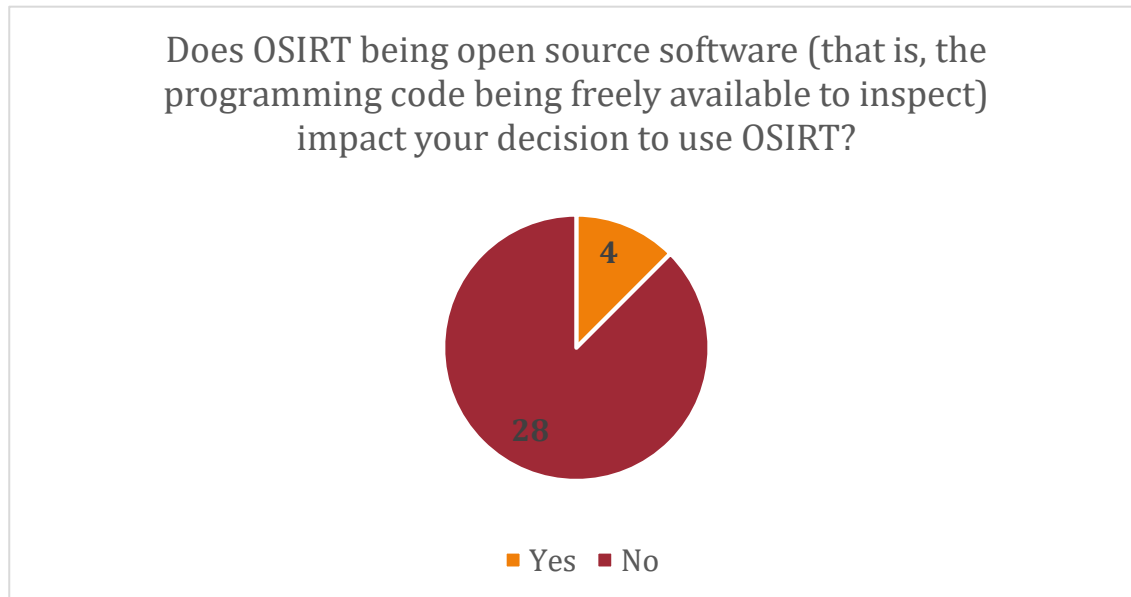


Figure 9.9 Does OSIRT being FLOSS software impact decision to use it

Of those who answered “yes”, a follow up question asked them why that was the case. One response was excluded as it appears they misunderstood the previous question (“Because it enables me to keep an audit trail of my research and I can refer back to the case as it has been saved.”). Two responses noted “transparency”, and that the source code can be “inspected” as reasons why it impacts their decision to use OSIRT. The other response was negative in nature and pointed out “there is no technical support” that has “made full implementation difficult within [their] agency”.

Section 9.5 discusses OSIRT as FLOSS software in more detail.

9.5 OSIRT as Free/Libre Open Source Software (FLOSS)

While it may be easy to dismiss the need for OSIRT to be a Free/Libre Open Source Software (FLOSS) product given the findings from section 9.4.11, there may be a deeper

reasoning for these results. Foremost, simply, respondents may not necessarily understand the difference between closed and open source software. On the face of it, why would a user prefer a product whose source code is hidden versus a product's whose source code is not. Perhaps for a ubiquitous product like Microsoft Windows or iOS, trust stems from its number of users, or the fact it is made by a multi-billion dollar company. However, does that trust extend to those products who are made by unknown individuals? While this answer could never have a quantitative result, it does raise some interesting questions; particularly surrounding FLOSS' integration in to UK public services.

This section looks at OSIRT as FLOSS and whether the decision to make OSIRT open source software had an impact upon its integration into law enforcement. Firstly, there is a review of FLOSS into broader public services within the UK followed by interviews from a police force that uses OSIRT.

9.5.1 FLOSS integration into UK public services

In 2012, the UK Government released a report acknowledging FLOSS "is not widely used in Government IT" (Cabinet Office and Home Office, 2012a). This is contrary to previously issued guidance, as early as 2004 that pushed for more governmental agencies to make use of FLOSS. Current governmental policy sees that FLOSS should be "actively and fairly consider[ed]" over its proprietary counterpart (Cabinet Office and Home Office, 2012b). During the UK Government's re-push for FLOSS integration, they released alongside their 2012 report a list of FLOSS alternatives to well-known proprietary systems (Cabinet Office and Home Office, 2012a). In November 2017, the UK Government once again stressed the use of open source software "to improve transparency, flexibility and accountability" (UK Government Digital Service, 2017) and provided a 15-point guide to evaluating the use of open source software.

Waring and Maddocks (2005) also highlighted that FLOSS was seldom used in the public sector, perhaps due to skills shortages, but those with a "degree of autonomy" may be more able and willing to integrate FLOSS. Law enforcement within the UK are allowed some choice, in which IT decisions, depending upon an officer's skill set, can be made on an individual level.

The potential reason for the slow uptake of FLOSS is that it may bring with it negative perceptions. From personal experience, it is not unusual to receive communications surrounding OSIRT's provenance and why the software is free-of-charge. Questions typically fall in to one of five categories: security/trust, maintenance, technical support, cost and training. These are five points will form the focus of the interviews surrounding OSIRT as a FLOSS product.

9.6 Method

9.6.1 Interviews

Interviews looked at OSIRT's integration and the impact of FLOSS into a police force with three participants being interviewed; an Inspector, Detective Constable and IT Administrator. All of who work at the same police force. The police service in this case-study has approximately 40 active OSIRT users. The three participants were chosen as they all have a different perspective when integrating or using software. Questions to these participants looked closer at OSIRT's integration as a FLOSS product and how it can make an impact. These questions looked at five key areas: Trust, maintenance, technical support, cost and training.

All interviews lasted between 15 and 45 minutes.

9.7 Interview results and discussion

9.7.1 Trust and security

A common question received in one form or another is "How can I trust this software?" this is an important question any user should be asking when using software, but it is particularly important on sensitive systems such as policing where evidential artefacts are being obtained. All three interviewees highlighted being able to trust software as being an important factor of usage. The Inspector said "We trust OSIRT because we've spoken to you, and we can contact you. If this was some software made by 'who knows' then it would be a different story". The IT administrator also highlighted the fact OSIRT being open-source made trusting "easier" and although they are "not an advanced programmer" just the thought of the source code being available provides peace of mind.

Without being a large software distributor, it is, understandably, hard for those to trust a product made by an individual, making OSIRT open source was an attempt to assuage those concerns. OSIRT is both linked to a university and has collaborative links with the College of Policing, aiding in abating trust issues.

9.7.2 Maintenance

Updating is a challenge that is faced by any development team, but as a lone developer working on an FLOSS project, this concern feels amplified by potential consumers. The IT administrator highlighted this initial concern surrounding OSIRT, “We need to ensure our systems are water-tight, so updates are important.” The Detective Constable highlighted the dynamic nature of their work and the importance of keeping abreast of current technological advances as a key driver for updates “It feels the nature of my work changes on a yearly basis, who knows what I’ll be working on next year, so having a tool that keeps on top of that, like OSIRT has been, is important to me”.

The Inspector also noted that updates were “important” but spoke about skills within the police service that may aid in development. Some police services within the UK are adopting ‘cyber specials’, a volunteer group with exceptional skills in areas of cybersecurity. The Inspector said that “Given that OSIRT is available [open-source] means we can look at giving the [cyber] specials tasks in updating OSIRT”. OSIRT, presently, has no developer community beyond the author so an opportunity to work with volunteers in policing roles provides a good opportunity to extend and maintain OSIRT.

9.7.3 Technical Support

While closely linked to ‘maintenance’ the ability to provide support and help if needed was an issue raised by all participants. The Detective Constable, who is a daily OSIRT user, highlighted the need to be able to reach out and how “scarce” technical support is, particularly for free tools. “The thing with paid for tools is that, as part of the contract, technical assistance is part of the cost, so we can reach out”. This officer felt that was not always the case with free tools, where there is no contact available. “I’ve had my fingers burnt before where I used some open source tool and it stopped working with an error

message, but I had no way of contacting the developer". The Inspector echoed this sentiment, also adding the ability to reach out and get support if needed was "crucial".

The IT administrator agreed with this, too, but said that this is "par-for-the-course" using FLOSS and that expectations of support should be lowered. "To me, this is the sole trade-off. You lower the initial costs, but may face larger ones supporting free software".

9.7.4 Cost

Unsurprisingly, the cost of OSIRT was a driving factor in its implementation within this police services' system. The Inspector said that they had looked at "a couple of other tools", however, the cost of these tools was "too high" with some of the tools being "£60-£150 a user per year." The Inspector also highlighted that buying licenses could be better spent, "If I wanted to roll that out, that would cost me thousands but I have OSIRT for free which means that budget can be spent on other things."

The IT administrator also noted cost and said "money does not necessarily mean better quality". While the administrator said that where proprietary software was used, they were in a position to look at FLOSS alternatives if needed. The administrator said that some forces "may not have this flexibility [to introduce FLOSS] due to policy, but things are changing."

The discussion surrounding policy is an interesting, if not inconsistent, one. Policies can range from all forces to force-wide to even local level and are hard to pin down as they are not necessarily released for public consumption. There is a policy in some forces, according to the IT administrator, that prevents software from being "networked" (that is, installed across all machines) where there is a lack of support option for that software. This does link back to the comment received in the OSIRT usage questionnaire, where a response noted "no technical support available which has made full implementation difficult within my agency". While "individual officers can request any software", to be installed, stressed the IT administrator, the decision to make it fully accessible across the network is still adjudged by 'policy'.

The Detective Constable was, seemingly, least averse to cost and instead highlighted the importance quality software was to deliver the "best service" whether the best software was free "shouldn't decide what's best for the best results, luckily OSIRT for me is the

best tool for the job”, but they “understood” why management would be forced to look at free alternatives. Given that integration of software is not a Detective Constable's concern, their response is not a particularly surprising one with regards to cost.

Software where there is no immediate charge may invoke a ‘try before you buy’ response as there is not a commitment to integrate the product if it does not work out.

Monetary costs are not the only considerations to any implementation of, or change to alternative, software. Further considerations include costs in time, deployment and training.

9.7.5 Training

One issue surrounding the use of more FLOSS products was the need to provide training on the new technology. This is not particularly a FLOSS issue, as any piece of software will require familiarisation. The Detective Constable spoke about the “comfort zone” and changing an officer's workflow may cause them to “resent” the new software; highlighting the need for a robust training plan to abate those concerns.

The Inspector highlighted additional training as a cost/benefit trade-off “Of course you get the software for free, but we have things in place already and replacing software means training, it means time, and we have to trade-off the cost of licenses versus the cost of training”.

OSIRT is fortunate in that it is used as the tool on the RITES course, providing officer's hands-on use over the five-days as part of a wider training package. Additionally, as part of OSIRT's development, tests are conducted by means of observations, SUS questionnaires and a cognitive walkthrough. Conducting these tests, arguably, enhance OSIRT's ease-of-use which may then lead to require less training for OSIRT itself.

9.7.6 Summary of interviews

This case study highlights experiences, thought-processes and issues faced by those using and making decisions when integrating software into systems. These interviews are also reflective of the conversations had with several law enforcement officials within the past,

and while anecdotal in nature, does support the need for, and successful implementation of, OSIRT in law enforcement systems.

9.8 Summary

Developing OSIRT has been a highly rewarding experience and has provided opportunities to deliver a useful tool for law enforcement. OSIRT's growth has seen it shift from a simple training tool to use across the globe, with a global userbase. While OSIRT's growth is exciting, it has brought with it additional challenges as highlighted in this chapter. OSIRT was written only with UK law enforcement in mind, and as such is British-centric in its design. Obviously, the nature of the Internet makes nothing localised and purposeful software will disseminate to wherever it finds a use, bringing with it new and unknown challenges.

Being an academic, sometimes it is easy to forget that software must be shipped and that people are going to be using it, and will need support. Thankfully, OSIRT is buoyed in the policing community with many questions answered before being contacted personally. That said, if OSIRT did not have that internal support it would be considerably harder to manage as an individual.

10 TRAINING OF LAW ENFORCEMENT OFFICIALS TO CONDUCT OPEN SOURCE RESEARCH WITH OSIRT

INTRODUCTION

To aid digital investigators in conducting open source research, the UK's College of Policing runs a five-day 'Researching, Identifying and Tracing the Electronic Suspect' (RITES) course. The RITES course provides an opportunity for LEOs, regardless of skill-level, to gain proficiency in lawfully obtaining intelligence and artefacts from the web. In addition to investigatory skills, the RITES course adopts the usage of the free and open source investigative software package Open Source Internet Research Tool (OSIRT); a tool designed specifically to assist in conducting open source research.

This chapter's objective is to understand how LEOs are trained to conduct open source research, and whether the training package and OSIRT is effective when officers are back on-the-job.

Samples of raw data for this chapter are in appendices E and F.

10.1 Background

10.1.1 Designing Training Courses for Law Enforcement and Applying Learning Styles

Similarly to courses structured for training law enforcement in digital forensic investigations (Genoe, Toolan and McGourty, 2014; Stephens, 2012), the RITES course requires an ability to problem solve, pay attention to detail and have a mindset for

investigation and intelligence. Considerations are directed by course aims “to provide investigating officers with the skills necessary to obtain, evaluate and use online information ... apply[ing] best practice in respect of proper authorization and recording processes for online investigations” (College of Policing, 2017).

For a number of years, police training programs adopted a “militaristic environment” (Birzer, 2003, p. 30) which a number of authors (Birzer, 2003; Haberfeld, Clarke and Sheehan, 2011; Vodde, 2009) state is not conducive to learning, as “it is essential that training is conducted in such a way as to be as meaningful as possible to the adult participants” (Birzer and Roberson, 2007, p. 226). The RITES course adopts both andragogic (i.e. self-directed learning and sharing of experiences) and pedagogic (i.e. dictating learning in the form of traditional lectures) approaches to learning which seemingly prove efficacious when training police officers (Birzer, 2003; Haberfeld, Clarke and Sheen, 2011; Queen, 2016). Tong, Bryant, & Horvath (Tong, Bryant and Horvath, 2009, p. 210) state that “training and learning styles need to reflect that uncertainty of police work and the principles that should inform practice.” Traditionally, lecture style approaches to educating learners are “almost always the most inefficient way of learning” (Grace, 2001, p. 125), and while it is unlikely for the RITES course to accommodate every style of learning, a concerted effort is made to engage their audience. By embracing modern approaches, College of Policing trainers afford the officers a better chance of applying their acquired skills to real-life scenarios.

10.1.2 Design of the RITES Course

The course is split into one to two-hour chunks of key topic areas, covering approximately five topic areas a day (Figure 10.1). Each topic area is then either proceeded or injected with practical sessions or discussion from the cohort, which is facilitated by the instructors. Practical sessions also include building upon a fabricated case using OSIRT over the five days. On the final day, the group members are examined by means of an unseen open source investigation. The artefacts they obtain through OSIRT from the ‘investigation’ are then applied to answer questions on a computer-aided, open book, multiple-choice examination. The course is then concluded with a reflection of the previous five days. Figure 10.2 represents the layout of the learning environment.

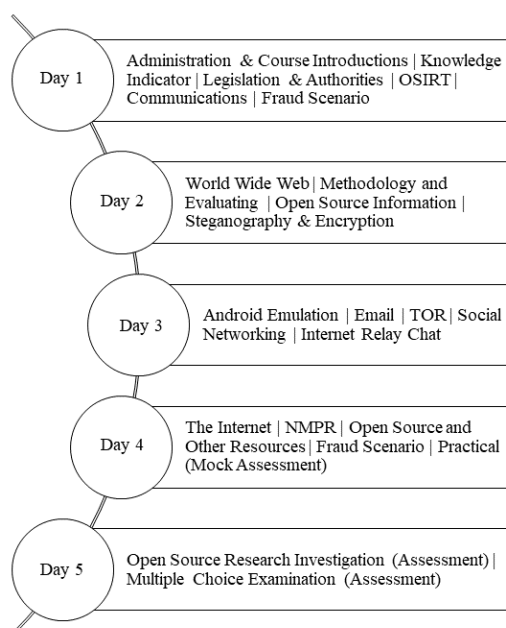


Figure 10.1 RITES course topics broken down per day

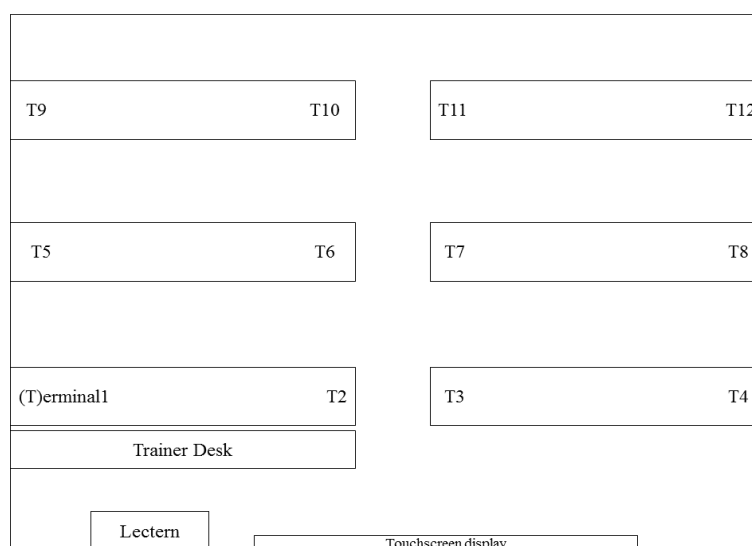


Figure 10.2 Room layout at the RITES course

10.1.3 Using Software for Investigative Work

In an ever-growing digital age, and with changing expectations in police competencies, LEOs require essential skills and abilities to conduct online investigations. However, the skill-level of officers requiring such training is diverse with many not being, or having had the need to be, skilful with computers during their daily roles. The requisite for software based solutions has a crucial element to aid the proficiency of conducting open

source research and go some ways towards making “officer[s] more efficient, more effective, more knowledgeable, and better able to spend [their] time ... and by improving reporting capabilities” Roberts (2011, as cited in Hess, Orthmann and Cho, 2013, p.16)

10.1.4 Using Kirkpatrick’s Training Evaluation Model

A number of courses within the policing context (Capacity Building and Training Directorate, 2012; Genoe et al., 2014; Stephens, 2012) have utilized Kirkpatrick’s evaluation model. Developed in the 1950s (Kirkpatrick and Kirkpatrick, 2006; Kirkpatrick and Kirkpatrick, 2016), it is now the “most widely used framework” due to its design and levelled implementation (Tamkin, Yarnall and Kerrin, 2002, p. 3). Furthermore, Kirkpatrick’s model encourages learner participation via four levels: Reaction, Learning, Behaviour and Results. The model is popular as it places value on learners’ views, suggestions and opinions. The four levels look at several key areas to evaluate effectiveness such as;

Reaction – Level 1: participants thoughts on the course, its relevance and their own engagement

Learning – Level 2: knowledge, skills and abilities (e.g. performance), attitudes and confidence

Behaviour – Level 3: changes in job behaviour due to training and the applicability of learned skills/content

Results – Level 4: impact of the training and content on the business

10.2 Methodology

A mixed method approach was adopted, using questionnaires, evaluations and observations. These methods were chosen due to their ease of mapping with Kirkpatrick’s evaluation model. Evaluations in this study included key questions to examine the courses effectiveness based on the Hybrid Kirkpatrick’s Evaluation tool (Kirkpatrick Partners, LLC, 2010), which provides example questions for levels one to four. For example, knowledge retention and applicability to real world environments, were sought through free-form answers and Likert scale statements. Limitations of Kirkpatrick’s model are abated by looking at the value of information across each level; avoiding the linear

approach criticized by Tamkin, Yarnall and Kerrin (2002). Using this approach ensures the most valued information of course effectiveness is collated. This study evaluates levels three and four from the perspective of attending officers; taking into consideration their experience, rank and own ability to assess their behavioural change, including the impact of the course on the working environment. Participants were made up of an opportunity sample of twelve serving LEOs attending a RITES course, containing six males and six females. Participant jobs ranged from Detective Constables and Sergeants, to Analysts. The average service time was sixteen years; with a minimum of 8 and a maximum of 26 years.

A pre-course questionnaire was completed electronically to gain insight into the participant's expectations of the RITES course and to establish current skill-levels at conducting open source research. Additionally, the questionnaire asked participants of any software they currently use to conduct open source research, if any. At the end of each training day, the cohort completed a paper-based questionnaire asking to evaluate each day's topic areas ("Easy" to "I'm lost"), the pace of the session ("Too slow" to "Too fast") and whether OSIRT was effective in that day's session ("Strongly disagree" to "Strongly agree"). The participants were also afforded an opportunity to freely express their thoughts for the day.

An electronic immediate post-course questionnaire was distributed on the final day; covering a range of areas such as course content coverage, course assessment and the applicability of OSIRT. All statements conformed to a ranked multiple-item Likert scale, and took a flipped phrased approach to reduce response bias (Field, 2006). Finally, eight weeks after course completion, an on-the-job questionnaire was distributed electronically to identify if the RITES course had an impact on their role.

Both immediate and delayed post-course evaluations contain multiple-item measures across the four levels of Kirkpatrick's model. In the case of this study, factor analysis was infeasible due to population size, however, Gliem and Gliem (2003) note the importance of calculating Cronbach's alpha for scale items. Cronbach's alpha is a popular statistical analysis to measure reliability among variables of interest (Tavakol and Dennick, 2011). Cronbach's alpha is adopted in this study to measure statements relating to the different

levels of the Kirkpatrick's model. Analysis of Cronbach's alpha was conducted using IBM SPSS 24.0.

Furthermore, for flipped phrased items and to prevent a negative impact on the reliability score, the negative statements were reversed before calculation. Common levels of internal reliability/consistency of alpha (α) were employed with acceptable values of 0.7, through to excellent values of $\alpha \geq 0.9$ (George and Mallery, 2003; Loewenthal and Lewis, 2015; van Griethuijsen *et al.*, 2015).

Observations were adopted providing instructors with the chance to look at each participant's level of engagement, demonstration of skills, through to how the course and OSIRT would be useful on-the-job. Mindful of the role the observer plays on the learner, considerations were made towards how the learner's behaviour can be affected, by the presence of an observer within the training environment. Hallenberg, O'Neil, & Tong (2016, p. 109) write that "Van Maanen describes four typologies" of a researcher. In this study the author, as an observer, can be classified as a 'fan', i.e., a researcher who is "interested in observing police practice as it happens" (Hallenberg et al., 2016, p.109). The observer kept a daily diary of events, with reflections made to correlate with learner comments and ratings from course evaluations.

