School of Management and Marketing

The Advanced Framework for Evaluating Remote Agents (AFERA): A Framework for Digital Forensic Practitioners

Richard Brian Adams

0000-0002-3839-0791

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

August 2023

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

The research presented and reported in this thesis was conducted in accordance with the National Health and Medical Research Council National Statement on Ethical Conduct in Human Research (2007) – updated March 2014. The proposed research study received human research ethics approval from the Curtin University Human Research Ethics Committee (EC00262), Approval Numbers HRE2020-0736 and HRE2022-0665.

Signature:

Richard Adams

Date: 17 August 2023

Acknowledgements

I extend my heartfelt gratitude to my Supervisor, Professor Peter Dell, for his unwavering support, expert guidance, and constant encouragement. His constructive criticism and scholarly insights have indelibly enriched the quality of this work.

My appreciation also goes to the esteemed members of my Thesis Committee, Dr. Sharyn Curran and Dr. Shirlee-Ann Knight, whose support and feedback have been instrumental throughout this journey.

I wish to express profound thanks to my beloved wife, Jane, whose unwavering support and selfless sacrifices have helped me throughout.

Lastly, my deepest appreciation extends to all the participants and interviewees who generously shared their time, experiences, and expertise. Your contributions have made this research comprehensive and profoundly meaningful.

This thesis stands as a testament to the support and contributions of these exceptional individuals and groups.

Abstract

Digital forensics (DF) is the process of identifying, preserving, analysing, and presenting digital evidence in a legally acceptable manner (McKemmish, 1999). Practitioners in this field can include law enforcement, military personnel, or commercial consultants. As the term "forensics" implies, this domain often pertains to cases that are likely to lead to criminal trials.

While it's not surprising that computers are frequently involved in criminal activities, those tasked with investigating such activities lack well-established standards, procedures, and frameworks for testing (as specifically required by courts) and evaluating their tools.

Despite calls for increased research into tool testing within digital forensics (Casey 2016; Kenneally 2005; Nikkel 2014) and the availability of testing frameworks and sample data (National Institute of Standards and Technology 2020; Scientific Working Group on Digital Evidence 2018b), only a limited number of published test results are accessible.

Slay and Beckett have argued that the scarcity of widespread tool testing is attributed to the "high workload and low resource environment" (Slay & Beckett 2007, p. 1) in which forensic computing practitioners typically operate. They also note that testing forensic tools is both costly and time-consuming, leading practitioners to rely on external validation studies.

Although the U.S. National Institute of Science and Technology (NIST)[1] has developed their Federated Testing Framework, it only offers practitioners limited resources to self-validate their tools through customised datasets for specific functions (like string searches), no such framework exists for collecting and processing live data from multiple endpoints. This is a fundamental aspect that has become more prevalent in digital forensic investigations.

Rather than restrict the focus to an incremental expansion of the resources available for ‘traditional’ static testing and evaluation, the objective of this research is to develop a comprehensive framework enabling digital forensic practitioners to assess the performance of distributed agent-based forensic tools designed for modern digital investigations. By offering this framework, the research aims to empower practitioners to provide court-required reliability assessments through evidence of their own tool testing. Furthermore, it allows practitioners to assess and compare various tools before making significant financial investments. Finally, the

[1] https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/federated-testing

framework can be used, with minor adaptions, as a relatively fast and simple method of testing and evaluating tools that do not rely on distributed remote agents.

This research employs the Design Science Research (DSR) methodology (Hevner et al. 2004), which is well-suited for creating a new framework (an artifact) for evaluating software tools, given its focus on designing solutions (Armstrong & Armstrong, 2010).

The resulting framework encompasses a recommended testing environment that is easy to set up, a dataset for testing, and two checklists outlining desired outcomes.

The Framework underwent review by a panel of expert digital forensic practitioners operating in diverse environments, including large commercial consultancies, law enforcement, military, and federal agencies. Their feedback has been overwhelmingly positive, contributing minor suggestions that have influenced the final version of the Framework.

1 TABLE OF CONTENTS

Table of figures

DEFINITION OF TERMS

Admissible evidence. Evidence that is introduced into a legal proceeding for consideration by a judge or jury.

Data carving. This is the process of attempting to extract data, such as files, from areas of a file storage system that are not in use (unallocated space). The process often uses file signatures as a means of locating potential 'valid' file data rather than relying on information from the file system itself (Raghavan, 2013)

Digital evidence. This refers to any artefact found in computer materials or their derivatives that could be considered useful in an investigation (Bashir & Campbell, 2015).

Digital forensics (DF). This is an applied science of identifying, collecting, and examining data from computer-generated sources to be used as evidence when investigating a crime or dispute (Erol-Kantarci & Mouftah, 2013).

Digital forensics professionals. These perform the extraction and analysis of data from electronic devices to collect information that may be used as evidence in an investigation or court case (Bashir & Campbell, 2015).

Endpoints. These are physical devices that connect to a computer network to exchange information.

Professional practices. In this study, professional practices refer to digital investigation procedures used to collect, evaluate, and preserve digital forensic evidence and the chain of custody of when investigating cybercrimes (Sabillon, Serra-Ruiz, Cavaller, & Cano, 2017).

Remote deployed agent. This is a software application that runs on endpoints. Typically, these are controlled directly by a central 'server' application, but they can also be independent running from a pre-configured set of instructions run on two or more systems that are physically and/or logically separated as part of an investigation. Typically, this is carried out concurrently across a network. This is also known by other terms such as 'remote live forensics' and 'triage', although the later tends to be less 'forensically vigorous'.

Standards. In this study, standards refer to the prescribed methods used by forensic investigators to obtain evidence. Standard methods include the use of empirical tests based on established theories and validated techniques proven to produce reliable and admissible evidence (Adams, Hobbs, & Mann, 2014).

Validation (typically synonymous with Reliability). Key aspects of validation are:

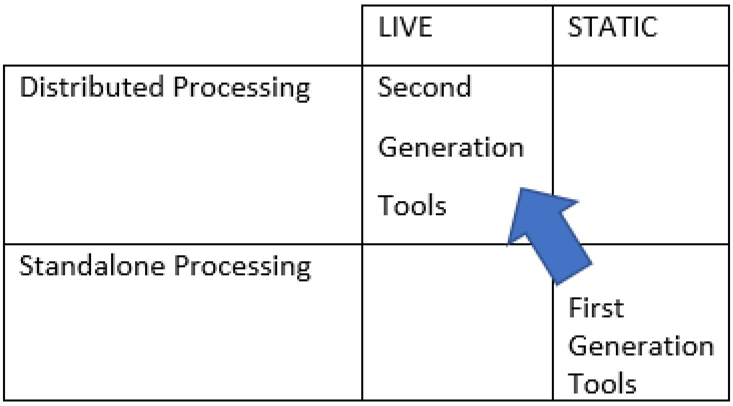
- Accuracy: Ensuring that the tools produce reliable and precise results. This involves comparing the output of the tools with known standards or manually verified results to check for any discrepancies.
- Reproducibility: Verifying that the tools consistently produce the same results when applied to the same evidence under the same conditions. Reproducibility is crucial in maintaining the integrity of the investigation and ensuring results can be independently confirmed.
- Completeness: Evaluating whether the tools can fully support the examination and analysis of various types of digital evidence, including different file formats, operating systems, and devices.

1 INTRODUCTION TO THIS RESEARCH

Digital forensics (DF) is defined as *"…the process of identifying, preserving, analysing and presenting digital evidence in a manner that is legally acceptable*" (McKemmish, 1999) and practitioners may be law enforcement, military personnel, or commercial consultants. As the 'forensics' element of the name suggests, this field involves cases that are likely to result in some form of criminal trial.

The fact that computers are often involved in criminal activities should come as no surprise to most people, however those tasked with investigating this type of activity do not have well-established standards, procedures or frameworks that can be used for testing and evaluating their tools and processes. One problem is the huge variety of different scenarios in which some malicious act can be carried out. This situation has led to 'computer crime' as a general term being split into many categories, each with their own environment and data of interest. Despite this segregation, there is often some overlap between categories.

A further problem in the field of digital forensics is the rate at which their 'target' environment changes, which meant that 'First Generation' digital forensic tools failed to keep up with, for instance, data being distributed across large networks and computers having to be processed while they are still running ('live') – the move shown by the direction of the arrow in Figure 1. This means that any test resources designed for these early attempts at digital forensic tools, where typically they were designed to process a forensic image of the data (a 'static' copy), now have limited application to later tools using remote agents as part of their processing of 'live' systems.



	LIVE	STATIC
Distributed Processing	Second Generation Tools	
Standalone Processing		First Generation Tools

Figure 1 The move from 'static' 'standalone' to 'live' 'distributed' processing.

This research was prompted by the author’s difficulty in finding a suitable framework for evaluating digital forensic tools that collect data from remote sources as part of a forensic investigation.

Two different panels of experts were involved in this research - The Expert Practitioner Panel and the Expert Panel. The Expert Practitioner Panel provided feedback on the high-level and low-level characteristics based on a draft framework document following initial interviews and therefore aided in the design. The Expert Panel evaluated the final proposed Framework and provided feedback.

1.1 DIGITAL FORENSIC EVIDENCE

Courts and organisations are having to deal with advances in technology that cause the amount and scope of digital data involved in everyday life to increase at an exponential rate (Goodison, Davis, & Jackson, 2015). This situation means that digital investigators are having to constantly adapt their processes and techniques as well as the tools they use.

From a court’s perspective, potential evidence presented or derived from electronic data is scientific evidence that is significantly different from that associated with more ‘traditional’ forms of evidence such as documents, fingerprints, and DNA analysis for which there are established standards and procedures to which the courts can refer (Smith, Grabosky, & Gregor Urbas, 2004; Stanfield, 2009).

To have digital evidence presented at court deemed legally acceptable it must be regarded as being reliable based on criteria established to assist with this determination, such as the Daubert Test for scientific evidence (Supreme Court of the United States, 2003). This is on the basis that judges *“…are obligated to ensure that only “good” science reaches the jury*” (Cheng, 2007) and a general concern amongst the legal profession that *“…evidence in a digital format is not to be trusted*…” (Mason, 2014, p. 1).

The fundamental principles of the Daubert Test for reliability are recognised and included in some form by many jurisdictions around the world (Edmond, 2010). A court must determine whether digital evidence being presented is scientific evidence that will assist the court, i.e., it is ‘reliable.’ To do this an assessment is made by the court that includes determining whether the tools and techniques used to produce the evidence can be (and have been) tested and whether their error rate(s) have been determined (Baggili, Mislan, & Rogers, 2007; Murff, Gardenier, & Gardenier, 2011).

1.2 TESTING VERSUS EVALUATION

It is important to distinguish between the typical testing that is performed on digital forensic tools (in line with other software applications) and the evaluation of these tools.

There are four key areas in which ‘testing’ and ‘evaluation’ differ in relation to a software application, namely:

1. **Purpose**
    - The primary purpose of testing is to identify and uncover defects, errors or other issues that arise from the use of the application. It has a ‘quality assurance’ aspect with the intention of improving reliability and performance.
    - The primary purpose of evaluation is to assess an application’s overall effectiveness, value, and its impact on other systems. It goes beyond identifying defects and focuses on analysing outcomes, benefits, efficiency, user satisfaction, and the achievement of objectives. Among other things, evaluation is used to inform decision-making.
2. **Scope**
    - Testing has a narrow scope and focuses on specific components, functionalities, or aspects of the application. It is typically conducted on a more detailed level, such as individual features or modules.
    - Evaluation has a broad scope and considers the overall performance and impact of the entire application. It looks at the interactions between different components within the environment(s) in which the application is to be used and assesses how well they work together to achieve the desired results.
3. **Timing**
    - Testing is an ongoing process throughout the development lifecycle. It starts early in the development phase and continues until the item is considered ready for release or implementation. For digital forensic practitioners, the courts also require them to undertake their own testing of their tools to ensure that they function as expected in key aspects, such as faithfully capturing data without altering it.
    - Evaluation is performed to assess the overall success and outcomes achieved.

4. **Stakeholders**
    - The stakeholders involved in testing are often developers, quality assurance (QA) teams, and technical experts responsible for identifying and fixing issues.
    - The stakeholders involved in evaluation are more diverse and may include project managers, clients, end-users, decision-makers, and other stakeholders. The evaluation process seeks input from different perspectives to provide a comprehensive view of the item being evaluated.

In summary, testing focuses on verifying and validating specific aspects of an application to ensure it meets defined requirements and performs correctly. Evaluation takes a broader view, assessing the overall effectiveness and impact of the application's use. Given that courts refer to 'testing' rather than 'evaluation' the emphasis for researchers and practitioners in Digital Forensics has been on the former, which leads Flandrin et al.et al.et al. to state that, with regard to evaluation techniques, these "…are virtually non-existent" (Flandrin, Buchanan, Macfarlane, Ramsay, & Smales, 2014) and so the development of a means of evaluating Digital Forensic tools must build on previous testing approaches.

1.3 DIGITAL FORENSIC TOOL TESTING

With respect to 'live' forensic tools, such as those that use remote agents, evaluation techniques "…are virtually non-existent" (Flandrin et al., 2014). However, even the current testing environment is behind what might be expected for an established scientific discipline. Digital Forensic practitioners need to show evidence of testing as it is a key requirement for tools and techniques being used to produce material that may be presented in court, especially given that judges and other members of the legal profession are unlikely to be familiar with the technological environment in which they are applied. This requires some definitive evidence of reliability (Horsman & Lyle, 2021; Hosseinian - Far et al., 2020; Jones & Vidalis, 2019).

In 2018 the Scientific Working Group on Digital Evidence (SWGDE) produced a document titled Minimum Requirements for Testing Tools used in Digital and Multimedia Forensics (Scientific Working Group on Digital Evidence, 2018). The purpose of the document is to provide organisations with a guide for validation testing to evaluate "*whether a tool or procedure performs as expected and to understand the limitations of tools*."

Within this document is a section relevant to this research that is titled 'Acquisition Tools' which deals with tools that are used for the physical or logical collection of data. Under this general heading is a sub-section that suggests that tools designed to provide a conduit between a forensic workstation and remote resources should be tested using a known dataset to act as a baseline capability for each tool, an approach supported by Yannikos et al. (Yannikos, Graner, Steinebach, & Winter, 2014).

The SWGDE document also refers to the National Institute of Standards and Technology (NIST) and their Computer Forensics Tool Testing (CFTT) programme, which is designed to assist law enforcement, but which is available to the public. Part of the CFTT is the Federated Testing project whose aim is to provide test suites for DF tool testing[2]. As of February 2023, the Federated Testing project consisted of the following tool categories:

- Disk Imaging
- Mobile Device
- Hardware Write Block
- Forensic String Search
- Forensic Media Preparation

The format of the testing is based around the use case of a single instance of a tool having a particular function tested in isolation against a specific data set containing specially created/modified artefacts. These artifacts will include the most current versions of various document types at the time of the data set creation.

The Federated Test project is an important initiative because with rapid changes in technology it is hard for practitioners and researchers to undertake the necessary testing in a timely manner such that they have a set of results that can be presented to the court to provide the required evidence of reliability. However, there have been test results for only 7[3] forensic tools in the Forensic String

[2]https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/federated-testing

[3] IPED v3.18.13 (May 2022)
Magnet Axiom Version 4.1.1.20153 (September 2020)
EnCase Version 8.09.00.192 (June 2020)
Access Data FTK Version 7.0.0.163 (April 2020)
BlackLight Version 2018-R4 (March 2020)
X-Ways Forensics Version 19.6-SR-4 x64 (March 2019)
Autopsy Version 4.6.0 (November 2018)

Search category (which is the closest category to this research) that were made available between November 2018 and February 2023.

Since 2010 NIST has been developing several ‘prototype’ datasets for tool testing as part of their Computer Forensic Reference Data Sets (CFReDS) project[4] but make the point that they are currently not designed for use in the Federated Testing project. In addition, the data set that represents the most complete environment (a PC and several removable media) is based around a Windows 7 PC image and so, as well as being an obsolete operating system, it contains many key document types that are also out of date.

It is therefore not viable to adapt current tool testing resources for evaluating the latest generation of forensic tools deploying remote agents.

1.4 REMOTE FORENSICS

The test scenarios previously mentioned involved a single instance of a tool running against a single data set. However, a more common real-life scenario often includes multiple systems. This is pointed out by Cohen et al., (Cohen, Bilby, & Caronni, 2011) who identify the problem encountered when undertaking DF investigations across an enterprise due to the number of remote machines (endpoints) that may be spread across a wide geographical area. They use the term ‘scalability’ in this context and suggest that it has several dimensions:

- The number of monitored assets.
- The number of forensic practitioners that can simultaneously investigate a single system.
- The number of objects (files, processes) that can be tracked.
- The average time to respond. Remote locations can mean that deploying trained investigators requires a long lead time.
- The average amount of time it takes to search all assets for indications of compromise.
- The variety and types of investigative operations that can be performed, such as keyword searches.

(Cohen et al., 2011)

[4] https://www.cfreds.nist.gov

In addition, tools that are employed in DF to locate and collect digital data have been struggling for many years to cope with a massive increase in data volumes (Irons & Lallie, 2014; Jusas, Birvinskas, & Gahramanov, 2017; Moser & Cohen, 2013). This issue has been compounded by other issues such as data being stored in the cloud (Adams, 2012; Mercuri, 2005; Peisert, Bishop, & Marzullo, 2008) and full disk encryption (Casey & Stellatos, 2008).

Thus, digital forensic processes have moved from being concerned purely with individual machines to having to deal with many machines that are all connected to the same infrastructure and at the same time coping with those machines storing ever more quantities of data.

Having recognised the problem of increasing data volumes and access to remote systems, Roussev and Golden (2004) made a case for remote DF. Their approach was based on remote processing from a central pool of data collected from the endpoints rather than processing at the endpoints themselves.

Several key developers of DF tools applied the concept of remote ‘worker’ processes for undertaking processing tasks (such as indexing) (Moser & Cohen, 2013) and, in line with the proposal of Roussev and Golden (2004), also based this around a central ‘coordinator’ machine.

In this approach, system files, ‘user’ files and emails of interest are identified by interrogating a central index that has been created from the digital data stored on the relevant systems (Attoe, 2016). The building of the central index is accomplished using ‘agents’ installed on all the endpoint machines that present their storage devices as network shares to a central computer running a process that interacts with the agents.

Figure 2 shows how this method is typically applied for installed agents. The central ‘examiner’ machine is employed with another machine that is the central storage point (although in some configurations they can be the same machine).

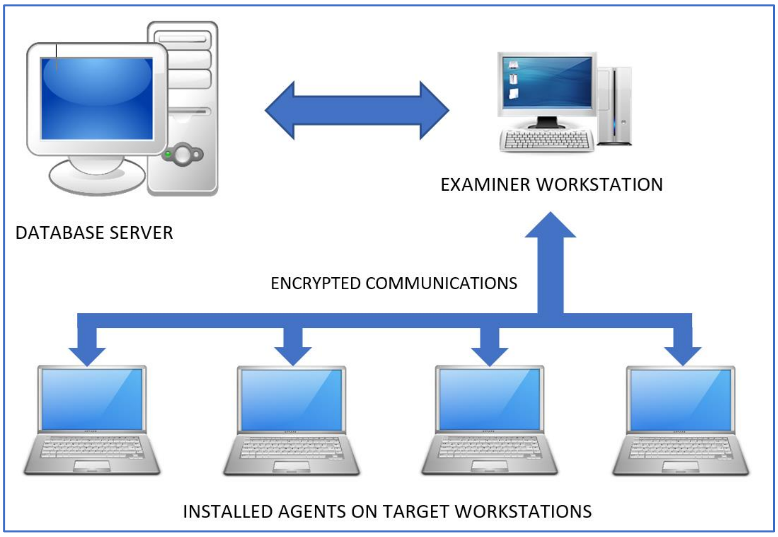


Figure 2 Example of an agent-based configuration (Edwards, 2019)

These agents create 'logical copies' of digital data (Attoe, 2016) and free investigators from the time-consuming collection processes of the past (Horsman, 2019). There are two types of these remote agent: those that must be installed on each computer and are controlled from a central processing point (Moser & Cohen, 2013), and agents that are deployed across the network without being installed but running in memory on each computer and sending back their results to a central point for review and further analysis (Adams, Mann, & Hobbs, 2017).

Depending on the type of deployed agent, the work is either carried out at a central point or at the endpoints themselves. In the first instance, the process of identifying the relevant data and collecting it relies on first creating an index of the data at the endpoints which can then be interrogated based on the required criteria before a command is issued to the installed agents to collect that data.

In the second approach, the non-installed agents are deployed with a set of instructions which then search the endpoint machines on which they are running (in memory) and then collect any relevant data back to a central point for analysis.

Time is usually an important factor to be considered during an investigation. This is not only in relation to the requirements of the litigation process but also the extent of the disruption caused by collecting data as part of that process (Adams, Hobbs, & Mann, 2013). Being able to automate a time-consuming activity has been of key interest to tool developers, practitioners, and the courts (Moser & Cohen, 2013; Pollitt, 2013).

1.5 Research Problem

Digital forensic practitioners need to be able to evaluate the tools that they use. This is not just because the courts require some measure of reliability when it comes to a determination of whether digital evidence being presented in court is admissible, but because there is typically a significant cost involved in the purchase of such tools and practitioners need to ensure that their limited budgets are used effectively to procure the best tool(s) for their needs.

Ideally, practitioners would like access to some form of reference data that compared the performance of all the available tools against a common data set that was kept up to date and relevant to the tasks undertaken by practitioners (Horsman & Lyle, 2021; Hughes & Karabiyik, 2020). This reference data would need to cover criteria other than functional performance, such as cost, ease of use, level of support etc. (Marshall, 2022; Mohiddin, Yalavarthi, & Kondragunta, 2019)

Unfortunately, despite calls for more research involving tool testing in DF (Casey, 2016; Göbel, Maltan, Türr, Baier, & Mann, 2022; Nikkel, 2014) and the existence of both testing frameworks and sample data (National Institute of Standards and Technology, 2020; Scientific Working Group on Digital Evidence, 2018) there are only a small number of published test results available and the data sets are typically out of date. In addition, the results focus on the ability of a tool to pass a series of tests and do not provide any information that would enable a practitioner to determine which of two tools achieving the same test results would be best suited for their needs, i.e., there is no form of evaluation undertaken.

There is another shortcoming in the NIST data which is that there is no method associated with collecting/processing data from multiple endpoints despite this becoming commonplace in digital forensic investigations.

Slay and Beckett recognise the "*high workload and low resource environment*" (Slay & Beckett, 2007, p. 1) in which digital forensic practitioners typically operate and point out that testing forensic tools is expensive and time consuming, meaning that when it comes to tool evaluation practitioners need a structured method to avoid wasting resources.

This issue leads to the statement of the Research Problem, which is that:

There is no defined method that enables a practitioner to undertake an evaluation of tools designed to collect data from multiple endpoints for digital forensic purposes.

This research is intended to address the Research Problem by providing such a method. This will simplify the evaluation of tools designed to collect data from endpoints for digital forensic purposes for the benefit of practitioners.

This will not only enable practitioners to provide evidence of their own tool testing as part of a reliability assessment but will also allow practitioners to evaluate and compare different tools prior to committing themselves to what is usually a significant financial outlay.

1.6 Research Aim, Questions and Objectives

1.6.1 Research aim

- To develop a method that will enable practitioners to evaluate remote agent-based forensic tools designed for use in DF investigations.

1.6.2 Research questions:

1. What components[5] are required of the method?
2. How can the identified components be combined into an effective artefact[6]?

1.6.3 Research objectives:

- Identify the necessary elements for evaluating remote agent-based tools for use in DF investigations.
- Identify or create one or more datasets for use in the evaluation of remote agent-based tools.
- Develop an artefact for evaluating remote agent-based tools for use in DF investigations.
- Identify an appropriate evaluation environment for use with the new artefact.
- Demonstrate the utility of the new artefact, data set and evaluation environment.

[5] The essential elements that need to be brought together and which help define the method.

[6] In the context of design science, an artefact refers to a purposefully created and designed object, system, or construct that embodies the outcomes of a design process.

1.7 RESEARCH SCOPE

There are many situations in which there is a need to undertake a search (and often a collection) of digital data. These situations can range from trying to locate a single lost document to identifying documents and emails meeting criteria that are related to some form of litigation process (eDiscovery). Rather than attempt to incorporate all facets of digital data search and collection, the scope of this research has been limited to DF because it involves activities that have the most stringent requirements for technical accuracy and reliability of results while involving the broadest range of digital data that includes system files as well as user-generated files.

DF focuses on the preservation and analysis of potential electronic evidence to a standard such that it can be used in a legal context (Kessler & Carlton, 2017; Marcella & Menendez, 2008; Scientific Working Group on Digital Evidence, 2014).

There are several other activities that involve the collection of digital data from multiple systems, such as incident response, electronic discovery, IT auditing and General Data Protection Regulation (GDPR) tasks. However, these tasks do not require a greater degree of reliability of the data and therefore, by focusing on DF requirements, the specific needs of these other tasks will be met and therefore they will not be considered separately in the narrative.

The scope of this research is also limited to the collection of data from computer systems running a version of the Microsoft Windows operating system given that in October 2022 it accounted for approximately 76% of all workstation operating systems (with OS X at around 19% and Linux at around 1.6%)[7]. However, the principles adopted for this research could be applied to other platforms.

Although forensic investigations may also involve data stored on cloud services these scenarios will not be included as part of this research.

[7] https://gs.statcounter.com/os-market-share/desktop/worldwide

1.8 OUTLINE OF THE THESIS

Chapter 1 - Introduction to the research

This chapter provides a summary of the DF environment and introduces the use of remote agents in these environments. The Research Problem is stated together with the Research Question, aims and Objectives.

The Research contribution section shows how this research fits into the broader field of DF research and states how this research will add to the body of knowledge.

The Research Scope comments on how DF requirements can be covered in the framework and describes how this research scope has been refined together with the reasoning behind this refinement.

Chapter 2 - Literature Review

This chapter provides a general review of current literature in relation to the location and collection of electronic artefacts required by digital forensic practitioners. This will lead to a review of previous research focussed on several of the key issues faced by these practitioners and various suggestions on how these might be overcome.

The background to remote agents and the characteristics that make them a suitable topic for research is expanded upon to put into context the research problem being addressed.

Chapter 3 – Methodology

This chapter identifies and justifies Design Science as the research methodology used in this research. It also identifies the Peffers et al. DSRP model that has been adopted and describes how it has been applied.

Chapter 4 – Artefact Design

This chapter incorporates the first part of the Design and Development stage of the DSR process model and covers the process in which the requirements that will form the basis of AFERA (the artefact for this research) are identified and the reasoning behind their selection.

Chapter 5 – Developing the Artefact

This chapter continues the ‘Design and Development’ stage of the DSR process model and builds upon the findings in previous chapters, the output from the Expert Practitioner Panel interviews and other authoritative resources to build the 'artefact' which, for this research, is AFERA for enabling digital forensic practitioners evaluate forensic tools that use remote agents.

Chapter 6 – Demonstration of the Framework

The DSRP model requires that prior to the Evaluation stage, a Demonstration stage needs to take place in which the artefact (in this case the Framework) is used to solve a problem in an appropriate context.

This activity helps to ensure that the Framework is practical, i.e., has utility. For the Demonstration stage a 'bench check' was undertaken using the Windows 11 and Office 365 Deployment Lab Kit as the operating environment and following the Framework's documentation.

Chapter 7 – Evaluation of the Framework

This chapter covers a central and critical part of design science research which is the evaluation of the design artefact, and the key aim of Information Systems (IS) design science research is 'utility'.

The main purpose of evaluation in DSR is to determine how well a designed artefact or achieves its expected environmental utility. A secondary purpose is to provide evidence that the theory leads to an artefact that has utility.

Chapter 8 – Conclusion

This chapter will discuss the outcomes from this research in relation to its stated aims and how it leads to future research opportunities. Its contribution to the field of DF will be stated.

A summary of the communications aspect of the selected methodology will be provided in relation to papers and other publications.

2 LITERATURE REVIEW

2.1 INTRODUCTION

This literature review starts by examining the literature relating to DF to place the research into context. Some of the fundamental attributes of DF are highlighted and discussed.

DF focuses on the preservation and analysis of potential electronic evidence to a standard such that it can be used in a legal context (Kessler & Carlton, 2017; Scientific Working Group on Digital Evidence, 2014).

Existing standards and guidelines relating to DF are identified that relate to the acquisition of digital evidence followed by a summary of the techniques that are being used, how they have evolved to cater for changes in technology and the environments in which they are applied, and how this has led to the development of remote forensic tools.

The need for some type of evaluation process that can be applied to tools used in a forensic environment is discussed, as well as the requirement for a suitable data set.

The chapter ends with a summary of the literature review leading to the problem being addressed in this research.

Scope

Because the technology involved in DF covers a wide spectrum it is necessary to focus on the most common situations and environments whilst being mindful to ensure that the development process for the creation of the artefact, the Framework, can be applied more broadly. In this research, networks based around the Microsoft Windows operating system are the intended environment in which the Framework will be applied.

Because the scope of this research limited to the collection of data from computer systems running a version of the Microsoft Windows operating system (accounting for approximately 76% of all workstation operating systems) literature relating to techniques and agents deployed in Macintosh and Linux networks has not been considered.

It is assumed that readers of this work will already be familiar with the digital forensic environment and therefore an in-depth description of standard practices has been avoided in the background section of this review.

2.2 MATERIAL COLLECTION AND MANAGEMENT

To identify the material relevant to the topic of this thesis the search facility of the Curtin Library Catalogue was employed (not restricted to any database) and was supplemented by Google Scholar and Academia.Edu resources.

The following search terms were used for both the Curtin Library Catalogue, Google Scholar, and Academia.Edu searches (specific databases included ProQuest, Scopus, ScienceDirect, ACM Digital Library, IEEE Xplore, SpringerLink and Taylor & Francis online):

'Digital AND forensics,' 'cyber AND forensics,' 'remote AND forensics,' 'triage AND forensics,' 'forensics' AND 'evaluating,' 'forensic test image,' 'forensic test data,' 'forensic datasets.'

EndNote X9 was used to manage the papers and other references with any non-PDF sources being converted to PDF (if appropriate) and incorporated into the EndNote library.

The references listed in papers identified using the above search terms were also reviewed.

The earliest date of publication was set at 2004 which was when Casey and Stanley published their paper entitled 'Tool review - remote forensic preservation and examination tools' (Casey & Stanley, 2004).

Based on the publication titles and abstracts the results for the first five search terms was reduced to 206 items. Although these results included papers that discussed the issue of effectiveness within a DF environment, a separate search for 'software effectiveness' was also undertaken through the same sources as for the previous DF and IR terms.

2.3 DIGITAL FORENSICS BACKGROUND

The aim for any branch of forensics is to present material that will be seen by a court of law as being reliable, i.e., it is of a standard whereby the court is happy to accept it as evidence. In general terms, evidence is *"…information that may be presented to a court such that it may decide on the probability of some facts asserted before it…"* (Hutton & Johnston, 2000).

The field of DF grew out of law enforcement and security agencies as they started to see more investigations involving computer systems and digital storage, thus much of the material they wished to present in court was in electronic format. McKemmish (1999) describes 'Forensic Computing' as "*…the process of identifying, preserving, analysing and presenting digital evidence in* *a manner that is legally acceptable*". The use of the term 'digital evidence' is useful as it captures all manner of electronic data that may take many forms and come from a variety of devices, from satellite navigation systems to fridges.

Collecting and examining electronic data in such a way that it could be presented in a court of law was a new concept before the use of computers became widespread. Prior to the advent of electronic records, the Common Law rule of 'Best Evidence' was the most widespread guide for admissibility of material in court, but this had been developed in the 18th Century when the material was in a physical form, such as paper records (Mason, 2007).

Electronic records being produced as the output of the activities of a digital forensic practitioner are copies of other records. For these copies to be presented as evidence many jurisdictions around the world have adopted some form of legislation that allows for a 'copy' of a record to be treated in the same way as the 'original,' for instance the Uniform Evidence Act in Australia (Argy, 2006).

2.3.1 Reliability

Producing electronic copies of material that may be used in evidence requires that material to meet the requirements of 'reliability.' Courts take the same stance with digital material as for any other type of material, such as fingerprints, placed before them as potential evidence. They will always focus on the way in which material presented to them has been collected and subsequently handled (Cohen, 2011). As such, they require evidence of some form of empirical testing of both the tools and techniques employed to produce the material (Horsman, 2019).

The issue of determining 'reliability' from a court's perspective is consistent across jurisdictions in relation to electronic copies of data, such that those copies of data may be considered admissible (i.e., accepted as evidence) if they meet three conditions, namely:

- "They were from the indicated source,
- they were acquired using proven tools and techniques, and
- they have not been altered since the time of acquisition."

(Steel, 2006)

Steel's first condition can be met through comprehensive documentation that may include photographs and diagrams as well as a record of equipment serial numbers and other identifying information.

In relation to Steel’s second condition, because DF lacks the same level of standards for tools and techniques that are found in more established areas of forensics, courts will often resort to the Daubert Test (Taipale, 2019). This test originated in the American court system, but the principles have been adopted in many other jurisdictions in a similar format. The test is named after the Daubert v Merrell Dow Pharmaceuticals case (Supreme Court of the United States, 2003) where a trial judge set out a framework for assessing ‘innovative or unusual scientific’ material which is often referred to as the judge having the role of ‘gatekeeper’ (Kessler & Carlton, 2017).

The fundamental elements of the Daubert Test are:

- “Whether the theory or technique in question can be and has been tested.
- Whether it has been subjected to peer review and publication.
- Its known potential rate of error along with the existence and maintenance of standards controlling the technique’s operation.
- The degree of acceptance within the relevant scientific community”

(Stephenson, 2003)

In practice, Steel’s final condition can be addressed using hash digests. One of the key functions built into many forensic image file formats is the ability to store a hash value, which is often referred to as a ‘digital fingerprint’ (Nichols, 2021). The hash value is calculated using one of several mathematical algorithms at the time of image creation and serves to identify the collection of data that comprise the forensic image. Identical hash values enable a digital forensic practitioner to validate, with a particular degree of confidence (depending on the type of hash used), that the processed and analysed data did not change from the time it was acquired from the original source (Horsman, 2019).

This research will provide empirical evidence that the technique of using remote agents to collect electronic data can be, and has been, tested using the framework that will be developed.

2.3.2 From Computer Forensics to Digital Forensics

With the growing demand to store data in digital form Chernyshev et al. (Chernyshev, Zeadally, Baig, & Woodward, 2017) recognise that DF has grown from being involved with desktop computer systems and laptops where it was more common to use the expression ‘computer forensics’. DF now incorporates a range of what they describe as being ‘sub disciplines.’ Of the sub-disciplines they mention, three are associated with hardware, namely computer forensics, network forensics and mobile forensics. Other sub-disciplines exist that are associated with storage structures, such as database forensics.

The use of the term ‘digital evidence’ in McKemmish’s definition of Forensic Computing (McKemmish, 1999) is also useful as it captures all manner of electronic data that may take many forms and come from a variety of devices, from satellite navigation systems to fridges. As Dell (Dell, 2018) notes, any device that can be accessed via the internet becomes a security risk which brings it into the realm of DF.

2.3.3 Outdated theories

Casey (2017) and some other members of the digital forensics community (Harichandran, Breitinger, Baggili, & Marrington, 2016; Horsman, 2020; Jones & Vidalis, 2019), suggest that the reason that here is little practical guidance for practitioners in relation to collecting data from remote sources is because current practices are based on outdated theories. Casey also comments that it is not possible to standardise practices due to the unique circumstances of each investigation and the continuing development of devices that contain digital data(Hosseinian - Far et al., 2020)

Horsman (2020) supports the contention of this research regarding a lack of guidance for remote data search and collection when he argues that the ACPO Principles need reviewing on the basis that they have remained essentially the same since being introduced in 1998. In addition to specifically mentioning the need to address tool validation and quality assurance Horsman suggests updating the principles to also consider privacy issues as well as recognising the fact that although some processes can be re-run, they will always return different results, such as with volatile memory capture.

This call for processes and guidelines to be updated is also supported by Kessler and Carlton (2017) who list four ‘myths’ that have been handed down from the early days of DF in relation to the practice of acquiring digital data in a ‘forensic’ manner. These myths are:

Myth 1 Never image (i.e., forensically copy) a running system.

Myth 2 Always use a write-blocker.

Myth 3 The destination media for the forensic copy must be sanitized prior to writing the files.

Myth 4 Hash algorithms are the proof of the integrity of a forensic copy.

Myth 1 requires that a target system must always be shut down before the creation of a forensic image. This is simply not practical in many situations such as critical file servers, cloud systems or systems with encrypted file system data where the appropriate password is not to hand.

However, in cases in which the machine holding the digital storage to be imaged has been shut down, there are two main approaches for creating a physical forensic image in such a way as to comply with Myth 2. The first is to remove the storage media and connect it via a hardware device (often referred to as a 'hardware write-blocker') that would prevent any alteration of the original data while it was copied into a special file that could be read by one of the forensic tools (Sankardas, Yan, & Lavenia, 2019).

The second approach that conforms to Myth 2 is to boot the computer using a boot CDROM or USB with a modified operating system (typically Linux-based) (Mohamed, Marrington, Iqbal, & Baggili, 2014a) which would by default prevent any data being written to attached storage devices (known as a type of 'software write-blocker') unlike a typical operating system that would automatically try to mount any file system found on a storage device, enable writing to that device and, in the case of MS Windows, actually write 'housekeeping' data to it.