10.3 Results and Discussion

10.3.1 Pre-Course Questionnaire Results

Challenges faced by participants when conducting open source research generally fell into one of three categories: the need to be trained in open source research, an absence of IT knowledge, or software tool 'overload'. Current software tool usage is consistent with feedback previously received in that officers use a varied array of software that is either free or built into the computer's operating system. Three respondents said they did not use any software, with one noting they do not have access to the necessary technology. No participants have previously used OSIRT as part of their investigations.

All participants said they prefer practical learning where a "realistic" and "hands-on" approach can be applied to real-life investigative scenarios. Responses show that expectations of learning were centred around having the necessary tools available to "research and capture" and "how to best use these practically" to "maximise [the] chances

of finding what [they] want to find”. Additionally, participants wanted to know “the ‘correct and best’ way of completing research” using “OSR techniques” that was both “safe [as well as showing] potential pitfalls when conducting OS research”.

Finally, “certification”, “knowledge” and “confidence” were stressed as attributes officers were wanting to achieve throughout the course. Other responses showed concern with monitoring their own digital footprint while conducting an open source investigation. Replies also showed that participants were using the course to pass knowledge and understanding back to colleagues in the working environment.

10.3.2 Daily evaluations and observer comments

10.3.2.1 Course pace and difficulty

Daily averages and the immediate post-course evaluation show the overall difficulty noted by most participants to be ‘Just Right’. Figure 3 demonstrates, overall, two learners felt the course was ‘Very Difficult’, speculatively this may have been linked with their perceived computer literacy (Figure 4) and three felt the course to be ‘A Little Tough’. These results are not unexpected, as observations showed a small number of the cohort readily admitting they were computer novices, one going as far to say they were a ‘technophobe’. Other comments lend themselves towards aspects of learning, where one respondent felt the course to be tough as their basic knowledge was poor, however, they emphasized that the trainers were helpful in assisting as much as possible, and being patient with them. The respondent felt these points helped make the course thoroughly enjoyable and “took a lot” from it.

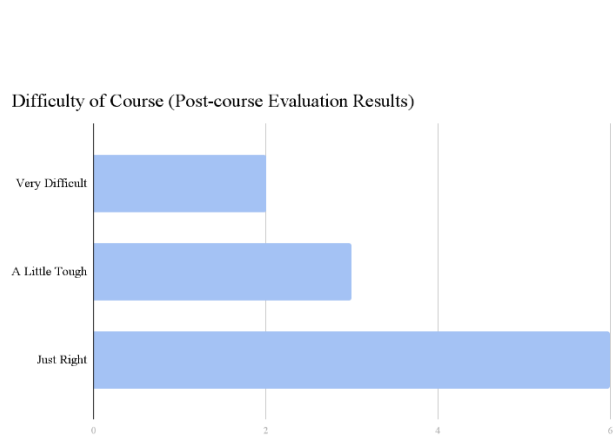


Figure 10.3 Overall Difficulty of the Course

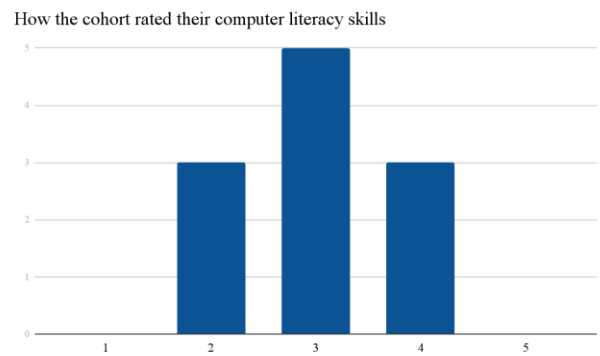


Figure 10.4 Cohorts' Rating of own Computer Literacy (1 ('not proficient at all') to 5 ('extremely proficient'))

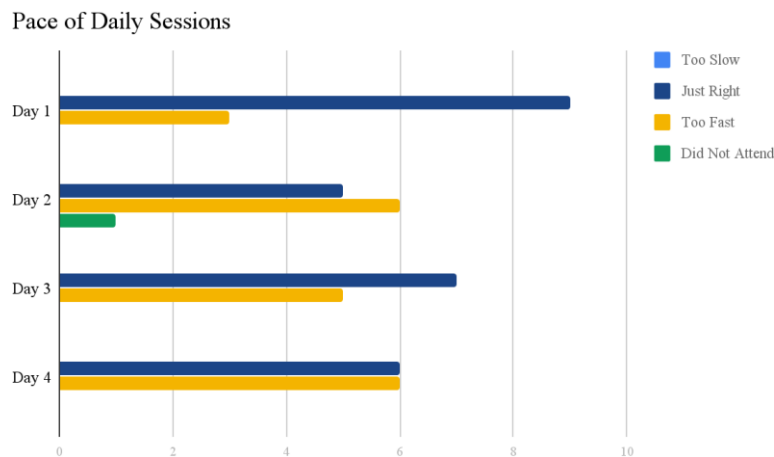


Figure 10.5 Perception of Daily Session Pace

The cohort throughout the week were engaged and responsive to interactive sessions. Additionally, observations showed that the trainers addressed issues with pacing, providing one-to-one guidance when needed. While the cohort were frequently split about pacing of the daily sessions, as seen in figure 5, pace was observed to be problematic on days where complex topics, such as encryption, were taught. Participants offered feedback in their daily evaluations for these challenging topics, one noting they “saw some people confused about terminology” and suggested that “perhaps ... more basic explanation[s]” could be provided. Given the technical complexity of some of the topics, it is understandable the cohort would find these difficult to immediately absorb. As with

any learning, the time taken to master and acquire knowledge differs per learner, and added with technical complexity of a topic, a “too fast” response would not be atypical given these circumstances.

Observations showed that there was good communication during these particularly tough sessions, with the use of analogies by the trainers making complex topics relatable to everyday life. One participant highlighted this in their comments, saying “comparing ‘digital’ to ‘real-life events’ assists in understanding”. Feedback also showed that although some sessions were “hard work”, they were still “very interesting” and “enjoyable”.

10.3.2.2 OSIRT

To capture the usage and effectiveness of OSIRT, officers were asked to rate the tool using a Likert scale, from strongly disagree to strongly agree, and provide comments based on the statement “OSIRT has been effective in today’s training” considered by the learners.

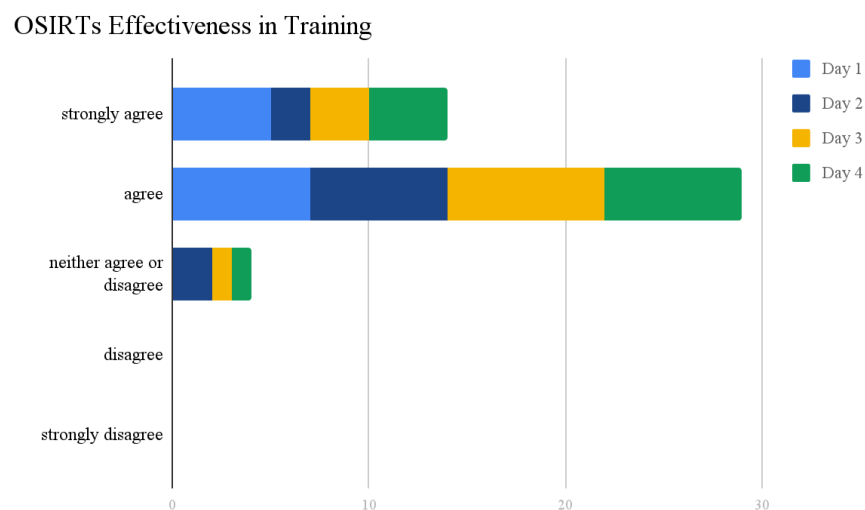


Figure 10.6 OSIRT's effectiveness

Results showed that OSIRT was successfully applied and received by learners throughout the course. Across the four days, which were analysed for OSIRT’s effectiveness, 91 percent felt they “agree” or “strongly agree” with the statement presented (Figure 6). Prior to the course, none of the officers had used OSIRT. Daily evaluations support this

assertion, with many officers using freeform answers to praise the tool noting its usefulness with comments such as: “it is extremely useful for structuring search and investigation process”, “[it is] very useful and makes things easy”, “it streamlines the process and makes it easier as an investigator”, “[it is] very very useful! - couldn’t have done it without OSIRT” and “everything can be done in OSIRT”.

In the post-course evaluation, learners were asked “Was OSIRT useful during the course?”. Everyone responded “yes”, expressing their praise for the tool with eight participants stating they would be using OSIRT to enhance the capabilities within their role for conducting OSR. Two participants expressed they were ‘unsure’ and one stated they would not be using the tool. The reasons for not being able to use OSIRT were concerns over IT restrictions. A positive response from the cohort on the toolkit also meant that OSIRT was mentioned as a specific skill they would apply back on-the-job and as an important aspect learned on the course. The toolkit satisfies several challenges noted by the learners in their pre-course questionnaire, for instance: the current state of use of a number of tools etc., where a number of participants noted the tool as “excellent” and “fantastic” which is “well designed” with participants “amazed” that the software is free.

10.3.3 Participant course evaluation

Eleven officers completed the immediate post-course evaluation, with eight from the same eleven officers completing the delayed post-course evaluation. Participants identified their attendance on the course was “to acquire new skills” (nine), “to improve current knowledge” (seven), “to familiarise [themselves] to train other in OSR” (five) and “to become certified in OSR” (four). One officer expressed the course was “mandatory”, with two others stating, “to use at work” and “to ensure those in my office with no training do not have to carry the responsibility of conducting and capturing open source research without that training” respectively. The course, at the time of writing, is the only accredited course in the UK to help officers conduct online investigations efficiently and with ease and knowledge of processes and relevant data. Findings demonstrate the course is delivered well, meeting expectations of officers.

10.3.3.1 Reaction – Level 1

Level one statements look at, for instance, the engagement of officers and relevance of training. When applying Cronbach's alpha to four statements categorized as 'reaction', an alpha (α) score of 0.70 was found; an acceptable reliability. Further to this, results from the immediate post-course evaluation demonstrate a strong percentage of officers who agree they took responsibility for their learning and that trainers enhanced the learning on the course.

Results from both evaluations showcase OSIRT's usefulness and effectiveness at helping investigating officers "capture online resources" as well as helping to retain and maintain audit trails. Respondents expressed that capturing and finding open source information was the most relevant information taken from the course, with all recalling OSIRT and evidential capture as their most memorable content.

To assess training satisfaction, participants were asked open-ended questions on whether anything could be improved on the course. Several yielded responses such as "no, it was pitched about right" and "no I liked it". While three officers felt the course could run longer due to the quantity of content covered. Others provided positive and constructive improvements, mentioning they would have liked more on topics such as social media, cryptocurrency and "more about research of an individual". Officers expressed no real issues, showcasing the courses effective delivery for this cohort.

Expanding on this, the delayed post-course evaluation also sought feedback to discover what topics could be added. Officers felt the course needed more on the "levels of open source research", "case law", "how websites are created" and "social media". Many of the suggestions will be considered for future delivery of the course.

10.3.3.2 Learning – Level 2

To achieve level two of Kirkpatrick's model (i.e., identifying learning and its effectiveness), several questions and statements focused on knowledge, skills, confidence, relevance and learning styles. A key element useful to identifying the effectiveness of the course content was to ask learners to pick three important concepts/topics they learned during the course. Results show that using OSIRT was the most mentioned topic (nine), followed by steganography (four) and social networking

(three). These topics were also specific skills which officers plan to use in their job when asked.

Eight statements covering aspects from quality of content, delivery and confidence of application were asked of participants. A tally of the collated responses for level two demonstrated that 86% of the cohort achieved learning on the course, with 91% feeling that there was sufficient time allocated to delivering the course content. Applying Cronbach's alpha shows a score of 0.88 across statements demonstrating a strong reliability between item correlation across eleven participants. Additionally, it was a strong indication that officers felt they learned skills transferable to the workplace.

To build a comparison between the immediate post evaluation questionnaire, officers were asked to identify what content they remembered the most. Mentioned were: OSIRT (1) and searching, capturing (6) and analysing (1) open source information. Although OSIRT was not explicitly mentioned by all officers, capturing open sources was mentioned by all. The only tool used on the course to capture evidential data was in fact OSIRT, so it can be inferred that OSIRT was an aid to their learning. Further testament to this are free comments provided which mention how it was "nice to discover OSIRT".

10.3.3.3 Behaviour – Level 3

So far results have shown participants were satisfied with the training and OSIRT, while demonstrating digestion of the subject matter. Level three is used to determine how much knowledge, skills and attitudes have been transferred following training and how on-the-job behaviour has consequently changed.

A strong consensus was illustrated by participants, in the delayed post-course evaluation, towards the practical application of course learning and OSIRT within four weeks. A few mentioned short delays due to work commitments, however, found the course materials sufficient in refreshing learning. Other officers noted no difficulties or "nothing unusual" when applying gained skills. One officer positively reflected by articulating "there were some things on the course [they] wondered why [they] were shown but it made more sense a few weeks after the course". Demonstrating development and maintenance of relationships between the training and business requirements.

Officers were then asked to consider and rate, using a Likert scale from 'little or no application' (recoded to 1) to 'very strong degree of application, and desire to help others

do the same' (recoded to 5), their on-the-job behaviour in accordance with course objectives. In the first instance, Cronbach's alpha returned a negative result. Field (2006) states that in cases where poor correlation between items [is found,] then some should be revised or discarded. In this instance two statements were removed leading to $\alpha = 0.76$ and demonstrating a strong internal reliability among the items.

The two statements excluded asked officers to consider their "Ability to navigate the web in order to capture and evaluate relevant data" and "Obtain familiarity with social networking sites". While these introduced a negative alpha, a breakdown of the statements demonstrates a positive impact from course back in the workplace. Results showed that each officer felt a strong degree of application (7) or very strong degree of application with desire to help others (1) with their ability to capture and evaluate relevant data. Furthermore, these concepts were formatively fed back by officers in free-form throughout post-course questionnaires.

A varied response was given for the statement "obtain familiarity with social networking sites", where three officers expressed a 'moderate degree of application' and five who expressed a 'strong degree of application'. The reason for this disparate response is not known, but speculatively it may be due to the statement's phrasing. For example, "moderate degree of application" for the statement "Obtain familiarity with social networking sites" does not align. On reflection, a statement such as "Usage of social networking sites" would have been less ambiguous.

10.3.3.4 Results – Level 4

This level looks at the impact of the course and OSIRT on the business through perspectives of the attending officers. Both post-course evaluations are used to assess 'results' e.g., the perceived, and resulting, impact of the application of learning to the job for departments and/or organization.

Immediate post-course evaluation found all, bar one, officers expressed the course would make a difference to the way they do their job. Officers expected to see positive impact in areas such as 'greater confidence in conducting OSR' and 'feeling better equipped to understand, speed-up and improve the OSR process'. Responses from delayed post-course evaluation corroborate this, finding OSIRT and capturing of open sources as the

main enhanced areas in officers' jobs. Course materials and OSIRT "slotted into [their] role quite nicely" and the "course ... help[ing] with some of the finer details". OSIRT's success as an investigative tool, its influence on officers' roles and asset to police departments was epitomized by one officer noting: "our team now uses OSIRT and the majority of us use it most days".

Officers saw improvements in most areas of their work, as demonstrated by Figure 10.7. Interestingly, only three respondents saw an increase in the quality of their work. The author speculate this is caused by professional bias, whereby officers may have felt the work they previously produced before the course to already be of high quality, and hence nothing to improve upon.

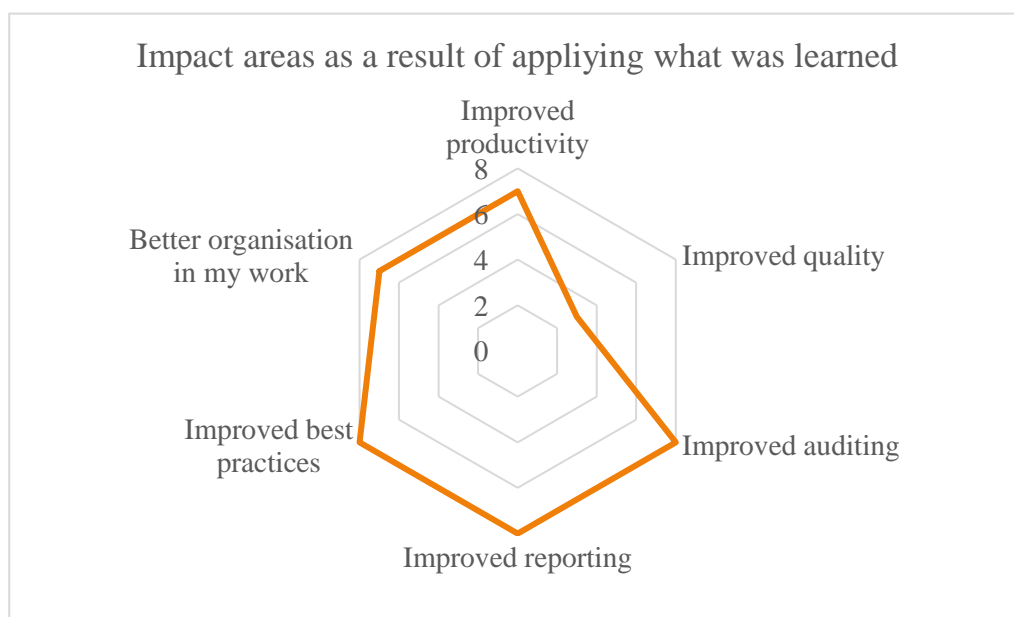


Figure 10.7 Impacts felt due to application of learning (number of responses that selected the 'impact area')

10.4 Discussion

The daily course evaluations represented well-rounded views that sessions matched the learning styles officers had noted in the pre-course survey. Occasionally, topics challenged a few of the cohort, but this was abated with trainers providing one-to-one sessions. Observations also confirmed that some of the cohort were forthright with their

IT abilities. This may explain why the advanced topics, such as encryption, were a challenge to those participants.

While the pre-survey showed little open source research experience among the officers, results indicated that all LEOs learned open source research skills during the course. This was highlighted by the fact that all the cohort passed the examination. For a majority of officers present, the overall pace of the program was just right for their learning style and speed. However, given the variety of skill-sets on the course, several participants did feel the course went a little fast for them. Suggestions for improvements to slow down the pace of certain sessions were relayed to trainers. Although these problem areas were identified, the consensus was the course provided a number of key topics and skill-sets which LEOs can utilize in the workplace. Results demonstrated that many turned back to their course notes and materials on-the-job, again showing the application of knowledge and skills learned.

The success of the RITES course was further strengthened with the use of OSIRT and its function in aiding open source research. Responses sought throughout this study, from daily surveys to direct and delayed post-course evaluations, saw the cohort provide positive responses to the tool's effectiveness. Further praise was vocalized by LEOs to the usefulness and ease-of-use of the tool, particularly for helping officers in its versatility and ability to methodically conduct open source research.

10.5 Limitations and future research

The main limitation of this chapter is the number of participants, a small group of officers. Future research will look at extending this research, looking at multiple cohorts of officers trained under the RITES course to analyse, compare and discuss findings toward the effectiveness of the course and OSIRT in helping investigating officers conduct open source research.

10.6 Chapter summary

This chapter looked to evaluate the overall effectiveness of the RITES course offered by the UK's College of Policing, OSIRT's integration into the course and its subsequent usage on-the-job by LEOs. Results showed the RITES course as an effectual training aid

to LEOs conducting open source research, and OSIRT as an effective tool for LEOs who conduct open source investigations as part of their role. Evaluation of the course took the approach of Kirkpatrick's model, where study responses showed knowledge transfer to real-life investigations, skill-sharing and the integration of OSIRT within their teams.

As the march of technology forges ahead, so must the education of those having to navigate its ever more complex wake. The police must evolve symbiotically with modern life to stay on top of the types of crime that now dominate the headlines. To grow effectively, their learning techniques and educational ethos must harness the most efficient teaching styles and tools; the RITES course and OSIRT is helping do just that. By engaging learners and diversifying their classroom experience, the police are encouraging the best retention for information. Incorporating OSIRT into this experience can improve the efficacy of learned skills in providing a successful and efficient tool. The RITES course and OSIRT are an ideal integration of modern learning and modern tools, to help police keep up with this modern world.

11 CONCLUSIONS AND FUTURE WORK

With people living an ever more public life online, open source research provides law enforcement with a useful tool in its investigative arsenal. While judicious use of open source research has many benefits, care must be taken to ensure investigations stay within the confines of the law and guidance. Even if the information is considered to be open and publicly accessible, it is not boundless. The debate of privacy versus security will inevitably be at the forefront of any discussion when collating information about individuals, and it is, in the author's opinion, a debate that can never fully be satisfied. Despite its irresolution, it is pivotal such questions are asked to maintain checks and balances.

OSIRT's integration and contribution to law enforcement has been successful and has seen OSIRT taken up by officers not just around the UK, but across the globe, along with being written into standard operating procedures for conducting open source research at several UK police forces. OSIRT is fully integrated into the College of Policing's RITES course, where it is has been, and continues to be, trained to hundreds of officers. Results from interviews, questionnaires and observations have shown OSIRT provides useful capabilities to officers, making a much-needed impact in their roles.

The following sections recap the goals of this thesis, summarises its findings and contributions along with a discussion of the limitations and future work of this research.

11.1 Goals, findings and addressing the research questions

This thesis had several research questions and aims, this section recaps and discusses the questions along with the goals and findings.

11.1.1 What constraints do law enforcement in the UK face when conducting open source research?

This question's aim was to look at legal, procedural and ethical issues surrounding conducting open source research. Chapter 2 reviewed definitions of what 'open source' meant and noted that definitions of open source vary depending upon the agency; but all stress that open source is publicly available. However, there is a strong debate surrounding what constitutes publicly available, and whether that automatically means that law enforcement are entitled to collect, analyse and store that data. Laws, such as RIPA (2000), are used if investigations are considered to be directed surveillance or are covert in nature. Yet, ACPO/NPCC guidance stresses it is unlikely for officers to require authorisation under RIPA for conducting open source research at lower levels. Although there is a gravitation towards requiring RIPA authorisation in newer advice, it is still not cast-iron by any means.

The impact of legislation on law enforcement to conduct open source research was analysed and discussed by means of conducting interviews with 22 law enforcement officials in chapter 4. It was evident from the interviews officers faced a legal and ethical minefield when conducting open source research, and a cautious and considered approach must be taken. This was especially noticeable when the line between "overt" and "covert" is not as clear as NPCC/ACPO guidance makes it first appear, and for some officers, it was better to be safe than sorry and to obtain a DSA under RIPA. New legislation, such as the GDPR and the subsequent Data Protection Act (2018), make provisions for ensuring collected data is managed, stored appropriately and eventually discarded.

OSIRT meets the criteria set out by ACPO/NPCC guidance and College of Policing training by maintaining audit trails, logging actions, ensuring officers justify their decisions when collecting artefacts and packaging it all within a case file for encryption; this also satisfies data protection legislation.

It is not possible to completely satisfy this question in a quantifiable way, but by providing a voice to those in law enforcement who are impacted by the problems discussed, and using their knowledge of their profession, it meant that a software tool could be generated to ensure they stay on the right side of the law once appropriate authorities were approved.

11.1.2 What do law enforcement need from a software tool when conducting opens source research?

This question aimed to gain an understanding of what law enforcement officials need to effectively conduct open source research from a practical perspective. To answer this question, a review of existing software tools trained by the College of Policing on the RITES course was conducted in chapter 2. In addition, a questionnaire was distributed to police forces to discover the type and range of tools used when conducting open source research; seen in chapter 4. Immediately it was evident officers used a range of tools that varied in price and quality, with little regard for standardisation. While the College of Policing attempted to standardise the toolset on its RITES course, officers would invariably bring their own preferred tools in; this would add to an already difficult dynamic during the course. Interviews with the lead high-tech crime trainer noted the confusion faced by the cohort from tool variance, and this led to the generation of a software specification for an all-in-one open source research tool; which ultimately became OSIRT.

Chapters 5 and 6 focused on the OSIRT prototype, which was an extremely useful method of requirements gathering and a way to obtain feedback from those officers who OSIRT is intended for. These chapters discovered a tool like OSIRT was needed by law enforcement and prompted the creation of a release version of OSIRT of which the development and discussion of its impact and contribution were made in chapters 7, 8, 9 and 10.

OSIRT's contribution to law enforcement has been impactful. OSIRT is now integrated in to several police forces' standard operating procedures for conducting open source research, as well as officers all around the country using it as their preferred open source research tool. This is in addition to being a pivotal and central aspect of the RITES course. OSIRT's contribution goes beyond policing, too, and is utilised by councils, Trading Standards and Food Standards Agency as well as individuals and private companies. OSIRT has reached across the globe and there is evidence OSIRT is used in Barbados, Spain, Portugal, Germany, Italy, Australia, Canada and USA to name a few.

11.1.3 What are the unique elements when engineering a software solution for law enforcement?

This question was one that was discussed and reflected upon throughout this thesis. As a question it is hard to quantify, but the experiences shown throughout the thesis and by providing a critically reflective discourse when discussing decisions provides an insight into challenges and unique elements. This research offered rich insight through unprecedented access to the College of Policing, police forces and other organisations; offering a unique understanding and perspective into building a bespoke software product.

With OSIRT being open source, one is left wondering what would have been the case had OSIRT adopted a different method to its development. For example, if OSIRT was closed source and created ‘for profit’, would the level of access that provided rich data for this research been as easily available? By creating OSIRT in an accessible way, such as including users in its development and by maintaining OSIRT as a free and open source product, it no doubt made a larger impact and contribution which offered the insights seen within this thesis.

11.1.4 How can developers involve users in the design process in a ‘closed’ environment?

This question was built upon from the start of this thesis and followed OSIRT’s creation throughout. The key aspect to be taken away from this question is that when working with those in jobs that are heavily gatekept, and whose time is scarce, it is imperative to ensure time with the participants is maximised and not wasted with excessive participation and micro-refining smaller details that can be ironed out later.

This research found observations, particularly over a period of a few days, to be the most effective method to build trust and gain a better understanding of the user’s needs; but they still only tell part of the story. When following a UCD approach, an aspect worth reiterating is the need to triangulate data collection methods; including both quantitative and qualitative approaches. An example of why only following one method is not optimal can be seen in the prototype results, where OSIRT scored highly in the quantitative SUS questionnaire, yet this did not reflect what was observed. It would have been easy to assume the SUS results were enough and OSIRT’s development would have stopped at the prototype stage; even with its flaws. In many regards, the qualitative aspects of this

research enhanced OSIRT the most and offered the richest sources of insight into the needs of the participants.

This research has been extremely fortuitous with the level of access that has provided a source of rich data; however, that did not come for free. Relationships and trust needed to be built. This was a process that took many months, if not years, to forge. It is important to remember that UCD does not stop with the data collection, it is a process throughout the entirety of the software's life. It is building trust and rapport by doing many smaller things, such as responding to e-mails and telephone calls, it is talking to participants throughout and involving them in the entire process.

11.1.5 How can law enforcement be effectively trained to conduct open source research?

Her Majesty's Inspectorate of Constabularies made clear there needs to be a drive to see police officers, regardless of role, to be able to conduct routine digital investigations. Chapter 10 showed the RITES course, in conjunction with OSIRT, provides officers with those core skills, even for those officers who perhaps consider themselves to be non-technical. While officers did find aspects of the course challenging, when officers went back on-the-job they were able to apply what they had been trained in real-life investigative situations and that OSIRT was a driving force behind it. The study in chapter 10 also showed the importance of looking at the whole picture in regards to training and to not focus just on a snapshot of the training process.

11.2 Critical review of thesis, limitations and reflection

11.2.1 Sample

While access to law enforcement officers was not an issue for data and feedback, those that responded to interview requests and/or fill out questionnaires were, arguably, 'fans' and users of OSIRT. This means that feedback focussed more on positive feedback from OSIRT fans. Access to those who do not use OSIRT, or do not like to use OSIRT, are harder to find because they are unlikely to reach out, or do not visit locations where OSIRT is discussed (e.g. Police knowledge exchange forums, such as POLKA).

11.2.2 User-centred design

User-centred design (UCD), overall, provided a solid method for capturing user requirements which aided greatly in OSIRT's development. However, several limitations of UCD were noted within this research, or specifically, how UCD was applied. Firstly, appeasing all users is not possible and there were occasions where contradictory feedback was given from users. There were stages during OSIRT's development where the developer tried to please everyone, but UCD as a method does not recommend to please all users all the time; only to integrate users into the development process as much as possible. Wanting to please all users was the fault of the developer, who did not want to upset or lose users. However, this attitude was not only impractical but also increased the time it took to release new versions of OSIRT. The lesson learned from this is to manage expectations and better establish why the user has requested such an addition.

This also links into the previously discussed Pareto's principle, or the 80:20 rule, whereby 80% of users only use 20% of the features. While the questionnaire results of feature usage were not quite 80:20, it was a remarkably close, given the sample size, of 70:30. The advice here is to be selective and do not be afraid to say "no" to users for fear of offending them.

UCD can be a time-consuming process. For this study, beta versions of OSIRT were used by the RITES course cohort. However, unless the cohort were being actively observed, it was uncommon to receive feedback about OSIRT from the cohort beyond some positive comments and SUS results. There would be an occasional bug report and feature request, but not much else beyond that.

11.2.3 Prototype software engineering methodology

Pressman (2014) warned of the prototype trap whereby the developer attempts to extend a throwaway prototype to be a working product; this is often a bad approach that leads to a broken product. There was a stage nearing the end of the OSIRT prototype where users were quite convincing that OSIRT only needed a few minor adjustments to get it fully working, and this led to nearly falling into the trap of extending the prototype. I can resoundingly state this would have been a terrible idea. As a developer, you know the limitations of your software better than anyone else and while it may be easy to get caught up in the positivity of your work, staying level-headed and reflective is a good

characteristic to have. The initial disappointment your users have will be far better than having a broken product no-one wants to use.

11.2.4 WinForms

To recap, WinForms (Windows Forms) is a GUI framework available in the .NET Framework first released in 2002. WinForms is well documents and well catered for with third-party APIs and provides simplified methods for building GUIs either by dragging and dropping or dynamically. However, WinForms has one key limitation in that its ability to scale on monitors with a higher DPI is imperfect. Developer updates on Windows 10 go some way to achieve better scaling for WinForms projects, however, many police systems are still on Windows 7. While the alternative technology in the framework, WPF (Windows Presentation Foundation), is more complex to integrate it does provide automatic scaling for different DPI monitors. WinForms and WPF are not particularly cross-compatible, so replacing one with the other will require a complete re-write.

11.2.5 FLOSS and licensing

OSIRT is a free and open source project that uses the MIT license. The MIT license is permissive in nature and allows for source code to be used commercially as well as code to be modified and distributed along as it included both a copyright statement and a copy of the license. The initial idea behind selecting this license was to encourage other developers to fork and enhance OSIRT, or for others to tailor OSIRT to their specific needs; this did not happen. Some projects¹⁷, however, did appear to use code from OSIRT in their paid-for products and not disclose it. Unfortunately, it is extremely difficult to provide evidence when the projects are closed-source. The lesson here is while your intentions may be good while creating open source software, you must be prepared for a commercial entity who may take your work and sell it: the MIT license does permit commercialisation.

¹⁷ As to not libel myself, details are vague.

11.3 Future work

11.3.1 Further usability testing

OSIRT has been user tested via beta tests and observed, with its impact and contribution to law enforcement well evidenced, further work should look at comparing OSIRT to different methods of conducting open source research using differing usability evaluation methods. These evaluations would be conducted within a usability lab environment comparing the differing open source collection using typical usability metrics such as error rate, learnability, task completion time, etc.

11.3.2 A general framework for collaborative software engineering

This research has shown a methodological approach in which a developer can work with law enforcement to create bespoke software. However, this approach extends beyond this specific domain; with the outcomes, knowledge and insight generated being applicable to scenarios outside law enforcement and open source research. Further research should look at the ethnographic and user-centred approaches taken within this thesis and how that could be made generalisable to other problems to create a general framework.

11.3.3 Framework for software tools that obtain publicly available information

OSIRT is one of several tools that allow users to capture and collect publicly available information, but it follows a UK-centric approach to capturing digital evidence. While OSIRT does see usage across the globe, individual countries will have their own laws regarding open source capture and what may be applicable in one country is not necessarily applicable in another. Future work can look at analysing legislation and guidelines surrounding the capture of publicly available information. The analysis may benefit from generating similar themes of what constitutes ‘valid’ open source capture and providing a general framework for software developers of open source research tools.

11.3.4 ISO 17025

While ISO 17025 appears to remain in state of flux for open source investigations, it is on the radar of the Forensic Science Regulator. What accreditation would entail for open

sources is unknown, but OSIRT will be ready to integrate the required changes to validate to ISO 17025.

11.3.5 OSIRT

OSIRT is an ongoing project and continues to be developed. Suggestions for additions roll in from across the globe and provide a continuous source of something to implement and integrate. While there are people using OSIRT and it remains useful for them, OSIRT will continue to be developed. The follow subsections look at areas in which OSIRT can be enhanced.

11.3.5.1 Mobile

With the rise of mobile devices, and the potential of a shift away from traditional desktop computing, open source research by UK law enforcement may need to see a change. Phone emulators are already used by some individuals, and the creation of an OSIRT-style tool for emulators would be the next logical step.

11.3.5.2 Cross-platform considerations

OSIRT is a Windows only application, I have occasionally received e-mails from people asking if there is a Linux or Mac version available. While Mono exists for C# development, CefSharp and the heavy use of interop services from the WinAPI cannot be trivially ported to Mono for cross-platform development. OSIRT in its current state cannot be ported, and there are no medium-term plans to do so, but the long-term goals do envision OSIRT as a cross-platform tool.

11.3.5.3 Internationalisation

OSIRT started as a tool with a focus solely on UK law enforcement. Given the nature of the Internet, though, it was inevitable that a useful tool would make its way outside the UK. OSIRT is not currently able to be easily internationalised but can be with additional time and resources. Offers have been received from Portuguese, Spanish and Catalonians to translate OSIRT.

11.3.5.4 Training

While there are no plans to commercialise OSIRT at the moment, there is discussion surrounding the training of OSIRT and the creation of training packages.

11.4 Concluding remarks

To conclude this thesis, an e-mail from Tim Lainsbury of Dorset Police, and lead of the South West's open source research training and integration, summarises this research:

As the police began to utilise the internet, social media and other sites to gain information about events, suspects, witnesses and more we quickly discovered that there were limited untested ways of obtaining evidence in a credible way and recording it so that it would stand the scrutiny of the court system. Joe Williams attended Open Source Training [RITES course] at the College of Policing, observing what the police were doing and decided to write his own software as a solution for the Police to use to capture information from the internet.

He engaged with the college and officers that had been trained in this skill to create a web browser that gave credibility, accountability and tools that has helped to make Open Source Research process easier and less labour intensive.

Joe has made himself available to all forces should they wish to use his product and has provided continuing support and updates, making this a bespoke product for the police to use. Since this product has become so useful I can't begin to imagine how much time it has saved officers from manually transcribing the process.

Joe has never charged the police a penny for this product despite it being integrated into other commercial software applications.

REFERENCES

Abras, C., Maloney-krichmar, D. and Preece, J. (2004) ‘User-Centered Design’, in *In Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications. Publications.*

Allen, S. (2004) *The Misunderstood Mutex, Ode to Code*. Available at: <https://odetocode.com/blogs/scott/archive/2004/08/20/the-misunderstood-mutex.aspx> (Accessed: 4 August 2018).

Andrew, C., Aldrich, R. J. and Wark, W. K. (2009) *Secret Intelligence: A Reader*. Routledge.

Association of Chief Police Officers (2012) ‘ACPO Good Practice Guide for Digital Evidence’. Metropolitan Police Service. Available at: http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf.

Association of Chief Police Officers (2013) ‘Online Research and Investigation’, 16 September. Available at: <http://library.college.police.uk/docs/appref/online-research-and-investigation-guidance.pdf> (Accessed: 1 December 2016).

Bangor, A., Kortum, P. T. and Miller, J. T. (2008) ‘An Empirical Evaluation of the System Usability Scale’, *International Journal of Human–Computer Interaction*, 24(6), pp. 574–594. doi: 10.1080/10447310802205776.

Bartlett, J. *et al.* (2013) ‘Policing in an Information Age’, *Demos*, pp. 1–42.

Bartlett, J. and Reynolds, L. (2015) *The state of the art 2015: A literature review of social media intelligence capabilities for counter-terrorism*. London: Demos. Available at: http://www.demos.co.uk/wp-content/uploads/2015/09/State_of_the_Arts_2015.pdf (Accessed: 11 January 2016).

BBC News (2016) 'Parents "asked police to take abuser son"', *BBC News*, 2 June. Available at: <https://www.bbc.com/news/uk-36435349> (Accessed: 3 August 2018).

BBC News (2018) 'Dark web paedophile jailed for 32 years', *BBC News*, 19 February. Available at: <https://www.bbc.com/news/uk-england-43114471> (Accessed: 3 August 2018).

Bennett, D. J. and Stephens, P. (2009) 'A cognitive walkthrough of Autopsy Forensic Browser', *Information Management & Computer Security*. Edited by S. M. Furnell, 17(1), pp. 20–29. doi: 10.1108/09685220910944731.

Bevan, N., Carter, J. and Harker, S. (2015) 'ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998?', in *Human-Computer Interaction: Design and Evaluation. International Conference on Human-Computer Interaction*, Springer, Cham, pp. 143–151. doi: 10.1007/978-3-319-20901-2_13.

Birzer, M. L. (2003) 'The theory of andragogy applied to police training', *Policing: An International Journal of Police Strategies & Management*, 26(1), pp. 29–42. doi: 10.1108/13639510310460288.

Birzer, M. L. and Roberson, C. (2007) *Policing Today and Tomorrow*. Pearson/Prentice Hall.

Boehm, B. W. (1987) 'Software Process Management: Lessons Learned from History', in *Proceedings of the 9th International Conference on Software Engineering*. Los Alamitos, CA, USA: IEEE Computer Society Press (ICSE '87), pp. 296–298. Available at: <http://dl.acm.org/citation.cfm?id=41765.41798> (Accessed: 27 February 2017).

Braun, V. and Clarke, V. (2006) 'Using thematic analysis in psychology', *Qualitative Research in Psychology*, 3(2), pp. 77–101. doi: 10.1191/1478088706qp063oa.

Brian Carrier (2015) *The Sleuth Kit*. Available at: <http://www.sleuthkit.org/>.

Burbick, R. (1998) *Software Engineering Methodology: The WaterSluice*. Stanford University. Available at: <http://infolab.stanford.edu/~burback/watersluice/watersluice.pdf> (Accessed: 27 February 2017).

Cabinet Office and Home Office (2012a) *All About Open Source - An Introduction to Open Source Software for Government IT*. 2.0, p. 28. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/78959/All_About_Open_Source_v2_0.pdf.

Cabinet Office and Home Office (2012b) *Open Source Software Options for Government*. 2.0, p. 55. Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/78964/Open_Source_Options_v2_0.pdf.

Capacity Building and Training Directorate (2012) *Transfer Evaluation*. Lyon, France: INTERPOL.

Carr, M. (1997) 'Prototyping and Software Development Approaches'. Available at: https://www.academia.edu/2563255/Prototyping_and_Software_Development_Approaches (Accessed: 26 February 2017).

Carroll, J., Mack, R. and Kellogg, W. (1988) 'Interface Metaphors and User Interface Design', in *Handbook of Human-Computer Interaction*. Elsevier Science Publishers, pp. 67–85.

Chromium (2010) 'A Year of Extensions', *Chromium Blog*, 9 December. Available at: <https://blog.chromium.org/2010/12/year-of-extensions.html> (Accessed: 22 March 2018).

CIA (2010) *INTelligence: Open Source Intelligence — Central Intelligence Agency, INTelligence: Open Source Intelligence*. Available at: <https://www.cia.gov/news-information/featured-story-archive/2010-featured-story-archive/open-source-intelligence.html> (Accessed: 9 August 2018).

Clarke, M. *et al.* (2015) 'A Democratic Licence to Operate: Report of the Independent Surveillance Review'. Available at: https://rusi.org/sites/default/files/20150714_whr_2-15_a_democratic_licence_to_operate.pdf (Accessed: 11 January 2017).

Cohen, G. L. and Sherman, D. K. (2014) 'The psychology of change: self-affirmation and social psychological intervention', *Annual Review of Psychology*, 65, pp. 333–371. doi: 10.1146/annurev-psych-010213-115137.

College of Policing (2017) *Researching, Identifying and Tracing the Electronic Suspect*. Available at: <http://www.college.police.uk/What-we-do/Learning/Professional-Training/digital-and-cyber-crime/Pages/Researching-Identifying-Tracing-Electronic-Suspect.aspx> (Accessed: 1 September 2017).

Crinnion, J. (1992) *Evolutionary Systems Development: A Practical Guide to the Use of Prototyping Within a Structured Systems Methodology*. Perseus Publishing.

Data Protection Act 1998, E. (no date) *Data Protection Act 1998*. Available at: <http://www.legislation.gov.uk/ukpga/1998/29/contents> (Accessed: 19 January 2017).

Davies, C. (2018) "'Sadistic" paedophile Matthew Falder jailed for 32 years', *The Guardian*, 19 February. Available at: <http://www.theguardian.com/technology/2018/feb/19/dark-web-paedophile-matthew-falder-jailed-for-32-years> (Accessed: 3 August 2018).

Denzin, N. K. (1973) *The Research Act: A Theoretical Introduction to Sociological Methods*. Transaction Publishers.

'Digital in 2016' (2016) *We Are Social UK*. Available at: <http://wearesocial.com/uk/special-reports/digital-in-2016> (Accessed: 13 July 2016).

dlemstra (2018) *dlemstra/Magick.NET: The .NET library for ImageMagick*. Available at: <https://github.com/dlemstra/Magick.NET> (Accessed: 5 August 2018).

Dover, R., Goodman, M. S. and Hillebrand, C. (2013) *Routledge Companion to Intelligence Studies*. Routledge.

Duong, L. (2009) 'Applying the "80-20 Rule" with The Standish Group's Statistics on Software Usage | Luu Duong's Blog', 4 March. Available at: <http://www.luuduong.com/archive/2009/03/04/applying-the-quot8020-rulequot-with-the-standish-groups-software-usage.aspx> (Accessed: 13 August 2018).

Edwards, R. (2013) *What is Qualitative Interviewing?* annotated edition edition. London : New Delhi: Bloomsbury Academic.

Evans, M. (2015) 'Police facing rising tide of social media crimes', 5 June. Available at: <http://www.telegraph.co.uk/news/uknews/crime/11653092/Police-facing-rising-tide-of-social-media-crimes.html> (Accessed: 11 January 2016).

Facebook (2015) *Terms of Service, Statement of Rights and Responsibilities*. Available at: <https://www.facebook.com/terms> (Accessed: 19 January 2017).

Facebook (2018) *Community Standards*. Available at: <https://www.facebook.com/communitystandards/misrepresentation> (Accessed: 1 September 2018).

facebook.com (2018) <https://www.facebook.com/terms.php>, Facebook. Available at: <https://www.facebook.com/terms.php> (Accessed: 4 March 2018).

Feldt, H. (2018) *DotNetZip.Semverd: A fork of the DotNetZip project without signing with a solution that compiles cleanly. This project aims to follow semver to avoid versioning conflicts. DotNetZip is a FAST, FR..* Available at: <https://github.com/haf/DotNetZip.Semverd> (Accessed: 3 August 2018).

Felipe, S. K. *et al.* (2010) 'Training Novices on Hierarchical Task Analysis', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. doi: 10.1177/154193121005402321.

Fisher, M. (2014) *Video Capture*. Available at: <http://graphics.stanford.edu/~mdfisher/VideoCapture.html> (Accessed: 11 January 2016).

Floyd, C. (1984) 'A Systematic Look at Prototyping', in Budde, R. *et al.* (eds) *Approaches to Prototyping*. Springer Berlin Heidelberg, pp. 1–18. doi: 10.1007/978-3-642-69796-8_1.

Forensic Science Regulator (2015) *Forensic Science Regulator Newsletter*. Newsletter 26. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/470526/FSR_Newsletter_26__October_2015.pdf.

Forensic Science Regulator (2017) *Codes of Practice and Conduct for forensic science providers and practitioners in the Criminal Justice System*. 4, p. 67. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/651966/100_-_2017_10_09_-_The_Codes_of_Practice_and_Conduct_-_Issue_4_final_web_web_pdf__2_.pdf.

Francis, B. (2016) *DotNetChromeTabs: A control that mimics the functionality of Google Chrome's tab strip*. Available at: <https://github.com/brandonfrancis/DotNetChromeTabs> (Accessed: 3 August 2018).