In relation to Myth 3, this was only a valid requirement in the early days when the forensic copy of the data was written directly to another device. In these situations, it would be easy to co-mingle previous data with the forensic image data due to the way that filesystems store their data. This is since when a file is deleted, or a disk is re-formatted the process just enables the re-use of the storage locations. This means that any existing data will be left on the device and any new data may only overwrite portions of the storage area.

Fortunately, digital data acquisition now involves the creation of special 'containers' that separate the forensic image data from the file system data on which they are stored so the requirement to wipe the forensic image storage medium is no longer necessary, and as Kessler and Carlton (2017) point out, in the case where forensic images are copied onto large network storage devices this would not be possible.

In relation to the fourth Myth, although hashes are commonly used in digital forensics to identify a string of data, such as a file, Kessler and Carlton (2017) point out that it is technically incorrect to state that a hash value is a unique identifier of a file due to a number of cases in which mathematicians have been able to falsify this claim (although this occurred in very specific circumstances). However, this does not reduce the value of the hash process but requires practitioners to quantify the reliability of the hash being used (as there are numerous different types of varying complexity). This might take the form of making a statement along the lines of "it is computationally infeasible for a file's contents to be altered while retaining the same hash value" (or by using more than one hash algorithm).

2.4 The Forensic data acquisition process and its challenges

The motivation for this research came about due to first-hand experience with the challenges of undertaking DF engagements in modern environments. These challenges are identified in the following sub-headings.

2.4.1 Remote data challenges

A typical enterprise will own many computer systems, from laptops and desktops to large, dedicated servers. Any of these systems could contain potential evidence in relation to an investigation, whether that be associated with some internal criminal activity, external attacks on the organisation or some other matter and would traditionally require a trained person to be physically present at each device to acquire potential evidence (Mark Scanlon, 2017).

When faced with potentially hundreds, if not thousands, of computers there is a problem with answering the question "does this computer contain information relevant to the digital investigation?" (Casey, 2013). Typically, practitioners have been forced to identify a subset of computers that will be analysed, but there are limits as to how many an individual practitioner can handle when following the traditional image-analyse approach.

Although theoretically being part of one network, there are numerous practical obstacles to overcome when seeking to acquire potential evidence from networked machines. In large organisations there can be many systems spread over a wide geographical area that may include systems located overseas (Quick & Choo, 2014). In the past, organisations providing DF services have had to coordinate the activities of teams of practitioners, potentially across different time zones. For many types of investigation, a piecemeal approach would not be appropriate given the potential for evidence to be deleted once knowledge of the investigation becomes known, which would be the case if all computers were not being processed at the same time.

The advent of the cloud takes the remote data challenge to another level (Dykstra & Sherman, 2012; Ruan & Global, 2013). Not only may the exact location of the data be unknown, but physical access may not be possible, e.g., when located in a secure datacentre not owned by the organisation (Adams, 2012). Even if physical access can be obtained, this will tie up resources because cloud computers often run as virtual machines on mainframe computers such that there is not a single 'box' associated with a particular cloud server or workstation and the host computer may well be the host to virtual machines for other businesses that are not part of the investigation and so cannot be taken offline.

For a DF practitioner, the key challenges associated with remote data comes down to two factors:

- The number of systems to be processed and
- The ability to access those systems.

Other factors also affect the ability of practitioners to process remote systems, such as the bandwidth of the connection to the remote systems, the resources available for processing, e.g., the amount of RAM allocated to the remote systems and the need to have minimal impact on the user(s) of the remote systems.

The increase in data volumes as described in the previous section also comes into effect and compounds the problem of dealing with remote systems.

A common feature of many forensic tools is the ability to undertake 'data carving'. However, this normally takes place on a forensic image because of the practical difficulties associated with carrying out this process on a 'live' system. These difficulties include:

- Data Overwriting: In a live system, data is constantly being read from and written to storage devices which can lead to the overwriting of deleted data blocks, making it difficult to recover the original content.
- Fragmentation: Deleted files are often fragmented with their data scattered across different locations on the storage device. Recovering and reconstructing fragmented files can be very resource-intensive on a remote system and the final result could be incomplete or corrupted data.
- Metadata Loss: When files are deleted, their metadata (such as file name, size, and timestamps) may also be overwritten. This can make it challenging to identify and properly attribute recovered data which can limit the value of any recovered data.

- Data Integrity: The live system's ongoing operations can affect the integrity of the data being carved. For example, a file being actively modified or deleted during the carving process can result in incomplete or corrupted data recovery.
- Performance Impact: Data carving from a live system can consume system resources and affect system performance, especially if it involves intensive disk I/O operations. This impact may disrupt the user of the live system.

2.4.2 Legal challenges

Although this research is focussed on specific technical challenges encountered by DF practitioners there is also a legal issue that relates to the search and seizure of potential digital evidence. This issue relates to the privacy of the entities that own the data that may be seized as part of an investigation and centres around the taking and accessing of data that is not covered by a search warrant or other legal document (Hong, Yu, Lee, & Lee, 2013; Woods, 2019).

Many jurisdictions around the world have some form of legislation that places limitations on what material can be taken as potential evidence. These limitations will often take the form of requiring some form of confirmation that data being seized is potentially relevant to an investigation and thus preventing devices being seized on mere suspicion (Stephen, 2012).

This creates problems when considering acquiring a forensic image of a device that has data from one or more legal entities that are not covered by a search warrant, e.g., on the file server used by a team of lawyers or accountants.

2.5 STRATEGIES FOR FORENSIC DATA ACQUISITION

This section of the literature review will identify research addressing the challenges for locating and acquiring digital data caused by a combination of storage capacity increases and extensive remote systems on networks (including the cloud).

The previous research can be split into two main topics. The first topic addresses ways of reducing the amount of data to be collected and so is relevant to this research. The second topic involves ways of handling large volumes of data that have already been collected, which is not relevant to this research and therefore not included in the literature review.

2.5.1 The move from physical to logical processing

The biggest change in the approach adopted by forensic practitioners with respect to acquiring digital data has been the increased use of techniques and processes for acquiring the data from 'live' systems.

As indicated earlier, the developments in information technology since the early days of DF have meant that a practitioner is no longer able to work in an 'ideal' environment but must make decisions based on the merits of the various acquisition options in the context of their investigation (Hosseinian - Far et al., 2020).

Shutting down and removing the data storage for processing or re-booting with a specialist operating system described in earlier paragraphs can be applied in some cases, but there are situations in which they are not appropriate. Originally these situations might have involved file servers or other devices that could not be shut down. In these cases, a 'forensic copy' (often called a 'logical copy') of the data would be acquired (Hosseinian - Far et al., 2020).

Logical copies of data are typically acquired in situations in which the device holding the digital storage could not be physically accessed (such as a laptop in a remote location) (Adams, 2012; Jones & Vidalis, 2019; Mohite & Ardhapurkar, 2015; Roussev, Ahmed, Barreto, McCulley, & Shanmughan, 2016) and so the required data might be copied across a network (Mark Scanlon, 2017).

With the growth in the number of file servers, storage volumes, networks and particularly the cloud, the live acquisition approach is becoming the preferred option in many cases rather than the traditional bit-stream images (Harichandran et al., 2016).

Other reasons to undertake a live acquisition approach on computer systems involve capturing data stored in volatile memory (Yang et al., 2017) and the potential to encounter full disk encryption (Hosseinian - Far et al., 2020). In addition, current procedures, such as imaging with hardware write-blockers, are very inefficient when dealing with RAID storage. Each disk must be removed from the array, write-blocked and imaged as individual disks which then must undergo complex RAID reconstruction from the individual images.

The most significant blow to the requirement to obtain a forensic (physical) image is around mobile devices, where it is not possible in many cases to directly access the storage and there is no facility for write-blocking. Worst still, from a 'traditional' DF approach, the way that the data is accessed may require changing the operating system of the mobile device for one that has been altered to allow data to be copied. This is potentially an irreversible modification to the device (Chernyshev et al., 2017).

In addition to the concept of a logical copy of digital data (Stanfield, 2009), McKemmish made allowance for those situations in which a physical image could not be created as part of his definition of forensic computing (McKemmish, 1999). This is also reflected in the ACPO Guidelines (Association of Chief Police Officers, 2012) and more recently in Horseman's proposed changes to those guidelines (Horsman, 2020).

The move away from routinely acquiring all the data on a storage device that, depending on the type of investigation, includes irrelevant operating system artefacts and file types, already provides a significant reduction in data volume (round 20GB for Windows 10 and at least 32GB for Server 2019 based on Microsoft requirements). Further reductions rely on identifying a sub-set of the remaining data that is deemed 'useful' or 'relevant,' and this is covered in the next section.

2.5.2 The concept of triage

It is often the case that digital forensic practitioners are restricted by some form of court order that prevents them seizing material without being able to show that it is relevant to their investigation. Coupled with the increase in data volumes and prevalence of digital storage devices together with the acceptance of 'live' data processing this has led to the adoption of a process that determines if the data contained on a device meets some criteria for being relevant to the investigation (Casey, 2013; Quick & Choo, 2014).

This process is known as 'triage,' which Koopmans (2010) defines as *"a process of sorting computer systems into groups, based on the amount of relevant information or evidence found on these computer systems".*

Moser and Cohen (2013) adopt a simple approach to triage and identify three groups of digital evidence using assumptions based on the location of the evidence, namely:

- The system is likely to contain crucial evidence, but it is unlikely that this evidence is in immediate danger of being destroyed.
- The system is likely to contain crucial evidence and there is an imminent danger that it may be destroyed.
- The system is not likely to contain relevant evidence and therefore can be ignored.

This classification reduces the volume of data being collected by either ignoring the contents of certain storage devices or putting off collection to a later date where the data is secure.

Pollitt (2013) points out that the concept of triage can include the examination stage as well as the acquisition stage and many collection tools already can filter data for collection based on a high-level examination of the data in situ that can include, among other criteria, the presence of certain terms (often called “keywords”).

Casey (2013) identifies issues of privacy and evidential integrity by the adoption of this type of “peek and seek” approach if there is no clear legal authority to undertake the activity. This concern is compounded by the fact that it is often ‘uninformed’ staff that undertake the task without having a clear understanding of the needs of the investigation and are therefore poorly placed to identify potentially relevant data (Pollitt, 2013). However, Hong et al. (2013) regard the triage process as having the benefit, when properly applied, of limiting the search and collection process from a privacy perspective without missing potentially relevant material needed by, in their example, law enforcement.

Shaw and Browne (2013) suggest an approach that automates data identification and collection to reduce the type of risk, highlighted by Casey (2013) of an inexperienced person missing potential evidence. However, they acknowledge that their approach is not suitable for use in the field and is designed primarily to enable less experienced staff to undertake some of the initial preparation of digital data that has already been seized. The approach involves booting a system from a forensic boot CD (CAINE in their case) and selecting various options that locate specific artefacts, such as URLs and ‘chat’ logs, as part of an ‘enhanced previewing’ process. They also allow for the use of keywords to identify potentially relevant data, but these keywords need to be verified as being suitable by an investigator who is familiar with the case.

The output of Shaw and Browne’s (2013) enhanced preview process is fed to an investigator as a small sub-set of the volume of data held on the storage device that has been processed. This means that the investigator does not have to spend time processing a large amount of data only to find that there is nothing there of relevance.

Dell Inc. took a different approach to eliminating the collection of data from devices that held nothing of relevance. In 2009 they launched a solution for DF that utilizes multiple instances of specialist hardware onsite and involves having a central data repository for storing and processing collected data which can then be shared by several investigators. The solution includes proprietary software together with the hardware in the form of touch-screen workstations. The workstations are used to create forensic images of systems but with the ability to preview the data onsite before the image has been fully created. It is possible to run searches and preconfigured filters to locate items that may hold relevant data (Dell Inc., 2009).

The idea is that at some point a decision is made on whether to continue the imaging process on a particular device or abandon it if no data of relevance has been located at that time. Those images that have been identified to contain relevant data are completed and then taken back to the lab and uploaded onto the central data repository for processing. This means the processing power of the machines in the lab can be used to best effect, i.e., they are not being used to process 'useless' data, and the low-performance (and cheaper) workstations can be deployed to the field where they have enough resources to undertake the triage process.

The concept of previewing data prior to collection is also supported by Rogers et al.(Rogers, Goldman, Mislan, Wedge, & Debrota, 2006) who focus on the time saved by adopting a 'high-level' approach to reviewing digital data held on a device to determine its relevance to an investigation. Their Computer Forensics Field Triage Model (CFFTM) was developed as a way of formalizing an investigative approach they had observed being used by a US Attorney's office and was validated by 20 State and local law enforcement officers. This time saving element of the triage approach is also supported by others (Casey, 2013)

Overill et al. (Overill, Silomon, & Roscoe, 2013) consider triage to include three distinct activities:

1. Pre-seizure – whereby the physical devices covered by the warrant and likely to be relevant to the investigation are identified and listed to create a 'shopping list'.
2. Search for the physical items on the 'shopping list' and seize them. This may also include portable devices attached to previously identified machines as well as data, such as logs, extracted from servers and other devices rather than the physical devices holding that data.
3. Post-seizure – undertake an examination of the data held on the seized devices to identify the existence (or otherwise) of data relevant to the investigation.

Overill et al.et al.et al. (2013) proposes a set of templates for different investigative scenarios that balance the potential value of data found on devices against the cost of the resources needed to obtain that data. The intention is to prioritise the processing of devices to achieve the most beneficial outcome for the least resource cost and thereby improve the efficiency of the triage activity. Overill et al.et al.et al. explain this as “front-loading probative value and back-loading resource utilisation” (Overill et al., 2013, p. 173)

An alternative to the initial ‘shopping list’ of devices or filtering approach is to index the data on the first processing pass. The index can then be interrogated to identify those data sources containing data of potential significance so that a more targeted collection can be made.

2.5.3 Triage in practice

A typical solution for triaging individual machines is to employ a specially configured boot device (usually Linux-based) which has a set of tools designed to identify the type of information that may be relevant to a particular type of case (Pollitt, 2013). An example of this is the Triage Investigator application from ADF Solutions that enables the creation of a boot device to be used on individual machines and the ability to select a ‘search profile’ (ADF Solutions, 2020).

Other similar non-commercial tools designed to assist in collection from individual ‘live’ systems have included **Windows Forensic Toolchest**[8] (last updated in 2014) and **Computer Online Forensic Evidence Extractor (COFEE)** (developed by Microsoft for law enforcement) but these seem to have been superseded in favour of various non-commercial boot environments that come with a set of utilities for undertaking examinations and investigations. Examples of this type of resource include:

Computer Aided Investigative Environment (CAINE)

CAINE is an open-source Linux platform that incorporates software tools and scripts within a graphical environment. As well as being a boot disk it is also possible to install CAINE on a workstation for use in lab analysis.

Kali Linux (Forensics Mode)

Kali Linux is an open-source Linux boot environment that comes pre-loaded with popular forensic tools for imaging and analysis.

BackBox Linux

[8] http://www.foolmoon.net/security/wft/index.html

BackBox Linux is built on Ubuntu and primarily designed to assist with a penetration testing and security assessment, although it also includes tools for use in forensic analysis.

WinFE

The most recent version of WinFE is a configurable Windows 10 boot disk that is based on the Windows Preinstallation Environment (WINPE). It is modified for forensic use and includes a range of tools.

2.5.4 Issues with boot disks

Some concerns have been raised regarding the use of forensic boot disks, especially if being used by inexperienced personnel (M. Scanlon & Kechadi, 2010). These concerns include:

- Non-functioning hardware.
- Passwords required for booting systems.
- Missing up to date drivers to support the newest hardware, especially those based on Linux distributions.
- DOS and Linux interfaces and systems being unfamiliar to a vast majority of forensic examiners.

Although booting individual systems using ‘forensic’ boot disks or running batch files to locate specific data was initially considered a viable solution, data storage is now so widespread that investigators often face the challenge of having to seize data from geographically spread servers/storage that cannot be physically accessed and large numbers of machines for each investigation (Mohamed, Marrington, Iqbal, & Baggili, 2014b).

Some concern has also been expressed regarding the forensic soundness of boot disks. Mohamed et al. (Mohamed et al., 2014b) undertook testing of three Linux distributions to identify what data, if any, was changed (or could be changed) through their use. Their results showed that some data was changed but this was limited to altered timestamps on a few system files. Critically however, no files were added to the disks being examined although Mohamed et al. (2014b) point out that it was possible to write data to the disk being examined when using the Knoppix boot disk.

While there are many solutions for forensically acquiring data from individual systems that provide the practitioner the ability to reduce the amount of data collected/examined using a triage approach, these require a large degree of practitioner involvement and crucially they do not scale up to processing multiple systems across a network. This challenge is covered in the next section.

2.5.5 TRIAGE - multiple systems

Notwithstanding the problems caused due to the volumes of data on individual devices, investigations involving networked systems (including the cloud) are a common occurrence and a challenge for investigators (Hosseinian - Far et al., 2020; Koopmans & James, 2013).

Roussev and Richard (2004) argued the case for remote digital forensic (DDF) tools at the DF Research Workshop held in 2004. They first identify the requirements for DDF software and then propose a framework designed to meet those requirements. However, their objective is to increase processing capacity and their approach is to use the co-ordinated processing power of multiple machines in the lab against data that has already been captured.

In relation to onsite deployment, O’Connor (2004) proposed a method for mass deployment of specialised boot media for triage purposes to address issues such as those described by Koopmans and James (2013). Their method would enable processing of numerous network systems via those systems’ network connections, rather than having to attend each system individually and use local media. His proposal utilised the Pre-boot Execution Environment (PXE) network boot function found in Intel firmware which became part of the Unified Extensible Firmware Interface (UEFI) standard in 2015.

On start-up, a client booting PXE identifies a PXE DHCP server and receives the necessary configuration information to join the PXE environment and download the boot resources. It therefore requires nothing other than a network connection to be able to boot into whatever operating system is required. For the purposes of triage, O’Conner (2004) uses the Knoppix distribution as an example, although he does acknowledge that Knoppix was not designed for forensic purposes. However, any suitable boot image can be used in this scenario.

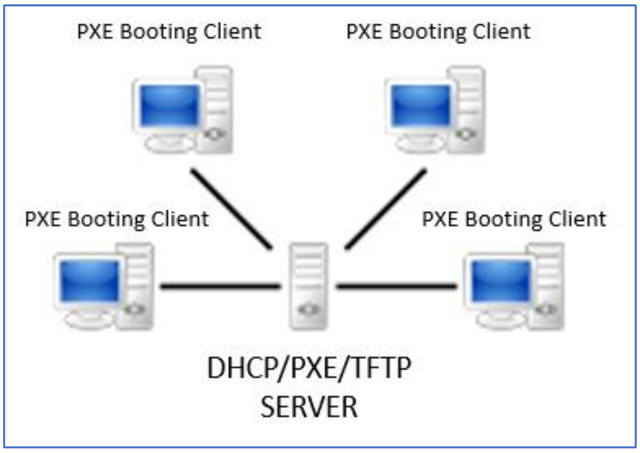


Figure 3 Pre-boot Execution Environment

While the process described by O'Conner (2004) does enable multiple systems to be booted with forensic tools available for triage purposes, it does also necessitate taking the client machines offline. In addition, there is limited capacity to automate the triage process which requires the creation of customised scripts and the setting up of a storage location for results.

Another remote forensics method, while not using the term 'triage,' was put forward by Gao et al. (2004) who proposed their Bluepipe Architecture to enable investigators to undertake "*on-the-spot DF*" which would provide, among other things, the ability to handle "*…an increased scale of investigative targets*". The approach relies on a client-server environment in which the Bluepipe Server process runs on the target machines while the Bluepipe Client runs on an investigators machine. Scalability is achieved using a proxy server that would allow search queries to be sent to multiple targets as shown in Figure 4. The proxy server would also allow remote investigators to process the target machines.

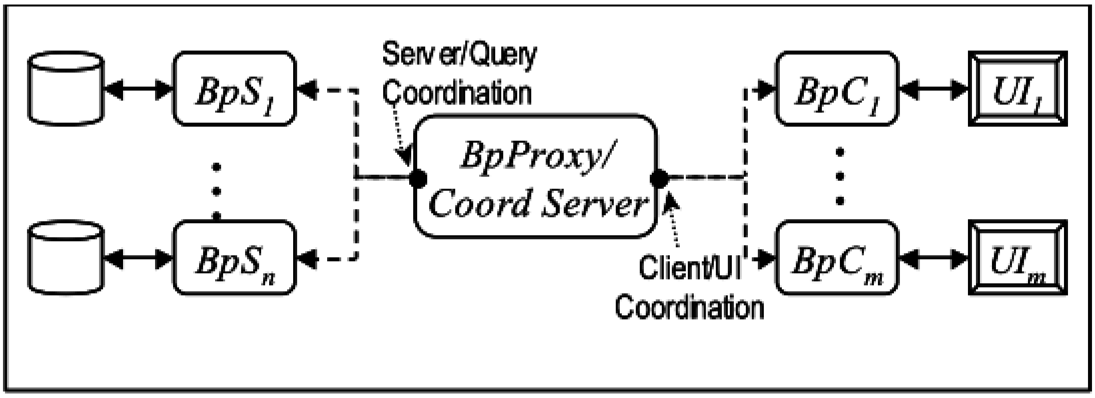


Figure 4 Bluepipe Architecture (Gao et al. 2004)

However, the Bluepipe Architecture (in the same fashion as that suggested by O'Connor (2004)) requires that the target machines be booted using a Linux distribution and so must be taken offline.

An alternative approach was proposed by Scanlon and Kechadi (2010) who suggest that rather than waste the valuable resource of a forensic practitioner going on site to undertake the imaging process, an automated client/server architecture could be deployed using a non-specialist law enforcement officer. This approach utilises an Ubuntu boot disk with a simple user interface that allows the selection of a device (or devices) for imaging using the DCFLDD tool. The data is then sent via the internet to the server where it is accessed and processed by the trained digital forensic staff.

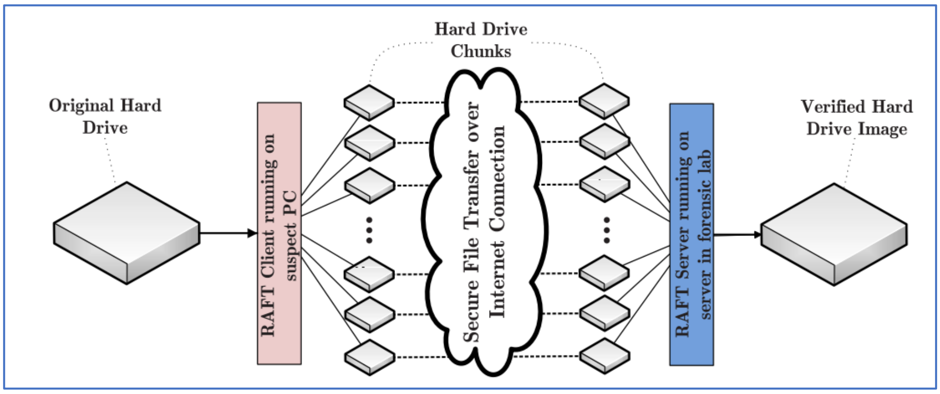

Hard Drive Chunks
Original Hard Drive
RAFT Client running on suspect PC
Secure File Transfer over Internet Connection
RAFT Server running on server in forensic lab
Verified Hard Drive Image

Figure 5 Overview of RAFT Imaging Architecture

Figure 5 shows how the RAFT system would be set up. Scanlon and Kechadi (2010) suggest that the system allows for multiple disk images to be collected in parallel and makes the best use of the time available for the trained digital forensic specialists. However, there are many difficulties with the proposed architecture. Not only are there likely to be technical issues (mentioned in a later section), which an untrained officer will be unable to address (not considered an issue by the authors), but the volume of data being transmitted across the internet would tie up an officer onsite for a considerable amount of time – far more than if the target drives were imaged locally. Although Scanlon and Kechadi planned to allow for live imaging (which would have reduced the data load) this was never developed.

The concept of having untrained officers undertake the data collection is supported by Hitchcock et al. (2016). However, while this proposal could potentially save some skilled practitioner time at the expense of tying up an 'ordinary' law enforcement officer it does not directly address large data volumes or remote data issues.

Typical onsite scenarios are referenced by Koopmans and James (2013) who cite cases handled by the Dutch Police that consisted of networks ranging from 8 to 45 computers. Although these cases were associated with law enforcement, they are also relevant to commercial enterprises (Wilkinson, 2019). Koopmans and James (2013) used special boot media but ran up against several problems that included defective CD players and systems that could not be made to boot from this media. These types of issues would be beyond the knowledge and experience of non-specialist officers.

Using their boot media in the 45-computer case, Koopmans and James (2013) had to address each of the systems in turn and noted that it was not until they had processed one of the few remaining systems that they located data that was relevant to their case. Had their warrant imposed a time restriction on their activities on site it is possible that this information could have been missed.

The time constraint issue is not limited to law enforcement. For instance, Anton Piller orders (used in English and many English-derived legal systems) are court orders that allow one party to enter another party's premises and undertake a search for items that might be used in evidence (including data contained on computer systems). These court orders typically contain a specific period during which the search can be carried out on a particular day.

Koopmans and James built on previous work (Acker, 2007; O'Connor, 2004; Robotti, 2009; M. Scanlon & Kechadi, 2010) for their 'Automated Network Triage' (ANT) proposal (Koopmans & James, 2013). Like O'Connor's (2004) approach, they also use Intel's PXE environment to boot a Knoppix image and run customised scripts that might involve keyword searches or identifying specific file types for collection.

Their approach requires the setting up of a private network using the organisation's infrastructure and a dedicated Linux server that will collect the results of the endpoint processing (Figure 6).

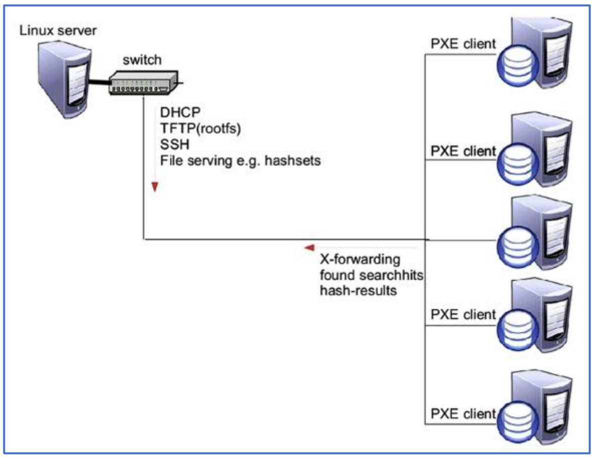


Figure 6 Automated Network Triage (ANT) configuration

While theoretically this approach has some merit, it is unlikely to be workable in practice for the same reasons identified for O'Connor's (2004) solution.

2.5.6 Criticism of triage approaches

Sondhi and Arora (2014) support the practice of indexing remote data based on the practices common in electronic discovery (e-Discovery), which they define as *"…a domain that involves detection and extraction of evidence from Electronically Stored Information (ESI) for civil litigation and other investigations"*. They point out that e-Discovery practitioners also face a problem with processing large datasets in a legal environment and have adopted solutions that migrate well to other domains facing the same issue, one of which is indexing the source data.

The idea behind the indexing of target machines is that instead of imaging or collecting all the data, an index of a system's contents is created. While this might take some time to create, once the index is built it is capable of being searched very rapidly. Using a keyword search as an example, all the systems containing one or more occurrences of the keyword can be identified and the relevant files collected through a separate process.

However, Roussev and Richard (2004) ask "Is it worth indexing the target?", and make the following points:

1. The effort of creating large indices is a waste of time given that regular expression searches can be completed in a fraction of the time it takes to build an index, of which only a small sub-set will contain any worthwhile data.
2. The investigator can be reviewing partial results as they arrive rather than having to wait for the indexing process to complete.

In a later paper, Richard and Roussev (2006) state that in some situations indexing can be useful as part of the pre-processing of a forensic image but the limited resources of individual workstations meant that it might take several days to complete. They also go on to identify other key weaknesses of the indexing approach in that it cannot accommodate regular expression searches or character strings not in the index. Commercial tools get around the workstation resources issue by having dedicated servers for creating and storing the indexes.

Adams et al. (2017) expand on the limitations of indexing, pointing out that because indexing engines can only process file types that they recognise (based on file extension) which means that not only can they miss unknown file types they can be fooled by incorrect file extensions (potentially due to malicious intent). They also identify filtering applied to the terms being indexed due to the length of the character string and the exclusion of 'noise' words as a way of reducing the size of the index.

Pollitt (2013) is critical of both the automated previewing of the potential evidence using some software with pre-set criteria and the physical "eyeball" approach whereby a human determines what is relevant. For the software approach he suggests that there is a lack of a consistent methodology, and the technique can be too restrictive and thereby miss relevant data. On the other hand, while he submits that same lack of methodology also applies to the 'human-centred' approach Pollitt (2013) also points out the risk that the human undertaking the decision making may not be an effective determiner of the relevance of the data they are examining.

According to Shaw and Browne (2013), the term 'DF triage' is poorly understood and they comment that the ACPO Guidelines are not clear on their advice with respect to this practice. For example, Horsman et al. (Horsman, Laing, & Vickers, 2014) use the term "High-level Triage" to mean prioritising physical items to be examined as part of an investigation while the examination of data on a device they call "Device Triage" but point out that they can both be regarded as 'triage'.

In a similar vein, Koopmans and James (2013) use the terms "physical triage" for the identification of physical devices likely to contain relevant information and "digital triage" for the examination of data stored on a device.

In support of Pollitt (2013), Shaw and Browne (2013) also identify several risks associated with current practice. Some of these risks they say are associated with processing and analysis, but in terms of the acquisition aspect of triage they suggest that there is a danger that evidence could be missed either through the lack of skills of the person undertaking the triage or because of some technical issue. A technical issue could be the inability of whatever tool is being used to access encrypted data or other files that are typically locked by the operating system.

The use of less experienced personnel to undertake the identification and collection process must also rely on suitable guides that provide detailed instructions to compensate for lack of experience but, as Horsman (2020) points out, these guides quickly become outdated due to the rapid advances in technology.

2.5.7 Remote agents

Moving on from the approach of having networked systems taken offline to be booted via a ‘forensic’ boot process, several key developers of DF tools applied the concept of remote ‘worker’ processes for undertaking processing tasks (such as indexing). This builds on the concepts introduced by Roussev and Richard[9] (2004), which are supported by other researchers (Beebe, 2009; Garfinkel, Farrell, Roussev, & Dinolt, 2009; Lillis, Becker, O’Sullivan, & Scanlon, 2016).

The critical difference between Roussev and Richard’s (2004) approach and that adopted by later developers however is that in the later approach the activities take place on ‘live’ systems. DF tool designers have adopted the remote identification and collection process using agents (Dykstra & Sherman, 2012; Moser & Cohen, 2013).

These agents create ‘logical copies’ of ESI (Attoe, 2016) and free investigators from the time-consuming collection processes of the past (Golden & Roussev, 2006).

Kebande and Venter (2018) promote the use of what they call an Agent-Based Solution (ABS) to address the challenge of collecting data from systems in the cloud in a forensic manner.

There are two types of these remote agent; those that must be installed on each computer and are controlled from a central processing point such as GRR (Moser & Cohen, 2013) and agents that are deployed across the network without being installed but running in memory on each computer and sending back their results to a central point for review and further analysis (Adams et al., 2017).

Time is always an important factor to be considered in DF, not only in relation to the requirements of the litigation process but also the extent of the disruption caused by collecting data as part of that process (Adams, Hobbs, & Mann, 2013). Being able to automate a time-consuming activity has been of key interest to tool developers, practitioners, and the courts (Moser & Cohen, 2013; Pollitt, 2013).

2.5.8 Intelligent agents

There have been other proposed approaches that use some form of remote agent in a ‘forensic’ context, mostly in relation to security/incident response activities, but these agents are designed to monitor and analyse certain data to detect and warn of a potential security event rather than be

[9] Who based their processing around a central ‘coordinator’ machine.

targeted at locating and collecting specific artefacts from remote computer systems (either a network or the cloud). In this respect they operate as a sophisticated Intrusion Detection System, and have been described as 'intelligent agents,' often with some level of autonomy or reasoning capability and some are designed for specific environments such as wireless networks or critical infrastructure (Jahanbin, Ghafarian, Hosseini Seno, & Nikookar, 2013; Kendrick, Criado, Hussain, & Randles, 2018).

An example of the properties envisaged for 'intelligent agents' are listed by Jahanbin et al. (2013) as:

- Autonomy: The agent possesses the capacity to act independently from its user, both in chronological terms and in the sense of adding intelligence to the user's instructions and exercising control over its own actions.
- Reactivity: The agent senses in and acts in its own surroundings.
- Proactivity: This refers to the agent's ability to exhibit goal-directed behaviour and take initiatives by itself to get closer to the defined goal, out of an external instruction
- by its user.
- Adaptability: The agent's capacity to learn and change according to the experiences accumulated.
- Continuity: An agent does not necessarily work only when its owner is sitting by the computer, it can be always active.
- Social ability: An agent is social software, which interact to other agents to do its job.
- Flexibility: The agent works proactively, that is directed by goals, but how it goes about to reach these goals may vary.
- Cooperation: The notion of cooperation with its user also seems to be fundamental in defining an agent, different from the one-way flow of information of ordinary software: intelligent agents are therefore true interactive tools.

These 'intelligent agents' are not relevant to this research as, despite the association with 'forensics' their output is not under the direct control and guidance of the investigator which contrasts with a remote installed agent, such as those provided by EnCase, FTK and Velocidex, that would typically:

- Execute some form of query sent from a central point (that may include some form of forensic analysis)

- Send back the results of the query directly to a central point.

2.5.9 Installed agents.

Figure 7 shows how remote agents are configured using Guidance Software's EnCase Endpoint Investigator software (where their 'servlet' can be run with up to 2,000 nodes (endpoints) connected simultaneously). There are two elements that make up an installation, the Examiner and the Secure Authentication for EnCase (SAFE) Server. The Examiner is a modified version of the Company's standalone forensic analysis tool to which the agents (which Guidance calls 'Servlets') installed on the endpoints connect via the authentication server, which handles access to the agents as well as enforcing role-based permissions and log-keeping (SC Magazine, 2016).

There are facilities that enable the agents to be installed at the endpoints that are dependent on the enterprise environment. Once installed, the Examiner provides Pathways (effectively templates) which guide the workflow for preparing the evidence in the investigation. For a 'full investigation' the Pathway includes five steps:

- Create a case.
- add evidence.
- audit your drive space.
- determine the time zone of your evidence.
- apply a hash library to your case.

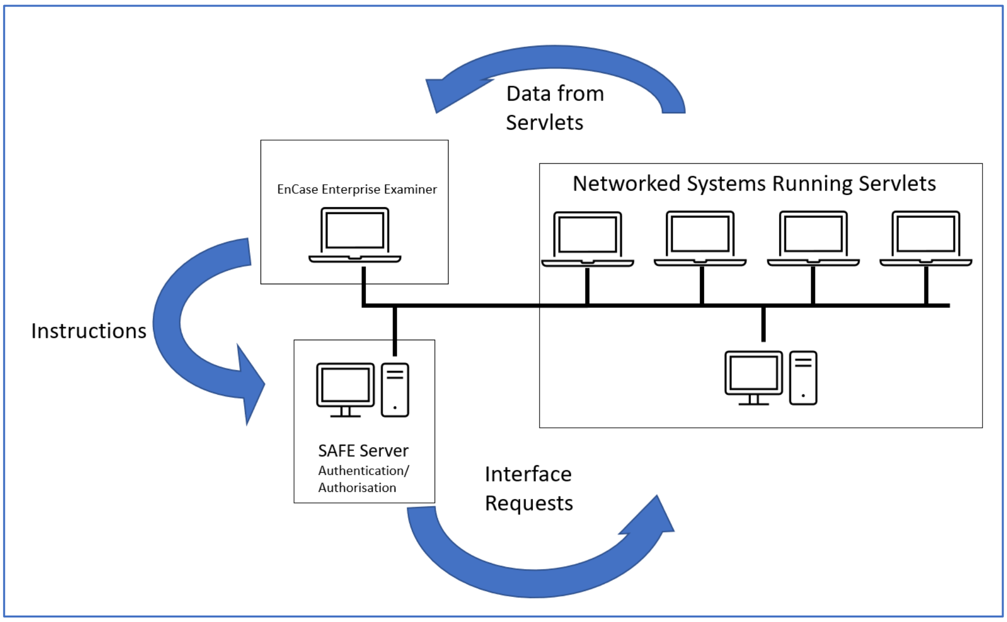


Figure 7 Simplified EnCase Configuration – installed 'servlets'.

Another of the well-established forensic tools that uses remote agents is FTK Enterprise which employs a similar structure consisting of:

- The AD Database – which can be either MS SQL Server or PostgreSQL and is used to store the processed metadata and undertakes user queries as well as other database-related activities (also used in the stand-alone forensic version – FTK Forensic Toolkit).
- The AD Enterprise Case Manager and Examiner – the Case Manager handles the administration aspects of, amongst other things, creating cases and users as well as the configuration settings for the database and agents. The Examiner is a version of the more 'traditional' forensic tool but modified to deal with data from the remote agents.
- The AD Enterprise Agent – this is deployed to the systems to be analysed and provides access to the Examiner.

A current (2023) example of a forensic tool that uses remote agents is Velociraptor[10]. This was developed from GRR (Cohen et al., 2011) and inspired by osquery which exposes an operating system as a relational database that enables it to be analysed using SQL queries.