Genoe, R., Toolan, F. and McGourty, J. (2014) 'Programming for Investigators: From Zero to Hero in 4 Days', in. *Cybercrime Forensics, Education and Training (CFET)*, Canterbury Christ Church University. Available at: https://www.researchgate.net/publication/271511408_Programming_for_Investigators_From_Zero_to_Hero_in_Four_Days.

George, D. and Mallery, P. (2003) *SPSS for Windows Step by Step: A Simple Guide and Reference, 11.0 Update*. Allyn and Bacon.

Gill, P. *et al.* (2008) 'Methods of data collection in qualitative research: interviews and focus groups', *British Dental Journal*, 204(6), pp. 291–295. doi: 10.1038/bdj.2008.192.

Gliem, J. A. and Gliem, R. R. (2003) 'Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales', in. *Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education*, Columbus, OH: The Ohio State University.

GOV.UK Digital Marketplace (no date) *Long Arm - Digital Marketplace, GOV.UK Digital Marketplace*. Available at: <https://www.digitalmarketplace.service.gov.uk/g-cloud/services/243226521515623#content%20and%20http://www.ii-solutions.co.uk/> (Accessed: 6 August 2018).

Grace, M. (2001) 'Continuing professional development: Learning styles', *British dental journal*, 191(3), pp. 125–128.

Gray, D. E. (2009) *Doing Research in the Real World*. 2 edition. Los Angeles: SAGE Publications Ltd.

Greenblatt, M. (2018) *Chromium Embedded Framework, Chromium Embedded Framework - bitbucket*. Available at: <https://bitbucket.org/chromiumembedded/cef> (Accessed: 5 March 2018).

van Griethuijsen, R. A. L. F. *et al.* (2015) 'Global Patterns in Students' Views of Science and Interest in Science', *Research in Science Education*, 45(4), pp. 581–603. doi: 10.1007/s11165-014-9438-6.

Haberfeld, M. R., Clarke, C. A. and Sheehan, D. L. (2011) *Police Organization and Training: Innovations in Research and Practice*. Springer Science & Business Media.

Hallenburg, K., O'Neil, M. and Tong, S. (2016) 'Watching the detectives: researching investigative practice', in *Introduction to Policing Research: Taking Lessons from Practice*. Abingdon, Oxon: Routledge, pp. 101–114.

Hassenzahl, M. and Tractinsky, N. (2006) 'User experience - a research agenda', *Behaviour & Information Technology*, 25(2), pp. 91–97. doi: 10.1080/01449290500330331.

Heale, R. and Forbes, D. (2013) ‘Understanding triangulation in research’, *Evidence-Based Nursing*, 16(4), pp. 98–98. doi: 10.1136/eb-2013-101494.

Hess, K. M., Orthmann, C. H. and Cho, H. L. (2013) *Police Operations: Theory and Practice*. Cengage Learning.

Hibshi, H., Vidas, T. and Cranor, L. F. (2011) ‘Usability of Forensics Tools: A User Study’, in *2011 Sixth International Conference on IT Security Incident Management and IT Forensics (IMF)*. *2011 Sixth International Conference on IT Security Incident Management and IT Forensics (IMF)*, pp. 81–91. doi: 10.1109/IMF.2011.19.

HMIC (2015a) *Online and on the edge: Real risks in a virtual world*. Available at: <https://www.justiceinspectorates.gov.uk/hmicfrs/wp-content/uploads/online-and-on-the-edge.pdf> (Accessed: 3 August 2018).

HMIC (2015b) *Real lives, real crimes: A study of digital crime and policing*. Available at: <https://www.justiceinspectorates.gov.uk/hmicfrs/wp-content/uploads/real-lives-real-crimes-a-study-of-digital-crime-and-policing.pdf>.

Hobbs, C., Moran, M. and Salisbury, D. (2014) *Open Source Intelligence in the Twenty-First Century: New Approaches and Opportunities*. 2014th edn. Palgrave Macmillan.

Holloway, I. and Todres, L. (2003) ‘The Status of Method: Flexibility, Consistency and Coherence’, *Qualitative Research*, 3(3), pp. 345–357. doi: 10.1177/1468794103033004.

Home Office (2014) *Covert human intelligence sources - code of practice*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/384976/Covert_Human_Intelligence_web.pdf.

Home Office (2017a) *Home Secretary gives £20 million boost to tackle online grooming*, GOV.UK. Available at: <https://www.gov.uk/government/news/home-secretary-gives-20-million-boost-to-tackle-online-grooming> (Accessed: 3 August 2018).

Home Office (2017b) *Police Workforce, England and Wales, 31 March 2017*. Available at: https://nls.ldls.org.uk/welcome.html?ark:/81055/vdc_100053138290.0x000001 (Accessed: 1 September 2018).

Housego, J. (2015) *NPCC Guidance on Open Source Investigation/Research*. Available at: https://www.suffolk.police.uk/sites/suffolk/files/003525-16_npcc_guidance_redacted.pdf.

Hulnick, A. and Valcourt, R. (1999) *Fixing the Spy Machine: Preparing American Intelligence for the Twenty-First Century*. Westport, Conn: Praeger Publishers.

IEEE Software (2018) ‘A Cambrian Explosion of DevOps Tools’, *IEEE Software*, March, pp. 14–17.

ImageMagick 7 (2018). ImageMagick Studio LLC. Available at: <https://github.com/ImageMagick/ImageMagick> (Accessed: 5 August 2018).

Information Commissioners Office (2018) *Data Protection Act 2018, Data Protection Act 2018*. Available at: <https://ico.org.uk/for-organisations/data-protection-act-2018/> (Accessed: 9 August 2018).

Internet Usage and 2015 Population in North America (2015). Available at: <http://www.internetworldstats.com/stats14.htm> (Accessed: 13 July 2016).

Internet World Stats (2017) *European Union Internet Users, Population and Facebook Statistics*. Available at: <https://www.internetworldstats.com/stats9.htm> (Accessed: 5 November 2018).

ISO (2005) *ISO/IEC 17025:2005 - General requirements for the competence of testing and calibration laboratories, International Organization for Standardization*. Available at: <https://www.iso.org/standard/39883.html> (Accessed: 9 August 2018).

ISO (2010) *ISO 9241-210:2010 - Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems, International Organization for Standardization*. Available at: <https://www.iso.org/standard/52075.html> (Accessed: 15 August 2018).

Jamshed, S. (2014) 'Qualitative research method-interviewing and observation', *Journal of Basic and Clinical Pharmacy*, 5(4), pp. 87–88. doi: 10.4103/0976-0105.141942.

JohanSt (2009) 'Microsoft Developer', *Problems with Flash-content in the WebBrowser control*, 6 August. Available at: <https://blogs.msdn.microsoft.com/johan/2009/08/06/problems-with-flash-content-in-the-webbrowser-control/> (Accessed: 3 August 2018).

John, B. E. and Marks, S. J. (1997) 'Tracking the effectiveness of usability evaluation methods', *Behaviour & Information Technology*, 16(4–5), pp. 188–202. doi: 10.1080/014492997119789.

kent.gov (2012) *Kent & Medway Information Sharing Agreement*, p. 34. Available at: <https://shareweb.kent.gov.uk/Documents/KELSI/Specialist%20Children%20Services/Integrated%20Processes/Toolkit/16%20%20Kent%20and%20Medway%20ISA.pdf>.

King, N. (2004) 'Using Templates in the Thematic Analysis of Text', in *Essential Guide to Qualitative Methods in Organizational Research*. London: SAGE Publications Ltd, pp. 256–270. doi: 10.4135/9781446280119.

Kirakowski, J. and Corbett, M. (1993) 'SUMI: the Software Usability Measurement Inventory', *British Journal of Educational Technology*, 24(3), pp. 210–212. doi: 10.1111/j.1467-8535.1993.tb00076.x.

Kirkpatrick, D. and Kirkpatrick, J. D. (2006) *Evaluating Training Programs: The Four Levels*. 3rd edn. San Francisco, California: Berrett-Koehler Publishers.

Kirkpatrick, J. D. and Kirkpatrick, W. K. (2016) *Kirkpatrick's Four Levels of Training Evaluation*. Association for Talent Development.

Kirkpatrick Partners, LLC (2010) ‘Kirkpatrick Hybrid Evaluation Tool Template’. Kirkpatrick Partners, LLC. Available at: <http://www.kirkpatrickpartners.com/Portals/0/Resources/Certified%20Only/Kirkpatrick%20Hybrid%20Evaluation%20Tool%20Template.docx> (Accessed: 27 September 2017).

Ladas, C. (2008) *Scrumban: and other essays on Kanban System for Lean Software development*. Saele, WA: Modus Cooperandi Press.

Laplante, P. A. and Neill, C. J. (2004) ‘The Demise of the Waterfall Model Is Imminent’, *Queue*, 1(10), pp. 10–15. doi: 10.1145/971564.971573.

Larman, C. and Basili, V. R. (2003) ‘Iterative and incremental developments. a brief history’, *Computer*, 36(6), pp. 47–56. doi: 10.1109/MC.2003.1204375.

Le Hégarret, P., Wood, L. and Robie, J. (2004) *What is the Document Object Model?* Available at: <https://www.w3.org/TR/DOM-Level-3-Core/introduction.html> (Accessed: 5 August 2018).

Leeuw, E. D. de, Hox, J. and Dillman, D. (2012) *International Handbook of Survey Methodology*. Routledge.

Lethbridge, T. C., Sim, S. E. and Singer, J. (2005) ‘Studying Software Engineers: Data Collection Techniques for Software Field Studies’, *Empirical Software Engineering*, 10(3), pp. 311–341. doi: 10.1007/s10664-005-1290-x.

Lewis, J. R. (1995) ‘IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use’, *Int. J. Hum.-Comput. Interact.*, 7(1), pp. 57–78. doi: 10.1080/10447319509526110.

Lewis, J. R. and Sauro, J. (2009) ‘The Factor Structure of the System Usability Scale’, in *Proceedings of the 1st International Conference on Human Centered Design: Held As Part of HCI International 2009*. Berlin, Heidelberg: Springer-Verlag (HCD 09), pp. 94–103. doi: 10.1007/978-3-642-02806-9_12.

Loewenthal, K. and Lewis, C. A. (2015) *An Introduction to Psychological Tests and Scales*. Psychology Press.

Marcus, A. and Gasperini, J. (2006) ‘Almost Dead on Arrival: A Case Study of Non-user-centered Design for a Police Emergency-response System’, *interactions*, 13(5), pp. 12–18. doi: 10.1145/1151314.1151328.

Mattox, J. (2013) *Why L&D needs Net Promoter Score*, pp. 135–140. Available at: [http://www.cedma-europe.org/newsletter%20articles/Inside%20Learning%20Technologies%20and%20Skills/Why%20L&D%20Needs%20Net%20Promoter%20Score%20\(Oct%2013\).pdf](http://www.cedma-europe.org/newsletter%20articles/Inside%20Learning%20Technologies%20and%20Skills/Why%20L&D%20Needs%20Net%20Promoter%20Score%20(Oct%2013).pdf) (Accessed: 6 August 2018).

MDN web docs (no date) *position*, *MDN Web Docs*. Available at: <https://developer.mozilla.org/en-US/docs/Web/CSS/position> (Accessed: 3 August 2018).

Microsoft (2011) *Windows IE custom download manager (CSIEDownloadManager) sample in C# for Visual Studio 2010*. Available at:

<https://code.msdn.microsoft.com/windowsdesktop/CSIEDownloadManager-8ab5d910> (Accessed: 11 January 2016).

Microsoft (no date) *AppDomain.UnhandledException Event (System)*. Available at: [https://msdn.microsoft.com/en-us/library/system.appdomain.unhandledexception\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.appdomain.unhandledexception(v=vs.110).aspx) (Accessed: 3 August 2018).

Mills, S. (2007) 'Contextualising design: Aspects of using usability context analysis and hierarchical task analysis for software design', *Behaviour & Information Technology*, 26(6), pp. 499–506. doi: 10.1080/01449290600740835.

Milne, J. (no date) 'Centre for CBL in Land Use and Environmental Sciences, Aberdeen University.', p. 1.

Ministry of Defence (2011) *Understanding and Intelligence support to Joint Operations*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/311572/20110830_jdp2_00_ed3_with_change1.pdf.

Moss, R. (2018) *Cyotek.Windows.Forms.ImageBox: The ImageBox is a custom control for displaying images. It supports zooming, scrolling, panning, region selection and much more!* Available at: <https://github.com/cyotek/Cyotek.Windows.Forms.ImageBox> (Accessed: 3 August 2018).

Mourrier, S. and Klawiter, J. (2012) *Html Agility Pack*. Available at: <https://htmlagilitypack.codeplex.com/Wikipage?ProjectName=htmlagilitypack> (Accessed: 11 January 2016).

Mozilla (2011) *How many Firefox users have add-ons installed? 85%!*, *Mozilla Add-ons Blog*. Available at: <https://blog.mozilla.org/addons/2011/06/21/firefox-4-add-on-users/> (Accessed: 22 March 2018).

Munro, T. (2017) *Maximising intelligence: How one force is looking to save thousands of working hours, Policing Insight*. Available at: <https://policinginsight.com/analysis/maximising-intelligence-one-force-looking-save-thousands-working-hours/> (Accessed: 6 August 2018).

Naumann, J. D. and Jenkins, A. M. (1982) 'Prototyping: The New Paradigm for Systems Development', *MIS Quarterly*, 6(3), pp. 29–44. doi: 10.2307/248654.

Nielsen, J. (1993) *Usability Engineering*. New edition edition. Morgan Kaufmann Publishers In.

Norman, D. (2013) *The Design of Everyday Things: Revised and Expanded Edition*. Revised, Expanded edition. New York, New York: Basic Books.

Norman, D. A. and Draper, S. W. (1986) *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Norman, D. and Nielsen, J. (no date) *The Definition of User Experience (UX)*, Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/definition-user-experience/> (Accessed: 15 August 2018).

Nowell, L. S. *et al.* (2017) 'Thematic Analysis: Striving to Meet the Trustworthiness Criteria', *International Journal of Qualitative Methods*, 16(1), p. 160940691773384. doi: 10.1177/1609406917733847.

Nurse, J. R. *et al.* (2011) 'Guidelines for usable cybersecurity: Past and present', in *Cyberspace Safety and Security (CSS), 2011 Third International Workshop on*. IEEE, pp. 21–26. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6058566 (Accessed: 11 January 2016).

Ofcom (2018) *Adults' Media Use and Attitudes Report*, p. 219. Available at: https://www.ofcom.org.uk/__data/assets/pdf_file/0011/113222/Adults-Media-Use-and-Attitudes-Report-2018.pdf.

Office for National Statistics (2018a) *Internet access – households and individuals, Great Britain - Office for National Statistics, Internet access – households and individuals, Great Britain: 2016*. Available at: <https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/bulletins/internetaccesshouseholdsandindividuals/2018> (Accessed: 23 August 2018).

Office for National Statistics (2018b) *Internet users, UK - Office for National Statistics, Internet users, UK: 2018*. Available at: <https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/internetusers/2018> (Accessed: 23 August 2018).

Office of Surveillance Commissioners (2015) *Annual report of the Chief Surveillance Commissioner to the Prime Minister and to the Scottish Ministers for 2014-2015*. Available at: <https://osc.independent.gov.uk/wp-content/uploads/2015/06/OSC-Annual-Report-2014-15-web-accessible-version.pdf> (Accessed: 11 January 2016).

Omand, D., Bartlett, J. and Miller, C. (2012) 'A balance between security and privacy online must be struck', *Magdalen House*, 136. Available at: http://www.demos.co.uk/files/_Intelligence_-_web.pdf?1335197327 (Accessed: 12 January 2016).

Oppenheim, A. (1998) *Questionnaire Design, Interviewing and Attitude Measurement*. New edition. London: Continuum-3PL.

PDFsharp: A .NET library for processing PDF (2018). empira Software. Available at: <https://github.com/empira/PDFsharp> (Accessed: 4 March 2018).

peSHIr (2009) *c# - Handling unhandled exceptions problem, Stack Overflow*. Available at: <https://stackoverflow.com/questions/406385/handling-unhandled-exceptions-problem/406473> (Accessed: 3 August 2018).

Pressman, R. S. and Maxim, B. (2014) *Software Engineering: A Practitioner's Approach*. 8 edition. New York, NY: McGraw-Hill Education.

Qu, S. Q. and Dumay, J. (2011) 'The qualitative research interview', *Qualitative Research in Accounting & Management*, 8(3), pp. 238–264. doi: 10.1108/11766091111162070.

Queen, C. R. (2016) *Effectiveness of problem-based learning strategies within police training academies and correlates with licensing exam outcomes*. Western Michigan University.

Regulation of Investigatory Powers Act (no date).

Reichheld, F. F. (2001) *The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value*. New edition edition. Boston, Mass: Harvard Business School Press.

Robson, C. (2011) *Real World Research*. 3rd edition. Chichester: John Wiley & Sons.

Rose, C. (2014) *Annual report of the Chief Surveillance Commissioner to the Prime Minister and to the Scottish Ministers for 2013-2014*. Available at: <https://www.ipco.org.uk/docs/OSC%20Annual%20Report%202013-14.pdf.pdf>.

Royce, W. (1970) 'Managing the Development of Large Software Systems', in *Technical Papers of Western Electronic Show and Convention*, Los Angeles, USA. Available at: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>.

Runeson, P. *et al.* (2012) *Case Study Research in Software Engineering: Guidelines and Examples*. 1 edition. Hoboken, N.J: Wiley-Blackwell.

Runeson, P. and Höst, M. (2009) 'Guidelines for conducting and reporting case study research in software engineering', *Empirical Software Engineering*, 14(2), p. 131. doi: 10.1007/s10664-008-9102-8.

Sauro, J. (2010) 'MeasuringU: Does Better Usability Increase Customer Loyalty?', *MeasuringU*, 7 January. Available at: <https://measuringu.com/usability-loyalty/> (Accessed: 6 August 2018).

Sauro, J. (2011) 'MeasuringU: What Is A Good Task-Completion Rate?', *What is a good task-completion rate?*, 21 March. Available at: <https://measuringu.com/task-completion/> (Accessed: 30 March 2019).

Sauro, J. (2012) *SUS Calculator Package*. Available at: <https://measuringu.com/product/suspack/>.

Sauro, J. (2013) '10 Things To Know About The System Usability Scale', 18 June. Available at: <http://www.measuringu.com/blog/10-things-SUS.php>.

Sauro, J. (2016) 'MeasuringU: 5 Ways to Use the System Usability Scale (SUS)', *MeasuringU*, 12 July. Available at: <https://measuringu.com/sus-five/> (Accessed: 16 August 2018).

Sauro, J. and Kindlund, E. (2005) 'Using a Single Usability Metric (SUM) to Compare the Usability of Competing Products', in *in Proceeding of the Human Computer*

Interaction International Conference (HCII 2005), Las Vegas, USA Making Sense of Usability Metrics: Usability and Six Sigma: (Sauro & Kindlund) p.10.

Schaurer, F. and Störger, J. (2013) 'The Evolution of Open Source Intelligence (OSINT)', *The Intelligencer - Journal of U.S Intelligence Studies*, 19(3), p. 4.

Scriven, O. and Herdale, G. (2015) *Digital Investigation and Intelligence - Policing capabilities for a digital age*. Available at: <http://www.nppcc.police.uk/documents/reports/Digital%20Investigation%20and%20Intelligence%20Policing%20capabilities%20for%20a%20digital%20age%20April%202015.pdf>.

Silverman, D. and Marvasti, A. (2008) *Doing Qualitative Research: A Comprehensive Guide*. SAGE.

Sommerville, I. (2006) *Software Engineering*: 8 edition. Harlow, England ; New York: Addison Wesley.

SourceForge (2017) *Oryon OSINT Browser*, SourceForge. Available at: <https://sourceforge.net/projects/oryon-osint-browser/> (Accessed: 24 August 2018).

Spencer, R. (2000) 'The Streamlined Cognitive Walkthrough Method, Working Around Social Constraints Encountered in a Software Development Company', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM (CHI '00), pp. 353–359. doi: 10.1145/332040.332456.

statista (2017) *Internet usage in Europe - Statistics & Facts*, www.statista.com. Available at: <https://www.statista.com/topics/3853/internet-usage-in-europe/> (Accessed: 5 November 2018).

Stephens, P. (2012) 'An Evaluation of Linux Cybercrime Forensics Courses for European Law Enforcement', in Clarke, N. and Furnell, S. (eds) *Proceedings of the Sixth International Symposium on Human Aspects of Information Security & Assurance (HAISA 2012)*. Plymouth University, pp. 119–128.

Taber, K. (2013) *Classroom-based Research and Evidence-based Practice: An Introduction*. Second edition. Los Angeles, Calif: Sage Publications Ltd.

Tamkin, P., Yarnall, J. and Kerrin, M. (2002) *Kirkpatrick and Beyond: A review of models of training evaluation*. Brighton, United Kingdom: The Institute for Employment Studies. Available at: <https://pdfs.semanticscholar.org/6845/52ac8528bfaed28fc8337a1a57b94c27aa39.pdf> (Accessed: 10 March 2018).

Tavakol, M. and Dennick, R. (2011) 'Making sense of Cronbach's alpha', *International Journal of Medical Education*, 2, pp. 53–55. doi: 10.5116/ijme.4dfb.8dfd.

Terms of Service - YouTube (2010) *Terms of Service - YouTube*. Available at: <https://www.youtube.com/static?gl=GB&template=terms> (Accessed: 4 March 2018).

The Chromium Projects (2018) *The Chromium Projects*. Available at: <https://www.chromium.org/> (Accessed: 5 March 2018).

‘The JAPAN Test’ (no date). Available at: https://www.kelsi.org.uk/__data/assets/pdf_file/0003/26706/Japan-Test.pdf.

Thompson, B. (1998) *Giving a Voice to Open Source Stakeholders: A Survey of State, Local & Tribal Law Enforcement*. Committee on Homeland Security. Available at: https://fas.org/irp/congress/2008_rpt/dhs-osint.pdf (Accessed: 13 July 2016).

Tofel-Grehl, C. and Feldon, D. F. (2013) ‘Cognitive Task Analysis–Based Training: A Meta-Analysis of Studies’, *Journal of Cognitive Engineering and Decision Making*, 7(3), pp. 293–304. doi: 10.1177/1555343412474821.

Tong, S., Bryant, R. P. and Horvath, M. A. H. (2009) *Understanding Criminal Investigation*. John Wiley & Sons.