Although Velociraptor uses a different type of connectivity to the endpoints than can be seen in EnCase and FTK, its configuration is similar with it having an examiner/administration machine, a central server and agents deployed to the remote systems as shown in Figure 8.

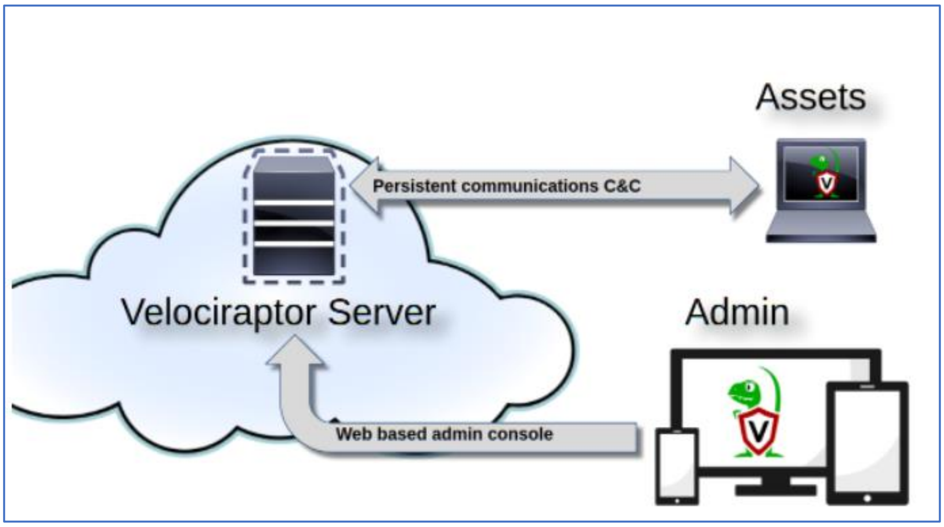


Figure 8 Velociraptor configuration[11]

[10] https://velociraptor.velocidex.com/velociraptor-e48a47e0317d

[11] retrieved from https://velociraptor.velocidex.com/velociraptor-e48a47e0317d.

2.5.10 Non-installed agents

While still executing some form of query and returning the results to a central point, Figure 9 shows the alternative approach where the agent is not installed but runs only in memory.

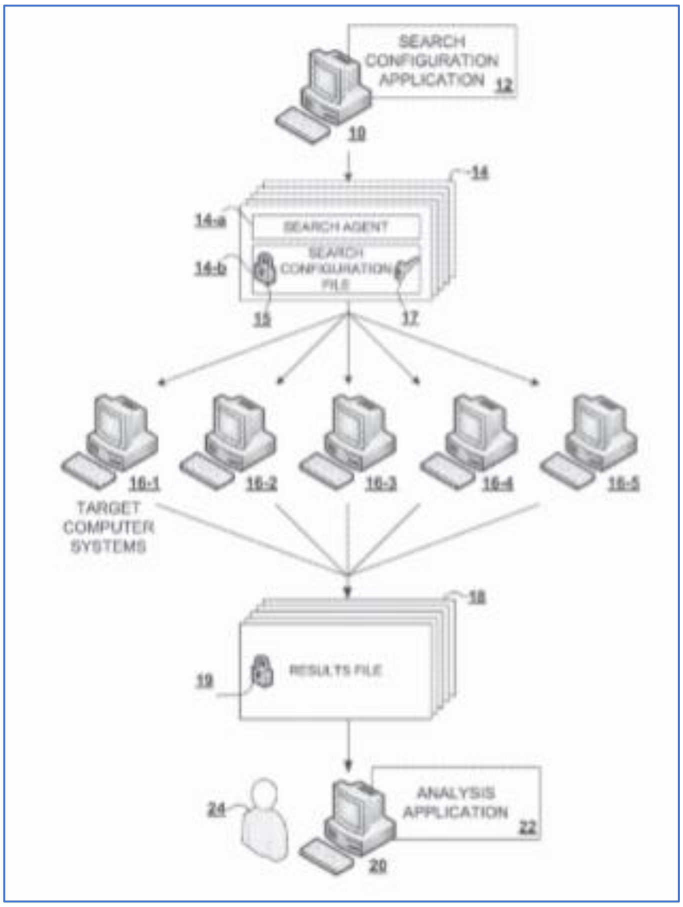


Figure 9 ISeekDiscovery non-installed agent processing[12]

Many of the tools designed to work forensically with remote systems have focused on a central processing point and rely heavily on indexing. ISeekDiscovery runs inside its own virtual environment situated entirely in memory. This feature means that it is possible for the application to be distributed across an unlimited number of target machines which can then be processed in parallel, with each instance being self-contained and with no central server required.

As an alternative to creating an index, ISeekDiscovery can search the raw data on a storage device without relying on the operating system to provide access to files, meaning that normally 'locked' files, such as email containers, can be processed. This means that processing across a large domain

[12] (https://patentimages.storage.googleapis.com/7e/a6/b7/e8cf8323dec9ae/US8392706.pdf)

can be undertaken while significantly reducing the volume of data being transported across the network compared to that involved in the ‘remote agent and indexing’ model.

2.6 Self-generated Test Images and Data

2.6.1 Forensig2

An alternative approach to providing pre-configured test images for forensic purposes is proposed by Moch and Freiling (2009). The authors describe a system that utilises script files (using Python and Qemu) to create virtual machines and simulate user behaviours such as the copying and deletion of files.

Moch and Freiling developed their approach for the teaching environment having discarded the alternative approaches being used which they identify as being:

- Manually generated – where the disk images are prepared using ‘artificial’ evidence (‘seed’ data)
- Honeypot generated – where images of systems that have been deliberately made available for compromise is used.
- Second-hand images – these are acquired from second-hand disks that have been purchased on the open market and rely on the fact that many people do not securely wipe their data before selling their disks.

The authors state that the problem with the other approaches is that they do not produce large volumes of data of sufficient practical relevance to the students or without placing a heavy burden on the instructors.

The Forensig2 process is shown in Figure 10. The input script produces a file system image with seeded ‘evidence’ and a report that defines the “ground truth,” i.e., a report detailing the ‘evidence’ seeded within the file system for later comparison with a student’s report.

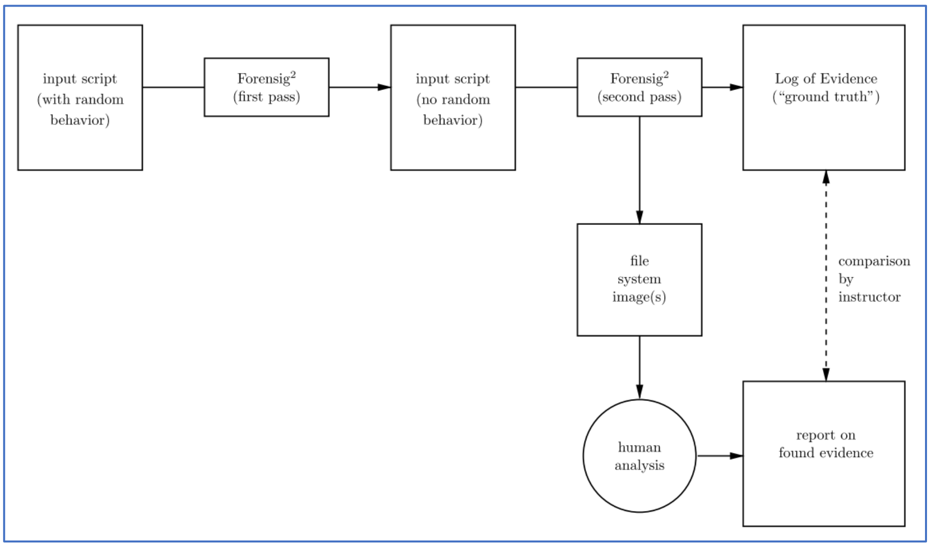

input script (with random behavior)
Forensig^2 (first pass)
input script (no random behavior)
Forensig^2 (second pass)
Log of Evidence ("ground truth")
file system image(s)
comparison by instructor
human analysis
report on found evidence

Figure 10 The Forensig2 Process (Moch &Freiling, 2009)

While the approach proposed by Moch and Freiling (2009) initially appears to hold promise for the generation of images that could be used by forensic practitioners to evaluate their tools against a 'known' set of data there are several drawbacks:

The solution requires practitioners to have a knowledge of Python scripts. This is likely to be less of an issue for those who are involved with incident response but is not a skill that could be assumed for all practitioners.

- The practitioner must construct the 'seed' data.
- The solution has only been demonstrated for Linux-based images.
- The project has not been updated since 2009 suggesting that there may have been practical issues with deployment of the approach.

The limitations noted here would seem to place restrictions on practitioners who would typically be encountering Windows-based machines. There is also a potential risk that the practitioner fails to execute actions in the correct sequence which could lead to 'contamination' of the created images which would make them unreliable as test data (Visti, Tohill, & Douglas, 2015).

2.6.1.1 Yannikos and Winter

A later method for automatically creating disk images is proposed by Yannikos and Winter (2013) in a paper titled Model-Based Generation of Synthetic Disk Images for Digital Forensic Tool Testing. In this paper the authors propose a framework for creating disk images containing simulated user activity without using virtualisation. Their framework is intended to assist in the creation of disk images based on "real-world" scenarios while removing the need for practitioners to have specific programming skill. It enables the creation of different disk images based on the same scenario without the need to make changes to the image generation process (summarised in Figure 11).

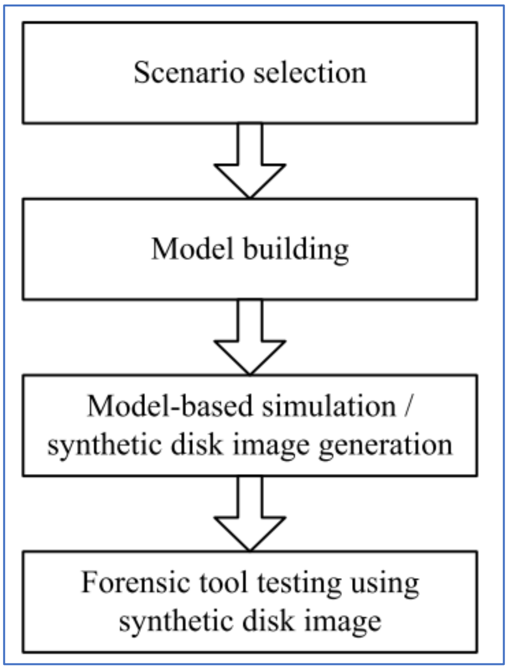

Scenario selection
Model building
Model-based simulation / synthetic disk image generation
Forensic tool testing using synthetic disk image

Figure 11 Generating a synthetic disk image (Yannikos & Winter, 2013)

The required model-building procedure is complex and requires the practitioner to follow an eight-step process:

1) **Subject definition**: For instance, users of a computer system with a hard disk.

2) **Global object definition**: (only two Global Object modules have been defined – *Synthetic Disk Image File* (mandatory) and *File Pool*). The practitioner needs to specify such parameters as the file system to be used, the number of null bytes and raw binary data to be written to the final synthetic image.

3) **Process definition**: The number and type of processes (sequences of actions to be performed) within the model are defined by the practitioner.

4) **Process sequence definition**: There are two types of processes:

1. Linear Processes - denotes a sequence of actions including one start action and one end action. Each action is performed exactly one time and has exactly one predecessor (except the start action) and one successor (except the end action).

2. Markov Processes - discrete-time processes used for simulating actions which result in synthetic data. Each Markov process within a model is defined as a set of states (actions) with state transitions and corresponding state probabilities as well as state transition probabilities. The state probabilities are defined while building the model and used to calculate the state transition probabilities with linear programming.

5) **Action definition**: There are nine implemented actions, and each action must be part of a previously defined process. The implemented actions are:

- Wait – do nothing.
- Create File System – only FAT16, FAT32 and EXT2 are supported.
- Create File – the file to be written to the file system can come from the file pool or contain pre-defined static data or random data.
- Delete File – this uses the file system for the synthetic disk image to remove a file and therefore updates the file's metadata.
- Write Raw Data – this writes data directly to the synthetic disk image without using its file system.
- Download File – this writes a file to the disk image in the same way as the Create File action but takes a random file from the internet as its source.
- Export Disk Image – exports the generated synthetic disk image.
- Export Disk Image Map – this provides detailed information in relation to the contents of the generated synthetic disk image.
- Import Disk Image – this imports an existing synthetic disk image (and its image map if available) for modification.

6) **Action transition definition:** Following on from the subject and action definitions, the different ways that one action can transition into another action within the same process are defined.

7) **Probability derivation**: The probabilities of certain actions being performed are defined. These can be derived externally through reference to statistics about file system behaviour, or internally, using assumptions regarding a scenario.

8) **Probability calculation**: The probability of one action transitioning into another action within the same process are calculated based on the previous step.

Notwithstanding the fact that the NTFS file system cannot be generated on the synthetic disk image, the lack of an operating system and associated artefacts limits the usefulness of this approach.

2.6.2 ForGe – Forensic Test Image Generator

An alternative approach to Yannikos and Winter (2013) is proposed by Visti et al. (2015) as a proof of concept for the rapid creation of synthetic images for forensic training and testing. Visti et al. (2015) recognise the benefits of creating virtual machine images and their tool is designed to use instructions for the image creation that are contained in a database. The tool is also designed to avoid the possible image contamination that is seen as a weakness of Forensig[2]. As for similar image creation tools, ForGe allows for the creation of an information sheet.

Visti et al. (2015)see a key benefit of their tool as being the provision of a user interface (although access to the created images is not available in the interface requiring the practitioner to copy them to another location via the command line). However, there is a requirement that the practitioners have some knowledge of the Linux platform as this is required to build and run the tool.

The image creation process features a 'case.' This is where the details of the selected file system are held as well as the number of images that are required to be built and their size, other file system parameters, root directory timestamps and timeline variance boundaries (Visti et al., 2015, p. 3). An overview of the inputs and outputs of ForGe are shown in Figure 12.

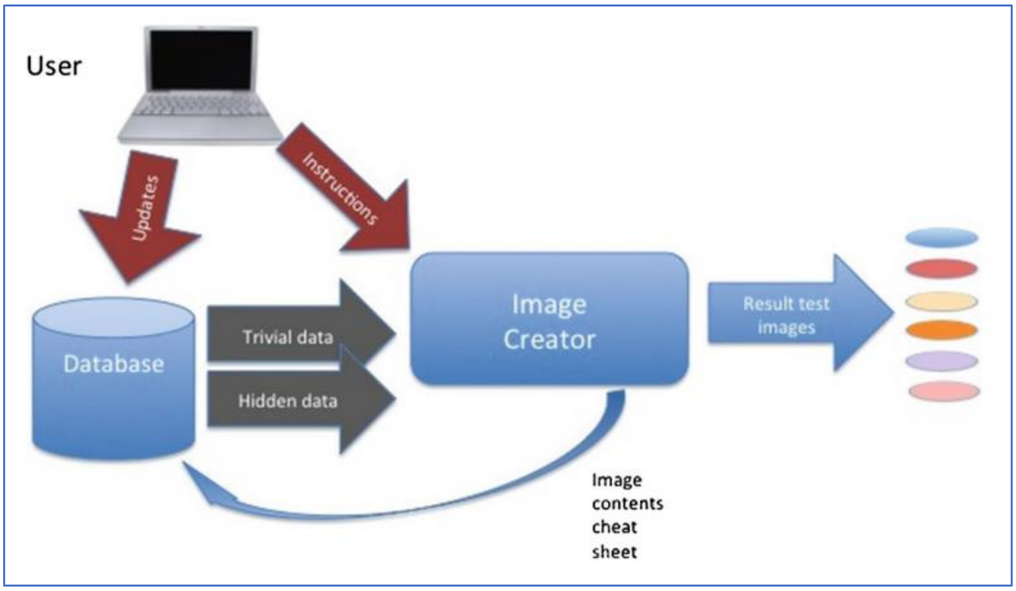


Figure 12 ForGe Inputs and Outputs (Vista et al.2015)

Definitions

- **Trivial file**: A file without any importance to the case. Trivial files are used in file system creation to place files on images according to a trivial strategy.
- **Trivial strategy**: An instruction related to case to place a random selection of trivial files in the image.
- **Secret file**: An important file to the case. – Secret strategy: An instruction related to case to place a single secret file using a hiding method to an image.
- **Action**: A simulated operation performed on a hidden file, related to timeline management. For example, action "rename" with a timestamp sets the timestamps to correspond to a rename action at the given time.

(Visti et al., 2015, p. 4)

Novel aspects of the ForGe design are random selections of 'seed' data for the synthetic images and the choice of various methods to hide data within the filesystem. The hiding methods are listed as:

- deleted file.
- extension changes
- alternate data streams
- concatenation of files
- file slack
- steganography
- unallocated space

The source code for ForGe is available on github.com[13] (as of August 2023) but has not seen any updates or changes since 2015, although a note from the developers suggest that they were working on web history and cache creation which would have added more value to the tool. While ForGe supports the NTFS and FAT filesystems it does not create an operating system or applications and their associated artefacts.

[13] https://github.com/hannuvisti/forge

2.6.3 Eviplant

Eviplant (M. Scanlon, Du, & Lillis, 2017) is designed to enable the efficient creation, manipulation, storage, and distribution of forensic images for the educational and training purposes. While not generating any new data the idea behind EviPlant is that a base image consisting only of the operating system can be downloaded and then some of its artefacts modified to create different test images which can then be distributed as packages containing details of the differences between the base image and the test image as well as the modified artefacts (and associated metadata).

Different challenges for students involve them downloading the base image and then integrating the evidence package to create the disk image that they will use for their analysis.

Eviplant consists of two core components: a "*diffing tool*" and an "*injection tool*." An instructor can manipulate the base image to simulate the type of activity that might occur in a real-world scenario and then use the diffing tool to identify the changes and create a package that can be applied to other copies of the base image using the injection tool.

As shown in Figure 13 the methodology features a range of artefacts as well as accommodating Linux and windows operating systems.

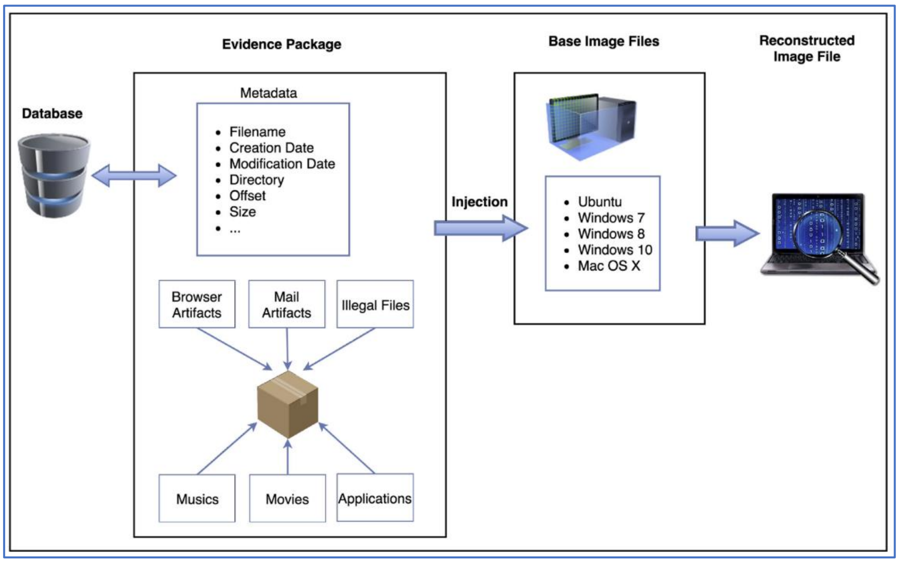


Figure 13 Eviplant methodology (M. Scanlon et al., 2017)

The authors comment that their approach can be used for digital forensic tool testing and validation as well as educational purposes but point out that the EviPlant system is a prototype and requires additional feature development, although they have carried out several tests of both the diffing and injection technologies on a Windows 10 system which they found to be successful.

Unfortunately, the diffing and injection tools are not publicly available for testing.

2.6.4 TREDE and VMPOP

Park (2018) proposed a proof-of-concept methodology to develop a synthetic corpus comprising of user-generated and system-generated reference data for practitioners working in the fields of digital forensics and security. The author defines synthetic data as "a*rtificially generated data for specific purposes, including but not limited to research, practice, education and tool validation*" which are divided into three groups:

"*Synthetic Test Data - If the purpose of generating a synthetic corpus is to test specific features in a group of tools, the generated corpus along with ground truth data will be synthetic test data for tool testing.*

Synthetic but Realistic Data - If the purpose of data generation is to establish a situation where a forensic examiner might encounter in an investigation, corpus creators can generate synthetic but realistic data that is typically associated with widely used OSes and applications.

Reference Data - All types of synthetic data may also be called reference data when they provide detailed information including ground truth data and related annotations available for reference."

(Park, 2018, p. 2)

As in earlier methodologies, Park (2018) uses virtualisation but rather than being Linux-centric he describes the use of the framework in a Windows environment that utilises system-generated reference data from the Windows operating system (Vista to Windows 10).

Park's (2018) aim is to describe a methodology that improves the efficiency of test dataset development that is practical and able to be employed for multiple purposes. The author demonstrates the proposed concepts with an example involving the creation of a windows Registry dataset.

TREDE stands for Two-class Reference Data dEvelopment with the two classes being user-generated reference data (UGRD) and system-generated reference data (SGRD).

To develop the UGRD, Park details a 5-step process:

1. Defining requirements

These will depend on the intended purpose for the resulting data and can be narrowly defined to limit the scope of the data generation or more broadly defined as required. This step serves to provide a basis on which to plan the other steps.

2. Setting up execution environments

   The execution environments can include operating systems, applications, external devices, hardware tools, and other associated resources.

3. Performing categorization

   The user-generated test data should be categorised based on the intended type and scope of the data relevant to a particular environment. This might include a particular file format in various forms, such as ‘normal,’ ‘normal with deleted data’ and ‘corrupted.’ Each category can have multiple classes and sub-classes.

4. Developing generation methods

The means by which the UGRD is generated need to be clearly defined and documented and may involve automated processes as well as manual data creation.

5. Executing tools and operations

This step involves running the automated tools, such as shell scripts, and any manual processes that are necessary to create the desired UGRD corpus.

Similar steps are involved in generating the SGRD:

1. Defining requirements

   As for the UGRD, this step involves defining the scope of the data generation but also requires the application of forensic knowledge of the relevant applications and operating systems.

2. Setting up execution environments

   This is the same as the second step for UGRD.

3. Defining user actions

The creator of the corpus will need to determine which user actions and their associated effect on the data they wish to incorporate. Knowledge of the characteristics of different operation systems (and different versions of operating systems) will need to be applied.

4. Building reference scenarios

   One or more scenarios are created to populate the target systems with realistic data for a given set of circumstances. The scenarios should contain detailed information to produce realistic user data such as a list of activities to be undertaken involving applications, user accounts and the like.

5. Populating target systems

   Once the scenarios have been developed the target systems are populated with the data that will become the SGRD corpus. Park (2018) proposes a practical strategy for automating the population of target systems based on virtualisation using a type-2 hypervisor and Python which is named VMPOP (Virtual Machine POPulation framework).

The overall workflow associated with TREDE is shown in Figure 14.

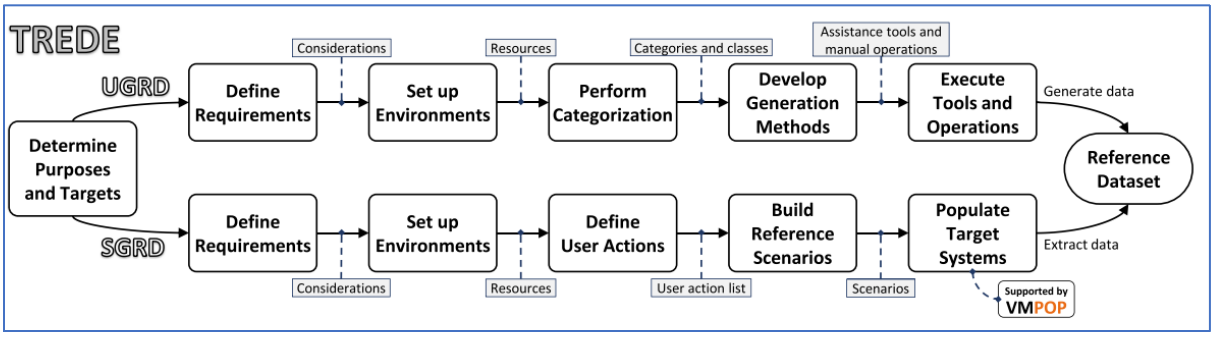


Figure 14 TREDE Workflow (Park, 2018)

The Virtual Machine Population Framework (VMPOP) was partly influenced by another project established in 2014 (ForGeOSI[14]) that was based around VirtualBox and Python which was intended to simplify the creation of virtual machines and automate data generation for forensic test and training purposes, although it was not completed.

VMPOP consists of five components (shown in Figure 15):

- HIS – Hypervisor Interface Subsystem

[14] https://github.com/maxfragg/ForGeOSI

- DES – Data Extraction Subsystem
- OAS – OS Automation Subsystem
- ULS – Universal Logging Subsystem
- EMS – Event Monitoring Subsystem

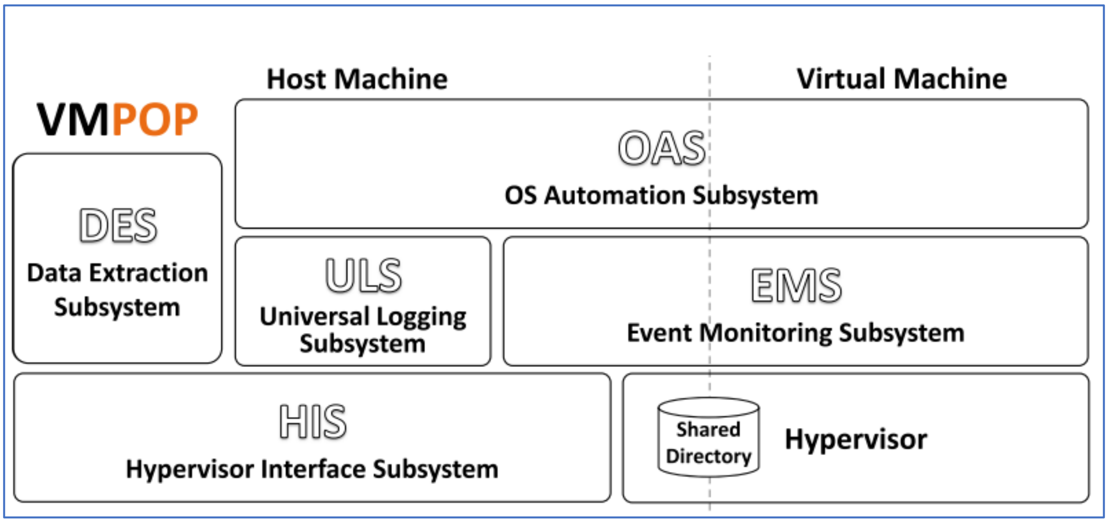

VMPOP
Host Machine
Virtual Machine
OAS
OS Automation Subsystem
DES
Data Extraction Subsystem
ULS
Universal Logging Subsystem
EMS
Event Monitoring Subsystem
HIS
Hypervisor Interface Subsystem
Shared Directory
Hypervisor

Figure 15 Design concept of VMPOP (Park, 2018)

- The Hypervisor Interface subsystem (HIS)

The HIS acts as a bridge between the subsystems of VMPOP and the instances of virtual machines. Its main function is to deliver requests from the OS Automation Subsystem (OAS) and the Event Monitoring Subsystem (EMS) using hypervisor APIs.

- The Data Extraction Subsystem (DES)

The DES handles the selective extraction of reference data from snapshots of virtual target machines or virtual disk formats to create the Reference Dataset Figure 16.

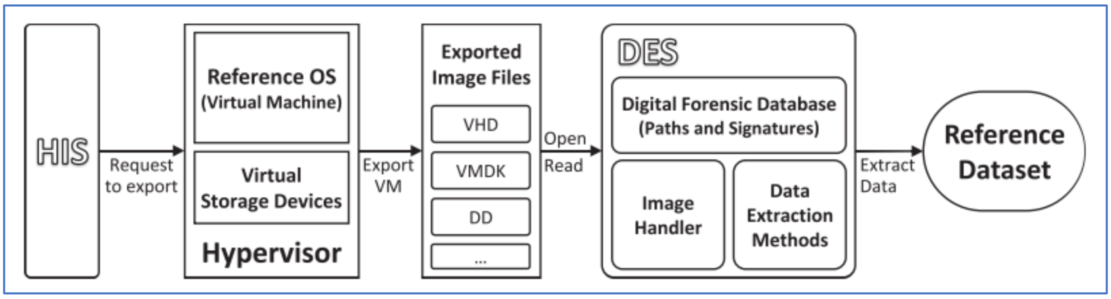


Figure 16 Event flows of the data extraction subsystem (DES) (Park, 2018)

- OS Automation Subsystem (OAS)

The role of the OAS is to provide automation of user activity inside the target virtual machine through a simple user interface. The subsystem consists of Operating system Automation Modules (OAM) each designed for a particular operating system. The subsystem relies on hypervisor APIs and shared storage.

- Universal Logging Subsystem (ULS)

The ULS is intended to process logging requests from the OAS (actions) and the EMS (events).

- Event Monitoring Subsystem (EMS)

The EMS is optional for the VMPOP concept. When enabled it can create a record of events within a target virtual environment that are associated with simulated user activities to produce 'ground truth' information for the reference data.

Although some example scripts are provided as well as the source code for the Python script *pyvmpop* deployment of TREDE/VMPOP requires significant work in setting up the environment, establishing scenarios, creating the reference data, knowledge of hypervisor APIs and Python scripts to tie the modules into the virtual machines.

2.6.5 The hystck Framework

Göbel et al. (2020) expanded the ideas of previous researchers in relation to the automatic generation of disk images containing simulated data with their hystck Framework. While they recognised that real-world datasets were useful to practitioners because they reflected the type of data they would be working with, the authors considered the potential lack of "ground truth" to be a significant limitation and agree with Yannikos and Winter (2013) that the solution is to produce synthetic datasets.

The hystck framework goes further than previous proposals for the creation of synthetic disk images in that it not only supports the automatic generation of operating systems but it also creates application artefacts through simulated human-computer activity (including synthetic network traffic) which Göbel et al. (2020) claim is indistinguishable from data generated in real life. A key feature of the framework is its moduler nature which enables the practitioner to simulate new applications (as well as generate new types of network traffic).

The framework is developed in Python to be platform-independent and employs virtualisation in the form of the open source Kernal-based Virtual Machine (KVM) that is based on the Linux kernal (which acts as a hypervisor), although Göbel et al. (2020) point out that other hypervisors which support *libvert* may be integrated into the framework, e.g. VirtualBox.

Hystck consists of two key client-server componants, the Framework Master (server-side) which is the physical machine host and the Interaction Manager (client-side) virtual machines as shown in Figure 17.

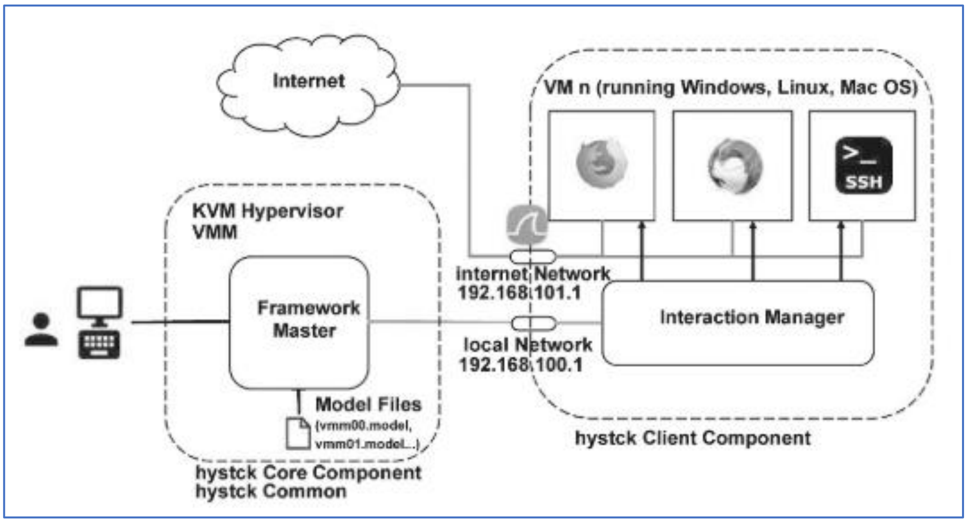


Figure 17 The hystck framework (Göbel et al., 2020)

The Framework Master manages the virtual machines and communicates via a local network connection with the Interaction Manager that controls the graphical user interfaces of the virtual machines to execute commands which generate the data for a particular scenario.

The virtual machines employ images based on templates that must be created in advance. Each virtual machine is configured so that each simulated user is working in an isolated environment.

The simulated user data is created using an 11-step procedure as described in the following table and illustrated in Figure 18.

Step	Action
1	The VMM class: • Functions as a setup environment to create and control guests. • Ensures that the default guest parameters (IP address, MAC address, template, tcpdump, etc.) are set to successfully clone templates. • Creates sockets on all the interfaces for the agents to listen on the guests.

2	The Guest class: - Loads the parameters from the constants.py configuration file. - Creates and controls the guests using the template files
3	The MAC addresses are linked to IP addresses and stored in the network configuration files for use by libvirt
4	Libvert creates the local and internet networks.
5	The Guest class: - Creates the virtual machines based on the templates using libvirt. - Creates a lock file
6	The Guest class causes each guest to load its user interaction model
7	- The user interaction models are executed in the guests causing the virtual machines to be started by libvirt. - The tcpdump tool is started on the host for each virtual machine to record network traffic.
8	The guest instances connect to the virtual machine.
9	The process flow of each user interaction model proceeds: - The interaction manager controls the graphical user interface - The interaction manager simulates the desired behavior using the operating system and its applications.
10	The simulation is complete: - The virtual machines are shut down by libvirt. - tcpdump stops capturing traffic
11	- The virtual machines and the local and internet network interfaces are deleted by libvirt. - The lock file is deleted.

(Göbel et al., 2020, pp. 9-11)

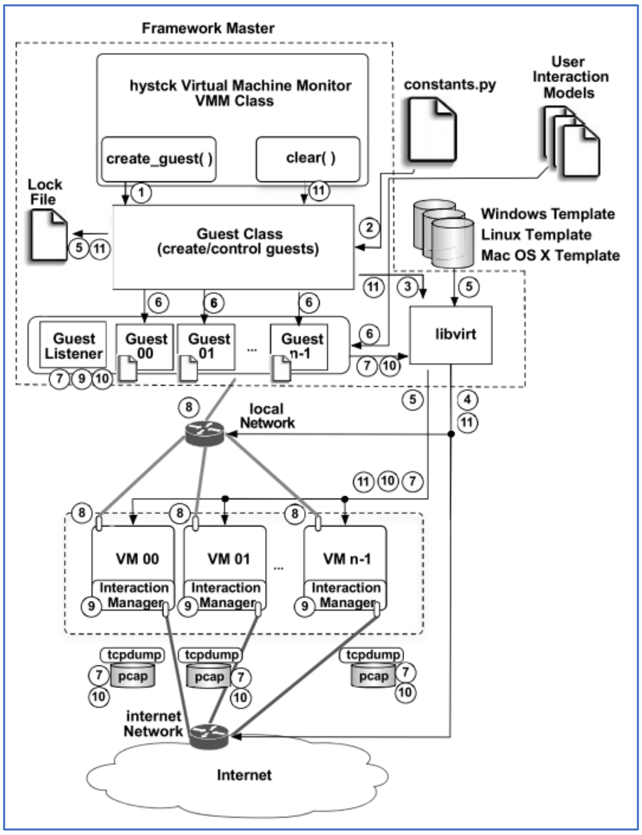
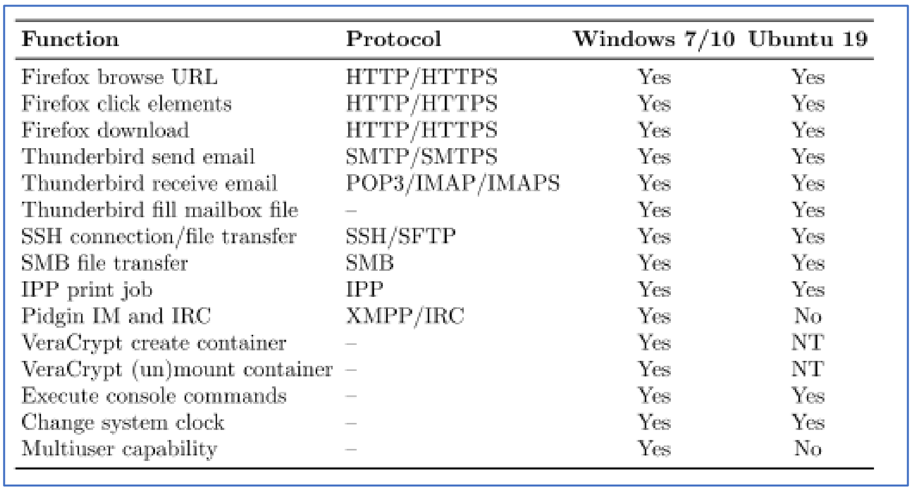


Figure 18 Hystck data synthesis procedure (Göbel et al. 2020)

The deployment and modification of images created using the framework is accomplished with a (large) base image and subsequent snapshots used to modify a copy of the base image with the new data.

The Hystck framework is claimed to be highly flexible and able to “partially automate forensic image generation” (Göbel et al., 2020) but its focus appears to be on creating network traffic and internet-based data as suggested by the framework features listed in Figure 19.