Tor (2018) *Tor Expert Bundle*. Available at: <https://www.torproject.org/download/download.html.en>.

Torgashov, P. (2018) *FastColoredTextBox: Fast Colored TextBox for Syntax Highlighting. The text editor component for .NET*. Available at: <https://github.com/PavelTorgashov/FastColoredTextBox> (Accessed: 3 August 2018).

Travis, D. (2011) *ISO 13407 is dead. Long live ISO 9241-210!*, *User Focus*. Available at: <https://www.userfocus.co.uk/articles/iso-13407-is-dead.html> (Accessed: 15 August 2018).

Tullis, T. S. and Stetson, J. N. (2004) ‘A comparison of questionnaires for assessing website usability’, in *Usability Professional Association Conference*, pp. 1–12. Available at: <http://home.comcast.net/~tomtullis/publications/UPA2004TullisStetson.pdf> (Accessed: 12 January 2016).

UK Government Digital Service (2017) *Be open and use open source - GOV.UK, Be open and use open source*. Available at: <https://www.gov.uk/guidance/be-open-and-use-open-source> (Accessed: 16 January 2018).

Vertex42 (2014) *Kanban Board Template, Vertex42.com*. Available at: <https://www.vertex42.com/ExcelTemplates/agile-kanban-board.html> (Accessed: 7 November 2018).

Vijayasarathy, L. R. and Butler, C. W. (2016) ‘Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?’, *IEEE Software*, 33(5), pp. 86–94. doi: 10.1109/MS.2015.26.

Vodde, R. F. (2009) *Andragogical Instruction for Effective Police Training*. Cambria Press.

W3C (2009) *W3C Document Object Model*. Available at: <https://www.w3.org/DOM/> (Accessed: 5 August 2018).

Waring, T. and Maddocks, P. (2005) ‘Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future’, *International*

Journal of Information Management, 25(5), pp. 411–428. doi: 10.1016/j.ijinfomgt.2005.06.002.

Wells, D. and Gibson, H. (2017) ‘OSINT from a UK perspective: considerations from the law enforcement and military domains’, in *Proceedings Estonian Academy of Security Sciences, 16 : From Research to Security Union*. Estonian Academy of Security Sciences, pp. 84–113. Available at: <https://digiriul.sisekaitse.ee/handle/123456789/2001> (Accessed: 3 August 2018).

Wharton, C. *et al.* (1994) ‘Usability Inspection Methods’, in Nielsen, J. and Mack, R. L. (eds). New York, NY, USA: John Wiley & Sons, Inc., pp. 105–140. Available at: <http://dl.acm.org/citation.cfm?id=189200.189214> (Accessed: 19 February 2017).

Wildemuth, B. M. (2009) *Applications of Social Research Methods to Questions in Information and Library Science*. Westport, Conn: Libraries Unlimited.

Williams, L. and Cockburn, A. (2003) ‘Agile software development: it’s about feedback and change’, *Computer*, 36(6), pp. 39–43. doi: 10.1109/MC.2003.1204373.

Zahabi, M. and Kaber, D. (2018) ‘Identification of task demands and usability issues in police use of mobile computing terminals’, *Applied Ergonomics*, 66, pp. 161–171. doi: 10.1016/j.apergo.2017.08.013.

APPENDICES

APPENDIX A –SUS: OSIRT PROTOTYPE.....	262
APPENDIX B – OBSERVATIONS: OSIRT PROTOTYPE.....	274
APPENDIX C – SUS: OSIRT RELEASE	287
APPENDIX D – OBSERVATIONS: OSIRT RELEASE.....	304
APPENDIX E – COGNITIVE WALKTHROUGH	318
APPENDIX F – INTERVIEWS	348
APPENDIX G – OBSERVATION TEMPLATE FOR RITES COURSE (CHAPTER 10)	372
APPENDIX H – SAMPLE OF DAILY QUESTIONNAIRES FOR RITES COURSE.....	375
APPENDIX I – IDownloadManager IMPLEMENTATION	400

APPENDIX A –SUS: OSIRT PROTOTYPE

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				✓
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

excellent product. Needs fine tuning.
 But with use by police and feedback
 I'm confident this will happen in
 a timely manner.

Thanks Joe.
 Mon. cop.

95

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.			✓		
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓			✓	
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

Having not used alternative products, I am unable to compare OSIRT against others. However, I found the principles of OSIRT & intended capabilities to be extremely efficient.

T3 (5)

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.			✓		
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.				✓	
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.			✓		

Please enter any comments you would like to make about OSIRT below:

would just need a backup
technical advise if having
problems. ie telephone no

72.5

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.	1				✓
I found the system unnecessarily complex.		✓			
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

Very good product. 

T5

6

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

Its an easy to use System and it does everything I want it to do. Just keep it updated !!

92.5

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.					
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

future development I would like to see

- ① Choose of Browsers
- ② History View - To allow a quicker return to a webpage.
- ③ Does OSIRT use Private Browsing?
If not consider this or have option to clear cache.

97.5?

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.			ONLY FOR FIXES		
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

PERSONALLY I THINK THIS IS AN EXCELLENT TOOL. ONCE ALL THE LITTLE BUGS ARE FIXED I WOULD FEEL 100% CONFIDENT TO USE THIS TOOL TO CAPTURE EVIDENTIAL PRODUCT. & PRODUCE EVIDENTIAL REPORTS. THIS TOOL IS AN ESSENTIAL TOOL FOR THIS TYPE OF WORK. DO BELIEVE THAT TECH SUPPORT REQUIRED TO FIX BUGS.

✓

85

T8

System Usability Scale (SUS) Questionnaire

level 6.

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.		✓			
I thought the system was easy to use.			✓		
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

A very useful tool that was easy to use.
The bugs that were identified were ironed out very quickly and it is definitely something I would make use of.

82.5

79

SUS Questionnaire

5/6

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.			✓		
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.			✓		
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

Good entry to use system

72.5

T90

SUS Questionnaire

6

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.		✓		✓	
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.		✓			
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

80

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

Very well thought out and user friendly system. Its great that its so adaptable and that the designer is happy to apply any ammendments to make the system the best it can be from the suggestions he has been given,

Good Luck!! :)

APPENDIX B – OBSERVATIONS: OSIRT PROTOTYPE

Participant	Actions	Quotes	Inferences
T1	Attempting to create a case & entering a name in the case ref field	"I don't understand why I'm getting an error here"	Field entry not clear Error message not clear * not clear what should be entered
T2	As above	—	As above
T7 T7	" —"	"What do I have to put here for it to work?"	As above
T5	" —"	"I know I have to be careful here"	" —"
T9	" —"	"Why am I getting an issue here?"	" —"

T1, T2, T5, T11 → Said it'd be better if case reference field had clearer instructions.

Participant	Actions	Quotes	Inferences
T11	As above (core creation issue)	"Give us a hand, here"	not clear (as above)
T14, T3, T1, T2, T7, T5, T6	Discussion about the core container.	"What if the core container was changed outside of OSIRT?" "An officer could accidentally delete this"	
		"Is there any way to encrypt this?"	
T1	Using the main browser, automated logging.	"This is a time Saver"	Automation = good!
T2		"I always used a pen & paper"	" — "

* these participants had a discussion & the participant quotes were taken from that discussion.

2)

3

277



Participant	Actions	Quotes	Inferences
T3	"	"This is great"	"
T4	"	"Very handy indeed!"	"
T5	"	"This is a useful feature, especially as I have to keep a manual log"	"
T7	"	"Automation makes my job much easier"	"
T8	"	"I used to have to manually enter these in Excel"	"

Participant	Actions	Quotes	Inferences
T9	"Automated logging of websites"	"Quick & easy"	"
T11	"	"That is good, I usually use pen & paper"	"
T5 (All but T11)* T1, 2, 3, 4, 5, 6, 7, 8, 9, 10 All but one participant had something negative to say about IE.	Asking about the underlying browser being IE	"Not a fan of IE, can it be Firefox?" "IE is so slow!" "What about viruses?" "We're actually not allowed to use IE as it's not secure" "Can you use Google Chrome?"	Stigma of IE, is perhaps take a look at the feasibility of other browser again.
All participants TODO!	Twitter failed to load - Error that version of IE needs to be updated.	"We really need Twitter to work, otherwise I cannot use OSIRT in actual investigations"	All websites must work. Highlighting the need to have other latest IE installed is crucial.

"Social media is a huge part of OSR, I capture a lot of data from them & that's not easy"

→ "I'm basically based on Twitter"



(+)

(5)

Participant	Actions	Quotes	Inferences
T1 & T2 & T4 T7 & T8	Download videos from websites (e.g. YT & FB)	"If we could download videos, this would be ideal" (T1) "Have you heard of Wondershare? They have the ability to download videos. It'll be great to see that in OS/RT" (T2)	Users would like way to obtain videos for evidence capture.
		T7 - "It would be really handy if we could download videos"	—
T4	Right-click on image in an attempt to save it	"I'm clicking on this image, but no option to save it shows up"	Bug w/ the calculation of the img element. T500: need to check this! *
T5, T2	Right-click on image. Image was wrapped around another element	"I'm pretty sure Firefox would be able to download this" "It should be possible to download this"	Bug with how the underlying document finds id of the wrapped img is an img... is it even possible. CHECK!!

7)

Participant	Actions	Quotes	Inferences
T11	Downloading an Excel spreadsheet from the RITES course website	"This is an unusual way to download things"	The pre-made all manager is unusual, look into making it friendlier.
T9	" ———"	"I have no idea what I'm meant to do here, I'm trying to download this"	" ———"
T3	" ———"	"How do I download, I pushed this button but nothing worked"	" ———"
T6	" ———"	→ observed having an issue but part @ TS helped him.	" ———"
T3	Website caused OSIRT to freeze.	"I was just loading this website & OSIRT froze"	check into source of this. perhaps some JS mem leak?

8

Participant	Actions	Quotes	Inferences
T5	Attempting to attach a file, but an unhandled Exception occurred	"I tried to attach this file, but this popped up."	Attachment logic needs a break. Exception occurred when trying to copy file.
T7	" ———"	"I think something bad happened"	" ———"
T5	The UX for the attachment	"The two pop-ups are a bit much, it would be nicer if this was all as one thing"	Good feedback will implement this.
T1	Case loading (Issues)	"I didn't know what to load"	The directory structure is not easy to understand What to load → look into self code file.
T9	" ———"	"What do I load?"	" ———"

(9)

have
blank
disc

Participant	Actions	Quotes	Inferences
T8	" (Case load)	"I loaded this container" → points to wrong folder	"
T10	"	"No idea what I'm meant to load, Sorry See"	"
A discussion was had about the case container, I suggested a case	file they loaded, like a Zip file, & a majority agreed this would be better → "less confusing"		
T3, T10, T11	Case loading +ve	"I think loading a case is quite simple" "This is easy enough for me, although I can see it confusing some people"	

Participant	Actions	Quotes	Inferences
T1	using video capture - confusion w/ UI.	"There are a lot of options here, do I need to change any of them or can I just record?" "What do I do?"	Bad UI, too complex.
T4	" ——— "		
T5	" ——— "		
T9 (T10, T7 & T8) R agreed.	Save ^{su} video capture 'complete' message	"Is there a chance of getting a 'capture' 'complete' message?"	Confirmation that video saved. Msg.
T5 *T2 & T7	Capturing sound when video capturing	"Does OSIRT capture sound?" "It would be very useful if we could get sound"	Make instructions for enabling stereo mic.
T6			

all but 2 terminals had issues with the default UI for video capture. To Do: need to fix !!!

Participant	Actions	Quotes	Inferences
T3	viewing image in image preview.	"Is there a way to zoom out so we can see the whole image?"	look a PictureBox control to see if scroll to zoom options exist.
T2 T1 T10 T8 T11	viewing image in preview coding (low-down)	"When I wanted an image I clicked log [ENTER] took a really long time to save it"	Large images, or images, may have memory leaking issues.
T4	viewing image in picu	"If it's a large image, I'd like to be able to zoom in on a particular part of it easily"	—
T9	" — "	"Is there a choice of a zoom option"	—
ALL Participants	Screenshot page that makes wx of frames	"Not a big deal, we can use snippet!" "I'd like the ability to capture TB W"	Issue surrounding frames, how do other tools deal with it?

Participant	Actions	Quotes	Inferences
T3 T9	Issue downloading .exe file. Tried to download exe	"I was testing out the downloader but an error appeared when I attempted to d/i cleaner"	Continued issues w/ the custom d/i Manager.
T1 / T4	Continued looking at Facebook, -Slow downs	"OSIRT has become really slow" "I took a screenshot of Facebook & now OSIRT is slow"	Is IE leaking memory? need to check. Slow is bad.
T9	Entering case notes	"It would be great if pressing Enter would add a note rather than adding a newline"	Add option to allow this. may prefer a new line being entered.
T5 T8	Saving as PDF in image preview " _____"	"We like to save screenshots as PDF, can that be added?" "Can we save as jpg"	look at adding more save format options

Participant	Actions	Quotes	Inferences
T1	viewing the audit log	"Could this be made bigger?"	Make use of vertical space for audit log
T2 T5 T7 T8 T9 T10 T4	" " agreed	"you have to scroll quite a bit to see"	
T11	" Sorting Audit log in chronological order	"you have to scroll, is there any chance that a search option is added?" "is there a way to sort these logs out of newest first order?"	Add search option, Add sorting option
T2 T7 others wanted this.	View previously captured files other than video images	"It'd be great if this file thing could be used to preview the videos & other files"	Look at adding previewer.

APPENDIX C – SUS: OSIRT RELEASE

This contains the hand-completed SUS results.

Order is arbitrary.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.				✓	
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.			✓		

Please enter any comments you would like to make about OSIRT below:

From my perspective as a complete novice to this kind of tool I found it quite easy to use + navigate + I know if I was using it on a day to day basis it would be an easy tool to get to grips with. I can definitely see a use for it in my new role.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					
I found the system unnecessarily complex.					
I thought the system was easy to use.					
I think that I would need the support of a technical person to be able to use this system.					
I found the various functions in this system were well integrated.					
I thought there was too much inconsistency in this system.					
I would imagine that most people would learn to use this system very quickly.					
I found the system very cumbersome to use.					
I felt very confident using the system.					
I needed to learn a lot of things before I could get going with this system.					

If things went wrong only.

Please enter any comments you would like to make about OSIRT below:

Good system - Like that the components are all on one, as opposed to using multiple tools.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

Very well put together. Thank you it will be a great help in the future.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.		✓			
I thought the system was easy to use.					
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.		✓			
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

Very good system which I will be looking to introduce to the work place.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

This system works perfectly in terms of what I am looking to achieve.

The log aspect is particularly pleasing as it allows me to build an audit trail without having to worry about a disclosure log. This is a key feature in saving time as well as preventing duplication.

The system is very user friendly and incorporates the set functions that I need in order to do my job and comply with the laws and guidance on open research

TI

SUS Questionnaire

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

Excellent product with enables investigation to take place in sig single location. which is auditable.
 Saves time on making extensive notes, and allows for rational to be added as you go.
 Some tweaks needed over course, but quickly resolved.
 long term recommendation, but would there be the same support in place in coming years??

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					/
I found the system unnecessarily complex.	/				
I thought the system was easy to use.					/
I think that I would need the support of a technical person to be able to use this system.	/				
I found the various functions in this system were well integrated.					/
I thought there was too much inconsistency in this system.	/				
I would imagine that most people would learn to use this system very quickly.					/
I found the system very cumbersome to use.	/				
I felt very confident using the system.				/	
I needed to learn a lot of things before I could get going with this system.	/				

Please enter any comments you would like to make about OSIRT below:

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				X	
I found the system unnecessarily complex.	X				
I thought the system was easy to use.					X
I think that I would need the support of a technical person to be able to use this system.		X			
I found the various functions in this system were well integrated.				X	
I thought there was too much inconsistency in this system.		X			
I would imagine that most people would learn to use this system very quickly.				X	
I found the system very cumbersome to use.		X			
I felt very confident using the system.			X		
I needed to learn a lot of things before I could get going with this system.		X			

Please enter any comments you would like to make about OSIRT below:

I would like to know more about how it might fit into my specific line of work/role (Victim identification in child exploitation online).

I would like to know how my reports/logs are 'set out' if I extract them for referral or dissemination to other agencies/officers.

TS

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.			✓		
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.		✓			
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

stability issues when working on multiple tabs.

Speed issues - bit slow on occasions.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.				✓	
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

*tech support + advice is always a necessity no matter how apparently simple the system is (e-mail/telephone helpline would suffice).

It is rare that a system used by the police is so straight forward + "idiot proof." I was particularly impressed by the various live-time upgrades - no need for costly + time consuming rebuilds + upgrades as we get from commercial providers.



System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.					✓
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				✓
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.					✓
I needed to learn a lot of things before I could get going with this system.			✓		

Please enter any comments you would like to make about OSIRT below:

I'm not particularly technical and once shown I found OSIRT to be simple to use and effective.

I particularly like the built in extract of who is, where the IP and the easy extraction of videos.

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓			4 ✓	
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.			✓		
I found the various functions in this system were well integrated.			✓		
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.		✓			
I felt very confident using the system.			✓		
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.	✓				
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.	✓				
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

Could have other uses, not just open source research.

SUS Questionnaire

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.	✓			✓	
I found the system unnecessarily complex.	✓				✓
I thought the system was easy to use.	✓				✓
I think that I would need the support of a technical person to be able to use this system.	✓			✓	
I found the various functions in this system were well integrated.	✓			✓	
I thought there was too much inconsistency in this system.	✓				✓
I would imagine that most people would learn to use this system very quickly.	✓				✓
I found the system very cumbersome to use.	✓				✓
I felt very confident using the system.	✓				✓
I needed to learn a lot of things before I could get going with this system.	✓				

Please enter any comments you would like to make about OSIRT below:

FIND FUNCTION FROM STANDARD BROWSER (CTRL + F) WOULD BE BENEFICIAL.

SUS Questionnaire

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.				✓	
I found the system unnecessarily complex.		✓			
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.				✓	
I thought there was too much inconsistency in this system.		✓			
I would imagine that most people would learn to use this system very quickly.				✓	
I found the system very cumbersome to use.	✓				
I felt very confident using the system.			✓		
I needed to learn a lot of things before I could get going with this system.		✓			

Please enter any comments you would like to make about OSIRT below:

SUS Questionnaire

System Usability Scale (SUS) Questionnaire

	1 Strongly Disagree	2	3	4	5 Strongly agree
I think that I would like to use this system frequently.					✓
I found the system unnecessarily complex.		✓			
I thought the system was easy to use.				✓	
I think that I would need the support of a technical person to be able to use this system.		✓			
I found the various functions in this system were well integrated.					✓
I thought there was too much inconsistency in this system.	✓				
I would imagine that most people would learn to use this system very quickly.					✓
I found the system very cumbersome to use.	✓				
I felt very confident using the system.				✓	
I needed to learn a lot of things before I could get going with this system.				✓	

Please enter any comments you would like to make about OSIRT below:

OSIRT has to be used ~~to~~ ^{that you} become more proficient.
An invaluable tool in the intelligence world.

APPENDIX D – OBSERVATIONS: OSIRT RELEASE

Course week Commencing:

*Video capture MUCH improved.

Monday.

Actions	Quotes	Inferences
wanting to save EXIF data (demoing OSIRT)	"How do I save EXIF data?"	No ability to save Exif data → Easy to add
User Trying to reload reload a case when OSIRT was already open w/ case	"I've lost my case"	OSIRT can have multiple instances → need way to remind user OSIRT already open & loaded
Issue with the SpellBox Control when entering a carriage return	"I'd like if I push enter for a new line to be entered in the note box"	Tough call, as some users may want it to enter note in to case notes. perhaps an option?
User attempt to go back (navigate) when not possible to do so	"I can't go back"	disable back+forward navigation buttons if can't go back+find.



this occurs several times during week

Introduction to OSIRT, not much done with OSIRT today beyond a 'guided tour'. Massive improvement from prototype.

Lots of positive comments about OSIRT → Lots saying how useful it will be for them.

Course W/C

Tuesday

Actions	Quotes	Inferences
Attempting to export a report as PDF.	"my report did not get created"	user had placed an invalid char into the file name field, & when the report export attempted to copy, it failed. There are checks for invalid file name, but further test required.
		
Closing the last remaining tab caused a fatal exception.	"I broke OSIRT"	Bug: if only one tab open then don't have the option to close it.
Navigating the HTML report, unable to go back Home for Certain Sections.	"Why doesn't this work?" (user clicks 'Home' button in report)	Bug: Check how HTML report navigation is generated.

OSIRT used very heavily today, very good stress test. There were a few minor bugs & usability issues discovered today that need to be addressed.



HTML reports could do with a TWEAK!

PTO →

W/C



Wednesday

Actions	Quotes	Inferences
Wrote Attempts to save image in image previewer but receives a file already ^{quote} message.	"I've had this error a couple of times now"	Ukr is receiving file already but message for the a freshly saved image??? <u>CHECK!!</u>
Cohort using Tor browser. Several participants ask if its possible to do this in OSIRT.		Ukr would also like the ability to conduct OSR in Tor → need to check if possible. Dark web is big feature also requested in photo interviews. Know how big of a deal TOR is to some researchers.
Report exported, officer asked if Case notes are integrated into report		<u>Placing case notes chronologically into the report is a good suggestion</u> → ACPO + disclosure
	"It'd be nice if there was an option to place case notes in order in the report"	

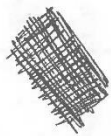
Not much OSIRT today apart from a 2+ hour session on social media. OSIRT much more stable on social media than prob. TOR? → possible? Perhaps...!

WIC

Thursday

Actions	Quotes	Inferences
OSIRT opened a couple dozen tabs & OSIRT crashed →	"OSIRT just crashed"	unsure... update: not properly disposing of tabs → fixed.
(Asked several times this week, Add a preview of image in Audit log)		can do this & will add it for images.
user attempts to find something on page	"Is there a find feature, a Ctrl+F"	Ability to find is important not automatically added in Coresharp unlike IE BL. feature added.
ex. user trying to find file they downloaded.	"how do I find that thing I just downloaded?"	user has to sift through Audit log → better to allow them to save where they want & make a copy.

Big OSIRT day, → well stress tested, only couple fixable bugs found.
End of day 1 hour + O.S investigation with OSIRT.



Course date week commencing:

Day:

Monday

Actions	Quotes	Inferences
user tried to uncheck items in audit log that were search for for report generation.	"It would be handy if we could remove parts we search & uncheck"	Agreed, would make sense to be able to uncheck searched items.
user asked if screenshots could be 'stamped' with URL	_____	Good suggestion & a useful for showing in court? Will see if ImageMagick provides the functionality.