Function	Protocol	Windows 7/10	Ubuntu 19
Firefox browse URL	HTTP/HTTPS	Yes	Yes
Firefox click elements	HTTP/HTTPS	Yes	Yes
Firefox download	HTTP/HTTPS	Yes	Yes
Thunderbird send email	SMTP/SMTPS	Yes	Yes
Thunderbird receive email	POP3/IMAP/IMAPS	Yes	Yes
Thunderbird fill mailbox file	–	Yes	Yes
SSH connection/file transfer	SSH/SFTP	Yes	Yes
SMB file transfer	SMB	Yes	Yes
IPP print job	IPP	Yes	Yes
Pidgin IM and IRC	XMPP/IRC	Yes	No
VeraCrypt create container	–	Yes	NT
VeraCrypt (un)mount container	–	Yes	NT
Execute console commands	–	Yes	Yes
Change system clock	–	Yes	Yes
Multiuser capability	–	Yes	No

Figure 19 Hystck framework features (Göbel et al. 2020)

The hystck ‘generator’ function takes instructions from a YAML configuration file to generate data. Details of the YAML configuration files are documented allowing users who are not familiar with Python scripts and the hystck codebase to modify the image creation process to suit their needs. In a similar approach to other researchers (Park, 2018; Visti et al., 2015), Göbel et al. (2020) also provide a ‘reporting’ function that keeps track of changes made during the production of the datasets.

An issue identified by Göbel et al. (Göbel et al., 2020) in relation to conducting detailed forensic analysis on the generated images is the trace evidence left in the image by utilising the framework, such as the installation of Python, the hystck source code, thrid-party libraries to provide additional funtionality, various scripts and network connections associated with monitoring of the virtual machine. The authors are seeking to address these issues in later research.

2.6.6 TraceGen

TraceGen is a framework for automating the generation of user actions on a Windows system in a consistent manner that is realistic, comprehensive, and auditable (Du, Hargreaves, Sheppard, & Scanlon, 2021). In a similar manner to the approach proposed by Park (2018), TraceGen simulates user actions with Python scripts run against target virtual machines via APIs. Du et al. (2021) suggest that complex scenarios can be built by combining scripts, for instance, to emulate wear-and-tear on the target system and that the resulting test images are indistinguishable for those created through normal use of a computer.

The authors claim that their proposal makes the following contributions:

- It provides a proof-of-concept framework for generating realistic user data inside a disk image.
- It provides several plugins for the framework to emulate user actions at different levels of complexity, from simple file copying, to a “Google research session” on a particular topic.
- It collects the network traffic generated by actions on the machine.
- It documents a method for validating the artefacts generated by automated user simulation against data generated by a human.
- It discusses in detail the lessons learnt from attempting to implement this framework including the intrusiveness of tools on the generated data sets. It also sets out a clear

> research agenda for extending this framework to provide major benefits in digital forensic education, research, and investigations.

(Du et al., 2021, p. 2)

The authors note that it is difficult to produce large quantities of realistic data that stand up to detailed forensic examination and therefore there is a need for "*synthetic, automated, realistic, digital forensics artefacts at scale*" (Du et al., 2021, p. 3). Their approach is to programmatically automate a user's activities within a virtual machine such that the operating system itself generates artefacts as opposed to trying to create the artefacts directly.

The design of the framework involves 'machine control' actions performed external to the VM (such as removing power), and 'user' actions initiated from both outside and inside of the VM.

For emulating user actions automatically, there are several options available:

- Using the Application Programming Interfaces (APIs) such as Win32 API.
- Basic mouse and keyboard control
- Graphical User Interface (GUI) interaction, e.g., using PyAutoGUI[15]
- Browser automation, e.g., using Selenium[16]

The overall process is shown in Figure 20 whereby the 'input' consists of a list of user actions and a script on the host machine controls the guest VM boot, shutdown, and other actions. Once the scripts have been run the 'output' is the resulting disk image together with a list of actions that have taken place on the disks, which is in line with other researchers (Göbel et al., 2020; Park, 2018; Visti et al., 2015).

[15] https://github.com/asweigart/pyautogui

[16] https://www.seleniumhq.org

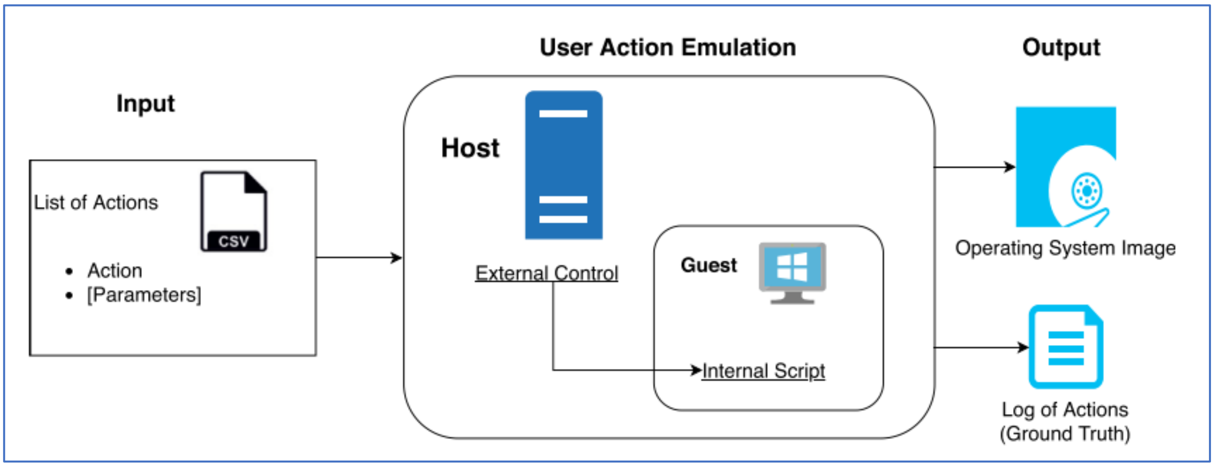


Figure 20 Image generation using TraceGen (Du et al. 2021)

Du et al. (2021) note that causing actions to occur inside the VM is a complex task and requires the Python interpreter to be running for anything except very basic tasks, such as opening an application. The authors accept that some of the activities they propose to emulate do not result in realistic artefacts without significant involvement of the user, such as using Python scripts to launch Google Chrome and inject a keyword into the address bar.

Du et al. (2021) identify two shortcomings in their approach:

1. Limitations of the internal scripts being run in the VMs that do not produce realistic artefacts when compared to those left by real user actions.
2. The approach of using external controls to run internal scripts requires a login session and therefore creates entries for the user account.

Additionally, as for some of the other proposals, to make anything other than basic actions within a VM a user is required to have some knowledge of Python scripting and APIs.

2.6.7 ForTrace

ForTrace is promoted by the researchers as addressing the need for a holistic approach to the generation of realistic data for digital forensics training data that can also be used for tool development and evaluation (Göbel et al., 2022). The approach is based on an extension of an earlier framework, *hystck* (Göbel et al., 2020).

ForTrace is composed of the two terms 'Forensics' and 'Traces' and aims to assist with generating "…*forensically relevant data sets, enabling the simultaneous generation of persistent, volatile and network traces*" (Göbel et al., 2022, p. 2). In line with previous researchers, the first step requires defining a model of a realistic system (or 'scenario') and then using automated functionality to simulate the actions of a user within a virtual machine. Many of the features of hystck are maintained in ForTrace, such as adopting client-server architecture, KVM, QEMU, YAMAL scripts, Python, and other open-source software as well as a dedicated network connection for communications.

A diagram provided by the researchers showing the high-level architecture of ForTrace (Figure 21) is very similar to that for hystck (shown in Figure 17) but with the addition of Guest Functionalities (and no connections from the Interaction Manager within the VM).

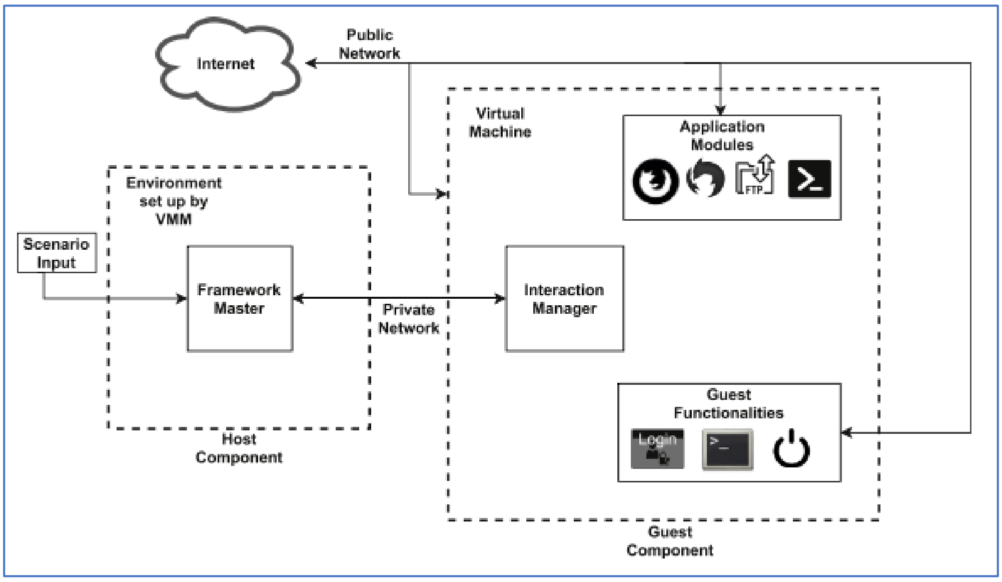


Figure 21 ForTrace framework architecture (Göbel et al. 2022 p.5)

The workflow consists of:

- Preparing and installing the framework on the host machine
- Creating a template for the guest VM
- Choosing and running a scenario

The key difference between hystck and ForTrace is the number of data synthesis features incorporated into the later, with the core features being:

- Elementary functions – such as shutting down the VM.
- Multi-user capability – examples are for multiple users being created and deleted or changing logged-in users.
- Anti-forensic capabilities – such as the deletion of registry keys

- PowerShell support – for example, connecting to a network share.
- Browser simulation – allows for emulating typical user web browsing activities.
- Email simulation – simulating user activity such as sending and receiving emails with attachments.
- Encrypted container support – provides basic VeraCrypt functionality such as mounting/dismounting encrypted containers.
- File transfer support – such as transferring files over the network.
- Printer support – allows the generation of printer traffic on the network.
- Malware synthesis – generates typical traces for common malware.

The ForTrace framework maintains a record of the 'ground truth' for each scenario in the form of an XML document that can be parsed into an HTTP format for easier review.

The authors point out that pre-configured Windows guests cannot be provided due to licencing restrictions, but details are given for setting up the necessary templates. Future enhancements include the expansion of existing modules to create traces of communication tools and adding more mail and browser applications.

2.7 Problem Statement

The problem being addressed in this research is that there is no method for evaluating forensic tools that use remote agents as part of their processing. This means that practitioners do not have a consistent way to validate the outputs or compare tools which they often rely upon and that are fundamental to their work.

2.8 Chapter Conclusion

This chapter has outlined the environment in which digital forensic and incident response practitioners operate as well as some of the tools and techniques they employ and how these have developed over time to address a changing technological landscape.

The literature review has found that while the environment, tools and techniques associated with digital forensics and incident response have seen significant changes in the last 20 years the supporting rigour of tool testing and validation has not kept up despite calls for more research and testing (E. Casey, 2016; Nikkel, 2014).

Furthermore, although it is regarded as an essential requirement of the courts, it is often not practical for practitioners to validate the forensic reliability of different tool outcomes during real-life processing of ESI given the time, resource, and access constraints (Guo, Slay, & Beckett, 2009).

Things have not changed since Slay and Beckett (2007) suggested the lack of widespread tool validation is due to the "high workload and low resource environment" (Slay & Beckett, 2007, p. 1) in which forensic computing practitioners typically operate (Franke et al., 2017). Slay and Beckett (2007) also point out that validating forensic tools is expensive and time consuming which leads to practitioners relying on independent validation studies by others.

The closest that the disciplines have come to a standardised testing framework is the NIST Federated Framework that enables practitioners to self-validate their tools for processing data contained in a single data set for certain functions, such as keyword searching. NIST provide the test data to be used in the validation of a tool's performance, but this is often based on old or obsolete file systems and file types.

Several researchers have proposed generic frameworks for tool testing or frameworks that target specific areas of practice, such as malware and mobile phone data. In these cases, practitioners must determine for themselves much of the criteria to be used for validating the performance of the tools and identify suitable test data.

The key findings of the literature review are:

- Despite the efforts of NIST, there are few test results available to DF practitioners in relation to DF tools and none of these results include any form of evaluation process.
- There are a lack of data sets available to DF practitioners that are both current and comprehensive.
- There are no frameworks that go beyond the testing of DF tools to assist with tool evaluation.
- The results of testing that are available do not include the processing of data on remote systems.
- Guides for building test platforms as part of an evaluation exercise are complex, time consuming, typically require specialist programming skills and do not cater for evaluating remote systems in a realistic environment.

- It was noted that there is a problem with creating pre-configured Microsoft Windows guest machines that is related to Microsoft's licence restrictions that restrict distribution as noted by Gobel et al. (2022) who adopted the use of templates and automated scripts for the operating system installation process to get around the problem.
- Other researchers adopted a similar approach to Gobel et al. (2022) by creating virtual machines based on a set of criteria. Some of these approaches provide an incomplete solution, but all of them focus on generating one or more individual machines that require a lot of effort from the user (particularly in creating a network that incorporates them) and in some cases specialist knowledge (such as Python scripting). In addition, the data incorporated into the virtual machines is mostly based on user activities from a limited number of applications and require scenarios to be developed together with the incorporation of other seed data.

Based on the literature review a Problem Statement has been made that encapsulates the focus of this research.

The next chapter will identify and justify the research methodology that has been adopted for this research.

3 METHODOLOGY

3.1 INTRODUCTION

Several methodologies that have been used in the field of information systems/technology were considered for this research, specifically:

Action Research: this involves an iterative process where researchers work collaboratively with practitioners to identify and address practical problems. It emphasizes reflection, action, and continuous improvement. It is suitable for problems in organizational and social settings that require practical and context-specific solutions (Avison, Baskerville, & Myers, 2001; Baskerville & Wood-Harper, 1996)

Case study research: this involves in-depth exploration of a particular case or cases to gain insights into a problem or phenomenon and typically incorporates extensive data collection and analysis. It is considered valuable for understanding complex, context-specific problems and generating rich qualitative data (Haamann & Basten, 2019; Stone, Kosack, & Aravopoulou, 2020).

Experimental Research: This involves manipulating variables to test hypotheses and establish cause-and-effect relationships. It is commonly used in scientific and empirical research where it is useful for problems that require controlled experiments and objective measurement of outcomes (Lankton & Luft, 2014).

Grounded Theory: This is a qualitative research method that aims to develop theories or concepts based on data analysis where uses iterative coding and categorization of data. It is considered effective for exploring and generating theories from empirical data in areas where existing theories are lacking (Urquhart & Fernández, 2013).

Mixed method: This combines both qualitative and quantitative research approaches and allows researchers to gather a comprehensive understanding of a problem. It is considered useful for addressing complex problems that benefit from a combination of qualitative and quantitative data (Sahaym, Vithayathil, Sarker, Sarker, & Bjørn-Andersen, 2023).

Design science: This solves a class of problems through the creation, and subsequent evaluation, of IT artefacts within a practical real-life context (Hevner & Chatterjee, 2010; Hevner, March, Park, & Ram, 2004; K Peffers et al., 2006). Artefacts include methods, models, instantiations, and constructs (Hevner et al., 2004). The central aim of design science is the creation of effective artefacts and utility (Hevner & Chatterjee, 2010; Hevner et al., 2004; Lee, 1989; McKay & Marshall, 2005; K Peffers et al., 2006; John Venable, 2006) and has been selected as the paradigm used for this research. The selection of Design Science rather than alternatives was made on the basis that it is particularly suited to the task of creating a new process model (artefact) for the testing of software tools due to its focus on designing solutions (Armstrong & Armstrong, 2010).

Of relevance to this research is the distinction Hevner et al., (2004) make between routine design and design science research. They state that routine design addresses organisational problems by drawing upon existing knowledge, whereas design science research finds new ways to address unsolved problems or addresses solved problems more efficiently or more effectively.

The aim of a design science researcher is to understand the problem that the artefact is intended to address and to consider how appropriate it would be for providing a solution. This is accomplished through the construction of an artefact and its subsequent use in a real-world context (Nunamaker, Chen, & Purdin, 1990). Researchers therefore aim to apply the artefact in a particular environment to improve an existing situation. They accomplish this by considering where the artefact is to be deployed and how it will be used (McKay & Marshall, 2005). Applegate (Applegate, 1999) promotes the idea of 'industry-relevant' research within the IS discipline as opposed to the more common adoption of the functionalist paradigm.

Design science is a viable alternative paradigm for IS research (Hevner & Chatterjee, 2010; John Venable, 2006) and Venable (2006) argues that researchers typically neglect to emphasise theory building and that theory should take the central role.

Using Venable's DSR Activity Framework (John Venable, 2006) (Figure 22) the 'Technology Invention/Design' associated with this research is a framework for evaluating deployed agents in the areas of DF and IR, although it has the potential to be of benefit in other areas.

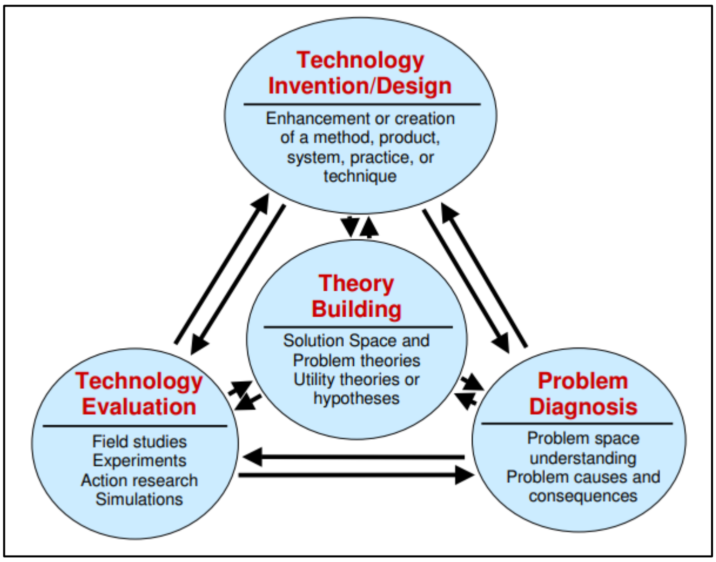

Technology Invention/Design
Enhancement or creation of a method, product, system, practice, or technique
Theory Building
Solution Space and Problem theories Utility theories or hypotheses
Technology Evaluation
Field studies
Experiments
Action research
Simulations
Problem Diagnosis
Problem space understanding
Problem causes and consequences

Figure 22 A DSR Activity Framework (Venable, 2006)

The thesis' literature review will provide the 'Problem Diagnosis' and drive the 'Theory Building' aspect of the research. Key elements of the thesis will be field studies and experiments that will comprise the 'Technology Evaluation' component. Following Venable's Framework, the various Activities will be iterative to refine the final artefact.

3.2 PROCESS MODEL FOR THE RESEARCH

The Design Science Research Methodology has been adopted for this research. This methodology incorporates the Design Science Research Process (DSRP) model that has frequently been used within information systems research and has been selected as the model for this research. The DSRP is a synthesis of common design process elements (Archer, 1984; Cole, Purao, Rossi, & Sein, 2005; Eekels & Roozenburg, 1991; Hevner et al., 2004; Nunamaker et al., 1990; Rossi & Sein, 2003; Takeda, Veerkamp, Tomiyama, & Yoshikawa, 1990; Walls, Widmeyer, & El Sawy, 1992). The DSRP aims to be consistent with prior literature and to deliver frameworks for undertaking design science research and for presenting design science research in Information Systems.

The DSRP developed by Peffers et al. incorporates six activities (Figure 23).

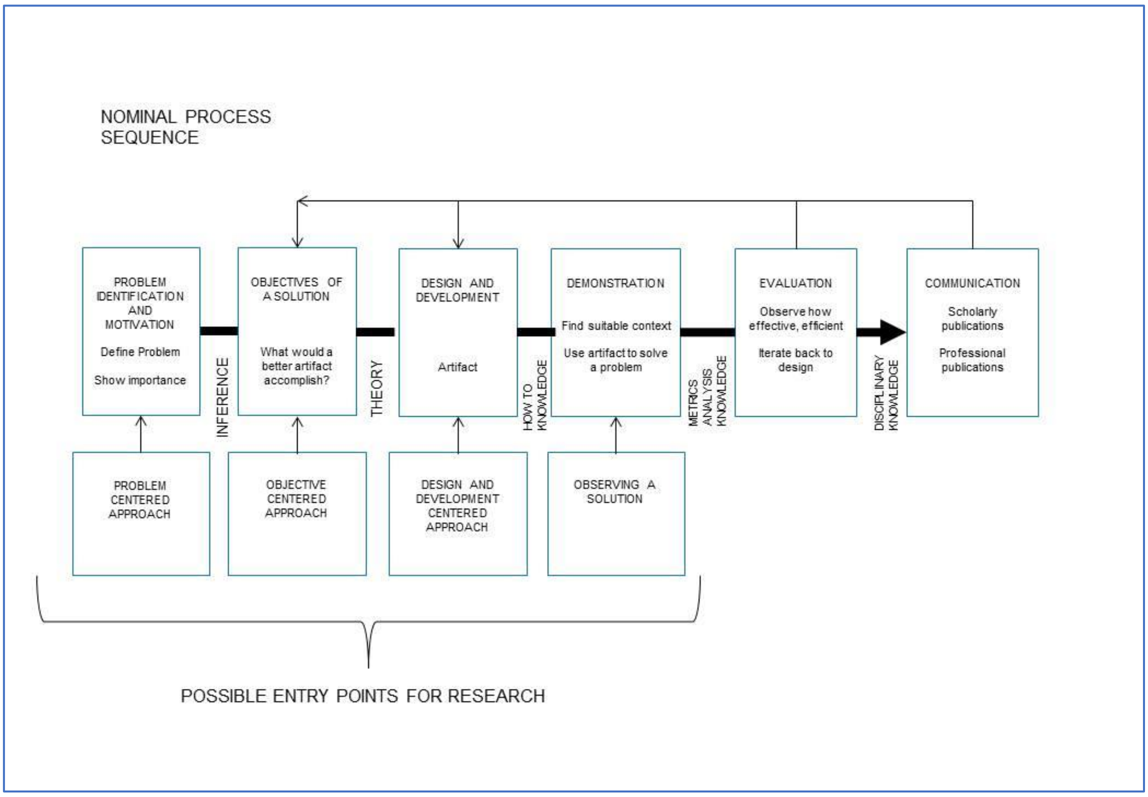

NOMINAL PROCESS SEQUENCE
PROBLEM IDENTIFICATION AND MOTIVATION
Define Problem
Show importance
INFERENCE
OBJECTIVES OF A SOLUTION
What would a better artifact accomplish?
THEORY
DESIGN AND DEVELOPMENT
Artifact
HOW TO KNOWLEDGE
DEMONSTRATION
Find suitable context
Use artifact to solve a problem
METRICS ANALYSIS KNOWLEDGE
EVALUATION
Observe how effective, efficient
Iterate back to design
DISCIPLINARY KNOWLEDGE
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
OBJECTIVE CENTERED APPROACH
DESIGN AND DEVELOPMENT CENTERED APPROACH
OBSERVING A SOLUTION
POSSIBLE ENTRY POINTS FOR RESEARCH

Figure 23 Design Science Research Process (DSRP) model after Peffers et al. (2007)

Brief descriptions of the activities in the DSRP and their relationship to this research are outlined below:

- **Activity 1 – Problem Identification and Motivations** - this involves establishing the problem to be addressed that enables the research to be justified based on the perceived benefits of the final output - the artefact. The background section of this research sets out the justification for this research which is supported by several experts in the appropriate domains. This will be expanded upon in the research itself.
- **Activity 2 - Objectives of a Solution** - the researcher defines the objectives of the research. These will be based on the problem to be solved. In this research the proposed new artefact will enable practitioners who collect ESI for DF to determine the effectiveness of different tools for use in their domain (following ethics approval).
- **Activity 3 - Design and Development** – this involves the creation of the artefact. For this research that will be AFERA.
- **Activity 4 - Demonstration** – this activity is where the artefact is used in a suitable environment to solve the identified problem. In this research AFERA has been used in both real-life and test environments.

- **Activity 5 - Evaluation** - the performance of the artefact was reviewed with reference to the stated objectives from Activity 2. This led to iterations of the process looping back to Activity 3.
- **Activity 6 - Communication** - the researcher adds to the body of knowledge through publication of their research. Peffers et al.(2007) identify the need for communication to publicise the problem that has been identified, its significance to the domain and the artefact that has been produced. Several papers will be published in appropriate peer-reviewed journals which, together with the published thesis, will be contributing to the body of knowledge.

3.3 DSRP ENTRY POINT

Peffers et al.(2007) identify several entry points in the DSRP where a researcher can begin their research process (Figure 23). The proposed research will have as its Entry Point the Problem-Centred Approach. The problem in this case will be the lack of a method for testing remote software agents designed to collect ESI for digital forensic, IR and electronic discovery purposes.

The utility of the method will be determined by running its final version in a real environment using different remote agents.

3.4 APPLYING THE METHODOLOGY TO THE RESEARCH PROBLEM

The following sections describe how the chosen methodology will be followed during this research by mapping the activities undertaken with the activities set out in the Peffers et al.(2006) DSRP model. However, a slight change to the DSRP model is adopted for this thesis to allow for feedback into the design and development activity of any issues encountered during several 'internal evaluation' iterations.

The additional explicit iteration has been described in email correspondence with Professor Peffers (K Peffers, 2011) and he indicated that it is consistent with the intentions of his model. The additional iteration and Peffers' comments are covered in the section on Activity 4 below.

Activity 1 - Problem identification and motivation

The problem being addressed in this research is that:

There is no defined method, such as a process or framework, that enables a practitioner to undertake an evaluation of tools designed to collect data from endpoints for digital forensic purposes.

Activity 2 - Objectives of a solution

The objective of this research is to develop an artefact that will enable practitioners to evaluate remote agent-based tools designed for use in DF investigations.

Activity 3 - Design and development

The literature review will enable this research to build on the contributions of other researchers. In addition, personal experience and interactions with other practitioners will be used to create the evaluation criteria that will form the basis for AFERA.

The top-level approach adopted for the Design and Development stage of the DSRP is:

1. Identify functional and non-functional criteria for evaluation tools designed to collect data from remote endpoints using agents.
2. Create a suitable data set that will form the basis of the test data.
3. Develop the framework based on the evaluation criteria and the test data.

These stages are now covered in more detail.

1. Identify functional and non-functional criteria for evaluation tools designed to collect data from remote endpoints using agents.

A series of interviews has been undertaken with leading academics and practitioners to help identify suitable functional and non-functional evaluation criteria. The Semi-Structured Interview Guide used for these interviews is included as an appendix to this thesis.

2. Create a suitable data set that will form the basis of the test data.

Based on the initial assessment criteria, a dataset has been adopted against which the remote agents can be evaluated. This dataset builds upon an existing NIST virtual machine image (modified with the permission of NIST) that is currently used for the testing of forensic tools in a standalone environment.

This virtual machine image can be replicated to form a small network against which the remote agents can be deployed. Although the virtual machine image will be the same for all the endpoints, the use of multiple endpoints will allow assessment of deployment times, bandwidth requirements and other features of the deployed agents to be evaluated.

3. Develop the framework based on the evaluation criteria and the test data.

Based on the outcome of the academic and practitioner expert interviews the initial iteration of the Framework has been constructed such that it can be used to evaluate a remote agent.

Activity 4 – Demonstration

The demonstration activity involves using AFERA on an instance of the test dataset comprising a single virtual machine. A slight modification has been made to the Peffers et al. (2006)., process flow which is circled in Figure 24. This is to allow for any outcomes of the Demonstration activity to be fed back to the earlier stages if required prior to the Evaluation activity taking place. This modification was discussed in a personal communication with Professor Peffers (K Peffers, 2011) (who commented that the DSRM encourages iteration) and it has previously been incorporated in earlier research where it was shown to have some utility (Adams, 2013) .

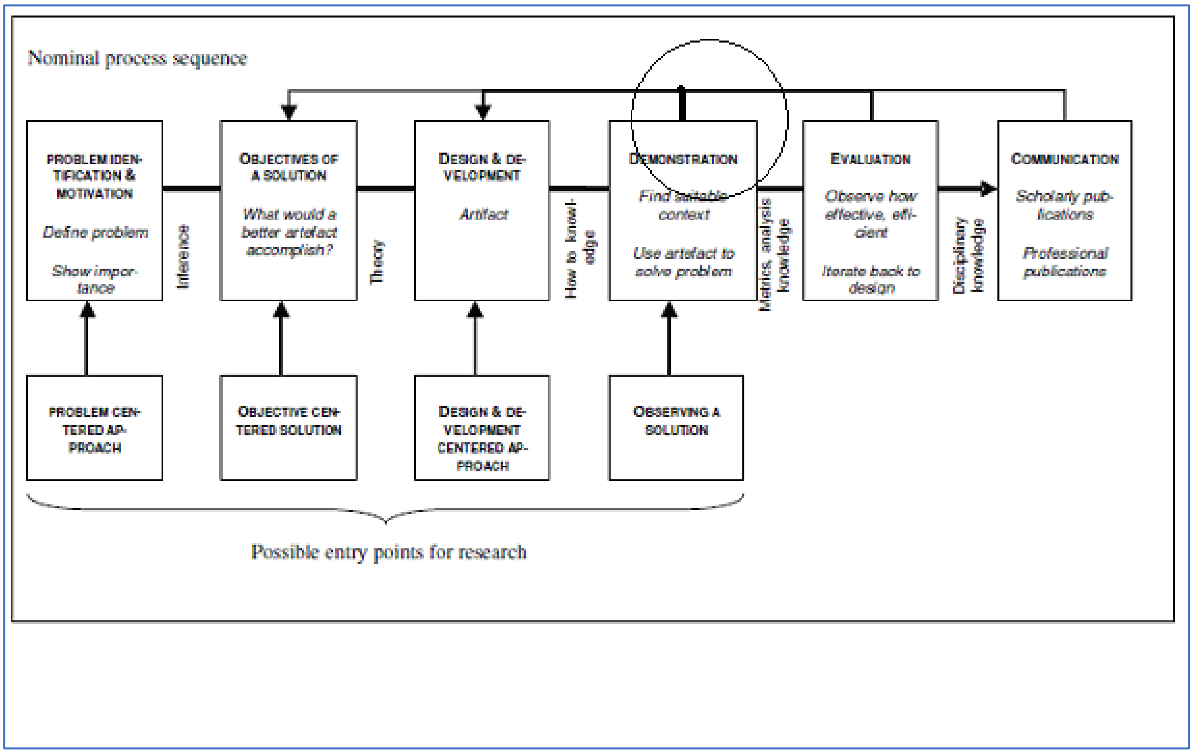

Nominal process sequence
PROBLEM IDENTIFICATION & MOTIVATION
Define problem
Show importance
Inference
OBJECTIVES OF A SOLUTION
What would a better artefact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, analysis knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
OBJECTIVE CENTERED SOLUTION
DESIGN & DEVELOPMENT CENTERED APPROACH
OBSERVING A SOLUTION
Possible entry points for research

Figure 24 Modification to the process flow of the DSRP model (after Peffers et al., 2006)

Activity 5 - Artefact Evaluation

The utility of the artefact will be assessed using a 'naturalistic' evaluation based on a case study approach. As an 'instantiation' the artefact evaluation criteria will be considered by an expert panel through interviews. The evaluation criteria to be addressed is:

- functionality
- usability
- reliability
- performance
- supportability

The issue of enhancing the credibility of this research through triangulation of data (Creswell, 2005) has been balanced with the practical aspects of obtaining quality feedback through in-depth reviews by authoritative reviewers. As Bruce (2007) points out, "*Ultimately, the number of data events is less important than the trustworthiness of the reporting*" and so despite the small number of reviewers planned they are all authorities within the appropriate field and will be given the opportunity to confirm that their feedback has been correctly interpreted and applied.

In line with the DSRP model, prior to submitting the artefact for evaluation, a 'demonstration' phase will be completed which will amount to a 'desk check' of the artefact by the researcher.

Activity 6 – Communication

Journal papers and this thesis constitute the communication activity. In addition, details of the research will be published on appropriate user group forums.

3.5 CHAPTER CONCLUSION

This chapter has identified and justified Design Science as the research paradigm used in this research. It has also identified the Design Science Research Methodology (DSRM) (K Peffers et al., 2006) that has been adopted, and described how it has been applied. Based on the DSRM, the next chapter will cover the design of the artefact.

4 ARTEFACT DESIGN

4.1 INTRODUCTION

This chapter incorporates the first part of the Design and Development stage of Peffers at al.'s DSR process model (2011; 2006) and covers the process in which the requirements that will form the basis the artefact for this research are identified and the reasoning behind their selection. The development of the artefact draws from existing theories and knowledge to produce a solution to a defined problem (Hevner et al., 2004).

The defined problem for this research is:

> *There is no defined method that enables a practitioner to undertake an evaluation of tools designed to collect data from endpoints for digital forensic purposes.*

The design of the artefact will follow the two-stage structure outlined below:

Stage 1 – Identifying the required features of the artefact. This will involve extrapolating the features being evaluated based on appropriate NIST and SANS digital forensic analysis documentation. In addition, expert practitioners will provide explicit requirements through a series of interviews.

Stage 2 – Selection of data sets for the evaluation environment. There are two sets of 'data' of interest in the evaluation of the remote forensic tools. The first set comprises of mostly application-generated data that has been developed by authoritative sources or is capable of being automatically generated, while the second set involves artefacts generated by the operating system through some structured process to ensure the appropriate artefacts are produced.

4.2 STAGE 1- IDENTIFYING THE REQUIRED FEATURES OF A REMOTE FORENSICS TOOL

4.2.1 Processing Categories

For this research, several categories have been created that are associated with artefact processing based on the high-level functionality required of a digital forensic tool (excluding the recovery of deleted files and processing Volume Shadow Copies) by reference to NIST and SANS test categories. The categories used in this research have been defined as:

- **Filesystem** – this includes all files, locked by the operating system or otherwise. This category also includes parts of the operating system itself, such as the MFT and USNJournal, as well as user-generated artefacts (although it does not include email containers, registry hives or event logs).
- **Registry** – all registry hives, including those used by applications.
- **Event logs** – all event logs.
- **Email** – all forms of email that are likely to be encountered. This includes individual MSG and EML files that have been saved to disk as well as email containers such as MBOX, PST and OST.

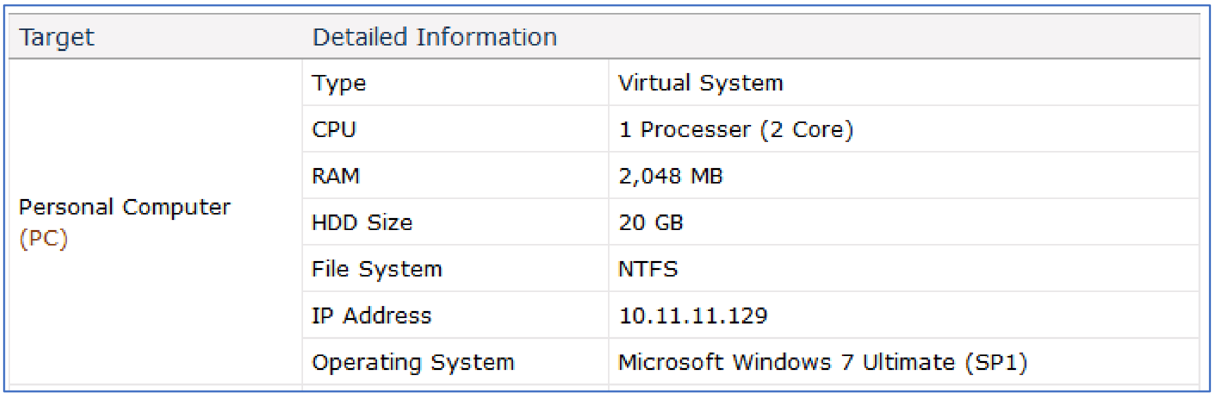
For this research, the processing carried out by the remote agent tools being evaluated involves getting the artifact, or a subset of the artifact, from the remote system to be processed by a central system. The processing may also involve a third-party tool, e.g., for memory analysis, or it may be processed into a usable form by the central system (or even by the agent itself).

4.2.2 Requirements Source 1: NIST CFReDS Data Leakage Case.

NIST makes available for public comment several prototype data sets for testing digital forensic tools[17]. One of these data sets, the Data Leakage Case[18], includes a complete virtual machine image of a PC. This fits in with the environment associated with this research, given that the tools being evaluated are intended to be run on remote PCs. NIST provide the following details in relation to the virtual machine image:

[17] https://www.cfreds.nist.gov/

[18] https://www.cfreds.nist.gov/data_leakage_case/data-leakage-case.html



Target	Detailed Information	
Personal Computer (PC)	Type	Virtual System
	CPU	1 Processer (2 Core)
	RAM	2,048 MB
	HDD Size	20 GB
	File System	NTFS
	IP Address	10.11.11.129
	Operating System	Microsoft Windows 7 Ultimate (SP1)

Figure 25 CFREDS Data Leakage Image

The purpose of the Data Leakage Case is to enable practitioners to learn about different types of data leakage and gain practice of using appropriate investigation techniques to answer 60 questions relating to the case. Although the NIST Data Leakage case data (in the form of forensic images) is too old for consideration as a data source for this research the functionality needed to answer Data Leakage Case questions (considered relevant for remote forensic tools as explained later in this chapter) has been used as the basis for creating a list of requirements for a deployed agent tool used for digital forensic or incident response investigations.

NIST provide a Scenario Overview for the Data Leakage Case data set:

"'Iaman Informant' was working as a manager of the technology development division at a famous international company OOO that developed state-of-the-art technologies and gadgets.

One day, at a place which 'Mr. Informant' visited on business, he received an offer from 'Spy Conspirator' to leak of sensitive information related to the newest technology. Actually, 'Mr. Conspirator' was an employee of a rival company, and 'Mr. Informant' decided to accept the offer for large amounts of money and began establishing a detailed leakage plan.

'Mr. Informant' made a deliberate effort to hide the leakage plan. He discussed it with 'Mr. Conspirator' using an e-mail service like a business relationship. He also sent samples of confidential information though personal cloud storage.

After receiving the sample data, 'Mr. Conspirator' asked for the direct delivery of storage devices that stored the remaining (large amounts of) data. Eventually, 'Mr. Informant' tried to take his storage devices away, but he and his devices were detected at the security checkpoint of the company. And he was suspected of leaking the company data.

At the security checkpoint, although his devices (a USB memory stick and a CD) were briefly checked (protected with portable write blockers), there was no evidence of any leakage. And then, they were immediately transferred to the digital forensics laboratory for further analysis.

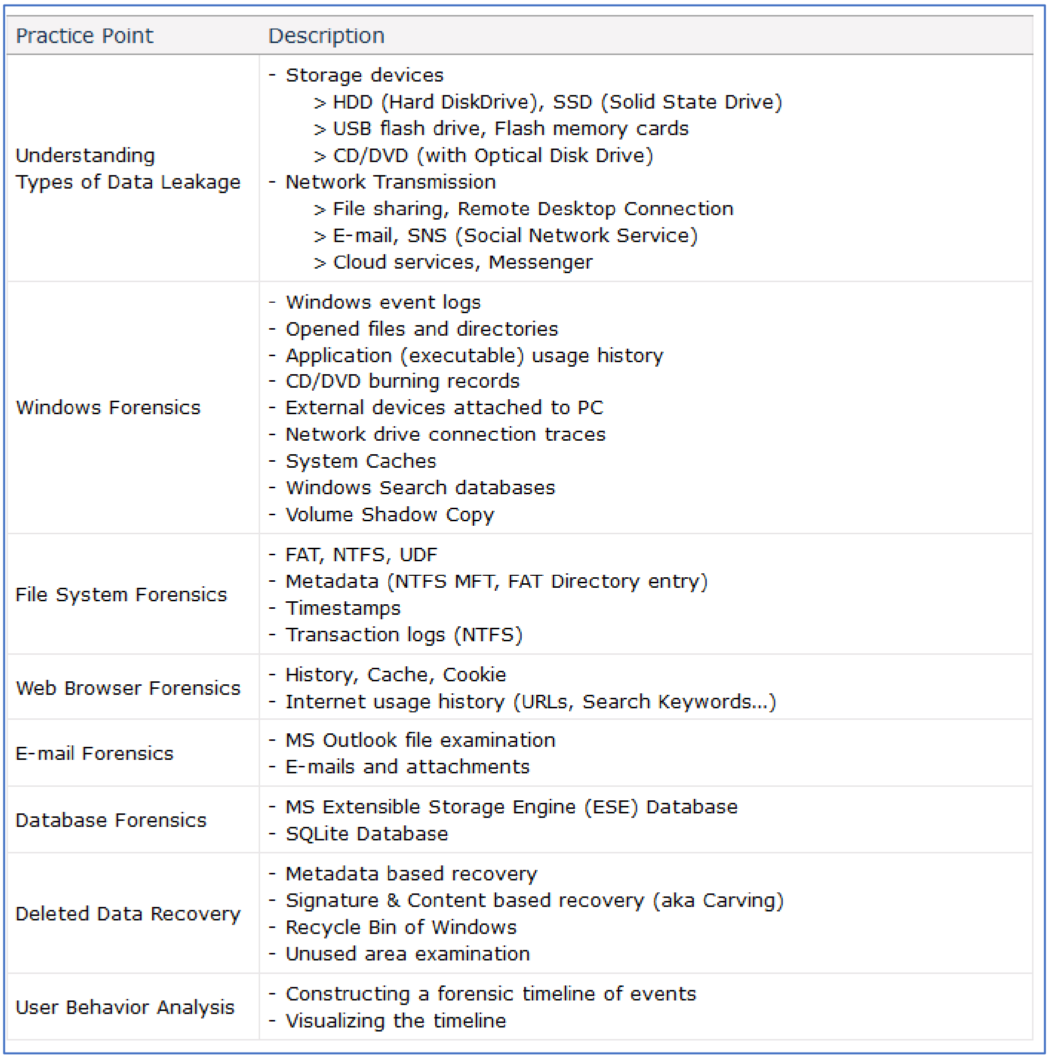
The information security policies in the company include the following:

- *Confidential electronic files should be stored and kept in the authorized external storage devices and the secured network drives.*
- *Confidential paper documents and electronic files can be accessed only within the allowed time range from 10:00 AM to 16:00 PM with the appropriate permissions.*
- *Non-authorized electronic devices such as laptops, portable storages, and smart devices cannot be carried onto the company.*
- *All employees are required to pass through the 'Security Checkpoint' system.*
- *All storage devices such as HDD (hard disk drives), SSD, USB memory stick, and CD/DVD are forbidden under the 'Security Checkpoint' rules.*

In addition, although the company managed separate internal and external networks and used DRM (Digital Rights Management) / DLP (Data Loss Prevention) solutions for their information security, 'Mr. Informant' had sufficient authority to bypass them. He was also very interested in IT (Information Technology) and had a slight knowledge of digital forensics.

In this scenario, find any evidence of the data leakage, and any data that might have been generated from the suspect's electronic devices."

The scenario includes a table containing Practice Points that it seeks to cover in relation to DF work, shown in Figure 26 below.



Practice Point	Description
Understanding Types of Data Leakage	- Storage devices > HDD (Hard DiskDrive), SSD (Solid State Drive) > USB flash drive, Flash memory cards > CD/DVD (with Optical Disk Drive) - Network Transmission > File sharing, Remote Desktop Connection > E-mail, SNS (Social Network Service) > Cloud services, Messenger
Windows Forensics	- Windows event logs - Opened files and directories - Application (executable) usage history - CD/DVD burning records - External devices attached to PC - Network drive connection traces - System Caches - Windows Search databases - Volume Shadow Copy
File System Forensics	- FAT, NTFS, UDF - Metadata (NTFS MFT, FAT Directory entry) - Timestamps - Transaction logs (NTFS)
Web Browser Forensics	- History, Cache, Cookie - Internet usage history (URLs, Search Keywords...)
E-mail Forensics	- MS Outlook file examination - E-mails and attachments
Database Forensics	- MS Extensible Storage Engine (ESE) Database - SQLite Database
Deleted Data Recovery	- Metadata based recovery - Signature & Content based recovery (aka Carving) - Recycle Bin of Windows - Unused area examination
User Behavior Analysis	- Constructing a forensic timeline of events - Visualizing the timeline

Figure 26 CFREDS Data Leakage Case Practice Points

Not all the Practice Points identified by NIST are directly relevant to the capabilities of a deployed agent whose task is to collect artefacts for further processing or make them available for remote processing. For instance, "User Behaviour Analysis" using some form of timeline approach is unlikely to be a feature of the remote agent but might be functionality available in a third-party tool against artefacts and their metadata collected by the remote agent.

Similarly, "Database Forensics" may involve processing of database artefacts collected using the remote agent rather than being accomplished through the remote agent itself. The allocation of categories associated with the functionality required of the remote agent (as described earlier) will be applied to specific (relevant) questions posed by NIST in their Data Leakage Scenario which will in turn address the high-level Practice Points.

For this research, the artefacts relating to the Data Leakage Case scenario have been identified and copied into a data set that can be copied to the endpoints on which the remote agents will be deployed. Some of the artefacts involve files that would typically be locked under the Windows file-locking functionality on a running system. As will be discussed later, the deployed agent's ability to access locked files will be tested separately.

NIST questions and practice points undertaken for this research.

Some of the questions relating to the Data Leakage Case have been ignored in relation to the identification of tool requirements, specifically:

- Question 1 – This is specific to the case and involves hashing and comparing images other than the 'main' PC image and is considered inappropriate for tools running in a live environment.
- Question 2 – The capturing of specific disk sectors (such as those containing the MBR) is not considered relevant for remote deployed agents running on live machines.
- Questions 47, 48, 49 & 50 – These relate to Volume Shadow Copies and are not considered relevant for remote deployed agents running on live machines.
- Questions 53, 54, 55,56 & 57 – These relate to external device images and data recovery. Where some of the data may be in the main PC image the functionality is covered by other questions (e.g., jumplists and shellbags).
- Questions 58, 59 & 60 – These relate to analysis of the collected data and do not provide any additional requirements for the remote deployed tools.

The Data Leakage questions, Focus Areas and Categories are included in Table 1 below, which shows the Scenario question number (as it appears on the NIST CFreDS website), the sixty questions themselves together with the Focus Area covered by each question and the Category allocated to each question.

Table 1 NIST Data Leakage Case Questions

CFReDS Original Number	Question	Focus Area	Category
1	What are the hash values (MD5 & SHA-1) of all images? Does the acquisition and verification hash value match?	Forensic practice	N/A
2	Identify the partition information of PC image.	Filesystems	N/A
3	Explain installed OS information in detail. (OS name, install date, registered owner…)	Registry	Filesystem, Registry

4	What is the time zone setting?	Registry	Registry
5	What is the computer name?	Registry	Registry
6	List all accounts in OS except the system accounts: Administrator, Guest, systemprofile, LocalService, NetworkService.	Registry	Registry
7	Who was the last user to logon into PC?	Registry	Registry
8	When was the last recorded shutdown date/time?	Registry	Registry
9	Explain the information of network interface(s) with an IP address assigned by DHCP.	Registry	Registry
10	What applications were installed by the suspect after installing OS?	Registry	Registry
11	List application execution logs. (Executable path, execution time, execution count...)	Prefetch, Jumplists	Filesystem
12	List all traces about the system on/off and the user logon/logoff.	Event Logs	Event Logs
13	What web browsers were used?	Registry	Registry
14	Identify directory/file paths related to the web browser history.	Browser Artifacts Filesystem	Filesystem
15	What websites were the suspect accessing? (Timestamp, URL…)	Browser Artifacts	Filesystem
16	List all search keywords using web browsers. (Timestamp, URL, keyword…)	Browser Artifacts	Filesystem
17	List all user keywords at the search bar in Windows Explorer. (Timestamp, Keyword)	Registry	Registry
18	What application was used for e-mail communication?	Registry	Registry
19	Where is the e-mail file located?	Registry	Registry
20	What was the e-mail account used by the suspect?	Registry	Registry
21	List all e-mails of the suspect. If possible, identify deleted e-mails.	Email Processing	Email
22	List external storage devices attached to PC.	Registry	Registry
23	Identify all traces related to "renaming" of files in Windows Desktop. (It should be considered only during a date range between 2015-03-23 and 2015-03-24.)	NTFS Journal File	Filesystem
24	What is the IP address of company's shared network drive?	Registry	Registry

25	List all directories that were traversed in "RM#2".	Filesystem	Registry
26	List all files that were opened in 'RM#2".	JumpLists, ShellBag	Filesystem
27	List all directories that were traversed in the company's network drive.	Registry, Filesystem	Registry, Filesystem
28	List all files that were opened in the company's network drive.	Registry, LNK files, JumpLists	Registry, Filesystem
29	Find traces related to cloud services on PC. (Service name, log files...)	Filesystem, Registry	Registry, Filesystem
30	What files were deleted from Google Drive? Find the filename and modified timestamp of the file.	Google Drive Transaction Log	Filesystem
31	Identify account information for synchronizing Google Drive.	Google Drive Transaction Log	Filesystem
32	What a method (or software) was used for burning CD-R?	Filesystem, Event Logs, NTFS Journal	Filesystem, Event Logs
33	When did the suspect burn CD-R?	Filesystem, Event Logs, NTFS Journal, Registry	Filesystem, Event Logs, Registry
34	What files were copied from PC to CD-R? (Hint: Just use PC image only. You can examine transaction logs of the file system for this task.)	Filesystem, NTFS Journal	Filesystem
35	What files were opened from CD-R?	JumpLists, LNK files, NTFS Journal	Filesystem
36	Identify all timestamps related to a resignation file in Windows Desktop.	Filesystem	Filesystem

37	How and when did the suspect print a resignation file?	Filesystem, Registry	Filesystem, Registry
38	Where are "Thumbcache" files located?	Filesystem	Filesystem
39	Identify traces related to confidential files stored in Thumbcache. (Include '256' only)	Thumbcache	Filesystem
40	Where are Sticky Note files located?	Filesystem	Filesystem
41	Identify notes stored in the Sticky Note file.	Filesystem	Filesystem
42	Was the 'Windows Search and Indexing' function enabled? How can you identify it? If it was enabled, what is a file path of the 'Windows Search' index database?	Registry	Registry
43	What kinds of data were stored in Windows Search database?	Microsoft ESE database	Filesystem
44	Find traces of Internet Explorer usage stored in Windows Search database. (It should be considered only during a date range between 2015-03-22 and 2015-03-23.)	Microsoft ESE database	Filesystem
45	List the e-mail communication stored in Windows Search database. (It should be considered only during a date range between 2015-03-23 and 2015-03-24.)	Microsoft ESE database	Filesystem
46	List files and directories related to Windows Desktop stored in Windows Search database. (Windows Desktop directory: \Users\informant\Desktop\)	Microsoft ESE database	Filesystem
47	Where are Volume Shadow Copies stored? When were they created?	Filesystem	N/A
48	Find traces related to Google Drive service in Volume Shadow Copy. What are the differences between the current system image (of Question 29 ~ 31) and its VSC?	Filesystem, VSC	N/A
49	What files were deleted from Google Drive? Find deleted records of cloud entry table inside snapshot. dB from VSC. (Just examine the SQLite database only. Let us suppose that a text-based log file was wiped.)	Filesystem, SQLite db.	N/A
50	Why can't we find Outlook's e-mail data in Volume Shadow Copy?	Registry	N/A
51	Examine 'Recycle Bin' data in PC.	Recycle Bin	Filesystem

52	What actions were performed for anti-forensics on PC at the last day '2015-03-25'?	NTFS Journal, Browser Artifacts, Microsoft ESE	Filesystem
53	Recover deleted files from USB drive 'RM#2'.	Data Carving	N/A
54	What actions were performed for anti-forensics on USB drive 'RM#2'? (Hint: this can be inferred from the results of Question 53.)	Data Carving, Filesystem	N/A
55	What files were copied from PC to USB drive 'RM#2'?	Filesystem, JumpLists, ShellBag, Data Carving	N/A
56	Recover hidden files from the CD-R 'RM#3'. How to determine proper filenames of the original files prior to renaming tasks?	Data Carving (signature and metadata)	N/A
57	What actions were performed for anti-forensics on CD-R ‘RM#3’?	Filesystem	N/A
58	Create a detailed timeline of data leakage processes.	Forensics Reporting	N/A
59	List and explain methodologies of data leakage performed by the suspect.	Forensics Reporting	N/A
60	Create a visual diagram for a summary of results.	Forensics Reporting	N/A

4.2.3 Requirements Source 2: SANS Forensic Analysis

The SANS poster associated with their DFIR course FOR500[19] was used as another authoritative reference. SANS splits the different artefacts into groupings associated with a particular activity commonly associated with digital forensics (and incident response) investigations. The groupings have been reproduced in the tables below.

In the first column is an identifier for the type of artefact involved. The second column identifies where the artefact may be located and in what form it exists. The final column is the category of associated with the functionality that a remote agent would need to possess to present the artefact for processing.

File Download

Artefact	Artefact Detail	Category
Open/Save MRU	Registry	Registry
Email Attachments	Email	Email
Skype History	Skype	Filesystem
Browser Artifacts	SQLite, Webcache	Filesystem
Downloads	SQLite, Webcache	Filesystem
ADS Zone.Identifier	Alternate data streams	Filesystem

Program Execution

Artefact	Artefact Detail	Category
UserAssist	Registry	Registry
Windows 10 Timeline	SQLite	Filesystem
BAM/DAM	Registry	Registry
Shimcache	Registry	Registry
Amcache.hve	Registry	Registry
SRUM	Registry, ESE database	Registry, Filesystem
Jump Lists	JumpLists	Filesystem
Last-Visited MRU	Registry	Registry
Prefetch	pf files	Filesystem

[19] DFPS_FOR500_v4.11_0121

Deleted File or File Knowledge

Artefact	Artefact Detail	Category
XP Search - ACMRU	Registry	Registry
Thumbcache	Thumbcache	Filesystem
Thumbs.db	Thumbs.db	Filesystem
IE/Edge file	History.IE5/Webcache	Filesystem
Search-WordWheelQuery	Registry	Registry
Recycle bin	Recycle bin	Filesystem
Last-Visited MRU	Registry	Registry

File/Folder Opening

Artefact	Artefact Detail	Category
Open/Save MRU	Registry	Registry
Recent Files	Registry	Registry
Jump Lists	Jump Lists	Filesystem
Shell Bags	Registry	Registry
Shortcut (LNK) Files	LNK files	Filesystem
Prefetch	pf files	Filesystem
Last-Visited MRU	Registry	Registry
IE/Edge file	History.IE5/Webcache	Filesystem
Office Recent Files	Registry	Registry

Account Usage

Artefact	Artefact Detail	Category
Last Login	Registry	Registry
Last Password Change	Registry	Registry
RDP Usage	Registry	Registry
Services Events	Event Logs	Event Logs

Logon Types	Event Logs	Event Logs
Authentication Events	Event Logs	Event Logs
Success/Fail Logons	Event Logs	Event Logs

External Device/USB Usage

Artefact	Artefact Detail	Category
Key Identification	Registry	Registry
First/Last Times	PnP Logs, Registry	Registry
User	Registry	Registry
PnP Events	Event Logs	Event Logs
Volume Serial Number	Registry	Registry
Drive Letter and Volume Name	Registry	Registry
Shortcut (LNK) Files	LNK Files	Filesystem

Browser Usage

Artefact	Artefact Detail	Category
History	History.IE5/Webcache, SQLite	Filesystem
Cookies	Cookies/SQLite	Filesystem
Cache	History.IE5/Webcache	Filesystem
Flash & Super Cookies (LSOs)	LSOs	Filesystem
Session Restore	JS, DAT, misc. files	Filesystem
Google Analytics Cookies	SQLite	Filesystem

NetworkActivity/Physical Location

Artefact	Artefact Detail	Category
Timezone	Registry	Registry
Cookies	Cookies/SQLite	Filesystem

Network History	Registry	Registry
WLAN event Log	Event Logs	Event Logs
Browser Search Terms	History.IE5/Webcache, SQLite	Filesystem
System Resource Usage Monitor (SRUM)	Registry, SRUM ESE db.	Registry, Filesystem

4.2.4 Requirements Source 3: Expert Practitioner Panel

Interviews with specialist digital forensic practitioners (as the Expert Practitioner Panel) have been conducted to ascertain the desired attributes of tools that employ remote forensic agents. Expert panels have been used in a variety of situations to assist in refining attributes/features of an artefact, for instance, in modelling and simulation applications (Balci, 2001), sustainable smart city development (Yadav, Mangla, Luthra, & Rai, 2019), software development and process modelling for forensic data acquisition (Adams, 2013). The practitioners used as the expert panel in this research are active in the following environments:

- Large resource companies
- Large financial institutions
- Academia (with experience in law enforcement)
- Law enforcement

Initial interviews focussed on identifying both the functionality required of the tools (and thus contributing to the need for forensic testing and validation) as well as non-functional attributes (enabling tools to be evaluated for potential adoption or comparison). Follow-up interviews took place, with different participants to the initial interviews, once the information gathered from the initial interviews had been used to create a draft framework.

Prior to the follow-up interviews, practitioners were provided with copies of the draft framework for them to consider against their own environment and experience.

The requirements are divided into 'high-level characteristics' and 'low-level characteristics.'

High-level characteristics

1. Speed – the time to complete the work is a crucial factor given that the skilled practitioner resources are expensive.
2. Language Support - the ability to support languages other than English is important and not very well supported. However, there was no consensus as to which languages should be supported by default.
3. Dependencies - consideration should be given in relation to the agent's dependencies, i.e., any other software that must be installed for the agent to perform.

4. Ease of use - in terms of general characteristics, the time it takes for a practitioner to learn how to use and deploy the remote agents is very important. We discussed how this might be determined, including having a group of practitioners undertake various tasks and rate their experience. Includes deployment method.
5. Scalability - the ability to scale is necessary as cases often involve multiple systems. This could have a significant impact on bandwidth, storage, and time as the number of concurrent agents increase.
6. Customisation – the ability to change the configuration of the tool means that unnecessary functions are not performed. Often, tools have functions that are run automatically and cannot be disabled when they are not needed for the case. This takes up time and effort.
7. Accuracy and Reliability – for forensic, i.e., court-related matters, it is essential that the output from the processing meets the standards necessary for it to be considered as evidence.
8. Auditability – For court purposes and in other situations, the ability is needed to audit the agent's processing, but not a blow-by-blow record of activities.
9. Results output options.
10. Cost – this is a big issue. It is often the case that free tools, while performing most tasks very well, are missing essential features that are only found in expensive commercial products. The high cost of some commercial products renders them unsuitable for a lot of cases.
11. Handling machines not on enterprise network.
12. Invisible to end user.
13. Minimal impact on end user.
14. Minimal impact on network bandwidth.
15. Can be run in the cloud.
16. Data sovereignty – must be Australian cloud for instance.

Low-level characteristics:

1. Search and process files

2. Search and process email:
    - MS OST and PST
    - MSG
    - EML

- TNEF
- MBOX

3. Search and process compressed files:
    - ZIP
    - TAR
    - RAR
    - 7Z

4. Search capabilities
    - Plain search
    - Wildcard search
    - Whole word search
    - Beginning of word search
    - End of word search
    - REGEX search
    - HEX sequence search

5. Capture capabilities
    - System memory
    - Swap file
    - Pagefile
    - Hibernation file
    - Registry hives (including amcache)
    - EVT & EVTX logs
    - SRUDB
    - Jumplist files
    - Prefetch files
    - Windows LNK files
    - Executables and DLLs
    - Windows system logfiles
    - NTFS $MFT

- NTFS $USNJournal
- Recycle bin.
- Windows Error Reporting (WER) artefacts
- RDP Cache
- Windows EDB (Search)
- Browser cache files
- Pre-configured collection categories

6. Identify encrypted/password protected files.

The Expert Practitioner Panel pointed out that in some instances, the remote tool will just collect artefacts that will be processed later using another specialist tool, examples of this would be the system memory and pagefile. In other instances, the tool will be required to process some of the artefacts prior to collection, an example of this would be collecting emails between certain dates.

Although it is common for a forensic tool to offer data carving as a feature, none of the members of the Expert Practitioner Panel suggested data carving given the problems identified in Section 2.4.1 Remote data challenges.

4.3 STAGE 2 - POTENTIAL DATA SETS (SEED DATA) FOR FORENSIC TOOL EVALUATION

Slay and Becket proposed a 'sustainable' model to address the issues that they identified as being the "*…high cost and time involved in validating a tool and the lack of verifiable and repeatable testing.*" (Slay & Beckett, 2007, p. 5)

Having pointed out that practitioners typically only use a small subset of the functions contained in a digital forensic tool, Slay and Becket (2007) support the idea of a deterministic suite of reference sets based on mapping of the critical functions of the discipline that have been identified using an extensible model as opposed to the traditional definitive model.

In this situation the tool (or process) used to produce the test results is independent of the mechanism that is used to validate the tool (or process). If the expected results are known for a particular function, then validation of a tool is reduced to providing a set of references with known results based on a set of metrics of accuracy and precision.

The flexibility of this approach is that if the needs of digital forensic practitioners can be identified in terms of critical functions (and associated parameters) with the expected results mapped as a reference set, then any tool can be validated against these elements independently from its original design intentions. The development of reference sets "*…has the added features of extensibility, tool neutrality, tool version neutrality and transparency*" (Slay & Beckett, 2007, p. 5) which are defined as:

Extensibility*: With a defined function, there exists a set of specifications for components that must be satisfied for the result of a function to be valid. That means as new specifications are found they can be added to a schema that defines the specification.*

Tool Neutrality*: If the results, or range of expected results, are known for a particular function, then it does not matter what tool is applied, but that results it returns for a known reference set can be measured.*

Tool Version Neutrality*: As a tool naturally develops over time, new versions are common, but the basic premise of this validation and verification regime means that the comments previously described for tool neutrality are also measurable.*

Transparency*: A set of known references described as a schema are therefore auditable and independently testable.*

The outcome of this approach is to focus the testing process such that practitioners needing to validate their tools can focus on the results obtained and determine if they are acceptable, rather than spending time and effort in attempting all functions under all conditions.

In a later paper, Guo et al. (2009) combine the 'Tool Neutrality' and 'Tool Version Neutrality' features of their original approach and add 'Detachability,' which allows for individual functions of a tool to be validated independently allowing for its partial utilisation prior to validation of the complete set of functions.

Flandrin et al.et al. (2014) suggest that existing methodologies for validation and testing are either too complicated to be useful or are limited in their coverage. They point out that in most cases the main evidence exists in the form of files, and therefore an essential part of any testing methodology needs to incorporate a standard digital corpus to prevent the testing data introducing a bias between different testers.

Garfinkel et al. (Garfinkel et al., 2009) identify the need for realistic test data in the forensic environment and comment that "*Having a reference set of representative corpora enhances the scientific evaluation of forensic methods beyond the obvious benefits of providing ready test data and enabling direct comparison of different approaches. Namely, it allows for the ground truth to be established using manual or otherwise time-consuming methods*", a view also supported by other researchers (Arshad, Jantan, & Abiodun, 2018; Baggili & Breitinger, 2015; Yannikos et al., 2014).

Not all electronic data that is relevant to digital forensics is unique to this environment. The contents of the registry, system-generated files and deleted data are potentially of interest in incident response due to its significant overlap of activities. Similarly, the contents of email and user-generated files may be relevant to researchers and practitioners such as those working in areas such as application development, electronic discovery (for legal matters), archiving and privacy audits.

In forensics, as for research, it is important that the results of testing are reproducible such that an independent researcher (or practitioner) can repeat the testing process and achieve the same results (Penrose, Macfarlane, & Buchanan, 2013). This is where having a stable test environment and appropriate test data is required (Garfinkel et al., 2009; Horsman, 2019; Yannikos & Winter, 2013).

However, finding suitable electronic data that can be reliably used for research (and, by extension, testing and evaluation) is challenging. The Third International Workshop on Building

Analysis Datasets and Gathering Experience Returns for Security held in 2014 contained a paper titled 'Are we missing labels? A study of the availability of ground-truth in network security research' (Abt & Baier, 2014) in which the authors identified that 70% of the papers they reviewed that had been submitted for conferences between 2009 and 2013 used data that had been manually compiled. The authors also found that most of the data created was not made publicly available.

While the research carried out by Abt and Baier (2014) focussed on network traffic data, other researchers identified the same issue with finding suitable data to use for evaluating forensic tools (Du et al., 2021; Göbel et al., 2020; Grajeda, Breitinger, & Baggili, 2017). In a paper presented at the 2017 Digital Forensics Research Workshop (DFRWS) titled 'Availability of datasets for digital forensics – and what is missing' (Grajeda et al., 2017) the researchers analysed 715 peer-reviewed research articles associated with digital forensics that had been published between 2010 and 2015. They assigned three attributes to the datasets that they identified:

Quality – correctly labelled, real-world (or close to real-world)

Quantity – sufficient data is provided to allow training/validation of approaches and tools.

Availability – allows other researchers and practitioners to replicate results.

In relation to the 'quality' of the datasets, the research showed that only 36.7% of the datasets they identified contained 'real-world' data, with the majority being generated as part of an experiment. In addition, as found in the earlier research by Abt and Baier (2014), only a small amount (3.8%) was made publicly available.

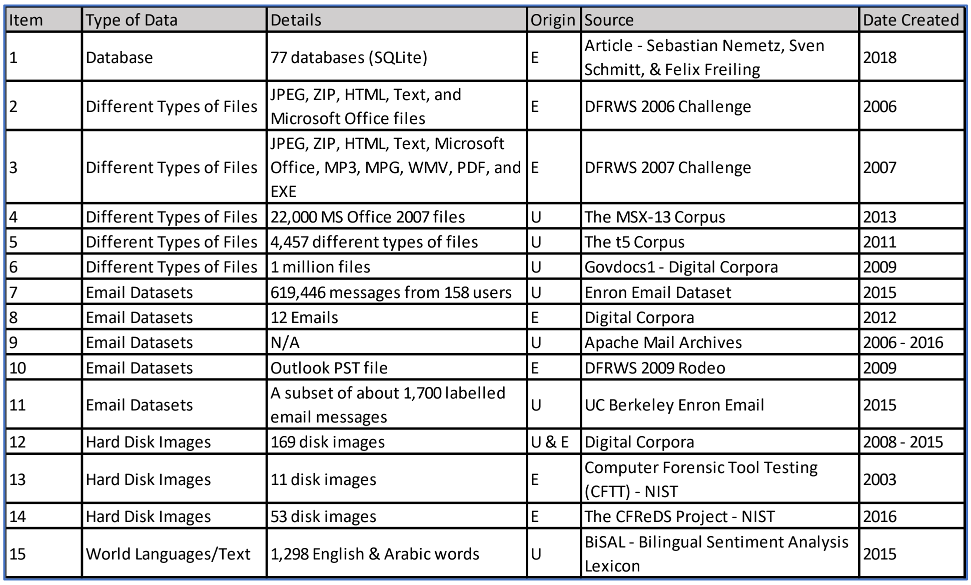
Following on from this research a website[20] was created by the University of New Haven as a resource for cyber forensic researchers which listed 82 data sets as of August 2023, with the most recent having been added in 2018. The information for the data sets includes a brief description of their content, their size, whether they were 'user generated,' 'experiment generated' or 'computer generated' as well as a link to the source and the date the data set was created.

Many of the data sets listed relate to aspects of digital forensics that are outside the scope of this research (namely remote agents on computers running a version of the Windows operating system) because they are associated with such areas as android devices, Apple-based devices, network packets, video game devices, malware, and chat logs. Table 2 shows a filtered list of data sets that

[20] https://datasets.fbreitinger.de/datasets/

might be relevant for use in this research on the basis that they could provide a basis for the endpoint systems on which to run the remote agents or provide 'seed' data:

Table 2 Subset of data sources - University of New Haven



Item	Type of Data	Details	Origin	Source	Date Created
1	Database	77 databases (SQLite)	E	Article - Sebastian Nemetz, Sven Schmitt, & Felix Freiling	2018
2	Different Types of Files	JPEG, ZIP, HTML, Text, and Microsoft Office files	E	DFRWS 2006 Challenge	2006
3	Different Types of Files	JPEG, ZIP, HTML, Text, Microsoft Office, MP3, MPG, WMV, PDF, and EXE	E	DFRWS 2007 Challenge	2007
4	Different Types of Files	22,000 MS Office 2007 files	U	The MSX-13 Corpus	2013
5	Different Types of Files	4,457 different types of files	U	The t5 Corpus	2011
6	Different Types of Files	1 million files	U	Govdocs1 - Digital Corpora	2009
7	Email Datasets	619,446 messages from 158 users	U	Enron Email Dataset	2015
8	Email Datasets	12 Emails	E	Digital Corpora	2012
9	Email Datasets	N/A	U	Apache Mail Archives	2006 - 2016
10	Email Datasets	Outlook PST file	E	DFRWS 2009 Rodeo	2009
11	Email Datasets	A subset of about 1,700 labelled email messages	U	UC Berkeley Enron Email	2015
12	Hard Disk Images	169 disk images	U & E	Digital Corpora	2008 - 2015
13	Hard Disk Images	11 disk images	E	Computer Forensic Tool Testing (CFTT) - NIST	2003
14	Hard Disk Images	53 disk images	E	The CFReDS Project - NIST	2016
15	World Languages/Text	1,298 English & Arabic words	U	BiSAL - Bilingual Sentiment Analysis Lexicon	2015

A review of the source reference links shown in Table 2 confirmed the following as of August 2023:

- Item 1 – The SQLite databases are available for download in a zip file.
- Item 2 – The DFRWS 2006 challenge data is available (via the Wayback Machine internet archive)
- Item 3 – The DFRWS Challenge data is available (via the Wayback Machine internet archive)
- Item 4 – The MSX-13 data is available.
- Item 5 - The t5 Corpus is no longer available (although it appears that this was just a subset of the Govdocs1 corpus).
- Item 6 – The Govdocs1 data is available.
- Item 7 – The Enron Dataset is available (although there are still various versions scattered across the internet).
- Item 8 – These emails are available as EML files that form part of a 2012 'insider threat' scenario.
- Item 9 - These emails are available, can be selected by domain, month, and year and then bulk downloaded in MBOX format.

- Item 10 - The DFRWS 2009 Rodeo PST is available (via the Wayback Machine internet archive)
- Item 11 – The subset of Enron emails is available as a TAR.GZ archive.
- Item 12 – The Digital Corpora disk images webpage was last updated in 2021. It is not clear where the figure of 169 disk images came from in Table 2, but multiple disk images are available including 750 images restricted to legitimate researchers.
- Item 13 – The link is to sourceforge.net (not NIST Computer Forensic Tool Testing) and the web page seems to have been updated several times since 2003 with additional images.
- Item 14 – There are 188 datasets available on the CFReDS website covering a range of different data types – including disk images (some of which are associated with DFRWS resources).
- Item 15 – An Excel spreadsheet is available containing lexicons in English and Arabic.

Using the information in Table 2 as a starting point a more detailed review of the data available for digital forensic testing was carried out. The results of this review are covered in the following sections with potential seed data and disk images being reviewed separately.

4.3.1 Forensic Test Data Repositories – potential seed data

Because of the Obama Administration's commitment to enabling public access to the results of publicly funded research some US agencies are supporting online repositories. The most relevant of these, for this research, are the NIST Computer Forensic Reference Data Set (CFReDS) initiative and the Information Marketplace for Policy and Analysis of Cyber-risk & Trust (IMPACT) project.

4.3.1.1 NIST

The CFReDS portal provides access to a range of data resources catering for different aspects of digital forensics. The data includes such resources as memory Images, disk images (some with associated scenarios), compressed DD images for deleted file recovery, user-generated registry files, SQLite databases, and a range of text files for testing search capabilities. In contrast to some of the other available data sets, many of those hosted by NIST have been recently produced.

The data sets are provided by NIST themselves as well as several other entities including commercial organisations, volunteer organisations, individuals, and academic institutions. Data sets are grouped by source and may contain several different types of data. The following is a list of some of the available data sets that can be accessed in addition to those provided by NIST:

- Cellebrite (2021) – iPhone and Android images
- VTO Labs – various IoT device data
- DFRWS (2006/7/8/9/11 via Wayback Machine Archives) – Game console images, USB Flash drive images, RAM dumps, Network traffic, Smartphone images, email (PST), various file types and fragmented files.
- Columbia University – Images and video files
- Arizona University – Video files
- University of Florence – Images
- Apache Software Foundation – emails (mbox)
- The MSX-13 corpus – random sample of MS Office 2007 files downloaded from the internet (docx, xlsx, pptx only)
- Nemetz, Schmidt & Freiling – SQLite databases

All the sources in the list were available via the links on the CFReDS website as of August 2023, with many of them duplicating data referenced in Table 2.

- Cellebrite (2021) – iPhone and Android images

This research does not involve mobile phone data, so these images are not relevant.

- VTO Labs – various IoT device data

This research does not involve IoT data, so these data are not relevant.

- DFRWS (2006/7/8/9/11 via Wayback Machine Archives) – Game console images, USB Flash drive images, RAM dumps, Network traffic, Smartphone images, email (PST), various file types and fragmented files.

Of the data made available from DFRWS, only the email and file-type data are relevant to this research. However, given the age of some of the data its usefulness for evaluating tools designed to be deployed to current (and future) live machines is limited.

- Columbia University – Images and video files

This research does not involve images and video files beyond the potential need to identify and capture by file extension or file type.

- Arizona University – Video files

This research does not involve video files beyond the potential need to identify and capture by file extension or file type.

- University of Florence – Images

This research does not involve images beyond the potential need to identify and capture by file extension or file type.

- Apache Software Foundation – emails (mbox)

The mbox email data is potentially useful as seed data.

- The MSX-13 corpus – random sample of MS Office 2007 files downloaded from the internet (docx, xlsx, pptx only)

The data from this source is potentially useful as seed data.

- Nemetz, Schmidt & Freiling – SQLite databases

4.3.1.2 Information Marketplace for Policy and Analysis of Cyber-risk & Trust (IMPACT)

This is an initiative of the US Department of Homeland Security (DHS) to facilitate the sharing of data that can be used by cyber security researchers, academia, government, and industry (although some data is restricted to academic researchers). The repository is intended to hold data associated with a "…large spectrum of devices and systems"[21].