Observer comments:

OSIRT worked very well during the interactive demo.

Cohort responded really well to it.


2 participants already use OSIRT!

More over →

Course date week commencing



Day: Tuesday


Actions	Quotes	Inferences
user to tried to search in address bar		This is a common feature in the browser, add this.

Observer comments:

OSIRT used a lot, but no issues as far as I could tell.
A couple of the cohort are quite vocal & joking about their computer literacy, but seem to get on w/ OSIRT.

Course date week commencing:

Day: Wednesday

Actions	Quotes	Inferences
Exporting report & trying to find how to export between dates.	"Is there a way to export between particular dates?"	This has been previously requested via e-mail... Been meaning to add this feature
TOR loading very slowly for some users. Causing frustration for some users.	Example: "It's not loading" Several other similar quotes.	TOR is quite slow, anyway, but it does appear slower than expected for these users.
Content menu items are a bit oddly ordered. Some users unsure.		Re-order content menu items to make a bit more sense. This has been observed several times this week.

Observer comments:

TOR very slow for some users, need to see what's causing that. Users getting very proficient with OSIRT & can find tools/features they need. Others are helping each other with it if not.

Course date week commencing: [REDACTED] Day: Thursday

Actions	Quotes	Inferences
User asked if report could be exported as XML.	"XML is very handy"	User is an analyst & says XML is very handy. Shouldn't be a problem.

Observer comments: End of day open source investigation went very well.
 Russell says OSIRT "has transformed" the course!

Course date week commencing

Day: FRIDAY

Actions	Quotes	Inferences

Observer comments:

End of week exam, OSIRT worked well & all Cohort successfully completed the test. Several of Cohort said they will definitely be taking OSIRT back with them.

Course date week commencing

Day: Monday

Actions	Quotes	Inferences
Error loading bookmark Manager for several users.	"my bookmarks Manager doesn't open"	Not sure... need to take a closer look. It didn't hook it up to all tabs...!!
unchecking individual items in audit log.	none; observed.	Need to add this before, implement an unchecked recheck all option.

Observer comments:

OSIAT has been through several courses since last observation, it's working "really well" according to Russell & has had a "huge impact" on the course. Russell mentioned they "rarely" have issues, & those are down to "user error".

Course date week commencing [redacted] Day: Tuesday

Actions	Quotes	Inferences

Observer comments: Good day for OSIRT, working well with NOTHING to report!

Course date week commencing [REDACTED] Day: Wednesday

Actions	Quotes	Inferences
Google v. slow loading in Tor & filled with so many captchas.	Russell: "Can we make the default search engine in Tor to be duckduckgo"	Google clearly hates Tor, so DDG is a good suggestion. DDG works <u>MUCH</u> better.

Observer comments: TOR working well, but a change to the default search engine is much needed as Google clearly does not appreciate TOR!

Course date week commencing

Day: Thursday

Actions	Quotes	Inferences
observation, not so much OSIRT-related. The cohort are shown HTTPack, which is a web-scraper → check if		ToDo: check if a way to save responses.
Cafsharp offers an interface or something to capture responses.		

Observer comments: OSI went well. Pretty quiet OSIRT week...

APPENDIX E – COGNITIVE WALKTHROUGH

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams



in bottom
Right

Action: 1st - create new can

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) Y
- (E) Y → ? About limits.
- (F) Y

(G) ? Default? if mouse doesn't
not really understood
(H) - once browse window opened
→ acceptable
→ may not understand as file locn

Q2: Is the action visible?

- (A) YES
- (B) YES - cursor on TEF
- (C) Y
- (D) Y
- (E) Y
- (F) Y

(G) Y
(H)

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y
- (D) Y
- (E) Y ? on limits
- (F) Y

(G) → Locn maybe seen as physical locn of
subject if search.
(H) Once display opened obvious. (Browse button pos)

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Status@ base ? might be easier @ top - most typical button
- (B) Y
- (C) Y
- (D) Y
- (E) None at the time for incorrect entry → ? on tab-off (hold exit) check validity.
- (F) (Y)
- (G) Confused as TEF is read only - have for use 'Browse' button. ? Think about linked words!!

once display opened (Y) - but on 'OK' file name path
'longer' than "Desktop" - may challenge immediate understanding
but "Desktop" is last part of path.

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: (1) (pt 2) (H) on words

Q1: Will the user realistically be trying to perform this action?

(H) N [?] pop up not working.
→ do not understand # fms.

(I) (Y)

(J) Y

(K) Y

Q2: Is the action visible?

(H) Y

(I) Y

(J) Y

(K) ~~Not immediately → busy filter, select first error.~~ Y

Q3: Will the user recognise the action as the correct one?

(H) (N) - Default to - SHA512 label ? [] change hash

(I) Y

(J) Y

(K) N - Not immediately → busy filter, select 1st error

Q4: Will the user understand the feedback (is it appropriate)?

(H) (N)

(I) Y

(J) Y

(K) ~~Extend~~ Can let pop up to "state cannot include spaces"
or ? TEF only allow select class.

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 2 - take screenshot

Q1: Will the user realistically be trying to perform this action?

- (A) y (iv) (y)
 (B) y
 (C) y
 (D) (i) y
 (ii) y
 (iii) y

Q2: Is the action visible?

- (A) y (iv) (y)
 (B) y
 (C) y
 (D) (i) y
 (ii) y
 (iii) y

Q3: Will the user recognise the action as the correct one?

- (A) y (iv) y
 (B) y
 (C) y
 (D) (i) y
 (ii) y
 (iii) y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) (y)
 (B) y
 (C) y - Many would initial "capturing" pop up/status ?
 (D) (i) y (only) (is Gif still even known?)
 (ii) y - on basis that they will leave alone.
 (iii) y
 (iv) y

(?) [Cancel] [Log] is [Log] [Cancel]
 in MS GUI. still

Cognitive Walkthrough – 4 Questions


David Bennett and Joseph Williams

Action: 3 - Current view screenshot

Q1: Will the user realistically be trying to perform this action?

(B) (N)
(C) (4)
(D) (4)

Q2: Is the action visible?

(B) (N) - hidden on drop down next to  camera icon
not easy to understand drop down
(C) (4)
(D) 4

Q3: Will the user recognise the action as the correct one?

(B) (4) - words OK.
(C) (4)
(D) 4

Q4: Will the user understand the feedback (is it appropriate)?

(B) 4 - All good
(C) (4)
(D) 4

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 4 - Current View Timed Screenshot.

Q1: Will the user realistically be trying to perform this action?

(B) Y

(C) Y

(D) Y - ? Default to 5 seconds?

(E) Y

(F) Y

(G) Y

Q2: Is the action visible?

(B) Y

(C) Y

(D) Y

(E) Y

(F) Y

(G) Y

Q3: Will the user recognise the action as the correct one?

(B) Y

(C) Y

(D) Y

(E) Y

(F) Y

(F) Y - just told what looks...

(G) Y

Q4: Will the user understand the feedback (is it appropriate)?

(B) Y

(C) Y

(D) Y

(E) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

(F) Y

- ? text could be larger / more spread out.
? [Cancel] [OK] on Dlg.

(G) Y

- consider in status not clearly visible? important or not?

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: S - Taking a snippet

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) Y
- (E) Y

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y
- (D) N – once started yes. not initially, but hand to do appropriately
→ cursor change
- (E) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Ambiguous → ? ... think about alternatives that / user
- (D) Y.
- (E) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y
- (C) Screen Grep. – not indication to click → drag to select.
? → cursor shape change is
→ a handled box could be used but would
be much slower.

- (D) Y.
- (E) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 56 - Extracting Links

Q1: Will the user realistically be trying to perform this action?

- (A) ☐ (F) 4 ~ ? Auto close?
- (B) ☐ 4
- (C) ☐ 4
- (D) ☐ 4
- (E) ☐ 4

Q2: Is the action visible?

- (A) ☐ (F) 4
- (B) ☐ N - But pop is a "normal action"
- (C) ☐ 4
- (D) ☐ 4
- (E) ☐ 4

Q3: Will the user recognise the action as the correct one?

- (A) ☐ (F) 4
- (B) ☐ 4
- (C) ☐ 4
- (D) ☐ → Q: Len and used len rather than [Log] at base of window.
- (E) ☐ Language mismatch
Save is log

Q4: Will the user understand the feedback (is it appropriate)?

- (A) ☐ (F) 4
- (B) ☐ 4
- (C) ☐ 4
- (D) ☐ 4
- (E) ☐ 4

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 7 - Saving image from webpage

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) (i) Y
(ii) Y

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y
- (D) (i) Y
(ii) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y
- (D) (i) Y
(ii) Y

Q4: Will the user understand the feedback (if it appropriate)?

- (A) Y
- (B) Y
- (C) Y
- (D) (i) Y
(ii) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 8- Reverse Image Search

Q1: Will the user realistically be trying to perform this action?

- A
- (B) 4
- (C) 4
- (D) As per user

Q2: Is the action visible?

- (B) N - As per discussion - add option
- (C) 4
- D

Q3: Will the user recognise the action as the correct one?

- (B) 4
- (C) 4
- (D)

Q4: Will the user understand the feedback (is it appropriate)?

- (B) 4
- (C) 4
- (D)

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 9- Visiting a URL

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y

Q2: Is the action visible?

- (A) Y
- (B) N - but still
- (C) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y
- (C) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 10 - viewing page source code.

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

? Does this need to be more like the course previewer

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 11 - Video Screen Capture

Q1: Will the user realistically be trying to perform this action?

- (A) y
- (B) y
- (C) y
- (D) y

(E) y

Q2: Is the action visible?

- (A) y
- (B) y
- (C) y
- (D) y

(E) y



Q3: Will the user recognise the action as the correct one?

- (A) y
- (B) Icon - play rather than record
- (C) Yes...
- (D) y

red + green

(E) y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) y

- (B) button closer to red dot - suitable? better? -> could it be brighter.

- (C) Yes.

- (D) y

(E) y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 12 - Video capture using marker window

Q1: Will the user realistically be trying to perform this action?

- (A) -
- (B) Y ~ probably
- (C) Y
- (D) Y
- (E) Possibly - may be thinking its started -- probably not then
- (F) Y
- (G) Y

Q2: Is the action visible?

- (A)
- (B) Y
- (C) Y
- (D) As far as any window 'window handles' are
- (E) Y
- (F) U
- (G) Y

Q3: Will the user recognise the action as the correct one?

- (A)
- (B) hopefully given use of camera
- (C) Y - video snippet? | m | do look at icon
- (D) Maybe
- (E) Maybe not
- (F) Y
- (G) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A)
- (B) Yes
- (C) Yes - "Marker window for video recording"?
- (D) Yes
- (E) Yes -> see T/H
- (F) U
- (G) Y



Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: B- Add a case note

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y
~~(F)~~

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y
~~(F)~~

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

(E) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 14 - Export case notes

Q1: Will the user realistically be trying to perform this action?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

- (E) Y
- (F) Y

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

- (E) Y
- (F) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

- (E) Y
- (F) Y

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y
- (C) Y
- (D) Y

- (E) Y
- (F) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 15 - view IP addresses associated w/domain & save

Q1: Will the user realistically be trying to perform this action?

- | | |
|-------|-------|
| (A) Y | (C) Y |
| (B) Y | (E) Y |
| (C) Y | (G) Y |
| (D) Y | |

Q2: Is the action visible?

- | | |
|-------|-------|
| (A) Y | (E) Y |
| (B) Y | (F) Y |
| (C) Y | (G) Y |
| (D) Y | |

Q3: Will the user recognise the action as the correct one?

- | | |
|-------------------|-------|
| (A) Y | (E) Y |
| (B) Y - tentative | (F) Y |
| (C) Y | (G) Y |
| (D) Y | |

Q4: Will the user understand the feedback (is it appropriate)?

- | | |
|-------|-------|
| (A) Y | (E) Y |
| (B) Y | (F) Y |
| (C) Y | (G) Y |
| (D) Y | |

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 16 - Attaching item to case.

Q1: Will the user realistically be trying to perform this action?

- | | |
|-------|-------|
| (A) Y | (E) Y |
| (B) Y | (F) Y |
| (C) Y | (G) Y |
| (D) Y | |

Q2: Is the action visible?

- | | |
|-------|-------|
| (A) Y | (E) Y |
| (B) Y | (F) Y |
| (C) Y | (G) Y |
| (D) Y | |

Q3: Will the user recognise the action as the correct one?

- | | |
|---|-------|
| (A) Y | |
| (B) - Maybe need a label for file icon over to assist | |
| (C) Y | (F) Y |
| (D) it - but could be focussed | (G) Y |
| (E) Y | |

Q4: Will the user understand the feedback (is it appropriate)?

- | | |
|-------|-------|
| (A) Y | (F) Y |
| (B) Y | (G) Y |
| (C) Y | |
| (D) Y | |
| (E) Y | |

Q: Multi-use of file icons in feedback for upload

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 17 - Viewing screenshot previously taken

- Q1: Will the user realistically be trying to perform this action?
- (A) Y ~ Do understand record keeping
 - (B) Y
 - (C) Y (F) Y
 - (D) Y
 - (E) Y

Q2: Is the action visible?

- (A) Y
- (B) Y
- (C) Y (F) Y
- (D) Y
- (E) Y

Q3: Will the user recognise the action as the correct one?

- (A) Y
- (B) N no images looking for images
- (C) ~ sort of Y (F) Y
- (D) Y
- (E) Possibly not - may double click on arrow to open it up.

Q4: Will the user understand the feedback (is it appropriate)?

- (A) Y
- (B) Y (F) Y
- (C) Y
- (D) Y
- (E) Y - Q : not a copy

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 18- Searching the audit log -

Q1: Will the user realistically be trying to perform this action?

- (A) 4
- (B) 4
- (C) 4
- (D) 4
- (E) 4

Q2: Is the action visible?

- (A) 4
- (B) 4
- (C) 4
- (D) 4
- (E) 4

Q3: Will the user recognise the action as the correct one?

- (A) 9
- (B) Maybe - could improve visibility of what it is ... quite far from Search
- (C) Maybe -> perhaps default to "all" or "complete"
- (D) 4
- (E) N -> Not clear that Go means go back -> hard to locate

Q4: Will the user understand the feedback (is it appropriate)?

- (A) 4
- (B) 4
- (C) 4.
- (D) 4
- (E) ✓ 4.

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 19- Export report as PDF

Q1: Will the user realistically be trying to perform this action?

- | | |
|------------------|-------|
| (A) Y | (F) Y |
| (B) Y | (G) Y |
| (C) Y | |
| (D) N | (H) Y |
| (E) Y | |

Q2: Is the action visible?

- | | |
|-------|-------|
| (A) Y | (F) Y |
| (B) Y | (G) Y |
| (C) Y | (H) Y |
| (D) Y | |
| (E) Y | |

Q3: Will the user recognise the action as the correct one?

- | | |
|---------------------|---|
| (A) Y |  (F) Y |
| (B) N → better icon | (G) Y |
| (C) Y | (H) Y |
| (D) Y | |
| (E) Y | |

Q4: Will the user understand the feedback (is it appropriate)?

- | | |
|-------|-------|
| (A) Y | (F) Y |
| (B) Y | (G) Y |
| (C) Y | (H) Y |
| (D) Y | |
| (E) Y | |

report format weak, hard to improve easily

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

→ A - C 27 H - M see (19)
100 100

Action: 20 - Export report w/options as PDF

Q1: Will the user realistically be trying to perform this action?

A) ~~D~~ Y

E) Y

F) Y

G) Y

Q2: Is the action visible?

D) Y - Language

E) Y

F) Y

G) Y

Q3: Will the user recognise the action as the correct one?

D) Y - Lang mismatch (loaded v websites loaded)
(webpage actions v website actions)

E) Y - ↑
(vids & attach op. order)

F) Yes, but focus is still on 'report selection' area. Perhaps.

G) Yes, but → print checkbox in audit log not immediately obvious to its functionality.

Q4: Will the user understand the feedback (is it appropriate)?

D) Y

E) Y

F) Y

G) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 21 - Shutting down case.

Q1: Will the user realistically be trying to perform this action?

- A) Y
- B) Y
- C) Y

Q2: Is the action visible?

- A) Y
- B) Y
- C) Y

Q3: Will the user recognise the action as the correct one?

- A) Y
- B) Y
- C) Y

Q4: Will the user understand the feedback (is it appropriate)?

- A) Y
- B) Y
- C) Y

Cognitive Walkthrough – 4 Questions

David Bennett and Joseph Williams

Action: 22 - open existing case

Q1: Will the user realistically be trying to perform this action?

A) Y E) Y
B) Y
C) Y
D) Y

Q2: Is the action visible?

A) Y D) Y
B) Y E) Y
C) Y

Q3: Will the user recognise the action as the correct one?

A) Y E) Y
B) Y
C) Y
D) Y

Q4: Will the user understand the feedback (if it appropriate)?

A) Y E) Y
B) Y
C) Y
D) Y

Cognitive Walkthrough – OSIRT

David Bennett and Joseph Williams

Details

Walkthrough conducted by: David Bennett and Joseph Williams

Scope of walkthrough: To test creating an OSIRT case, using common functions such as screen capturing and report generation, then exporting an OSIRT report once complete. The case is then re-archived after shutting OSIRT down. Our persona will be a Police Officer who has experience in using computers and a web browser.

General Scenario

A Police Constable has been tasked with conducting Open Source Research online to gather intelligence on [REDACTED] a [REDACTED] OSIRT (Open Source Internet Research Tool) is to be used in order to conduct this investigation.

Walkthrough

1. Creating a new case - 1A

- A o Click the "Create New Case" button
- B o Enter Investigating Officer details as 'DCJ Williams'
- C o Enter Investigating Agency details as 'CCCU'
- D o Enter Operation Name details as 'Op CCCU CW'
- E o Enter Case Reference details as 'cog-walkthrough' *← change to*
- F o Enter Evidence Reference details 'djb-jjw-999'
- G o Case Location: Click the Browse button and select Desktop
- H o Select the SHA512 hash function *→ talking point 1 why do we even have this?*
- I o Enter "Open source investigation into [REDACTED] in Notes section
- J o Click Next button
- K o *Correct error on Case Ref to cog-walkthrough*

2. Taking a Screenshot and saving as png¹

- A o Enter [REDACTED] into the Google search bar
- B o Click the search result [REDACTED]
[REDACTED] a [REDACTED] web address to open [REDACTED] staff profile
- C o Press the Capture Screenshot button
- D o When the Image Previewer opens
 - (i) Leave the File Name as the suggested name
 - (ii) Select .png from the file type menu
 - (iii) Enter [REDACTED] staff profile png" in the Notes section
 - (iv) Click the Log button

3. Taking a Current View Screenshot and saving as png

- (A) o On the same webpage...

¹ These steps can also be used to save pdfs and jpgs.

- B ○ Click the arrow next to the Capture screenshot button to open more screenshot options
- C ○ Click the Current View Screenshot button
- D ○ When the Image Previewer opens
 - (i) ■ Leave the File Name as the suggested name
 - (ii) ■ Select .png from the file type menu
 - (iii) ■ Enter "██████████ current view png" in the Notes section
 - (iv) ■ Click the Log button

- 4 • Taking a Current View Time Screenshot and saving as png
- A ○ On the same webpage...
 - B ○ Click the arrow next to the Capture screenshot button to open more screenshot options
 - C ○ Click the Current View Timed Screenshot button
 - D ○ Enter 5 seconds
 - E ○ Click the OK button *more cursor to "about us" to get tooltips + want*
 - F ○ When the Image Previewer opens
 - (i) ■ Leave the File Name as the suggested name
 - (ii) ■ Select .png from the file type menu
 - (iii) ■ Enter "██████████ staff profile current view png" in the Notes section
 - (iv) ■ Click the Log button

- (5) • Taking a Snippet and saving a png
- A ○ Click the Go Back button in the menu bar
 - B ○ Click the arrow next to the Capture screenshot button to open more screenshot options
 - C ○ Click Snippet
 - D ○ Click and drag a rectangular area around the "██████████" Google search result
 - E ○ When the Image Previewer opens
 - (i) ■ Leave the File Name as the suggested name
 - (ii) ■ Select .png from the file type menu
 - (iii) ■ Enter "Snippet google search png" in the Notes section
 - (iv) ■ Click the Log button

- (6) • Extracting and saving links on webpage
- A ○ On the Google search page...
 - B ○ Right click anywhere on the page
 - C ○ Click 'Extract all links on page' from the context menu
 - D ○ Save the links by clicking the save button
 - E ○ Click OK on the 'Save Successful' MessageBox
 - F ○ Close the source code viewer window

- 7 • Saving image from webpage - linkedin
- A ○ Select ██████████ LinkedIn profile from the Google search result
 - B ○ Right click on the profile image on the LinkedIn website
 - C ○ Click 'Save image' from the context menu

- D ○ When the Image Previewer opens
 - (i) ■ Enter "Save image" in the Notes section
 - (ii) ■ Click the Log button

- 8 • **Reverse Image Searching using TinEye**
- A ○ On [redacted] LinkedIn profile
 - B ○ Right click on the Ministry of Justice logo, (where [redacted])
 - C ○ Click 'Reverse image search using TinEye'
 - D ○ Take full page screenshot of TinEye results that opened
 - (i) ■ See previous steps for saving screenshot.
 - (ii) ■

- (9) • **Visiting a URL**
- (A) ○ Enter [redacted] wordpress.com into the address bar
 - (B) ○ Press the Enter key on the keyboard
 - (C) ○ Wait for page to load

- 10 • **Viewing and Saving page source code**
- (A) ○ Right click on the webpage loaded in 'Visiting a URL'
 - (B) ○ Click 'View page source' button
 - (C) ○ Click the Save button in the window that displayed
 - (D) ○ Click OK on the Save Successful MessageBox
 - (E) ○ Close the source code viewer window

- 11 • **Video Screen Capture**
- A ○ Visit [redacted]
 - B ○ Click the 'Start video capture' button
 - C ○ Scroll down the page for 5 seconds
 - D ○ Click the 'Stop video capture' button
 - E ○ When the Video Previewer opens
 - (i) ■ Enter [redacted] Twitter feed video' as a Note
 - (ii) ■ Click the Log button

- 12 • **Video Screen Capture using Marker Window**
- A ○ Stay on [redacted] Twitter feed
 - B ○ Click the arrow 'More video options' next to the 'Start video capture' button
 - C ○ Click the Marker Window button
 - D ○ Resize the Marker Window so it's the width of the Tweets (i.e. the Marker Window should only contain Tweets within it and nothing else)
 - E ○ Click 'Start video capture'
 - F ○ Scroll the webpage for 5 seconds
 - G ○ When the Video Previewer opens
 - (i) ■ Enter [redacted] Twitter feed video using marker window' as a Note
 - (ii) ■ Click the Log button

- 13 • **Add a Case Note**
- (A) ○ Click the 'Add note to case' button

- (B) ☐ Enter 'Started looking at Tweets for evidence' in the note section
- (C) ☐ Click 'Add Note' button
- (D) ☐ Enter 'Finished looking at Tweets for evidence' in the note section
- (E) ☐ Click 'Add Note' button

14 • Exporting Case Notes as PDF

- A ☐ Click the 'Export as PDF' button
- B ☐ Enter 'example-case-notes' as File name in the Save As dialog
- C ☐ Click Save
- D ☐ Inspect case notes as PDF
- E ☐ Close the PDF reader window
- F ☐ Close the Case Notes window

15 • View IP Addresses associated with domain and save

- A ☐ Visit <http://canterbury.ac.uk>
- B ☐ Click the Tools option in the menu bar
- C ☐ Click the What's the IP? Button
- D ☐ Wait for window to open
- E ☐ Click the save button
- F ☐ Click OK on the Save Successful MessageBox
- G ☐ Close the Whols? window

more advanced.

16 • Attaching an item to the case

- A ☐ Click the 'Attach item to this case' button
- B ☐ In the window that opened, select the 'Browse' button
- C ☐ Browse to 'example.pdf' located on the Desktop + select (ok)
- D ☐ Enter 'Attaching evidence to case' as the Note + click
- E ☐ Click 'Attach File' button
- F ☐ Wait for file to be copied to case container
- G ☐ Click 'Close' button

17 • Perform and save Whols? information

- ☐ Visit <http://canterbury.ac.uk>
- ☐ Click the Tools option in the menu bar
- ☐ Click the Whols? button
- ☐ Wait for window to open
- ☐ Click the save button
- ☐ Click OK on the Save Successful MessageBox
- ☐ Close the Whols? window

1748 • Viewing a Screenshot previously taken²

- A ☐ Click the Audit Log button
- B ☐ Select Website Actions Tab

² This can also be applied to Attachments & Videos tab

- C ○ Click row with an id of 3
- D ○ Wait for image to load in the File Previewer
- E ○ Click the image in File Previewer
- F ○ Inspect and close the file
- G ○ Repeat previous steps for id numbers 1, 4 and 5. ✓

18 • Searching Audit Log

- A) ○ Open Audit Log
- (B) ○ In the search bar, type: Canterbury
- (C) ○ In the tab selection menu, select 'All Tables'
- (D) ○ Click the Search button
- (E) ○ Click the Audit Log button to go back to the main Audit Log

1920 • Exporting a report as PDF³

- A ○ Click the Audit Log button
- B ○ Click the 'Export report' button
- C ○ Select 'Top Secret' from the GSCP Stamp drop down
- D ○ Click the Browse button
- E ○ Select Desktop from the now opened file dialog
- F ○ Click the 'Export report with artefacts as PDF' button
- G ○ Inspect the report that opened
- H ○ Close the report

21 • Export a report with options as PDF

- A ○ Click the Audit Log button
- B ○ Click the 'Export report' button
- C ○ Select 'Top Secret' from the GSCP Stamp drop down
- D ○ Uncheck Loaded checkbox
- E ○ Uncheck OSIRT actions checkbox
- F ○ Click the Websites Actions tab
- G ○ Uncheck the 'print' checkbox for id numbers 2,3 and 5
- H ○ Click the Browse button
- I ○ Select Desktop
- J ○ Click the 'Export report with artefacts as PDF' button
- K ○ Inspect the report that opened
- L ○ Close the report
- M ○ Click the Audit Log button to go back to the main Audit Log

2122 • Shutting case down

- A ○ Click the close button
- B ○ Click Yes when asked if you want to close the current case
- C ○ Wait for case to close

2223 • Open existing case

- A ○ Double click OSIRT icon on the desktop
- B ○ Click Load case button

³ This can also be applied to exporting as HTML, XML and CSV (note XML does not automatically open)

- ☐ C Select cog_walkthrough.osr from the Open File Dialog
- ☐ D Click Open
- ☐ E Click Open Case

APPENDIX F – INTERVIEWS

These participants have provided consent that these interviews can be displayed fully transcribed in this appendix. Most participants provided consent only for quotes and not entire transcripts to be made available.

The order of these interviews are entirely arbitrary.

Interview 1

Joseph Williams (JW): So you're happy with the reasons we're conducting this interview, all the ethics have been cleared...

Participant: Absolutely

JW: And you're happy?

Participant: No problem.

JW: Wonderful.

JW: We'll start with, softball question, what made you interested in joining the police force to begin with?

Participant: I always wanted a job that was not predictable. So the same thing where I wouldn't have to do the same thing day after day. When I went to work, I didn't know what I was going to encounter; I wanted the unpredictability.

JW: What did you start as? Public Order?

Participant: I started as a response unit, so, general policing duties. Uniform, in the job, out in the cars and attending any incident that was an emergency or non-emergency.

JW: So when did you move into Open Source Research?

Participant: This has been quite a recent thing for me, I would say within the last 2 years and it's because of the role I do and what we ask some of the people I work with to do. It works with the open source, but also on the covert side of things.

JW: Did you have a speciality, which is why you were picked for the role? Or did you want to do something different?

Participant: I like to do something different, I like to learn new things. I would class myself as being computer literate, but not tech savvy. So I want to improve myself.

JW: Why do you need to conduct Open Source Research in your role?

Participant: The necessity for my role is purely from an intelligence perspective, so it's finding things out in order to inform a policing plan and a policing project.

JW: Interesting that you mention the word intelligence there, and what I noticed that perhaps within UK law enforcement, is that the term Open Source Intelligence isn't used as much. Are you familiar with the term OSINT?

Participant: I am familiar with OSINT.

JW: What is the difference between Open Source Research and OSINT?

Participant: That's a very good question.

JW: I genuinely don't know.

Participant: We have an OSINT team and I had a two-day presentation from them. From my perspective, they tend to look at it from an investigative background. Yes, there are intelligence dividends that come out of it, but in the main they will look at it, they will harvest the information.

JW: So is it more of a team look? So intelligence is more of an analysis from a team...

Participant: Yes.

JW: Whereas Open Source Research is gathering information...

Participant: Very much bespoke.

JW: With not as much analysis from your peers.

Participant: No [In agreeance with previous statement].

JW: Would that be a fair assessment?

Participant: I think that's a fair assessment. And what my role would require me to do is look at what I've got, and what I can actually achieve, based on, first of all, the information I have been given but then how do I go and quantify it, how do I risk assess, how do I quality assure it and then what do I do with it? So there's a process I go through, and it is bespoke to each piece of information that you get. So that's where the Open Source stuff is vital for me. It allows me the opportunity, say, I've been given a piece of

information about Joe Williams, I want to find out if I can corroborate that information in some way, because if I can corroborate it that means I can break it out. I have to be very careful in my role that I don't put the people to talk to me at harm. So, we grade things, as to who knows about it, who can we tell. So if I can go to Open Source Research, and it says "Joe Williams goes to the 'Boot and Flogger' every Friday, gets tanked up and drives home". If that's the information I've been given from one specific intelligence thread, can I find it elsewhere?

JW: So how do you go about that? What tools?

Participant: We have a stand-alone covert computer, that we use, so we have no footprint, from the police perspective and it would be simple Open Source Research. So if we know it's Social Media, we use Social Media. If we're able to, sort of, tap into other areas of local interest, say newspaper articles or Internet...

JW: Is there any particular software that you use?

Participant: We don't use particular software.

JW: So you just use a standard web browser? Firefox?

Participant: Yeah.

JW: And how do you document?

Participant: We have a... We're very old fashioned. This is why I think OSIRT is going to become extremely handy. We've got a book [to write in by hand].

JW: Do you see any advantages of automatically logging versus the book?

Participant: 100%. It's been the standalone thing that I've taken from the presentation [presentation on OSIRT] and from working on the system today is that ability to look at things, document it, take what I want from it, but then not have to then say "Right, I've done this and I've done this" and I'm writing down everything that I'm having to do. It's there, it's documented for me, it's auditable so if my superiors want to see what I've been doing. If the CPS [Crown Prosecution Service] want to look at it at some stage from an evidential perspective. Or, if we're being looked at, my department tends to get looked at

by the OSC [Office of Surveillance Commissioners]. If they're coming and inspecting us, and they're saying "What processes do you have to control and audit the information you come across?". I can then say, "This is what I've been working on, this is my investigation, this is my intelligence case at the moment and this is what I've done" and they can review it.

JW: Were they [OSC] satisfied before with just the pen and paper? Were there any issues that have happened?

Participant: It wasn't reviewed this year, the review we had this year. It focused in online stuff we're doing, but not Open Source Research.

JW: I was going to ask how effective these tools are, but if it's just pen and paper... How effective would you rate it?

Participant: If it's written down, it's evidential and it's also auditable. It's not as effective, it's not as time effective and unless you're writing things down verbatim you're not going to be as accurate as you are with the audit log you can have on OSIRT.

JW: Can OSIRT be applied within your current role?

Participant: 100%. Very easily. And, actually, it's going to... If we're able to move over and use it, I don't foresee any problems with doing that, it will make my job easier it'll also satisfy the concerns that my superiors, in particular my authorising officer, has regarding online work.

JW: So how do you see that integration of OSIRT in your working role?

Participant: It's quite a simple sort of transition that we move away from our current system, which is to use pen and paper to record things, and straight into using the OSIRT application. The note [case notes] tab is particularly relevant to me, as it allows me the opportunity to put through what my thought processes are and then apply my rationale and then what the outcome has been because of that.

JW: How do you feel about current legislation in regards to Open Source research? Do you feel it's effective?

Participant: The difficulty I have with the current legislation is that it's a bit of a grey area with regards as to what's guidance and what's legislation. I think that's the crux of the matter. There's a necessity to clarify the situation so we know we can do and what we can't do. I always border on the cautious side, I'm cautious by nature, but I look at it from my job perspective I have to make sure I've gone through all the right processes. So being able to go through all the right processes and using a tool, such as OSIRT, to do so I think satisfies my professional and moral standing on what I should be doing.

JW: How do you feel Brexit could affect your ability to conduct Open Source Research in the future, given the Human Rights Act could no longer be a factor.

Participant: Personally I don't see ECHR is going to change, I think we'll have an incarnation of it. In my opinion, there is no way human rights lawyers are going to allow themselves to be wrote out of the ability to earn copious amounts of money. So, therefore, there will be some form, some incarnation of ECHR so I don't think it's going to have a big effect.

JW: So when we look at these laws, RIPA 2000, Data protection Act 1998, the Human Rights Act as well, that's 1998. We look at those years, then compare it to when Social Media was created. Facebook, 2006 I believe, Twitter, later again, all these platforms come considerably after these laws that are used. How do you integrate these laws into Social Media?

Participant: It's very difficult. It's an extremely pertinent question. Especially in the line of business that my team works on. RIPA doesn't actually fit around social media, it doesn't focus on it, it's never been catered for. So, in effect, what we're doing at the moment is sometimes by trial and error, sometimes by trail blazing, you're going through and as long as you're saying you've audited it and you've considered every available possible contingency at what you need to do, that's good enough at the moment. I think there needs to be a re-write [of RIPA], because we are in that social media age.

In order to do either from an investigative or from an intelligence perspective, there needs to be clarity on what you can can't do.

JW: So that's why you're cautious to begin with, that's why you like to log everything, because it is so grey.

Participant: Absolutely. It's looking at whether or not, looking at it from my job, which is agent [?] handling, so working with Covert Human Intelligence Sources is the person I'm asking to be a CHIS to do some work online are they a CHIS or are they undercover? And there's a big debate going on at the moment, as to which side that actually fits on. And that's why RIPA needs to take that into account.

JW: Definitely.

JW: Thank you very much indeed.

[End of interview]

Interview 2

Me: Hi. Thank you very much, [Participant], for being a willing participant in this interview. The first question is a softball question, is: why were you interested in joining the Police service?

Participant: Because I think I watched too many police programmes as I was a child, and - Beverly Hills Cop actually, one of them – and I liked the style of policing; being down to earth, being on the streets, and making a change. And I think that's one of the reasons why I wanted to be a police officer.

Me: Beverly Hills, so something... is that 90210?

Participant: No. Beverly Hills Cop. Eddie Murphy, he's a funny character, and I... I'm not saying I break rules, but I like the way he stands for the street, you know.

Me: Yeah. And does that influence the way you police, as well? Well, obviously, not in that way, but...

Participant: I keep a... I'm from the area I work, so I think I have sort of connections with the people I deal with. In not... not the way they act, but the way they've maybe been brought up. So I think it helps me being a police officer.

Me: Definitely. How much experience do you have conducting open source research?

Participant: I initially done a course about eighteen months ago. It's a level two open source course, which is a very kind of basic course. I can't add anyone, or... I just look... YouTube, Facebook, Twitter -- I have a false persona already. That's kind of it, no further really.

Me: So eighteen months ago, that's when you were trained...

Participant: Yep.

Me: And you were tried just to go on the internet, and look at websites.

Participant: Yeah, websites.

Me: And not prior, before that? You just unwillingly – you didn't know you were conducting open source research.

Participant: Yes.

Me: But this is when you officially trained, so to speak.

Participant: Yeah.

Me: Okay. Oh, why do you need to conduct research?

Participant: My role in the Met, I'm on a gangs unit, most of the gangs use social media, to incite violence. This is done by using YouTube making videos going into other gang members' territories, insulting other gang members. There's so much information you can get off these videos and posts on Twitter, and Facebooks. It's a gold mine.

Me: So it's a way to gather evidence for gang... you deal with gang-related violence.

Participant: Yes, that's right yeah.

Me: So what do you need from a software tool to conduct open source research? So your dream, basically.

Participant: To what I need from a tool is... go on to a website or, say, Twitter, and manage to capture everything I need in one go, with one click and it's saved, and then it's encoded, encrypted, without me doing it. Also, our policies in the [police force], I know other forces change it, is you've got to document everything you do step-by-step, with open source... this is to be transparent, so if we ever go to court with the information we have we can go back to our notes and it can show how we got to that page. Or, just showing how we led up to it. With your app what we've used, you've showed me that it has the capability of storing every different URL we go to, which saves a lot of time. And you also have a log on there, which I can keep up-to-date as well, so there's no need for a pen and paper next to my terminal anymore.

Me: So you used to use a pen and paper. And you can... can you see OSIRT integrating into your current role?

Participant: Yes, yes I can. All I need to do, as we use a standalone computer, is probably after I finish with a case is to load all the data onto a CD and file it somewhere for any...

Me: So, OSIRT being self-contained, you don't need to install it. And you have self-contained case file. That's quite useful for you because you have to use a standalone computer.

Participant: Yes, yes. So if I...

Me: So portability is important.

Participant: Yeah, exactly.

Me: So if I... sorry I cut you off there... because you have to...?

Participant: Because we have to document it, and store it... I mean it's on the computer anyway, but we can transfer it to a disk and it can be stored for a, however amount of time, and it's always there, so it's perfect.

Me: So you can see OSIRT working quite well with what you do.

Participant: Yes. Yep.

Me: Gang-related crimes, in an evidence gathering form.

Participant: Yeah. Also, the OSIRT tool where you can capture the YouTube videos. Prior to this I've had a lot of issues capturing the sound and picture at the same time. Other apps I've used, you get free trials for a little while and... money's tight these days. You don't get the funding to have these apps, even though they still want the work done. So having that tool...

Me: So having a free tool, that's got to remain free, is ideal.

Participant: Yep.

Me: What other tools do you use? So you mentioned some tools there, what tools were they, to conduct research prior to OSIRT.

Participant: FastStone Capture, I used that...

Me: Oh yes.

Participant: I've used all sorts of tools. I've gone all over the internet looking for... I use free trials...

Me: So you have to use free trials.

Participant: Yep, I have used, I do still use YouTube-to-MP3...

Me: Is that an online, er...

Participant: ... MP3 and MP4, it's an online... yeah, where it converts YouTube videos, but it's only, it does only a certain amount size of the video. It might not capture it all. So, it's handy.

Me: So, you mentioned the Met has documentation. Is this country-wide procedures? Or are these just procedures specific... obviously...

Participant: It could be... it's what the Met brought in with policies, how to record things using open source. I don't know about other constabularies, what they do to, to record how they get to a certain page, or how they located a suspect. I don't know...

Me: So there's no standardisation to speak of when conducting open source research.

Participant: It's a grey area.

Me: Because it's all policy-based.

Participant: Yeah. Yep.

Me: So you mentioned these tools, Fast... was it...?

Participant: FastStone capture.

Me: Yeah, I am familiar with that. How effective are these tools? You mentioned that the YouTube downloader wasn't always effective because it cut off. What about these other tools?

Participant: FastStone Capture, they were good, but like I said it charges you after a trial.

Me: Yeah. What was it about that tool that you quite liked?

Participant: It done everything what your tool does basically. It's obviously set up a little bit different, but.... What you'd do, you'd open FastStone Capture up separately from using say Google, but with yours you have to use... OSIRT, you'd have to start up OSIRT to go into Google, so that would run separate to Google. You just open it up when you need it, but obviously you wouldn't get the logs or anything so you'd still be writing up the logs.

Me: How much time do you think you spend hand-writing those logs?

Participant: Probably, I'd say, 80% of the time documenting what I'm doing, and 20% actually doing the work. So it's the time...

Me: Do you... so with OSIRT do you feel you could... it's more time researching.

Participant: Yes; more time researching, less time writing everything down.

Me: Yeah you're doing well answering these questions before I've asked them. It's nice. So you mentioned standards, you mentioned documentation... does current legislation provide enough for officers like yourself to conduct open source research?

Participant: I think it's a grey area still. As we've studied earlier, regarding how many times you can visit a certain person before it comes...

Me: It says... it says repeated, but there's no number.

Participant: No, exactly. I think it's got to be more specific. Because if someone does make a complaint, I don't know... I don't think it would be covered with what we do. Because like I said, I look at gang members, and I look at certain gang members more than I do others, because there's more prevalent ones that make videos, or Tweet more, or... just interact with other gang members, goading them on. So am I breaking the law? No, not at the moment, because of the way I interpret it.

Me: What law do you use to...

Participant: It... it'd be the Human Rights... peoples' personal, like, rights...

Me: Do you need to use RIPA at all?

Participant: Um, RIPA, yes that'll come under RIPA as well.

Me: So human rights... and... so the Human Rights Act...

Participant: It's more... it's respect isn't it. But obviously I'm looking at them more, and I'm obviously directing my searches towards them more. Because they're the more prevalent ones. So, am I... am I breaching?

Me: Are you breaching... that's the... well that's... what social media has blurred, I think. Perhaps before social media we had forums, and there were aliases, so... would that... that could... is that breaching there, well no, because they were using aliases. But now we have social media, and Facebook, and... yeah, it's hard isn't it, because is that... okay, perhaps their profile is open. I don't know. Is there still an expectation of privacy? And as you say, it's a grey area.

Participant: Yes. I feel that it needs to be more looked into, and given a definitive answer.

Me: Definitely.

Participant: With that, that will challenge a lot of work I do because I'll be getting, you know, I'll be asking for RIPA authorities all the time. So if they change the legislation to "Look once, after that..., or, "look twice, after that you need authority."

Me: Yeah. Is it... so if we look at, we'll look at RIPA and Human Rights Act. Human Rights Act is 1998, RIPA is 2000. And you do a lot of work on Twitter and Facebook.

Participant: Yes.

Me: And this social media... RIPA and Human Rights Act pre-date social media. But yet, you're... well not being forced to use it, because it's all you have to use, but you have to use these laws. Do you ever feel uncomfortable using old legislation?

Participant: Just like most of the laws, look how old all the laws are.

Me: Well, yeah...

Participant: They're all old.

Me: That's true. Does... but it's being implemented into modern... so when it comes to social media...

Participant: That's completely it, yeah. It doesn't match.

Me: How does it affect you, though. Do you ever stop and think "Am I doing this right?"

Participant: Yeah, of course you do. But you're trying to make the job work. And that's when, probably, every single person in the room makes the job work, because that's

what we've been doing for years. But, is that right? Erm... no, because we'll be the ones getting in trouble...

Me: How do you cover yourself, so to speak?

Participant: By doing these logs. This is how I do. Logs... I don't think I'm doing anything wrong by looking at open sites, going onto YouTube, because it is for everyone. I'm not intruding any more than any other person can do, at the moment. So I'm not going in, adding them as a friend, talking to them, trying to get information out of them. They're giving me that information, it's out there. I wouldn't even need to go onto Twitter, I could just put their Twitter name in Google, and it will come up what they've tweeted. So, you know, that's how I feel... I'm not actually intruding that much. But in another person's eyes, I am.

Me: And that's because of... and that's just down to the cloudiness of the law.

Participant: Exactly. Exactly. And that's how I feel on the matter.

Me: Thank you for that. It's a good answer, so thank you. Um, I suppose because Brexit - we've spoken about the Human Rights Act - is that going to make any difference to you in two years. So you mentioned the Human Rights Act here, that's an EU law.

Participant: That's going to stay in, so...

Me: How... that's going to stay in because the British government's going to...

Participant: Yes, yes.

Me: Okay. So you can't see it making any difference at all.

Participant: No.

Me: What about... have you ever been... have you ever gone "Oh, that blasted Human Rights Act is preventing me from doing something."

Participant: No. Not... not really.

Me: It hasn't been a hindrance to you.

Participant: No, it hasn't really been a hindrance to me because I don't think I infringe on it that much.

Me: Do you want to clarify 'that much'?

Participant: Just, I don't... I don't do anything that I shouldn't do. You know?

Me: Yeah. You have to use old laws.

Participant: Old laws...

Me: On new technology. And you do the best you can.

Participant: Yeah, exactly. Like I said earlier, I don't do anything that breaches their human rights. I don't communicate with them, find out any information, where they live or you know – what they're doing.

Me: So you see yourself as just any person using the internet when you're conducting research.

Participant: Yes. I'm not myself, I'm a different person. I'm portraying myself as a completely different person, so I'm not me. I'm a member of the public. On a standalone computer I'm not even... er, attributed to the Met police. So...

Me: Moving back to OSIRT, is there anything that you'd like to see added?