Contributors to the repository are listed as:

- Carnegie Mellon University (CMU)
- Centre for Infrastructure Assurance and Security (UTSA/CIAS)
- Colorado State University (CSU)
- DARPA
- DHS Cybersecurity and Infrastructure Security Agency (DHS CISA)

[21] https://www.dst.defence.gov.au/our-technologies/information-marketplace-policy-and-analysis-cyber-risk-trust-impact

- Galois, Inc. (Galois)
- Georgia Tech (GT)
- Massachusetts General Hospital
- MIT Lincoln Laboratory (MIT)
- Naval Postgraduate School (NPS)
- Parsons, Inc
- SKAION
- UCSD - Centre for Applied Internet Data Analysis (CAIDA)
- University of Southern California-Information Sciences Institute (ISI)
- University of Wisconsin (UW)

Other providers include the organisers of the National Collegiate Cyber Defense Competition (nccdc.org).

The data is all available via a keyword search and/or filter based on provider. However, the data is mostly associated with identifying malicious network activity and includes:

- Databases of internet marketplace information
- Packet capture logs in various formats from different environments (including medical)
- Email in various formats (mostly mbox)
- Firewall logs

Although the stated intention is to hold a wide variety of data, so far, the content has a heavy bias towards network traffic and is therefore an unlikely candidate for providing seed data to be placed on the endpoints unless there is a particular email file format not included elsewhere.

4.3.1.3 Digital Corpora - Real Data Corpus (RDC)

Having identified the need for ‘real’ data, Garfinkel et al. (2009) created the Real Data Corpus which they describe on the website as being a resource for:

- Developing and validating forensic and data recovery algorithms and tools.
- Developing and validating document translation software.
- Exploring and characterizing real-world computing practices, configuration choices, and option settings.
- Studying the storage allocation strategies of file systems under real-world conditions

The corpus contains more than 750 images from purchased disks and are split between those relating to "US Persons" and those relating to "non-US Persons". Unfortunately for typical practitioners, this corpus received funding from the U.S. Government and as such *"…use of the RDC is limited to bonafide researchers operating under the oversight of an Institutional Review Board that has a DoD Assurance"*[22]. This restriction makes this corpus unsuitable for providing a basis for images that could be used as network endpoints.

4.3.1.4 Digital Corpora – NPS-2010-emails

Digital corpora provide a disk image that is intended to be used for tools that use string searches to find email addresses. There are 30 different email addresses spread across multiple documents using different coding. Both Apple and Windows-based documents are included – mainly Microsoft Word 2008 (for Apple) and Microsoft Word/PowerPoint/Excel 2007 (for Windows) together with PDFs.

The format of the image means that it will need to be mounted and the potential seed data extracted separately. Given its focus on email addresses and encoding this is a poor candidate for providing seed data.

4.3.1.5 Digital Corpora - Scenarios

A collection of scenarios featuring multiple disk images (including portable devices), network traffic and memory dumps is provided. The scenarios and descriptions listed on the site as of January 2022 are:

- 2008 M57-Jean – A single disk scenario involving the exfiltration of corporate documents from an executive's laptop.
- 2008 Nitroba University Harassment Scenario – A fun-to-solve network forensics scenario.
- 2009 M57-Patents – A complex scenario involving multiple drives and actors set at a small company over the course of several weeks.
- 2012 National gallery DC – a fictional attack on the National Gallery DC, foiled in 2012.

[22] https://digitalcorpora.org/corpora/disk-images/real-data-corpus

- 2018 Lone Wolf Scenario – A scenario involving the seizure of the laptop of a fictional person planning a mass shooting.

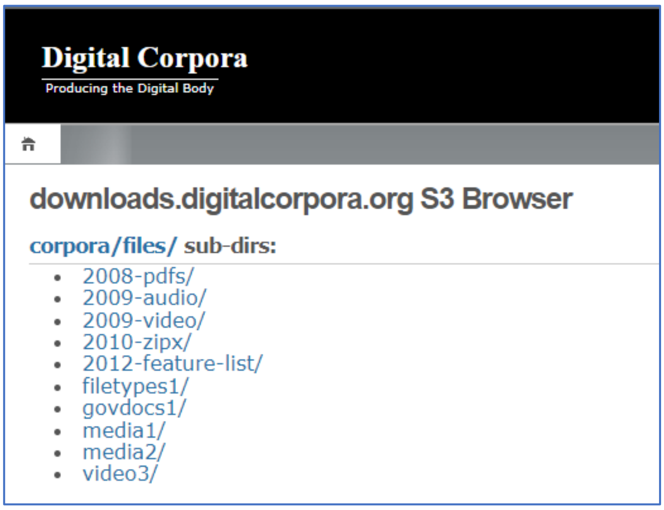
Although the main page was last updated in November 2021 most of the images are more than 10 years old and the most recent is based on a scenario requiring analysis of cloud artefacts.

4.3.1.6 Digital Corpora – Govdocs1

In 2010 Digital Corpora created a corpus of 1 million documents that are made freely available for research as well as being free for distribution (subject to disclaimer). These documents were collected from web servers in the .gov domain using randomly chosen search strings.

4.3.1.7 Digital Corpora – File Types

Collections of various file types are made available for download as shown in Figure 27 (which includes the Govdocs1 data mentioned above. The 'filetypes1' sub directory contains a collection of zip files sorted and named by file extension, a sample of which is shown in Figure 28



Digital Corpora

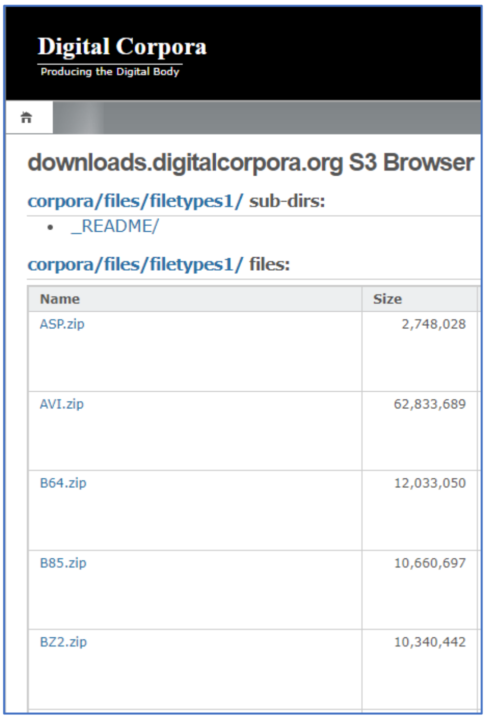
Producing the Digital Body

downloads.digitalcorpora.org S3 Browser

corpora/files/ sub-dirs:

- 2008-pdfs/
- 2009-audio/
- 2009-video/
- 2010-zipx/
- 2012-feature-list/
- filetypes1/
- govdocs1/
- media1/
- media2/
- video3/

Figure 27 Digital Corpora - file types



Digital Corpora

Producing the Digital Body

downloads.digitalcorpora.org S3 Browser

corpora/files/filetypes1/ sub-dirs:

- _README/

corpora/files/filetypes1/ files:

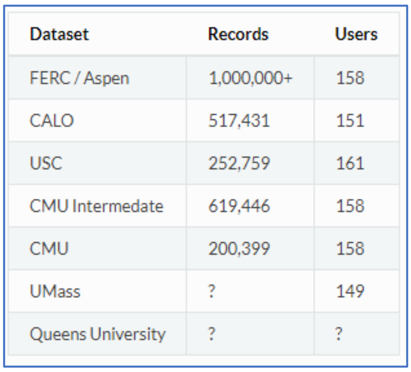
Name	Size
ASP.zip	2,748,028
AVI.zip	62,833,689
B64.zip	12,033,050
B85.zip	10,660,697
BZ2.zip	10,340,442

Figure 28 Digital Corpora - zip files for each file extension

4.3.1.8 Enron Corpus

Probably the most well-known set of real data relating to email is that of the Enron Corpus[23]. This comprises entirely of email linked to (mostly) senior management within Enron taken from their email servers by the Federal Energy Regulatory Commission (FERC) after Enron's collapse in 2001.

There are several versions available online although there are differences between the number of users involved and the number of emails contained in the data set depending on the source (as shown in Table 3).



Dataset	Records	Users
FERC / Aspen	1,000,000+	158
CALO	517,431	151
USC	252,759	161
CMU Intermediate	619,446	158
CMU	200,399	158
UMass	?	149
Queens University	?	?

Table 3 Enron Email Datasets from enrondata.org

[23] https://www.cs.cmu.edu/~enron/

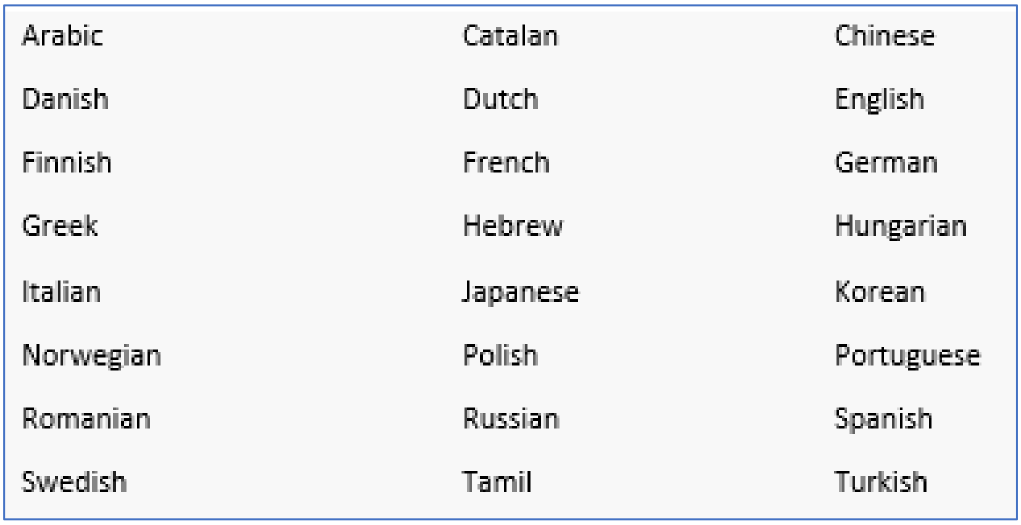
Over the years there have been several updates to the corpus to remove personal or sensitive data. The first release of the corpus was in 2004, followed by other versions in 2009 and 2011 with the current version being released to the public in 2015. The latest version does not include attachments and has had 23 messages removed since the previous version.

The format of the emails also differs between versions and sources, although the most useful format for use as seed data for this research is the 148 custodian PST files made available by EnronData.org. This data set comes in the form of a 734MB 7z archive that decompresses to 8.6GB.

4.3.1.9 EDRM Internationalization Data Set

The EDRM Global Advisory Council are focussed on electronic discovery and provide data sets for practitioners to practice and test their tools. One of these data sets is the EDRM Internationalization Data Set which is provided on their website[24]. This data set contains over 700MB of email from an Ubuntu mailing list archive that consists of email data from 23 languages.

The languages are:



Arabic	Catalan	Chinese
Danish	Dutch	English
Finnish	French	German
Greek	Hebrew	Hungarian
Italian	Japanese	Korean
Norwegian	Polish	Portuguese
Romanian	Russian	Spanish
Swedish	Tamil	Turkish

4.3.1.10 EDRM File Formats Data Set

The latest version of this data set[25] is provided in a zip file. It is structured into folders whose contents are referenced in an accompanying spreadsheet. However, there seems to be some mixing

[24] https://edrm.net/resources/data-sets/
[25] Downloaded 15 August 2020 from https://edrm.net/resources/data-sets/edrm-internationalization-data-set/

of file types within the folders with the ‘bmp’ folder also containing files with the ‘cdr’,’ico’ and ‘cur’ extensions.

The file date on the folders is 1 January 2009 and so the file formats are too far out of date to be useful as seed data for this research (although there is mention on the EDRM website stating that there is a project to create a new generic data set).

4.3.2 Summary of potential seed data review

The review has shown that not only are there very few sources of potential seed data that would be suitable for testing remote forensic tools in a Microsoft Windows environment, but most of this data is also out of date with respect to many of the file formats used.

A further limitation is the fact that they are collections designed for a limited purpose and because many contain ‘realistic’ data they do not provide a broad range of file types. For instance, they may contain only a single version of a particular MS Office document (which have changed formats from binary to XML-based) or, in the case of emails, contain a limited number of different formats (typically only a single mail format such as a PST or individual MBOX files).

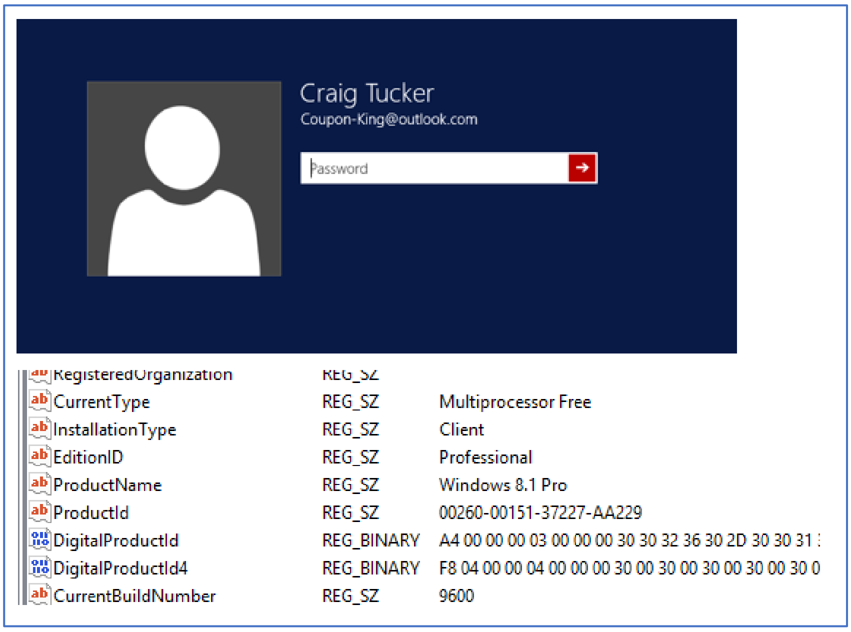
4.3.3 Potential endpoint images

4.3.3.1 California Cybersecurity Institute (CCI)

The Institute provides practice resources (available for downloading online) for their digital forensic training[26]. The computer images include “Laptop Image” (5.4GB) and “Additional Practice Image” (7.0GB) that have been compressed into zip files.

The Laptop Image can be mounted using Arsenal Image Mounter and a virtual machine created in windows Hyper-V. However, as can be seen in Figure 29, the image is that of a machine running the obsolete Windows 8.1 operating system, which renders it unsuitable for this research.

[26] https://cci.calpoly.edu/2019-digital-forensics-downloads



Craig Tucker
Coupon-King@outlook.com

RegisteredOrganization	REG_SZ	
CurrentType	REG_SZ	Multiprocessor Free
InstallationType	REG_SZ	Client
EditionID	REG_SZ	Professional
ProductName	REG_SZ	Windows 8.1 Pro
ProductId	REG_SZ	00260-00151-37227-AA229
DigitalProductId	REG_BINARY	A4 00 00 00 03 00 00 00 30 30 32 36 30 2D 30 30 31
DigitalProductId4	REG_BINARY	F8 04 00 00 04 00 00 00 30 00 30 00 30 00 30 00 30 0
CurrentBuildNumber	REG_SZ	9600

Figure 29 The CCI Laptop Image converted to a virtual machine.

4.3.3.2 NIST

With respect to disk images that could potentially be used as the basis for endpoints in a test environment, NIST have produced some prototype test data sets of which the most useful are those associated with the Hacking Case and the Data Leakage Case. What makes the two hard disk images that form part of the two cases of interest is the set of questions associated with the images based on the seeded data they contain and the scenario for which they were created. However, the Hacking Case data is dated 2004 and the more recent Data Leakage Case has an image of an obsolete Windows 7 operating system dating from 2015.

4.3.3.3 Digital Corpora – NPS Test Disk Images[27]

Links are provided on the Digital Corpora website to a set of disk images that have been created for the purpose of testing and evaluating tools used in computer forensics. The images are freely available and the website states that all *"These images are free of non-public Personally Identifiable Information (PII)"* as well as being free of copyright restrictions (and where copyright exists this has been made freely available by the copyright holder).

As of January 2022, the following images (and their descriptions) are available from the website:

[27] https://digitalcorpora.org/corpora/disk-images

- nps-2009-canon2 — A set of images taken on with a Canon digital camera that can be used to test basic file recovery, fragmented file recovery, and file carving.
- nps-2009-casper-rw — An ext3 file system from a bootable USB token that had an installation of Ubuntu 8.10. The operating system was used to browse several US Government websites.
- nps-2009-hfsjtest1 — A test image of a journaled HFS system in which the data from a previous version of a file can only be recovered from the HFS journal.
- nps-2009-ntfs1 — A test image of an NTFS file system including unfragmented and highly fragmented files stored in raw, compressed, and encrypted directories. The decryption key is provided.
- nps-2009-ubnist1 — The FAT32 file system from which the nps-2009-capser-rw disk image was extracted.
- nps-2009-domexusers — This is a disk image of a Windows XP SP3 system that has two users, domexuser1 and domexuser2, who communicate with a third user (domexuser3) via IM and email. Two versions of this disk image will be provided:
- nps-2009-domexusers – The full system, distributed as an encrypted disk image.
- nps-2009-domexusers-redacted – The full system with the Microsoft Windows executables redacted so that they cannot be executed.
- nps-2010-emails — is a test disk image consists of 30 different email addresses, each one stored in a different document with a different coding scheme.
- nps-2014-usb-nondeterministic – this is a series of disk images that were made from a USB storage device that produced different data each time it was read. The original submission ZIP file and narrative are presented, as well as E01 files that were created by extracting the raw files from the ZIP image and re-encoding them.

As can be seen from the descriptions, most of the images are over 10 years old and consist of data from a camera, USB devices and computer systems and disks containing several different file systems. The only operating system mentioned are the obsolete Windows XP SP3 and Ubuntu 8.10 (current version is 22.04 LTS).

4.3.4 Summary of potential endpoint images

The available ‘forensic images’ are not suitable for use as endpoint systems as they are either of the wrong operating system or are too out of date (or both) and would involve the user in a lot of effort trying to bring them up to current versions of Microsoft Windows and subsequently incorporate them into a network for evaluation. Therefore, to develop a comprehensive framework such as AFERA, a new dataset needed to be created.

4.4 CHAPTER CONCLUSION

This chapter has covered the Design and Development stage of the DSRP and has involved the identification of the required features of a remote forensic tool by drawing on information provided by two authoritative sources (NIST and SANS) as well as a panel of expert practitioners (Stage 1).

The required features from the NIST and SANS sources were determined by reference to those artefacts that were involved in their testing or forensic analysis procedures associated with 'standard' (i.e., not remote) forensic tools. For the requirements of forensic practitioners, they explicitly identified the required features which were divided into two categories, 'functional' and 'non-functional'.

Stage 2 of this chapter examined the availability of potential seed data and numerous corpora were identified and examined for their suitability. It was found that these data suffered from several limitations and often consisted of relatively old file formats or were designed for specific purposes that limited the range of file types that were included. It was also found that the available endpoint images are not suitable for incorporating into an evaluation framework.

The next chapter builds upon the information gathered in this chapter to develop the artefact.

5 DEVELOPING THE ARTEFACT

5.1 INTRODUCTION

This chapter continues the ‘Design and Development’ stage of the DSRP model and builds upon the findings in previous chapters, the output from the Expert Practitioner Panel interviews and other authoritative resources to build the ‘artefact’ which, for this research, will be a new framework for enabling digital forensic practitioners evaluate forensic tools that use remote agents. The framework will be called the Advanced Framework for Evaluating Remote Agents (AFERA).

AFERA will include:

- Recommendations for setting up and configuring an evaluation environment in which to deploy the remote agent tools being evaluated.
- A suitable corpus designed to provide the necessary artefacts for testing the functionality of the remote agent tools.
- Criteria for evaluating the remote agent tools.

This chapter will use the following structure:

- Selection of the tool environment
- Building a suitable evaluation corpus
- Creation of evaluation criteria

5.2 SELECTION OF THE TOOL EVALUATION ENVIRONMENT

On the basis that the scope for this research is limited to MS Windows networks the previous chapter focussed on the available MS Windows system images that could potentially serve as endpoint machines for the evaluation environment of AFERA. The review found them to be unsuitable for several reasons, particularly because of their use of old/obsolete versions of operating system. Similarly, previous attempts to enable the creation of system images was also found to be unsuitable given their complexity, programming requirements and the time it would take to build a suitable network with them.

A review of Microsoft Developer resources was undertaken to determine if Microsoft might provide a suitable solution (or one that could be adapted) for use with AFERA through their

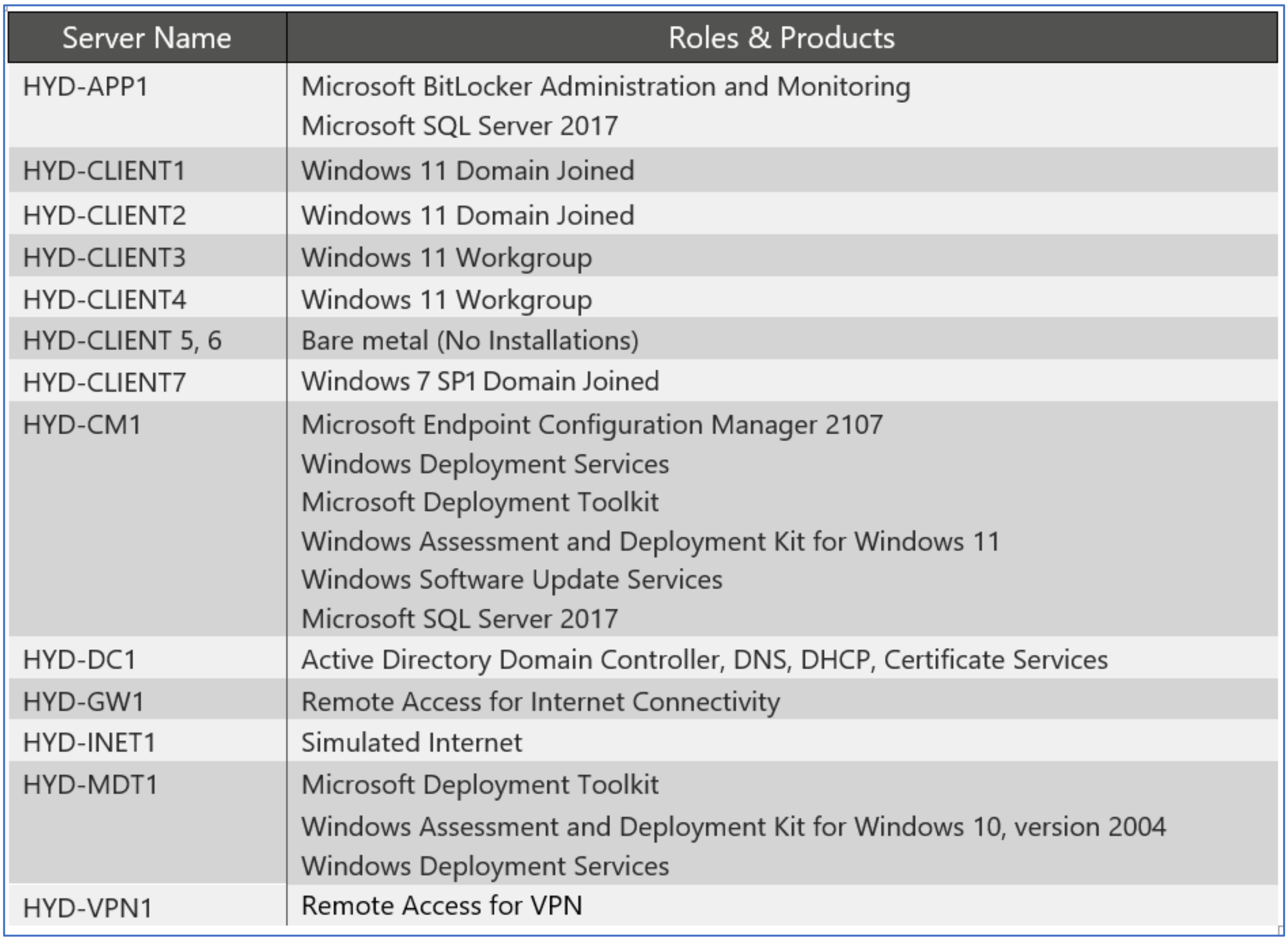
Evaluation Center. During this review, the Windows Deployment Labs were identified as being a suitable environment for practitioners to evaluate remote deployed forensic tools. The key reasons that this environment has been selected are:

- It is free and publicly available – this makes it easy for practitioners to obtain.
- Both Microsoft Windows 10 and Windows 11 environments are available and include the associated Windows Server operating systems, with the latest Lab version being updated twice a year – this ensures that it is always relevant to the environment in which the tools are likely to be deployed.
- It utilises Microsoft Hyper-V virtual machines that can be run on a single host – this helps practitioners by limiting the requirement for physical machines.
- It sets up a domain network with typical servers and stand-alone machines automatically – this makes it easier for practitioners to use and provides an environment that can be tailored if necessary to match specific configurations.
- It does not require significant resources in terms of memory, storage, or processing power.

The Windows 11 and Office 365 Deployment Lab consists of the following:

- Windows 11 Enterprise
- Windows 7 Enterprise Service Pack 1, Version 6.1
- Microsoft Endpoint Configuration Manager 2107
- Windows Assessment and Deployment Kit for Windows 11
- Microsoft Deployment Toolkit
- Microsoft BitLocker Administration and Monitoring 2.5 SP1
- Windows Server 2022 Standard Evaluation

The virtual machines that are created provide the roles and products shown in Figure 30:



Server Name	Roles & Products
HYD-APP1	Microsoft BitLocker Administration and Monitoring Microsoft SQL Server 2017
HYD-CLIENT1	Windows 11 Domain Joined
HYD-CLIENT2	Windows 11 Domain Joined
HYD-CLIENT3	Windows 11 Workgroup
HYD-CLIENT4	Windows 11 Workgroup
HYD-CLIENT 5, 6	Bare metal (No Installations)
HYD-CLIENT7	Windows 7 SP1 Domain Joined
HYD-CM1	Microsoft Endpoint Configuration Manager 2107 Windows Deployment Services Microsoft Deployment Toolkit Windows Assessment and Deployment Kit for Windows 11 Windows Software Update Services Microsoft SQL Server 2017
HYD-DC1	Active Directory Domain Controller, DNS, DHCP, Certificate Services
HYD-GW1	Remote Access for Internet Connectivity
HYD-INET1	Simulated Internet
HYD-MDT1	Microsoft Deployment Toolkit Windows Assessment and Deployment Kit for Windows 10, version 2004 Windows Deployment Services
HYD-VPN1	Remote Access for VPN

Figure 30 List of virtual machines included in the Deployment Lab

The accompanying guide with the Lab includes instructions for setting up and configuring cloud access if required.

Microsoft describe the resources produced by the Deployment Lab as an isolated environment which *"…is ideal for exploring deployment tool updates and testing your deployment-related automation"*[28].

Horsman (2018) comments that the environment under which the testing takes place is critical as it influences the three variables of accuracy, repeatability and applicability required for forensic work. The use of the Windows 10 or 11 and Office 365 Deployment environment addresses Horsman's repeatability, and applicability variables by:

1. Having an automated installation process (repeatability).

[28] https://docs.microsoft.com/en-us/microsoft-365/enterprise/modern-desktop-deployment-and-management-lab?view=o365-worldwide

2. Generating the latest versions of Windows networked systems that have been pre-configured by Microsoft to emulate a generic enterprise Windows domain structure and which can form the basis for replicating a specific environment (applicability).

Horsman's remaining variable, accuracy, is addressed by using known data with pre-determined outcomes that should be produced by the remote agents (discussed later).

The selection of a virtual environment based on the Windows 10 or 11 and Office 365 Deployment Lab is in line with the approach of previous researchers (reviewed earlier) for the generation of digital forensic test environments as well as the teaching of digital forensic practitioners through the provision of specially constructed virtual test environments (Han, Harries, & Brown, 2013; Maximov & Karasik, 2014) with the suggested benefits being:

- Portability – able to be run on the students' own computers or departmental machines.
- Low cost – after the initial setup time has been expended setting up the environment no other costs are incurred (using open-source platforms).
- Have no security constraints – the environment is self-contained and therefore the user can be assigned super-user privilege without having the potential to cause harm on the hosts system.
- No compatibility issues with other software
- It is easy to restore the original settings (useful if "learning by breaking" is involved)

Virtualisation has also been proposed as a practical tool for evaluation of malware forensic tools (Elisan, 2015; Malin, Casey, & Aquilina, 2008). Kennedy (Kennedy, 2017) points out that when reviewing the functioning of malware using a specific forensic tool in a virtual environment it provides the benefits of scalability and throughput while also being quick to reset.

Digital forensic professionals are now exposed to virtual environments on a regular basis both during the course of their work and in their training making it a suitable choice of environment for the implementation of AFERA (Barrett & Kipper, 2010; Mrdovic, Huseinovic, & Zajko, 2009).

5.3 BUILDING A SUITABLE EVALUATION DATA SET (SEED DATA)

Shiaeles et al. (2013) referenced NIST and SANS in relation to defining the metadata and artefacts that should be involved in testing forensic tools. Following on from Shiaeles et al. (2013), and to address any potential interpretation and researcher bias in relation to the data collection and analysis criteria (Easterbrook, Singer, Storey, & Damian, 2008), this research uses criteria from three sources in the development of AFERA to define the functional requirements of a digital forensic tool that uses remote agents (as discussed in the previous chapter):

- Source 1: Requirements for investigating a simulated 'real' case were obtained by using the data and questions from a current NIST CFReDS forensic tool test scenario – The Data Leakage Case.
- Source 2: Generic requirements for DFIR investigations were obtained from the Artifact Analysis information associated with the SANS Windows Forensic Analysis course FOR500.
- Source 3: Practitioner requirements were obtained through a series of interviews with expert practitioners working in a variety of environments.

Given that the aim is to be able to evaluate a tool that uses remote agents to process/collect data from multiple endpoints the mechanics of how the tool operates is immaterial in relation to building the seed data corpus. For instance, in relation to searching, the same seed data can be used regardless of whether a tool processes an artefact remotely with the agent to identify any search hits or it uses the agent to create an index at a central point that is then used to identify any search hits.

For this research, if a tool 'processes' an artefact this means that the artefact can ultimately be searched and viewed. A tool may not undertake all the processing itself but may involve another tool or utility after the artefact has been identified and collected. The view may be a sub-set of the artefact, i.e., an individual email from within a PST file, or the entire artefact such as a registry hive.

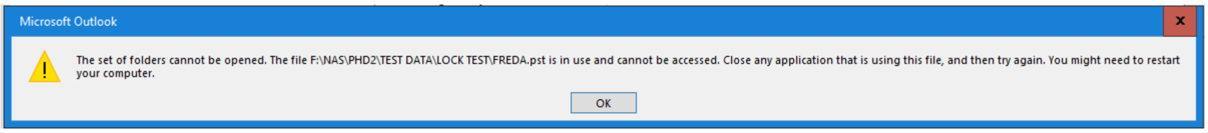
5.3.1 File locking

An important difference between processing artefacts from a 'dead' image and those from a 'live' system is that on a running machine the operating system will lock (and hide) some of the operating system artefacts that may be of interest to a digital forensic practitioner. Furthermore, for a machine in the process of being operated by an end user there may be other files that are locked by various applications, for example, PST or OST email containers, that employ a temporary 'lock file'.

For a forensic tool to work effectively on a live system it needs to be able to bypass the locking mechanisms on all artefacts. For testing purposes, the fact that the endpoint system is running means that the operating system is already locking certain artefacts. However, to check that application-generated files that are locked can be processed, it would be ideal if the relevant applications are running on the endpoints with some files open that contain known search terms (particularly for email containers).

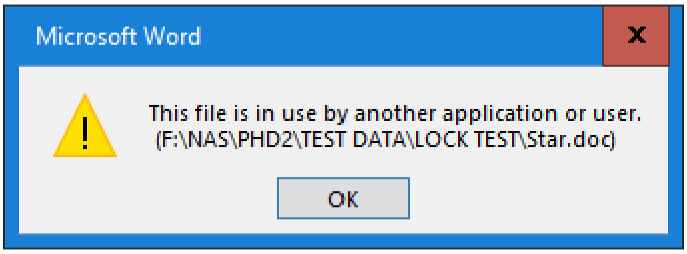
A simple alternative to installing and running numerous applications on the endpoints to test the ability of a remote agent to bypass the file locking mechanism was investigated. The investigation identified the command-line utility 'FileLocker'[29] which was tested on the Windows 11 and Office 365 Deployment Lab virtual machines and found to be effective.

FileLocker was used to lock a PST file included as part of the proposed seed data which caused a message to be displayed when attempting to access the container through Outlook:



The set of folders cannot be opened. The file F:\NAS\PHD2\TEST DATA\LOCK TEST\FREDA.pst is in use and cannot be accessed. Close any application that is using this file, and then try again. You might need to restart your computer.

Similarly, attempting to open a Word document using Microsoft Word that had been locked with FileLocker also caused an error message to be displayed:



This file is in use by another application or user. (F:\NAS\PHD2\TEST DATA\LOCK TEST\Star.doc)

The FileLocker utility was tested on Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows 7, 10 and 11 operating systems using Administrator credentials. It was also tested on both Microsoft Windows application files (Word, Excel etc.,) and LibreOffice files (Writer, Calc etc.,). In all cases it worked correctly without causing any errors.

5.3.2 Foreign language handling

In Chapter 4 the processing requirements were divided into categories for the SANS and NIST sources and two 'characteristics' for the Expert Practitioner Panel's requirements (high-level and

[29] https://www.jensscheffler.de/filelocker

low-level). Within the high-level characteristics were two items that directly affect the contents of the seed data:

1. The ability to be able to search languages other than English.
2. The ability to identify encrypted/password protected files.

No specific languages were mentioned in the Expert Practitioner Panel interviews, but in a draft document NIST refer to five languages as part of their language testing in their Federated Testing programme:

1. Unicode Chinese/Japanese ideograms
2. Unicode Korean Hangul
3. Unicode Japanese phonetic Kana
4. Unicode Russian Cyrillic
5. Unicode right-to-left Arabic

NIST provide test data in the form of 'text' files together with a separate file allowing for copy/paste to ensure that the strings are entered correctly. The approach used by NIST will be adopted to meet the language search requirements of the Expert Practitioner Panel.

5.3.3 Password-protected files

An assortment of password-protected and encrypted files will be provided to meet the identification requirements of the Expert Practitioner Panel. These will be versions of test documents that have been password-protected and/or encrypted using their native application.

5.3.4 Addressing specific requirements from the Expert Practitioner Panel

Requirement 1 - Search and process files

No specific files were mentioned. As discussed in Chapter 4, test corpora should contain multiple versions of the same file in different versions of an application. Furthermore, taking MS Word as an example, in addition to being able to find a seeded search term in the main body of a document, a forensic tool should also be able to locate the term within:

- The document metadata
- An embedded OLE object (if applicable to the file type) which should iterate through three levels.

Only a ‘plain’ search term will be incorporated to determine if different file types can be searched on the basis that if a tool can find the term in all the locations identified above in a ‘plain’ search it will find any other search term its search engine can locate in those same locations, for example using a ‘wildcard’ option which will be tested separately.

File types excluded from the evaluation of the search functionality include those that were identified by the Expert Practitioner Panel as needing to be captured (see ‘Capture capabilities’ section below).

Requirement 2 - Search and process email:

No detail in relation to the content of the email was mentioned by the Expert Practitioner Panel. In addition to having a seeded search term within and email itself, a forensic tool should also be able to locate the term within attachments to that email.

Requirement 3 - Search and process compressed files:

A selection of compressed file formats will be included in the seed data based on the NIST Container Types data set.

Requirement 4 - Search options

The plain text tests that will be established for Requirement 1 will be extended to test the tool’s functionality in terms of different search options.

As part of their string search test data NIST provide Unicode strings that can test a forensic tool’s ability to recognise Normalization Form D (NFD – Canonical Decomposition) and Normalization Form C (NFC – Canonical Decomposition followed by Canonical Composition). These are relevant for testing the handling of composite characters when searching (such as those with an accent). These NFC and NFD strings will be included as part of the seed data.

Tests for the following search types will be developed:

- Plain search
- Wildcard search
- Whole word search
- Beginning of word search
- End of word search
- REGEX search

- HEX sequence search

Requirement 5 - Capture capabilities

The capture functionality will determine a tool’s capability to locate and capture files that may be locked by the file system and/or require a particular file type to be correctly recognised.

- System memory – switch from dynamic
- Swap file – new Windows 10
- Pagefile
- Hibernation file
- Registry hives
- Event logs
- SRUDB
- Jumplist files
- Prefetch files
- Windows LNK files
- Executables and DLLs
- Windows system logfiles
- NTFS $MFT
- NTFS $USNJournal
- Recycle bin
- Windows Error Reporting (WER) artefacts
- RDP Cache
- Windows EDB (Search)
- Browser artefacts

Requirement 6 - Identify encrypted/password protected files.

In many types of investigation, the presence of these files is of interest, and they will need to be collected for separate processing.

This requirement is met by ensuring that the deployed agent can ‘flag’ encrypted, or password protected files. This could be in the form of a report for follow-up action or be associated with the ability to detect, categorise, and capture these types of files.