Participant: Off the top of my head, no. I think it's good... it covers it all. I mean, you have the... you can take snapshots, you can take videos, you can download off of YouTube. You've got your logs, you've got your notes.

Me: So it does everything you need, as someone who conducts open source research.

Participant: Yep.

Me: Well these other tools... you couldn't speak to any developer.

Participant: No.

Me: Do you feel that you could get in touch with me and feel that you could make a change to OSIRT ...

Participant: Yes, I feel that you're very approachable and you've helped us out loads, and when there's been bugs or anything you've been straight on it. So...

Interview 3

Joseph Williams (JW): I'll start with a softball question; what made you interested in joining the police force to begin with?

Participant: I was at university studying artificial intelligence and computer science and I got bored of computers. So one of the lads, he wasn't doing my course but in my accommodation, was a bobby for the met [Metropolitan Police] who was on a three-year secondment to do a degree. I know "I quite fancy that", so in my final year I applied to join GMP [Greater Manchester Police] and got in. I applied to see how it goes, but the more and more I got through the process the more and more I thought "I fancy doing this", because I thought I always have my degree to fall back on and I've never really looked back.

JW: How much experience do you have conducting Open Source Research (OSR)?

Participant: Depends. I was the Intel [Intelligence] [REDACTED] where we, as the Intelligence Sergeant, run open source work and we would do stuff with Facebook and stuff like that and I recently run the regional cybercrime unit for the [REDACTED] which is an amalgamation of the five forces: [REDACTED], where our main role is to investigate cybercrime. So your DDOS attacks, malware, network intrusions that sort of stuff and as part of that we do a lot of open source and that type of work. I have a researcher on my team who specialises in open source research and she is trained to a very high level and her trade craft is to a very high level into relation to open source.

I do, and have done bits [open source] in the past, and the reason I'm on this course is because I haven't got the tick in the box.

JW: So how many years' experience would you say you've done Open Source Research?

Participant: I would say, probably about, 5 or 6 years between different roles.

JW: But as you say, you just need that tick in the box.

Participant: Yeah. In conscious of the ISO potentially coming into this world. We can no longer get away with, and we're doing a lot of work on the forensic side of things ISO

wise to get our lab up to speed, because I have my own forensic examiners as well. The reason I'm here is because this is the only course that's recognised nationally as a level 3 course. There are lots of different courses out there, but this is only one I could find that could give you the rubber stamp is here [College of Policing]. If I'm stood in court, and they decide to question my competency. The first question is "Do you have the relevant training". I would hope the College would come along and go "This is what we trained".

JW: Why do you need to conduct OSR in your role?

Participant: Because the majority of our offenders are online, and I think if you're targeting a subject the main part of that research on that subject nowadays, no matter where they're from, open source is one of the many facets you should be looking at. In my world, it's one of the main ones, because my offenders are all over the world, I may never get to the physical body, I may never get to arrest because they could be in any country. So the thing you're looking at is, what is their online profile, what are they doing online.

Hackers do like to boast. So they do like to have that, you know, they tag what they're doing. So whether it would be your organisations, your lizard squads, your individual hackers, they're putting themselves out there; they're proud of their work.

JW: You mention that open source is a substantial amount of your work. As a rough percentage, how much of the evidence do you gather comes from open sources?

Participant: On my team of six, I have a dedicated open source researcher, who's role is to conduct open source research.

JW: So you have a person on your team who is 100% dedicated to open source research?

Participant: Pretty much. They do closed source research to, such as PNC, but her speciality is open source and the majority of her time is online doing open source research.

JW: OK. What about yourself?

Participant: Well, I'm the manager so I do bits and bobs myself. The majority of that work is managing and understanding what they're doing. If a job kicks off, and the reason I'm

here is that if something happens, there's a live attack. One thing you initially start doing is the open source research. That's where a lot of your stuff comes in. It's all hands to the pumps. There's no use me doing that, and not being able to evidence it, so hence me being here.

JW: You touched upon intelligence, so you're familiar with Open Source Intelligence (OSINT) and Open Source Research.

Participant: What I meant there was, I may look at something for intelligence purposes or am I looking at it for evidential purposes. Which are two different things.

JW: What is the difference?

Participant: For something to be intelligence, it has to be graded. For example, I may see something online and I would capture that, but it may never become evidence. So it's not something I'm capturing, I'm not saying I'm ever going to be at court saying "I'm doing this", but I've captured it because it's intelligence, it points me in the right direction.

JW: So who grades that? Is that your personal opinion, or is that with your peers?

Participant: When I say grading, there is a grading we do in law enforcement. It's across all intelligence, it's not just limited to open source or to computer... Say you ring up Crimestoppers, say "I've just seen this happen" and that intelligence would be given a grading. You grade the person, how reliable is that intelligence, and apply a handling code.

JW: Just to clarify, just for me, you have Open Source Intelligence (OSINT) and you have OSR. What is the difference?

Participant: I don't think there is.

JW: you don't think there's any difference at all?

Participant: Again, it's terms that are bandied about. Different organisations across the board you go in to, different terms will mean different things. To me, OSR and OSINT is the same thing.

JW: Do you think there's more analysis for OSINT in comparison to OSR? Do you think there's more hands in the pot in regards to OSINT?

Participant: No. Again, it comes down to terminology. It depends what you're looking at, what level of analysis you go into and what you're doing. It's all open source work.

JW: What tools, as in software tools, do you currently use to conduct OSR?

Participant: We use a number of capture tools, SnagIt, Camtasia. [REDCATED – Open Source Researcher's name] uses a lot of tools they have come across. We don't, at the moment, have an overarching package that does it all, like your package here, but it is something we're looking into.

JW: Say Snagit and Camtasia, they are for capturing still and live (video) images. How do you document, how do you maintain an audit log?

Participant: Just using Excel or physically writing them down.

JW: Did you have much say in these tool's development?

Participant: No. Not for that, no. Other bits of software we do [have a say].

JW: Are you allowed to name them?

Participant: No. They're forensic tools.

JW: When it comes to Camtasia and Snagit...

Participant: They're just off the shelf products.

JW: What do you need from an OSR tool? What is the dream?

Participant: The dream for me, would be something that allows me to VPN out and I can choose where I'm going, I can choose what my mac address is, and I can choose what operating system I'm showing to the world and that I have control of that. I want it look like I'm in France, I want my computer to look like Windows XP.

JW: That sounds almost bordering on the covert side, you're hiding your identity.

Participant: A lot of the stuff we're doing [in class] is based around covert work. So you are, the minimum you'll do, is you'll be running it through an IP address that isn't attributed back to the police. If you're doing OSR, you'll be running it under a covert banner. So you might be looking at Facebook, or whatever, but you're trying to obfuscate [inaudible] that may be because you're trying to hide yourself from Facebook because of their tracking tools and stuff like that, or it may be because you're hiding yourself from the offender because you're going to their website. You don't want them to see that it's the Leicestershire police IP address coming through.

If you take it a level further, depending on where you're going, again these are the type of investigations we potentially may get involved in, if anyone does look at you, you want to portray a certain character. So that may be, well, I'm running Linux, I'm doing this, I'm doing that. Or you're actually running Linux, but want to be shown as running Windows 10. It's not something everyone would need, and one of the things we're looking at is that base level, where the officer probably doesn't have the trade craft, so they would just click on something and it would automatically... They may VPN out, but it doesn't matter where they're popping out- it may show UK or whatever. Then you want something higher level, and they can start choosing that type of stuff.

JW: It's interesting how advanced those techniques are. You haven't chosen some of the simpler techniques such as "click this and take a screenshot" ... You've chosen VPNs.

Participant: Yeah, the world that I investigate, the people using computers are highly computer proficient. You have to try and be one step ahead of them. Obfuscate yourself online, it's no good just saying "I have an IP address". When you're talking about the perfect system, that would be the base of that system.

I quite like your system, to be honest with you. I've seen different ones. And, you know, it's that what you want it to do is log every single website you visit, and some of them don't, some of them don't capture everywhere you're going, you want it to capture, as yours does. I'd also want to be able to capture a video in its native format, if possible, because if capture what's on the screen like yours does, the resolution may not be there. Down the line, that resolution may become very important because if it's a 4K video, say if it's child abuse, you may be looking around the room to see if the plugs are European plugs, so it's likely this offense hasn't occurred in the UK. That type of stuff you may want to pull out from [a video]. Right click on a video, and say "Right, I want to download

that. I want to store it.” That sort of stuff. You want to be able to log it, you want to be able to take the notes, and the big thing is you want to package it all up very nicely with what you want in there. I’d like to be able to annotate stuff as well, so you’ve got a website up there. You’ve got the original website there, and I love the fact you hash that. But then I also want to be able say “Right, this is what’s interesting to me”.

One of the frustrations I have, whether it be from computer forensics or this type of stuff, is once you’ve got that everything is really hard, because there’s nothing that does it all for you. You want to take that capture, bang, and then create the report. The report will either be on a disk, or whatever.... I did write down a number of suggestions [for OSIRT].

JW: We can have a chat about it after.

Participant: For me, the report should be encrypted, or an option to automatically encrypt that report, because there may be sensitivity on that report. If I’m then handing that report to the CPS and it’s already encrypted, then if that disk is lost with that report on, then there’s at least some level of security there. As a manager, I may want that turned on permanently, so my staff can’t export data out that isn’t encrypted. It just puts that layer of security on. Tools like this do sit of specialist units, but there is a big push to push it out to general policing. You’ve got to make OSR daily business. Police officers already check PNC, they’ll do their research on somebody, but that research doesn’t already go on to the online world. Well, you know, a lot of people put their whole worlds online. Criminals will sometimes put their whole lives online, and it has to be business as usual, it should no longer sit with the specialist teams.

JW: You say OSR currently sits with specialist teams, you think it needs to be pushed out more into your average... So your PC sitting at their desk, open source needs to become part of their routine as well.

Participant: Yes. And I say, we are pushing towards that. We just put a tender out, it’s on something called BlueLight which is why I can talk about it, we’ve gone out to companies saying “develop us an open source tool that will allow us to have an icon on the officer’s desktop, double clicking on that icon and it puts them out securely, not showing out as Leicestershire”. They’re not going to do Facebook friend requests, that sort of level. There’s no interaction, but for doing that general OSR they can go out, as

long as they have the relevant authorities, because they have to put all that in. So, again, on your case thing you might want to have a box for, you know, authorities.

JW: I have the general notes box at the front there, to cover anything missed. Each constabulary has a different...

Participant: Yeah, exactly. It's difficult, because we do have 43 different ways of doing things. Even within some forces it's 2 or 3 [laughs], because this is all new, we are still learning this.

JW: So there isn't much standardisation between constabularies?

Participant: We're working on it. And, I'm not trying to blow my own trumpet, but EMSON is five forces working together as one. The forces have realised you need to collaborate and we're doing quite a lot of work to pull it all together.

JW: If we take a look at what you said about downloading videos, if we take YouTube for example, their terms and conditions state you're not allowed to download videos, has that ever been a concern to you that you're breaking websites terms and conditions?

Participant: No. No. We're doing it for lawful purposes. So I'm not downloading somebody who's uploaded the latest Avengers movie for personal use, I'm doing it for lawful purposes. Usually, if you're doing that, you have your DSA [Directed Surveillance Authority] in place, and if you're doing anything surveillance wise, you have to have one of them in place. Like if I was in the World, I can't just say "I'm going to have a look at Joe", I can't just follow you around, potentially I'm breaching your human rights. The DSA allows me to do that, so we've got that level of protection. Having ghost accounts breaks Facebook's terms and conditions, but again, we have DSA.

JW: Has there ever been an incident where a ghost account has been chopped?

Participant: They'll chop them, but whether that would ever be an issue at court, I don't think so, because it'll be Facebook having to take you to court.

JW: And it'll be civil as well.

Participant: It'll be civil court, yeah. Again, it comes down to necessity and proportionality test. RIPA, when it first came in, councils were using it to trap people dog

fouling and stuff like that. That's not what the legislation was there for. So I'm comfortable for that type of investigations we're running, I would quite happily stand up in front of a judge and say "yeah, ok, I've downloaded that video from YouTube but I done that because that video contained this evidence linked to this job". So, again, it comes down to necessity and proportionality.

JW: We'll come back to RIPA in a second. We'll go back to OSIRT for now. I'm wondering how OSIRT could be applied in your role? How would you see its integration?

Participant: To integrate it, it would be very simple. Because at the moment, as I say, we use lots of different tools. So, you know, the way... I'm going to pass it [Researcher's name] who's my researcher and I'm going to ask her to have a look at it. She's looked at lots and lots of different products and she'll give me an honest opinion and if she says it's good, then we'll take that on and pass it to other members of the team.

JW: If she says it's not good, please pass on the feedback.

Participant: To be honest with you Joe, I'd be quite happy for you to come up and see us.

JW: That'd be nice, actually.

Participant: Again, we have our own R&D team, so linking you in with them and pushing this type of thing forward... We like to get involved in projects. There's a need for these tools, but again, everybody's need is a little bit unique. You can have something that's generally good enough to do everything, or do you go for bespoke.

JW: That was one of the reasons I released the source code. Someone may say "this sort of does what I need, but it needs to do this" and they can extend it.

You mentioned RIPA earlier, that's 2000, Data Protection Act 1998, Human Rights Act 1998 then when we look at social media, Facebook 2004, Twitter 2006... How does legislation that pre-dates social media, how do you use legislation such as RIPA that was created in 2000 to integrate.

Participant: With difficulty. Everything's a grey area. Because it's all still quite new, and there's been very few stated cases, a lot of the stuff is someone's opinion. So it'll be either

the cab (?) manager's opinion or the designated persons, or my opinion of where we can go with something and it's very hard to get a definitive answer about who can do what and what authorities you need to go where and to do certain things. Because technology changes so quickly, legislation is struggling to keep up. So if you've seen the new Bill that coming up, that's already getting out of date, so you're in that situation where you're constantly playing catch up. When RIPA was wrote, you had telephones, voicemail, and you had letters and maybe e-mails. You've now got WhatsApp, Facebook Messaging, things are peer-to-peer encrypted all of these things, and I understand why companies have done it, but everything makes our life that little bit more difficult. Us trying to exploit the investigative opportunities that are out there, a lot of times, I can do it because bad guys are doing it but legislation says I can't do it. So it's getting that balance right between what we can do, what we can't do, and legally what we can do if that makes sense. As a nerd, you want to push the boundaries but then you've got to constantly be saying to yourself "Can I do this? Is there a framework I can put this under?" So when I go to my boss, I can say "We want to do this, we want to try that" and we haven't always got that.

JW: May as well ask... Brexit, we've voted to leave the European Union, how do you see that effecting your ability to conduct OSR? Particularly in regards to the Human Rights Act.

Participant: Human Rights Act, I think something similar will come along anyway so I don't see that being a massive problem. One of my big issues is I work with the NCA and with Europol and with my international partners, cybercrime as a whole is a worldwide issue, there are no boundaries, there are no borders. If we're careful as a nation, we will make life twice as hard for ourselves and if we make life harder for ourselves, we become more and more of a target for international criminal gangs, because they'll look at Britain and say "you're a soft target", because actually I can sit in France and do this, and you can't get me.

JW: So you still envision there will still be collaboration between the EU?

Participant: Yeah, if you look at Europe at this moment in time. Europol isn't just Europe, Canadians have a footprint in there, Australia has a footprint in there. So we're going to have to become one of them countries to get our footprint in there. So, for me, for us not to do that, to put up the walls and go "no, no, no", it doesn't work because cybercrime and computer crime is borderless.

JW: So Brexit for you as a LEO isn't the end of the world?

Participant: No.

[End of interview]

APPENDIX G – OBSERVATION TEMPLATE FOR RITES COURSE (CHAPTER 10)

Training Observation Form

Date:

Day of Course:

Day:

No. of participants:

Trainer(s)/Facilitator(s):

Observer:

Session Focus:

Description:

Design and Planning

e.g. briefing, session structure, organisation, appropriate to level/learners, links to course etc.

Observations:

Communication

*e.g. pace, clarity, mannerisms, feedback, slides, handouts, signposting etc.
e.g. indicate instructional resources (for example hands-on, audio-visual, printed etc.)*

Observations:

Use of resources and teaching/learning methods

e.g. appropriate and effective use of technology, space, methods/approaches, support students' learning and supports aims etc.

Observations:

--

Learner Engagement

<i>e.g. learners' attention, participation, interactivity, questions, feedback, confidence, management, awareness, needs, learning style, is learning enhanced by the facilitator, does the course hold the learners' interests, does the course seem relevant to the learners, learner satisfaction, listening, writing, reading, computer use (engaged), interaction (among learners/with trainer(s)) etc.</i>
--

Observations:

Trainers Activities

<i>e.g. activities of presenters and participants in the session (for example: focus of formal presentations, description of problem-solving activities, reflections, assessments); how well did the facilitator monitor the session/exercise;</i>
--

Observations:

Strengths

<i>identifying where there were areas that worked well in the session</i>

General Comments/Observations

Anything which is not covered in the above

Notes on Training Environment

e.g. description of space and arrangement, technology etc.

Notes about the group of learners

Anything which is not covered in the above

APPENDIX H – SAMPLE OF DAILY QUESTIONNAIRES FOR RITES COURSE

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Slower pace.

rites TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

No

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

No

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

rites training course - day 1 survey

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Just a list of acronyms, some
are unfamiliar.

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☐ Yes

☒ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

more 100 breaks! (5 mins)

rites training course - day 1 survey

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Would help to have pre knowledge re phone analysis (for candidates - not for you necessarily to teach)

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

NO

rites training course - day 1 survey

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

NO

UTES TRAINING COURSE - DAY 1 SURVEY

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

A bit more background information about how information (like the emails) would have been captured as part of the scenario.

rites Training Course - Day 1 Survey

Were the course aims and learning objectives well defined at the start of the day?

☒ Yes

☐ No

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
Knowledge Indicator	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Legislation and Authorities	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Open Source Internet Research Toolkit (OSIRT)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Communications Single Point of Contact (SPOC) Communications Data Investigator (CDI)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Art Scenario	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

NO.

Is there anything else which could be done to improve today's training?

SLOWER PACE.

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> N/A

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Is there anything else which could be done to improve today's training?

rites training course - day 2 survey

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Is there anything else which could be done to improve today's training?

→ Student missed aspects of today due to illness.

rites TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Is there anything else which could be done to improve today's training?

rites TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow ☒ Just Right ☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Is there anything else which could be done to improve today's training?

rites training course - day 2 survey

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Is there anything else which could be done to improve today's training?

rites training course - day 2 survey

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

No.

Is there anything else which could be done to improve today's training?

No

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

no

Is there anything else which could be done to improve today's training?

Very Power Point heavy → more practical examples

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> N/A

How did you find the pace of today's sessions?

☐ Too Slow

☒ Just Right

☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

I Saw some people confused about
some terminology like SHA256 or
although Shawn had to use it, this could
perhaps use more or basic explanation

Is there anything else which could be done to improve today's training?

As above

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> NOT DONE

How did you find the pace of today's sessions?

☐ Too Slow
 ☐ Just Right
 ☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Nothing further to add today, as there was one topic we didn't cover.

Is there anything else which could be done to improve today's training?

rites training course - day 2 survey

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right

A bit
☒ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Couldn't have fit anything more in my brain

Is there anything else which could be done to improve today's training?

~~Is~~ I personally need much more basic teaching as QR codes / hashing / cache mean nothing to me - not necessarily your problem but possibly send these "basic" details as a bit of a pre read?

UTES TRAINING COURSE - DAY 2 SURVEY

Please rate the difficulty of the following topics:

	Easy	Just Right	A Little Tough	Very Difficult	I'm Lost
World Wide Web	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Methodology and Evaluating Open Source Information	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Steganography and Encryption	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
NMPR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did you find the pace of today's sessions?

☐ Too Slow

☐ Just Right



☐ Too Fast

Were there any other topics you think should have been included in today's session, or any you would like to know more about which have been addressed so far?

Technical Issues caused me to struggle at times keeping up although it has helped to get back on track.

Is there anything else which could be done to improve today's training?

NO

APPENDIX I – IDownloadManager IMPLEMENTATION

IDownloadManager.cs

```
[ComVisible(false), ComImport]
[Guid("988934A4-064B-11D3-BB80-00104B35E7F9")]
[InterfaceType(ComInterfaceType.InterfaceIsIUnknown)]
public interface IDownloadManager
{
    [return: MarshalAs(UnmanagedType.I4)]
    [PreserveSig]
    int Download(
        [In, MarshalAs(UnmanagedType.Interface)] IMoniker pmk,
        [In, MarshalAs(UnmanagedType.Interface)] IBindCtx pbc,
        [In, MarshalAs(UnmanagedType.U4)] UInt32 dwBindVerb,
        [In] int grfBINF,
        [In] IntPtr pBindInfo,
        [In, MarshalAs(UnmanagedType.LPWStr)] string pszHeaders,
        [In, MarshalAs(UnmanagedType.LPWStr)] string pszRedir,
        [In, MarshalAs(UnmanagedType.U4)] uint uiCP);
}
```

DownloadManagerImpl.cs

```
[System.Runtime.InteropServices.ComVisible(true)]
[System.Runtime.InteropServices.Guid("bdb9c34c-d0ca-448e-b497-8de62e709744")]
public class DownloadManagerImpl : IDownloadManager
{
    private Facade facade;

    public IEDownloadManager(Facade facade)
    {
        this.facade = facade;
    }

    public int Download(IMoniker pmk, IBindCtx pbc, uint dwBindVerb,
int grfBINDF,
        IntPtr pBindInfo, string pszHeaders, string pszRedir, uint
uiCP)
    {
        // Get the display name of the pointer to an IMoniker
        interface that specifies
        // the object to be downloaded.
        string name = string.Empty;
        pmk.GetDisplayName(pbc, null, out name);

        if (!string.IsNullOrEmpty(name))
        {
            Uri url = null;
            bool result = Uri.TryCreate(name, UriKind.Absolute,
out url);

            if (result)
            {
                WebDownload manager = new WebDownload(facade);
                manager.FileToDownload = url.AbsoluteUri;
                manager.Show();
                return 0;
            }
        }
        return 1;
    }
}
```

IServiceProvider.cs

```
[ComImport, ComVisible(true)]
[Guid("6d5140c1-7436-11ce-8034-00aa006009fa")]
[InterfaceType(ComInterfaceType.InterfaceIsIUnknown)]
internal interface IServiceProvider
{
    [return: MarshalAs(UnmanagedType.I4)]
    [PreserveSig]
    int QueryService(
        [In] ref Guid guidService,
        [In] ref Guid riid,
        [Out] out IntPtr ppvObject);
}
```