5.3.5 Creation of evaluation criteria

Typically, there are two types of requirements identified in Requirements Engineering. The features and capabilities that an entity, such as a system, should exhibit are classified as functional Requirements (FR), while quality constraints such as security, usability and performance are classified as non-functional requirements (NFR) (Becker, Tebes, Peppino, & Olsina, 2019).

Both functional and non-functional requirements have been incorporated into the Framework as part of a more comprehensive tool evaluation process. This research therefore goes beyond addressing the essential need to test and validate forensic tools in relation to core functionality in that it creates a framework that enables practitioners to evaluate different tools by assessing both functional and non-functional criteria.

The high-level practitioner requirements are accommodated by the Framework as non-functional requirements and are summarised below:

1. **Accuracy and reliability**

   For forensic, i.e., court-related matters, it is essential that the output from the processing meets the standards necessary for it to be considered as evidence.

   By using a known data set and a specific set of search criteria the ability of a remote agent to locate and collect artefacts can be measured simply by confirming that the correct number of files/emails are identified and collected.

2. **Scalability**

   The ability to scale is necessary as cases often involve multiple systems. This could have a significant impact on bandwidth, storage, and time as the number of concurrent agents increase.

   This is a complex parameter to determine. Many of the tools that deploy remote agents suggest that there are no limitations on the number of agents deployed. However, in practice the volume of data coming to a central point, for instance during the creation of an index, will be limited by the network infrastructure.

   As discussed by Duboc et al. (Duboc, Rosenblum, & Wicks, 2007) there is a lack of "…a clear, consistent and systematic treatment of scalability" and they state that the term is poorly defined and poorly understood within computer literature. They also point out that this makes the task of comparing different claims from different sources very difficult and declare that scalability "…is, in general, a highly complex attribute of systems requiring sophisticated analysis techniques."

Duboc et al. (2007) developed a framework for "…precisely characterizing and analysing the scalability of a software system" which treats scalability as a problem in multi-criteria optimization. Their framework was applied to different system designs and was (in keeping with their comments) somewhat sophisticated.

Rygg et al. (Rygg, Brataas, Millstein, & Molle, 2013) undertook scalability testing in relation to Microsoft Lync services to establish optimal provisioning of virtualised hardware. This is the opposite scenario of the testing environment in which the remote agents will be used in this research as the provisioning of hardware resources will be fixed such that a comparison of different software can be made based on its effect on the system performance. However, Rygg et al.'s (2013) scalability testing considered resource under-provisioning, which would be the case if a remote agent deployment were saturating one or more of the system resources and would be characterised by a degradation in end user service 'Quality of Experience' and the creation of a 'bottleneck.'

The Quality of Experience was measured by Rygg et al. (2013) using a compound metric that they called the Mean Opinion Score (MOS). This consisted of transport layer values (such as packet loss) and 'payload parameters' relating to the quality of audio service measured by such things as noise level and echo.

The results of the Rygg et al. (2013) study considered four 'bottleneck resources':

- Memory – the point at which memory was being swapped out to disk.
- Network – network utilisation
- Disk – in relation to disk utilisation.
- CPU – the CPU utilisation was closely correlated with system performance.

Jogalekar and Woodside (2000) considered scalability within remote systems and produced a metric based on cost-effectiveness which they describe effectiveness as a function of a system's throughput and quality of service. They point out that many remote systems must be scalable to cater for such things as differing numbers of nodes, data volumes and geographical coverage.

They state that scalability is not simply the ability to operate, but the ability to operate efficiently while maintaining an acceptable level of service within the stated range of environments in which they will be used. However, one of the key aspects of the framework they propose separates the evaluation of 'quantity of work' from 'quality of service.'

This separation is relevant to the subject of this research as the concept of quality of service equates to the time taken for the search/collection process for a given number of remotely deployed agents. This is an indirect measure that captures the effect of bottlenecks in resources caused by inefficient processing by recording longer times to complete a given task. In relation to consideration of quantity of work, one tool may cause more work to be undertaken than another tool, for instance due to the technology of its database engine running on the server, but the ability to process the output from the deployed agents may be slower than that of the other tool.

Jogalekar and Woodside (2000) sought to identify a strategy for scaling up the resources for remote systems to remove bottlenecks, whereas this research is only focussing on identifying the relative likelihood that different tools may cause bottlenecks within a particular environment.

Confais et al. (Confais, Arslan, & Parrein, 2022) identify three domains that limit the efficiency of data location across multi-site environments. These are stated as scalability, the ability to deal with changes to network topology and constraints in the data naming process.

Confais et al. (2022) disagree with Jogalekar and Woodside's use of the Quality-of-Service metric and instead consider that in the distributed data environment certain database and connection metrics are more appropriate. However, they make a statement that "Scalability is the ability of a process, a network, or a software program to grow and meet the increased demand without any observable performance degradation" which, regardless of the metric used to measure performance, is a useful generic definition that is consistent with the other researchers.

While accepting that scalability is a "complex attribute" the intention of this research is to provide practitioners with a simple framework with which to evaluate the performance of different remote agents for the purpose of digital forensics and incident response investigations. The identification of bottlenecks is a key feature of previous research around scalability (Confais et al., 2022; Duboc et al., 2007; Jogalekar & Woodside, 2000; Rygg et al.,

2013) and therefore, in addition to having a simple 'time to complete task' metric, this research will also utilise Rygg et al.'s (2013) bottleneck resources to identify where potential limitations of scalability might become apparent for a particular tool.

The Framework utilises five workstations on which the remote agents are deployed and require the network load to be monitored as well as the total time taken to complete a specific task. This will provide a comparison against other tools.

The use of five workstations is not arbitrary. Personal experience has shown that when there are three remote agents transferring data to a server it is common for other agents to be paused until a 'slot' becomes free. Having five agents deployed should replicate this situation.

3. **Minimal impact on end user.**

   This functionality will be inferred by the requirement to undertake monitoring of the local system resources given that a remote agent that causes a high CPU or memory usage is likely to have a detrimental effect on user applications and therefore impact the user.

4. **Speed**

   The time to complete the work is a crucial factor given that the skilled practitioner resources are expensive.

   The Framework will suggest recording the elapsed time for deployment and the 'evaluation run,' i.e., setting the main task for the tool. This will enable a useful comparison of tools that 'queue' deployed agents to reduce impact on network bandwidth or due to restrictions imposed by an indexing engine against those tools that have less impact on network bandwidth, more efficient indexing, or no indexing function.

5. **Dependencies**

   Consideration should be given in relation to the agent's dependencies, i.e., any other software that must be installed on the endpoints or the 'server' for the agent to perform.

   A free-text section has been allocated for recording any other software that has to be installed and/or system configurations that are required to deploy the agents.

6. **Ease of use**

   In terms of General Characteristics, the time it takes for a practitioner to learn how to use and deploy the remote agents is very important.

The Framework is not intended to provide a structured assessment of this aspect of the requirements; therefore, a free-text section has been allocated for the practitioner to record their experience of deploying the agents.

7. **Customisation**

   The ability to change the configuration of the tool means that unnecessary functions are not performed. These options might include the ability to select/deselect:

   - System files, e.g., registry, event logs, USN Journal
   - ‘Ignorable’ files, e.g., operating system files
   - Image and multi-media files
   - Deleted files.
   - Large files such as virtual hard disks

8. **Auditability**

   For court purposes and in other situations, the ability is needed to audit the agent’s processing, but not a blow-by-blow record of activities.

   This requirement might be accommodated using logs associated with the deployed agent’s activity.

9. **Results output options**

   The form in which data is collected is important, such as in encrypted and hashed containers to maintain evidence integrity.

The forensic environment requires that the integrity of data collected as part of an investigation is always maintained. Typically, forensic software will create a ‘forensic physical image’ or ‘forensic logical image’ that serve as repositories for collected artefacts. Deployed agents are unlikely to be used to acquire physical images of remote systems but will require the ability to create ‘forensic grade’ containers in which to store the artefacts that may have originated in different parts of the remote system’s filesystem.

10. Cost

This was highlighted as being a big issue. It is often the case that free tools, while performing most tasks very well, are missing essential features that are only found in expensive commercial products. The high cost of some commercial products renders them unsuitable for a lot of cases.

Although cost is an important consideration for potential buyers of forensic tools, the Framework being developed in this research will only require consideration of this aspect of the tool's evaluation rather than attempt to introduce a form of criteria, such as 'value for money.'

11. Handling machines not on enterprise network

The ability to run the deployed agent from, for example, a USB device that can be mailed/couriered to and from the remote location forms part of the evaluation framework.

12. Invisible to end user

The ability to undertake covert search and collection processing is an important feature, especially when undertaking investigations that may involve company staff. By implication, a tool that is invisible to the end user does not require them to undertake any actions for the tool to operate. For instance, if a tool cannot process email while the user has their email client open, the activities of the deployed agent would not be 'invisible' to that user.

In addition, if the resource requirements of a deployed agent are such that they cause the remote systems to slow down to the extent that they interfere with the normal work of the end users this would also prevent them from being categorised as being invisible to the end user.

This functionality is evaluated in the Framework through the monitoring of the resource utilisation at the endpoints and noting any pre-requisites or other requirements needed to deploy the agent.

13. Cloud Deployment capability

Deployment to the cloud can be evaluated using the Framework but the full process involves the creation of a test tenant for Azure AD and some trial licences.

14. Data sovereignty

This requirement is only relevant for hosted solutions where the collected data is stored/processed in the cloud. The Framework includes consideration of the location of any stored cloud data.

15. Minimal impact on network bandwidth

The Framework requires monitoring of the network utilisation of the systems on which the remote agents have been deployed. This will provide a rough indication of the theoretical total number of endpoints that can be processed on a given network for a given bandwidth.

5.3.6 Translating high-level requirements into the Framework.

The high-level requirements were renamed 'non-Functional Considerations' as a set of criteria, and several of the original high-level requirements were moved to a set of criteria named 'Functional Considerations' as they could be included in specific tests. The remaining non-Functional Considerations are:

1. Speed – the time to complete the work.
2. Dependencies - consideration should be given in relation to the agent's dependencies, i.e., any other software that must be installed for the agent to perform.
3. Ease of use - in terms of General Characteristics, the time it takes for a practitioner to learn how to use and deploy the remote agents.
4. Customisation – the ability to change the configuration of the tool such that unnecessary functions are not performed.
5. Auditability – the ability to audit the agent's processing.
6. Results output options.
7. Cost – the cost of licencing.
8. Handling machines not on enterprise network.
9. Invisible to end user.
10. Can be run on systems in the cloud.
11. Data sovereignty – for instance, output data must be capable of being restricted to Australian cloud storage.

The previous high-level items that have been moved to the Functional Considerations Criteria are:

- Scalability - the ability to run multiple concurrent search/collections. This may be estimated by having the forensic tool search through the entire file system while monitoring the network load at the central collection machine.

- Minimal impact on end user – for the Framework this will be inferred by the resource requirements at the endpoints running the remote clients.
- Identify encrypted/password protected files – the seed data will contain encrypted files for identification by the tool.
- Language Support - the ability to support languages other than English. The seed data will include the key languages used by NIST as previously identified.

Accuracy and Reliability may be determined by reference to the results of the combined functional tests.

5.4 CHAPTER CONCLUSION

The key topic of this chapter has been the development of AFERA (the DSR artefact) which consists of an evaluation environment, a set of instructions for using AFERA, a set of evaluation criteria and an associated data set (the seed data).

This chapter has identified an appropriate environment for forensic practitioners to use when evaluating forensic tools that may be deployed across a network. This environment is relatively easy to create and will remain relevant to systems likely to be encountered by forensic practitioners given that it is maintained by Microsoft to the latest Windows build and is updated every 6 months.

A set of instructions were created (Appendix 1 – The Framework Guide Document (initial version)) describing how the required features may be assessed using the evaluation environment and the data set. The two sets of criteria are included as Appendix 2 - The Non-functional Criteria (initial version) & Appendix 3 - The Functional Criteria (initial version).

The next stage of the DSRP model is the Demonstration stage which involves a ‘bench test’ of the built artefact where it can be assessed for its potential usefulness. This stage will also help to refine the artefact prior to giving it to the practitioners who will make up the Expert Reviewers Panel.

6 DEMONSTRATION OF THE FRAMEWORK

6.1 INTRODUCTION

The DSRP model requires that prior to the Evaluation stage, a Demonstration stage needs to take place in which the artefact (in this case the Framework) is used to solve a problem in an appropriate context (J. Venable, Pries-Heje, & Baskerville, 2017).

This activity helps to ensure that the Framework is practical, i.e., has utility. For the Demonstration stage a 'bench check' was undertaken using the Windows 11 and Office 365 Deployment Lab Kit as the operating environment and following the Framework's documentation.

The hardware used for the evaluation was a server (the Demonstration Server) with the following specification:

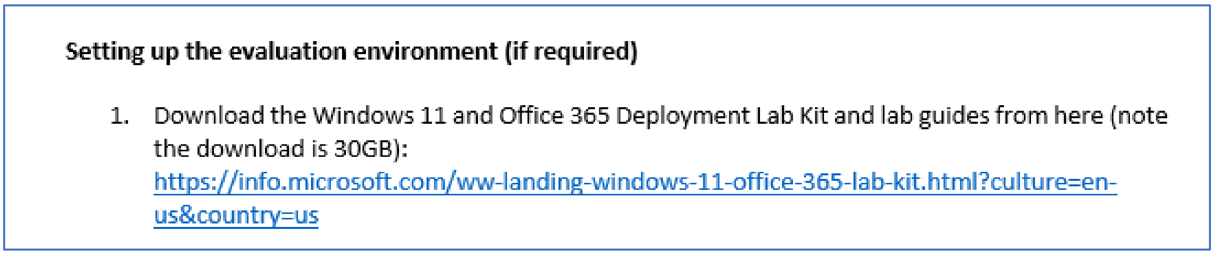
- Intel i7 4-core CPU running at 3.5GHz
- 32GB RAM
- 125GB SSD system disk
- 1TB Seagate HDD available for the evaluation
- Microsoft Server 2016 Standard operating system

This machine was a re-purposed digital forensic workstation and from the researcher's experience it represents an unexceptional system for a digital forensics lab[30].

6.2 DOWNLOADING THE EVALUATION ENVIRONMENT

The Framework suggests using the Microsoft Windows 11 and Office 365 Deployment Lab Kit as the operating environment, so the first step for the Demonstration stage of this research was to visit this Microsoft site using the link provided in the document 'Using the Framework' (Figure 31).

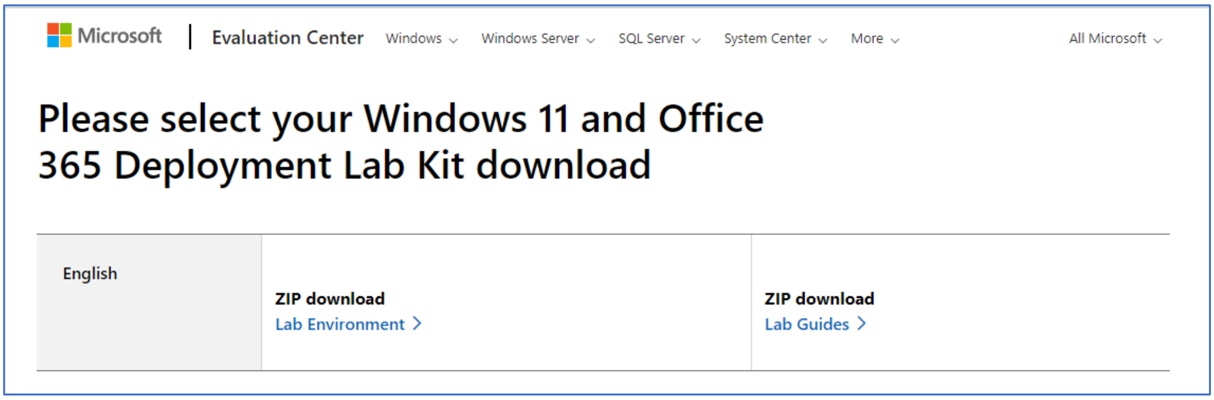
[30] https://siliconforensics.com/products/forensic-workstations.html



Setting up the evaluation environment (if required)

1. Download the Windows 11 and Office 365 Deployment Lab Kit and lab guides from here (note the download is 30GB): https://info.microsoft.com/ww-landing-windows-11-office-365-lab-kit.html?culture=en-us&country=us

Figure 31 Extract from Using the Framework document.



Please select your Windows 11 and Office 365 Deployment Lab Kit download

English	**ZIP download** Lab Environment >	**ZIP download** Lab Guides >

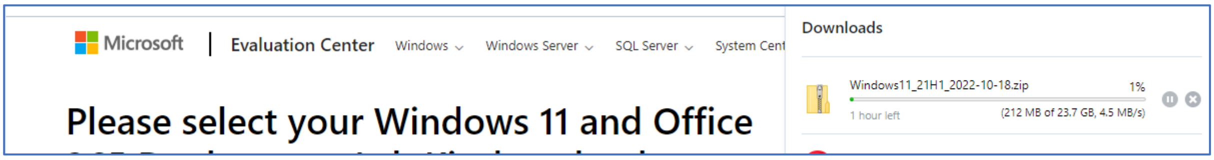
Figure 32 Microsoft site for Deployment Lab Kit

In relation to the pre-requisites for running the Lab, Microsoft provides the following note on the download site[31]:

> Please use a broadband internet connection to download this content and allow approximately 30 minutes for automatic provisioning. The lab environment requires a minimum of 16 GB of available memory and 150 GB of free disk space. For optimal performance, 32 GB of available memory and 300 GB of free space is recommended.

The Demonstration Server met the 'optimal' requirements for running the Lab. The site requires registration to get access to the Deployment Lab Kit. Following registration, the Zip file took 65 minutes to download on a broadband network tested at 54 Mbps download speed based on the results provided by M-Lab and speedtest.net. The Lab Guides were also downloaded, but these took only a few seconds. This makes acquiring the environment for running evaluations a relatively trivial exercise.

[31] https://docs.microsoft.com/en-us/microsoft-365/enterprise/modern-desktop-deployment-and-management-lab?view=o365-worldwide



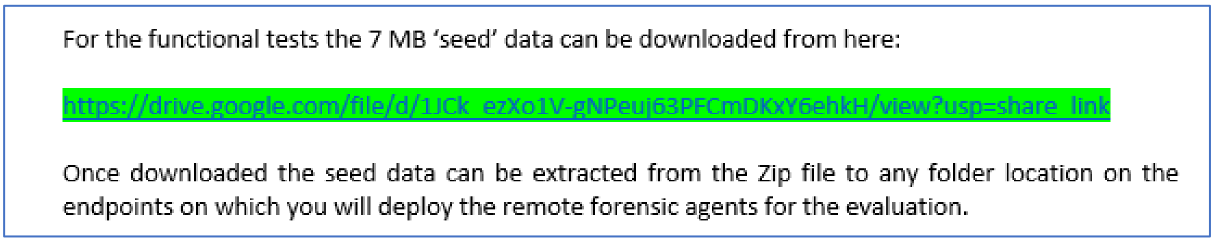
Please select your Windows 11 and Office

Windows11_21H1_2022-10-18.zip 1%
1 hour left (212 MB of 23.7 GB, 4.5 MB/s)

Figure 33 Downloading the Lab Kit Zip file.

6.3 Downloading the Seed Data

The seed data was downloaded using the link provided in the document Using the Framework (Figure 34). This process took a few seconds (Figure 35 & 36).

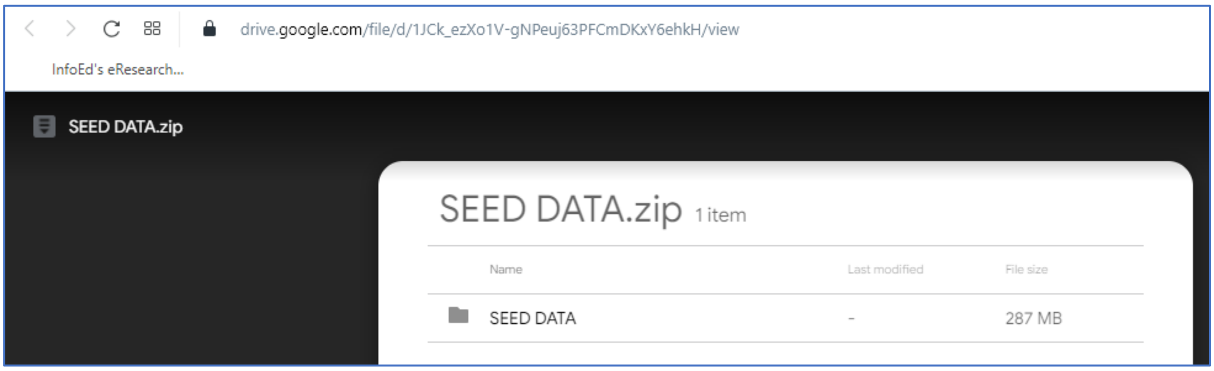


For the functional tests the 7 MB 'seed' data can be downloaded from here:

https://drive.google.com/file/d/1JCk_ezXo1V-gNPeuj63PFCmDKxY6ehkH/view?usp=share_link

Once downloaded the seed data can be extracted from the Zip file to any folder location on the endpoints on which you will deploy the remote forensic agents for the evaluation.

Figure 34 Extract from Using the Framework document.

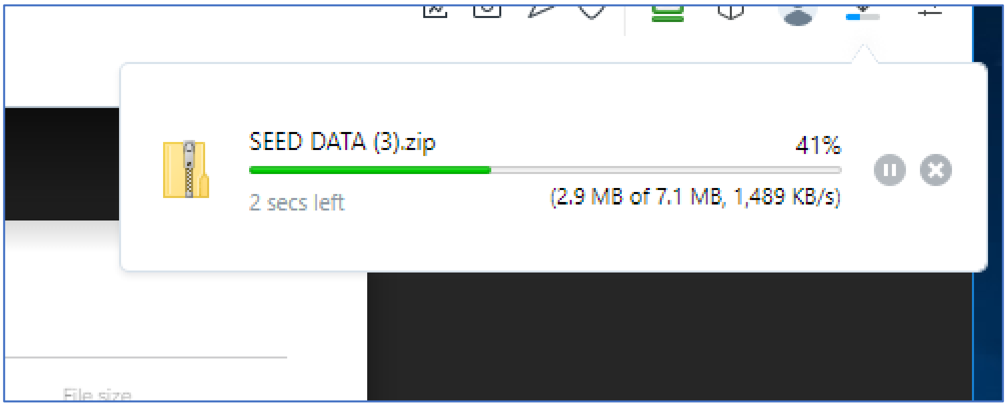


drive.google.com/file/d/1JCk_ezXo1V-gNPeuj63PFCmDKxY6ehkH/view

SEED DATA.zip 1 item

Name	Last modified	File size
SEED DATA	-	287 MB

Figure 35 Target of seed data download link



SEED DATA (3).zip

41%

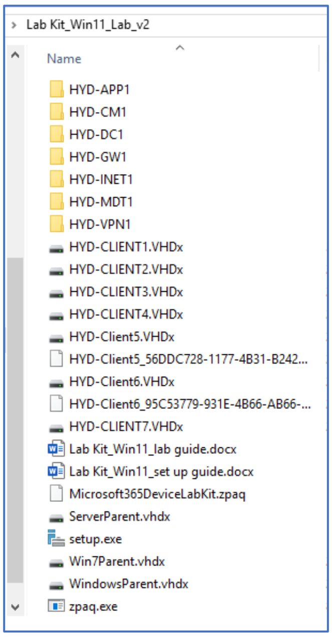
2 secs left

(2.9 MB of 7.1 MB, 1,489 KB/s)

Figure 36 Downloading seed data.

6.4 Setting up the evaluation environment

Having downloaded the Lab Kit, the instructions in the 'Lab Kit Win 11 set up guide' document were followed, starting with extracting the zip file to a suitable location and running the setup process. This took 2hrs and 15 minutes to complete (a relatively insignificant amount of time), at which time the contents of the installation folder contained the virtual machines shown in Figure 37.



> Lab Kit_Win11_Lab_v2

Name

HYD-APP1
HYD-CM1
HYD-DC1
HYD-GW1
HYD-INET1
HYD-MDT1
HYD-VPN1
HYD-CLIENT1.VHDx
HYD-CLIENT2.VHDx
HYD-CLIENT3.VHDx
HYD-CLIENT4.VHDx
HYD-Client5.VHDx
HYD-Client5_56DDC728-1177-4B31-B242...
HYD-Client6.VHDx
HYD-Client6_95C53779-931E-4B66-AB66-...
HYD-CLIENT7.VHDx
Lab Kit_Win11_lab guide.docx
Lab Kit_Win11_set up guide.docx
Microsoft365DeviceLabKit.zpaq
ServerParent.vhdx
setup.exe
Win7Parent.vhdx
WindowsParent.vhdx
zpaq.exe

Figure 37 Extracted Lab zip file contents.

Executing 'setup.exe' added the Lab virtual machines as shown in the Hyper-V Manager list of virtual machines (Figure 38).

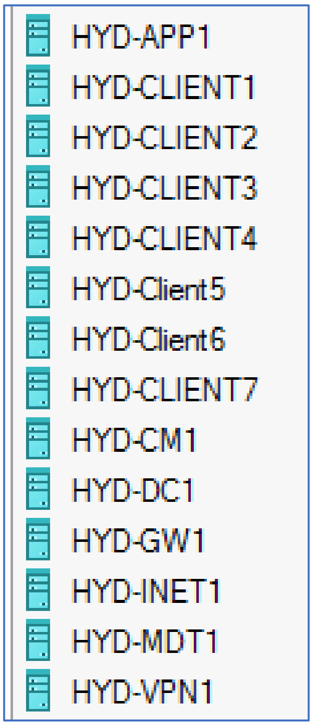
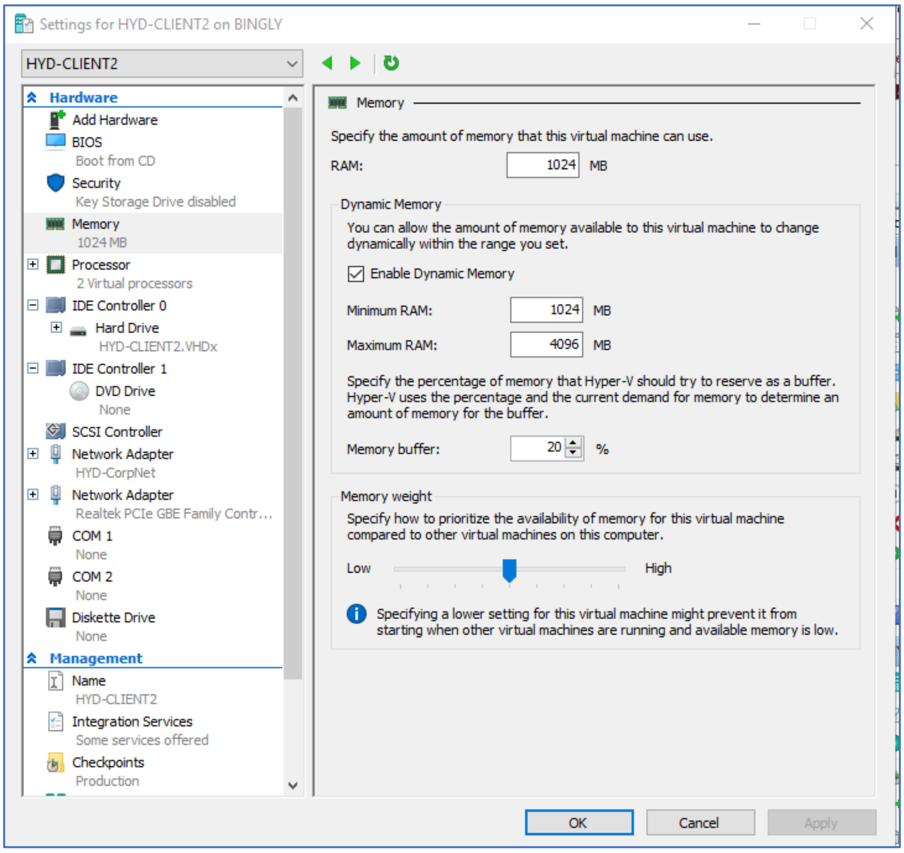

HYD-APP1
HYD-CLIENT1
HYD-CLIENT2
HYD-CLIENT3
HYD-CLIENT4
HYD-Client5
HYD-Client6
HYD-CLIENT7
HYD-CM1
HYD-DC1
HYD-GW1
HYD-INET1
HYD-MDT1
HYD-VPN1

Figure 38 Virtual machines created using Lab setup.

6.5 RUNNING THE EVALUATION ENVIRONMENT

The default settings for the Lab virtual machines have them configured to the private network HYD-CORPNET with only 2 virtual processors assigned and dynamic memory enabled (Figure 39). An entry was made to the record of this evaluation activity to add a requirement to the Using the Framework document to disable dynamic memory and set it to 4GB, both to allow for the capture of RAM[32] and to mitigate any effects of memory load balancing that might occur during the evaluation process.

[32] Previous testing had shown that RAM capture tools assume the RAM to be static and fail when they encounter dynamic memory.



Settings for HYD-CLIENT2 on BINGLY

HYD-CLIENT2

Hardware

- Add Hardware
- BIOS
  Boot from CD
- Security
  Key Storage Drive disabled
- Memory
  1024 MB
- Processor
  2 Virtual processors
- IDE Controller 0
  - Hard Drive
    HYD-CLIENT2.VHDx
- IDE Controller 1
  - DVD Drive
    None
- SCSI Controller
- Network Adapter
  HYD-CorpNet
- Network Adapter
  Realtek PCIe GBE Family Contr…
- COM 1
  None
- COM 2
  None
- Diskette Drive
  None

Management

- Name
  HYD-CLIENT2
- Integration Services
  Some services offered
- Checkpoints
  Production

Memory

Specify the amount of memory that this virtual machine can use.

RAM: 1024 MB

Dynamic Memory

You can allow the amount of memory available to this virtual machine to change dynamically within the range you set.

☑ Enable Dynamic Memory

Minimum RAM: 1024 MB

Maximum RAM: 4096 MB

Specify the percentage of memory that Hyper-V should try to reserve as a buffer. Hyper-V uses the percentage and the current demand for memory to determine an amount of memory for the buffer.

Memory buffer: 20 %

Memory weight

Specify how to prioritize the availability of memory for this virtual machine compared to other virtual machines on this computer.

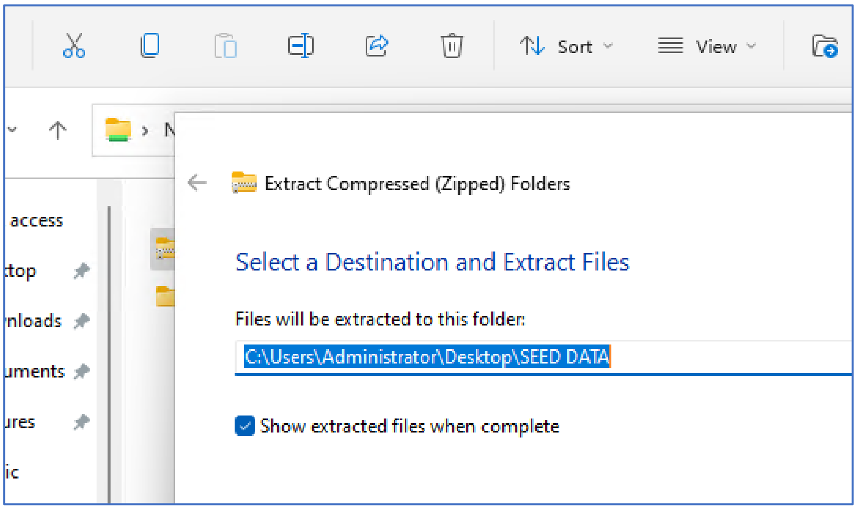
Low — High

Specifying a lower setting for this virtual machine might prevent it from starting when other virtual machines are running and available memory is low.

Figure 39 Example of default settings for Lab virtual machines

As the Lab is only being used as a testing platform the document 'Lab Kit Win11 lab guide' was not needed as it covers setting up and configuring the various virtual machine workstations and servers for specific use cases. An entry was made to the record of this evaluation activity to add a reference to this Microsoft document in the Using the Framework document to alert practitioners to the information available to configure the Lab machines to cater for more advanced environments.

As a starting point, the domain controller server (DC1) and two of the workstations (HYD-CLIENT1 and HYD-CLIENT2) were started (both running Windows 11). The server DC1 has a share already configured, 'mdop', and it was here that the seed data zip file was placed after a simple 'copy and paste' from the downloaded folder. On each of the workstations the seed data was extracted to the desktop from the server shared folder (Figure 40).



Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

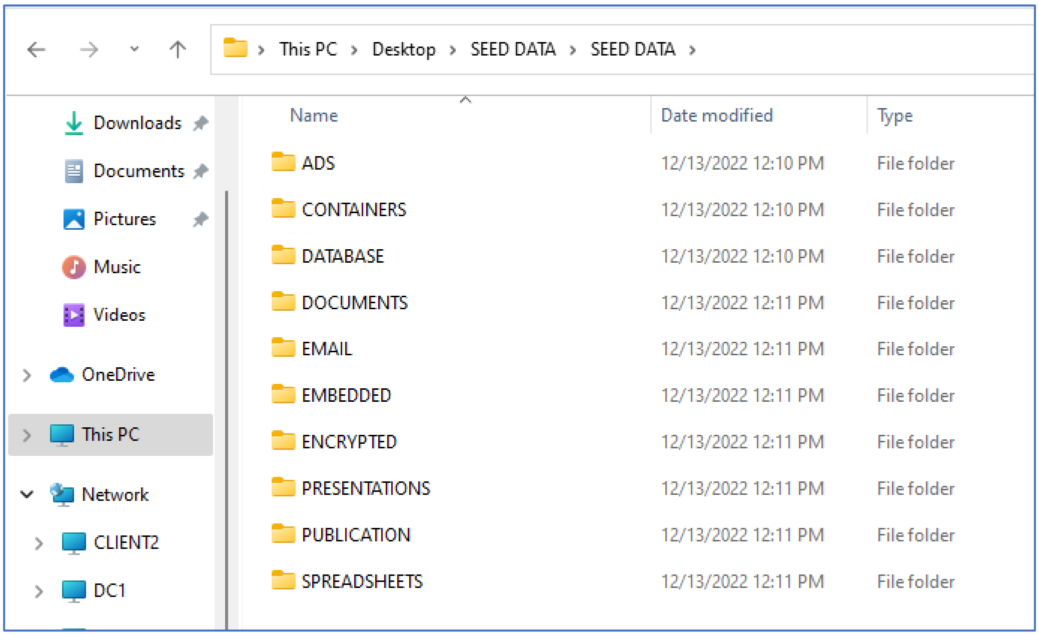
Files will be extracted to this folder:

C:\Users\Administrator\Desktop\SEED DATA

☑ Show extracted files when complete

Figure 40 Extracting seed data on a workstation.

The extraction process took less than 2 minutes for both workstations and resulted in the seed folders being stored on the desktop for the logged-in user (Figure 41).



This PC › Desktop › SEED DATA › SEED DATA ›

Name	Date modified	Type
ADS	12/13/2022 12:10 PM	File folder
CONTAINERS	12/13/2022 12:10 PM	File folder
DATABASE	12/13/2022 12:10 PM	File folder
DOCUMENTS	12/13/2022 12:11 PM	File folder
EMAIL	12/13/2022 12:11 PM	File folder
EMBEDDED	12/13/2022 12:11 PM	File folder
ENCRYPTED	12/13/2022 12:11 PM	File folder
PRESENTATIONS	12/13/2022 12:11 PM	File folder
PUBLICATION	12/13/2022 12:11 PM	File folder
SPREADSHEETS	12/13/2022 12:11 PM	File folder

Figure 41 Seed data folders on a workstation

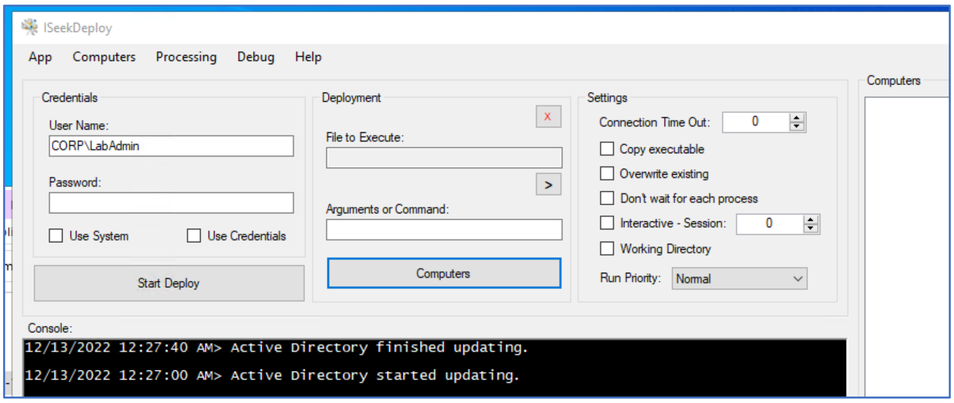
6.6 Evaluation of Forensic Tool

The tool chosen for the evaluation uses ‘independent’ remote agents that are not installed but run based on instructions provided by a configuration file (although any tool could have been used for this exercise).

A checkpoint was created on the two workstation VMs prior to running the first evaluation.

Many of the non-Functional Considerations can be obtained from technical literature associated with a particular tool, with only the unstructured ‘speed’ and ‘ease of use’ results being based on running a tool. Therefore, only the Functional Considerations were involved in the Demonstration activity to assess the utility of the Framework based on its use of the Windows Lab environment, the sets of criteria and associated seed data.

The agent deployment tool was copied to the ‘mdop’ share on the server DC1. This tool serves as a graphical user interface for PsExec. The deployment tool was run on the server DC1 causing it to load the contents of the Active Directory (Figure 42). This process took around a minute to complete.



User Name: CORP\LabAdmin

Password:

□ Use System □ Use Credentials

File to Execute:

Arguments or Command:

Connection Time Out: 0

□ Copy executable

□ Overwrite existing

□ Dont wait for each process

□ Interactive - Session: 0

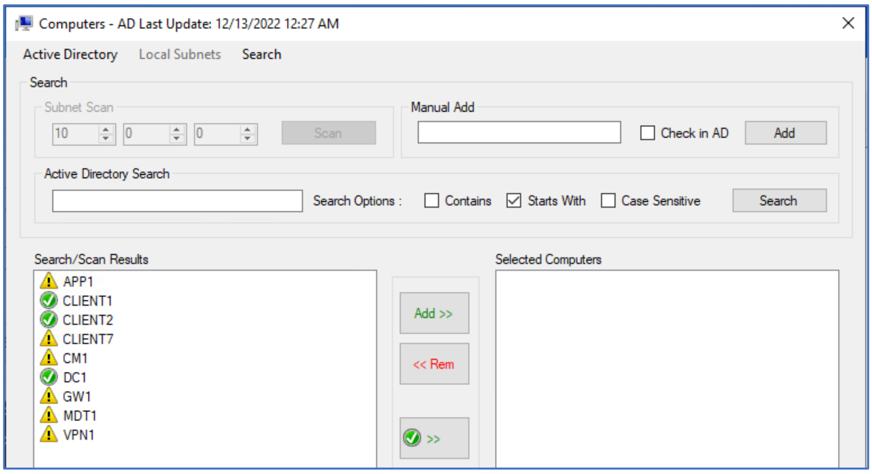
□ Working Directory

Run Priority: Normal

```
12/13/2022 12:27:40 AM> Active Directory finished updating.
12/13/2022 12:27:00 AM> Active Directory started updating.
```

Figure 42 Deployment tool - AD loading

The list of computers was selected which showed the three running computers on the test domain (the server and two workstations) (Figure 43).



Computers - AD Last Update: 12/13/2022 12:27 AM

Subnet Scan: 10 . 0 . 0

☐ Check in AD

Search Options : ☐ Contains ☑ Starts With ☐ Case Sensitive

Search/Scan Results:

- ⚠ APP1
- ✓ CLIENT1
- ✓ CLIENT2
- ⚠ CLIENT7
- ⚠ CM1
- ✓ DC1
- ⚠ GW1
- ⚠ MDT1
- ⚠ VPN1

Figure 43 List of AD computers

The configuration utility was installed on the server DC1. This process took 20 minutes. The results of the searches/collections were configured to be stored on the server DC1.

6.6.1 Functionality Check List

The following table shows the results of using the AFERA in a simulated ‘real-life’ situation whereby a forensic tool that uses remote agents is checked against the functional requirements when being run in the evaluation environment.

Test#	Criteria	Expected	Actual
1	Plain text metadata search for the term ‘FREDA’	2	2
2	Plain text search for the term ‘casseopeia’	54	53
3	Plain text search for the term 中国 東京 (Unicode Chinese/Japanese ideograms)	1	2
4	Plain text search for the term 서울 (Unicode CJK Korean Hangul)	1	1
5	Plain text search for the term スバル みつびし (Unicode CJK Japanese phonetic Kana)	1	1
6	Plain text search for the term Сибирь (Unicode Cyrillic (Russian))	1	1

7	Plain text search for the term الكسكس (Unicode RTL (Arabic))	1	1
8	Plain text search for the term Mäuse (Normalized Form D)	1	1
9	Plain text search for the term Mäuse (Normalized Form C)	1	1
10	Wildcard search for the term *glycerin	1	0
11	Whole word search for the term glycerin	0	0
12	Beginning of word search using the term Nitro	1	1
13	End of word search using the term glycerin	1	1
14	Regex search using the expression: \d{3}-\d{2}-\d{4} (to find 333-22-4444)	3	3
15	Hexadecimal sequence search using: 6F6C6576616E	1	1
16	Case sensitive search using the term Craig TuCkeR	1	1
17	Search embedded documents – Level 1 (Term: Embedded Level1)	1	1
18	Search embedded documents – Level 2 (Term: Embedded Level2)	1	1
19	Search embedded documents – Level 3 (Term: Embedded Level3)	1	1
20	Search email using the term “Buffalo Gap”	12	12
21	Identify encrypted files.	2	2

Notes

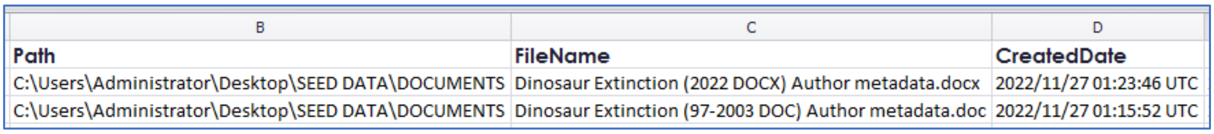
Test 1 - Plain text metadata search for the term ‘FREDA’.

As a reference point the tool was deployed and configured to search across the entire system disk in all types of files for the term ‘FREDA’. This resulted in an unacceptable processing time. Upon investigation, it was found the slow speed was because the VM contains numerous large WIM, CAB and other compressed files that must be decompressed to be searched.

Although a search in practice might involve the entire disk, it is often the case that system directories and certain file types are excluded to quickly identify user-related files that may be relevant. Digital forensic tools typically can exclude items to be searched based on a range of criteria, or conversely, allow for certain folders/file types to be targeted while excluding everything else.

Given that the purpose of the seed data is to test the ability to apply different searches to specific file types there is no need to introduce the time delay caused by searching across the entire disk (although it may produce false positives) when the intention is to provide a quick method of testing.

A note was made to include an instruction to use the inclusion/exclusion capabilities of the tool under test to focus on the seed data directories (this was also the setting used for the rest of the 'Demonstration' tests).

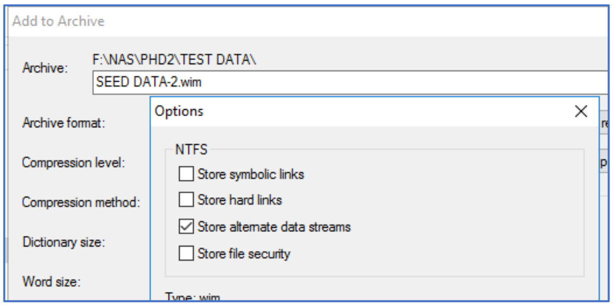


B	C	D
Path	**FileName**	**CreatedDate**
C:\Users\Administrator\Desktop\SEED DATA\DOCUMENTS	Dinosaur Extinction (2022 DOCX) Author metadata.docx	2022/11/27 01:23:46 UTC
C:\Users\Administrator\Desktop\SEED DATA\DOCUMENTS	Dinosaur Extinction (97-2003 DOC) Author metadata.doc	2022/11/27 01:15:52 UTC

Test 2 - Plain text search for the term 'casseopeia'.

Selected only the seed folder.

Referring to the sub-totals for each type of document it was found that there was a hit missing that was in the ADS folder. It was verified that the alternate data stream had not been maintained during the process of placing the seed data into a zip container. An investigation of internet sources identified a way to maintain the alternate data streams by saving the files with 7zip into a 'wim' container (that does not have compression) and selection the 'Store alternate data streams' option (**Error! Reference source not found.**).



Add to Archive

Archive: F:\NAS\PHD2\TEST DATA\
SEED DATA-2.wim

Archive format:
Compression level:
Compression method:
Dictionary size:
Word size:

Options

NTFS

☐ Store symbolic links
☐ Store hard links
☑ Store alternate data streams
☐ Store file security

Type: wim

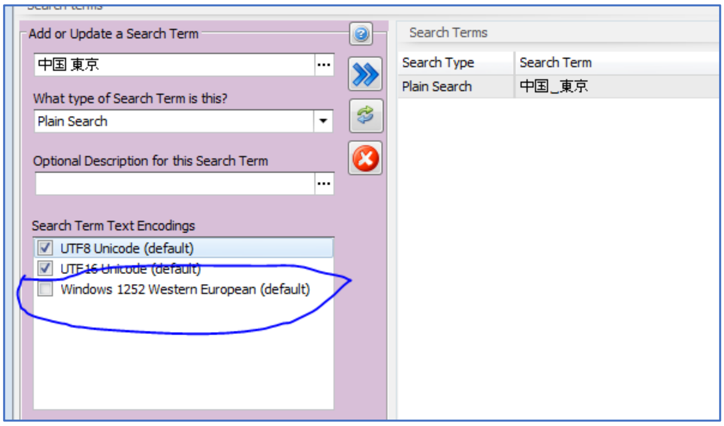
Figure 44 Option to store alternate data streams.

Further testing was carried out using the WIM container which was then zipped and unzipped to the endpoint clients and the test re-run, which then gave the expected results.

A note was made to require practitioners to download and use 7Zip to extract the seed data from the WIM container once it is extracted from the compressed zip container.

Test 3 - Plain text search for the term 中国 東京 (Unicode Chinese/Japanese ideograms)

The first foreign language search term was copied from the electronic copy of the Evaluation Criteria document. On entering the search term into the tool's search dialogue, a warning message appeared indicating that the term could not be used with the Windows 1252 Western European coding, so this was unchecked leaving UTF8 and UTF16 Unicode encodings (**Error! Reference source not found.**).



Add or Update a Search Term

中国 東京

What type of Search Term is this?
Plain Search

Optional Description for this Search Term

Search Term Text Encodings

☑ UTF8 Unicode (default)
☑ UTF16 Unicode (default)
☐ Windows 1252 Western European (default)

Search Terms

Search Type	Search Term
Plain Search	中国_東京

Figure 45 Removing Windows 1252 encoding.

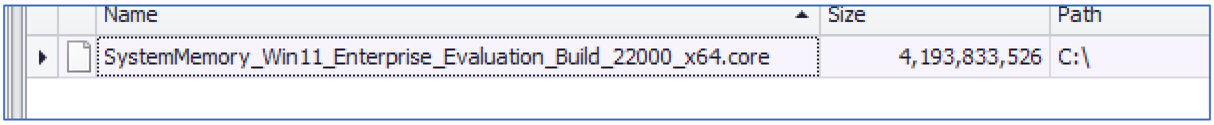
There was a mistake in the Criteria which had not included a PDF that had been created from the DOCX file containing the responsive term.

A note was made to add the PDF document to the list of expected results.

Capture capabilities

Capture system Memory

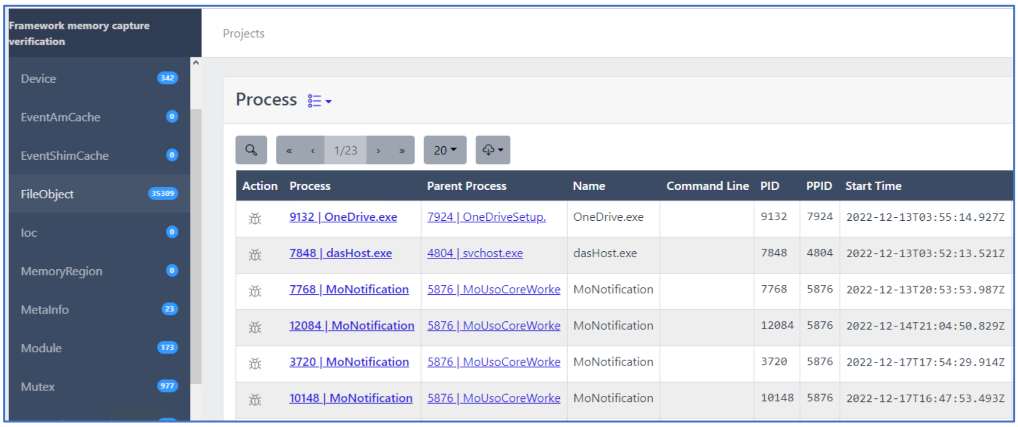
The option to capture system memory was selected and the tool was run on the endpoints. A file was created that had automatically been named in accordance with the Windows build details (which is important when processing memory captures as it determines the template to use in order to be able to interpret the memory contents) (Figure 46).



Name	Size	Path
SystemMemory_Win11_Enterprise_Evaluation_Build_22000_x64.core	4,193,833,526	C:\

Figure 46 Example of captured memory image

The forensic memory analysis utility Trufflepig Nexus was used to determine if the captured memory was valid and capable of being processed. Prior to processing the captured memory file needed to have the extension 'DMP' added. As shown in Figure 47, Trufflepig Nexus was able to correctly identify artefacts within the captured memory image without reporting errors.

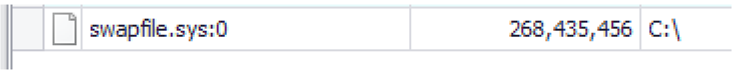


Action	Process	Parent Process	Name	Command Line	PID	PPID	Start Time
	9132 | OneDrive.exe	7924 | OneDriveSetup.	OneDrive.exe		9132	7924	2022-12-13T03:55:14.927Z
	7848 | dasHost.exe	4804 | svchost.exe	dasHost.exe		7848	4804	2022-12-13T03:52:13.521Z
	7768 | MoNotification	5876 | MoUsoCoreWorke	MoNotification		7768	5876	2022-12-13T20:53:53.987Z
	12084 | MoNotification	5876 | MoUsoCoreWorke	MoNotification		12084	5876	2022-12-14T21:04:50.829Z
	3720 | MoNotification	5876 | MoUsoCoreWorke	MoNotification		3720	5876	2022-12-17T17:54:29.914Z
	10148 | MoNotification	5876 | MoUsoCoreWorke	MoNotification		10148	5876	2022-12-17T16:47:53.493Z

Figure 47 Verifying memory capture.

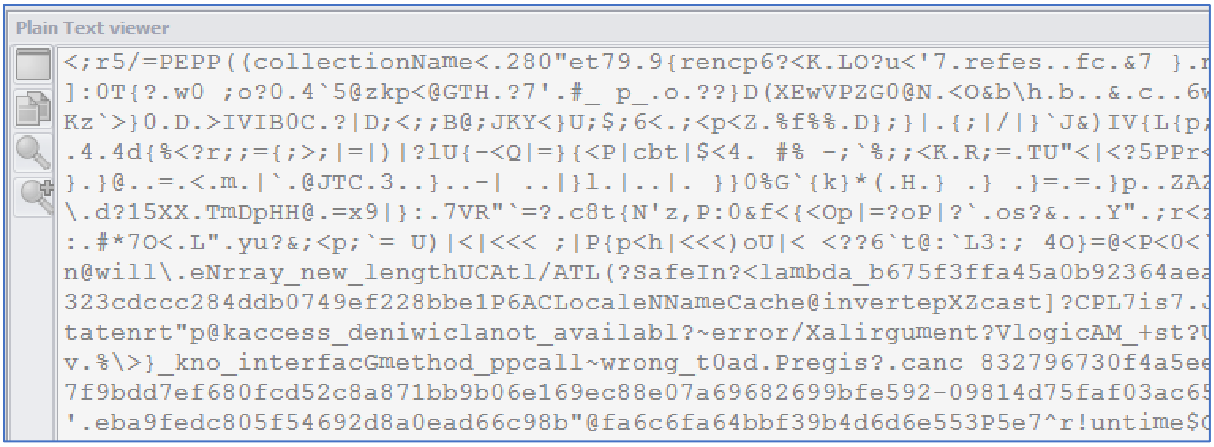
Capture Swap File

The Swap file was successfully captured.



swapfile.sys:0	268,435,456	C:\

Reviewing the raw file suggested that it contained valid data, but this was not verified at this stage.

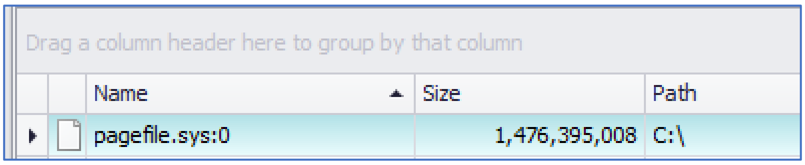


Plain Text viewer

```
<;r5/=PEPP((collectionName<.280"et79.9{rencp6?<K.LO?u<'7.refes..fc.&7 }.r
]:0T{?.w0 ;o?0.4`5@zkp<@GTH.?7'.#_ p_.o.??}D(XEwVPZG0@N.<O&b\h.b..&.c..6w
Kz`>}0.D.>IVIB0C.?|D;<;;B@;JKY<}U;$;6<.;<p<Z.%f%%.D};}|.{;|/|}`J&)IV{L{p;
.4.4d{%<?r;;={;>;|=|)|?lU{-<Q|=}{<P|cbt|$<4. #% -;`%;;<K.R;=.TU"<|<?5PPr<
}.}@..=.<.m.|`.@JTC.3..}..-| ..|}l.|..|. }}0%G`{k}*(.H.} .} .}=.=.}p..ZAZ
\.d?15XX.TmDpHH@.=x9|}:.7VR"`=?.c8t{N'z,P:0&f<{<Op|=?oP|?`.os?&...Y".;r<z
:.#*7O<.L".yu?&;<p;`= U)|<|<<< ;|P{p<h|<<<)oU|< <??6`t@:`L3:; 4O}=@<P<0<`
n@will\.eNrray_new_lengthUCAtl/ATL(?SafeIn?<lambda_b675f3ffa45a0b92364aea
323cdccc284ddb0749ef228bbe1P6ACLocaleNNameCache@invertepXZcast]?CPL7is7.J
tatenrt"p@kaccess_deniwiclanot_availabl?~error/Xalirgument?VlogicAM_+st?U
v.%\>}_kno_interfacGmethod_ppcall~wrong_t0ad.Pregis?.canc 832796730f4a5ee
7f9bdd7ef680fcd52c8a871bb9b06e169ec88e07a69682699bfe592-09814d75faf03ac65
'.eba9fedc805f54692d8a0ead66c98b"@fa6c6fa64bbf39b4d6d6e553P5e7^r!untime$(
```

Capture Pagefile

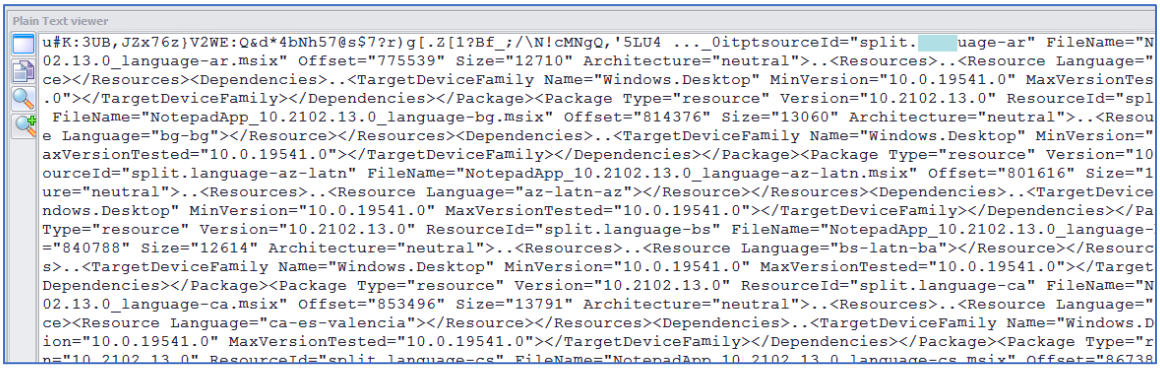
The Pagefile was successfully captured.



Drag a column header here to group by that column

Name	Size	Path
pagefile.sys:0	1,476,395,008	C:\

Reviewing the raw file suggested that it contained valid data, but this was not verified at this stage.

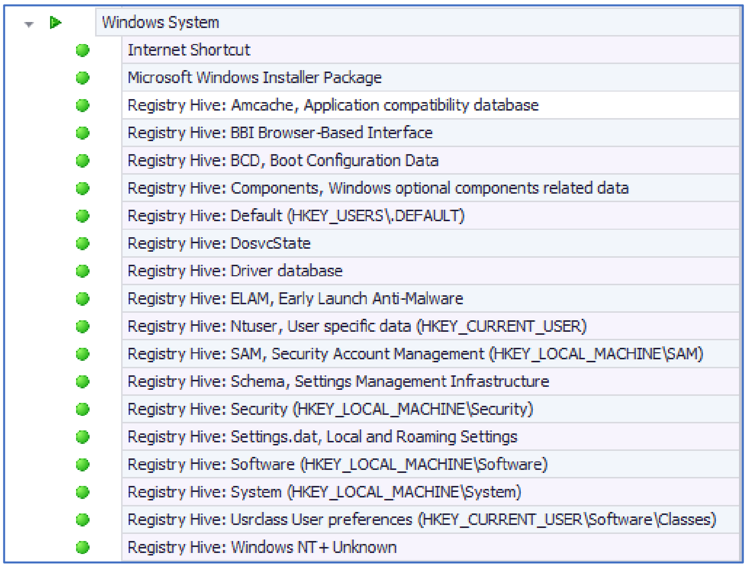


Plain Text viewer

```
u#K:3UB,JZx76z}V2WE:Q&d*4bNh57@s$7?r)g[.Z[1?Bf_;/\N!cMNgQ,'5LU4 ..._0itptsourceId="split.    uage-ar" FileName="N
02.13.0_language-ar.msix" Offset="775539" Size="12710" Architecture="neutral">..<Resources>..<Resource Language="
ce></Resources><Dependencies>..<TargetDeviceFamily Name="Windows.Desktop" MinVersion="10.0.19541.0" MaxVersionTes
.0"></TargetDeviceFamily></Dependencies></Package><Package Type="resource" Version="10.2102.13.0" ResourceId="spl
 FileName="NotepadApp_10.2102.13.0_language-bg.msix" Offset="814376" Size="13060" Architecture="neutral">..<Resou
e Language="bg-bg"></Resource></Resources><Dependencies>..<TargetDeviceFamily Name="Windows.Desktop" MinVersion="
axVersionTested="10.0.19541.0"></TargetDeviceFamily></Dependencies></Package><Package Type="resource" Version="10
ourceId="split.language-az-latn" FileName="NotepadApp_10.2102.13.0_language-az-latn.msix" Offset="801616" Size="1
ure="neutral">..<Resources>..<Resource Language="az-latn-az"></Resource></Resources><Dependencies>..<TargetDevice
ndows.Desktop" MinVersion="10.0.19541.0" MaxVersionTested="10.0.19541.0"></TargetDeviceFamily></Dependencies></Pa
Type="resource" Version="10.2102.13.0" ResourceId="split.language-bs" FileName="NotepadApp_10.2102.13.0_language-
="840788" Size="12614" Architecture="neutral">..<Resources>..<Resource Language="bs-latn-ba"></Resource></Resourc
s>..<TargetDeviceFamily Name="Windows.Desktop" MinVersion="10.0.19541.0" MaxVersionTested="10.0.19541.0"></Target
Dependencies></Package><Package Type="resource" Version="10.2102.13.0" ResourceId="split.language-ca" FileName="N
02.13.0_language-ca.msix" Offset="853496" Size="13791" Architecture="neutral">..<Resources>..<Resource Language="
ce><Resource Language="ca-es-valencia"></Resource></Resources><Dependencies>..<TargetDeviceFamily Name="Windows.D
ion="10.0.19541.0" MaxVersionTested="10.0.19541.0"></TargetDeviceFamily></Dependencies></Package><Package Type="r
n="10.2102.13.0" ResourceId="split.language-cs" FileName="NotepadApp_10.2102.13.0_language-cs.msix" Offset="86738
```

Capture Registry Hives

Registry hives were captured (including amcache) as shown in the extract of the captured files in Figure 48.

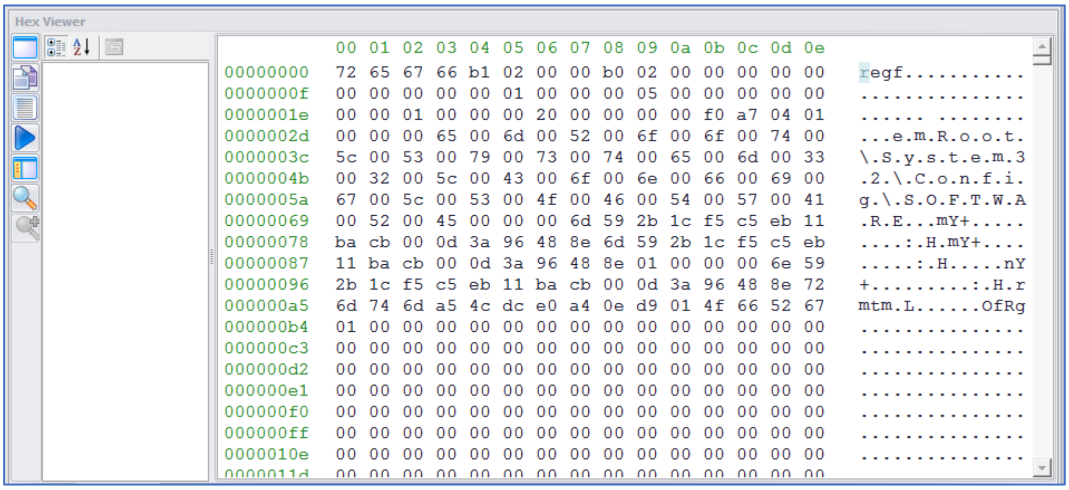


Windows System

- Internet Shortcut
- Microsoft Windows Installer Package
- Registry Hive: Amcache, Application compatibility database
- Registry Hive: BBI Browser-Based Interface
- Registry Hive: BCD, Boot Configuration Data
- Registry Hive: Components, Windows optional components related data
- Registry Hive: Default (HKEY_USERS\.DEFAULT)
- Registry Hive: DosvcState
- Registry Hive: Driver database
- Registry Hive: ELAM, Early Launch Anti-Malware
- Registry Hive: Ntuser, User specific data (HKEY_CURRENT_USER)
- Registry Hive: SAM, Security Account Management (HKEY_LOCAL_MACHINE\SAM)
- Registry Hive: Schema, Settings Management Infrastructure
- Registry Hive: Security (HKEY_LOCAL_MACHINE\Security)
- Registry Hive: Settings.dat, Local and Roaming Settings
- Registry Hive: Software (HKEY_LOCAL_MACHINE\Software)
- Registry Hive: System (HKEY_LOCAL_MACHINE\System)
- Registry Hive: Usrclass User preferences (HKEY_CURRENT_USER\Software\Classes)
- Registry Hive: Windows NT+ Unknown

Figure 48 Captured object list - Registry hives.

Reviewing the hex contents of the captured hives suggests that they are valid (as shown in an example for the Software hive in Figure 49).

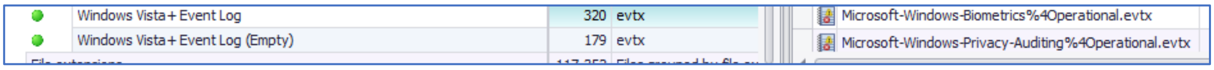


```
          00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
00000000  72 65 67 66 b1 02 00 00 b0 02 00 00 00 00 00   regf...........
0000000f  00 00 00 00 00 01 00 00 00 05 00 00 00 00 00   ...............
0000001e  00 00 01 00 00 00 20 00 00 00 00 f0 a7 04 01   ...... ........
0000002d  00 00 00 65 00 6d 00 52 00 6f 00 6f 00 74 00   ...e.m.R.o.o.t.
0000003c  5c 00 53 00 79 00 73 00 74 00 65 00 6d 00 33   \.S.y.s.t.e.m.3
0000004b  00 32 00 5c 00 43 00 6f 00 6e 00 66 00 69 00   .2.\.C.o.n.f.i.
0000005a  67 00 5c 00 53 00 4f 00 46 00 54 00 57 00 41   g.\.S.O.F.T.W.A
00000069  00 52 00 45 00 00 00 6d 59 2b 1c f5 c5 eb 11   .R.E...mY+.....
00000078  ba cb 00 0d 3a 96 48 8e 6d 59 2b 1c f5 c5 eb   ....:.H.mY+....
00000087  11 ba cb 00 0d 3a 96 48 8e 01 00 00 00 6e 59   .....:.H.....nY
00000096  2b 1c f5 c5 eb 11 ba cb 00 0d 3a 96 48 8e 72   +.........:.H.r
000000a5  6d 74 6d a5 4c dc e0 a4 0e d9 01 4f 66 52 67   mtm.L......OfRg
000000b4  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
000000c3  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
000000d2  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
000000e1  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
000000f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
000000ff  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
0000010e  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ...............
0000011d  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Figure 49 First sectors of captured Software hive

Capture EVT and EVTX Files

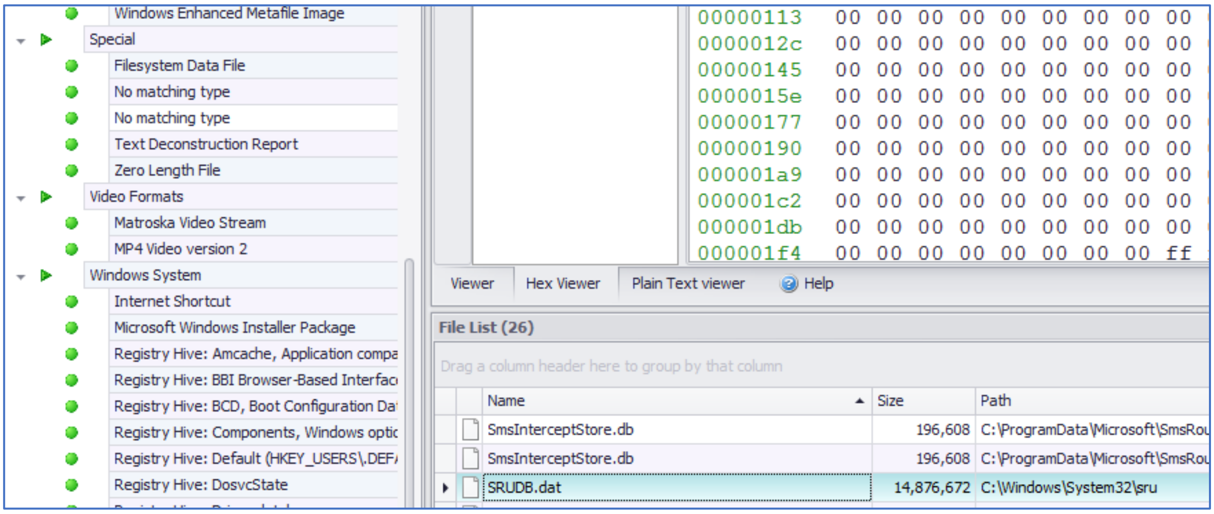
EVTX files were captured.



Windows Vista+ Event Log	320	evtx	Microsoft-Windows-Biometrics%4Operational.evtx
Windows Vista+ Event Log (Empty)	179	evtx	Microsoft-Windows-Privacy-Auditing%4Operational.evtx

Capture SRUDB

The SRUDB was captured under the category ‘Special’.



- ● Windows Enhanced Metafile Image
- ▶ Special
  - ● Filesystem Data File
  - ● No matching type
  - ● No matching type
  - ● Text Deconstruction Report
  - ● Zero Length File
- ▶ Video Formats
  - ● Matroska Video Stream
  - ● MP4 Video version 2
- ▶ Windows System
  - ● Internet Shortcut
  - ● Microsoft Windows Installer Package
  - ● Registry Hive: Amcache, Application compa
  - ● Registry Hive: BBI Browser-Based Interfac
  - ● Registry Hive: BCD, Boot Configuration Da
  - ● Registry Hive: Components, Windows optic
  - ● Registry Hive: Default (HKEY_USERS\.DEF/
  - ● Registry Hive: DosvcState

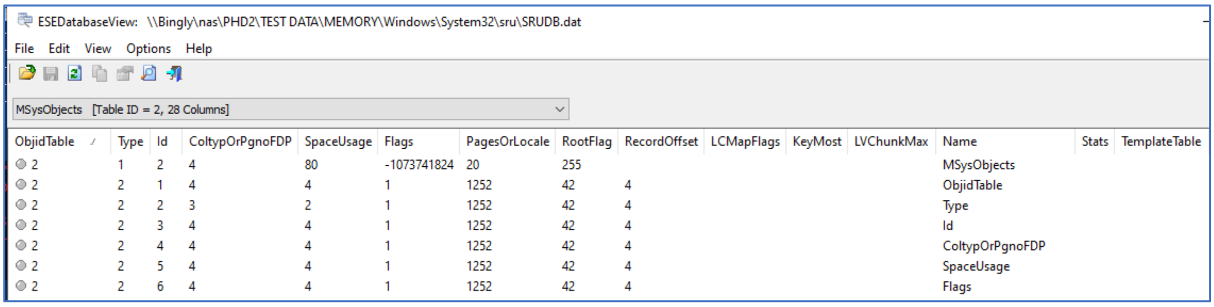
```
00000113  00 00 00 00 00 00 00 00 00
0000012c  00 00 00 00 00 00 00 00 00
00000145  00 00 00 00 00 00 00 00 00
0000015e  00 00 00 00 00 00 00 00 00
00000177  00 00 00 00 00 00 00 00 00
00000190  00 00 00 00 00 00 00 00 00
000001a9  00 00 00 00 00 00 00 00 00
000001c2  00 00 00 00 00 00 00 00 00
000001db  00 00 00 00 00 00 00 00 00
000001f4  00 00 00 00 00 00 00 00 ff
```

File List (26)

Drag a column header here to group by that column

Name	Size	Path
SmsInterceptStore.db	196,608	C:\ProgramData\Microsoft\SmsRou
SmsInterceptStore.db	196,608	C:\ProgramData\Microsoft\SmsRou
SRUDB.dat	14,876,672	C:\Windows\System32\sru

The captured file was extracted and processed without error using the utility ESEDatabaseView from Nirsoft (Figure 50).



ESEDatabaseView: \\Bingly\nas\PHD2\TEST DATA\MEMORY\Windows\System32\sru\SRUDB.dat

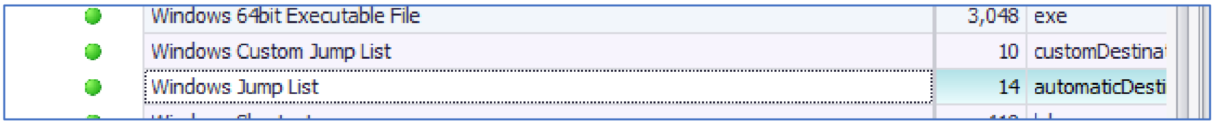
MSysObjects [Table ID = 2, 28 Columns]

ObjidTable	Type	Id	ColtypOrPgnoFDP	SpaceUsage	Flags	PagesOrLocale	RootFlag	RecordOffset	LCMapFlags	KeyMost	LVChunkMax	Name	Stats	TemplateTable
2	1	2	4	80	-1073741824	20	255					MSysObjects		
2	2	1	4	4	1	1252	42	4				ObjidTable		
2	2	2	3	2	1	1252	42	4				Type		
2	2	3	4	4	1	1252	42	4				Id		
2	2	4	4	4	1	1252	42	4				ColtypOrPgnoFDP		
2	2	5	4	4	1	1252	42	4				SpaceUsage		
2	2	6	4	4	1	1252	42	4				Flags		

Figure 50 Sample of processed SRUDB

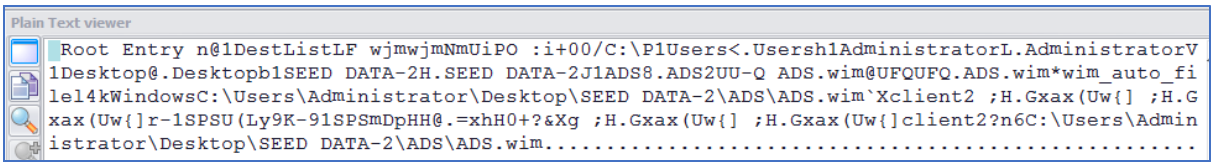
Capture Windows Jumplists

Both automatic and custom jumplists were captured.



●	Windows 64bit Executable File	3,048	exe
●	Windows Custom Jump List	10	customDestina
●	Windows Jump List	14	automaticDesti

A sample of the jumplists was examined and the data appeared to be valid Figure 51



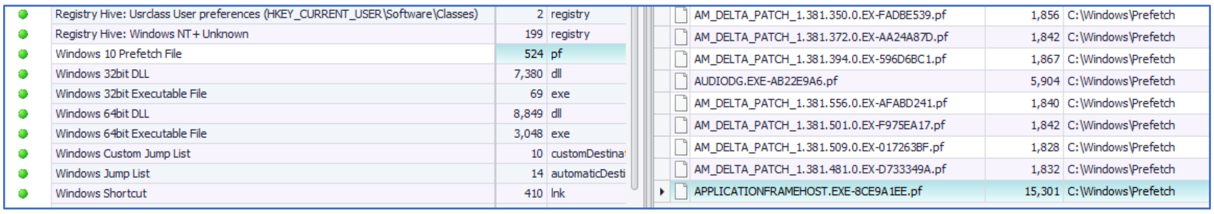
Plain Text viewer

```
Root Entry n@1DestListLF wjmwjmNmUiPO :i+00/C:\P1Users<.Usersh1AdministratorL.AdministratorV
1Desktop@.Desktopb1SEED DATA-2H.SEED DATA-2J1ADS8.ADS2UU-Q ADS.wim@UFQUFQ.ADS.wim*wim_auto_fi
lel4kWindowsC:\Users\Administrator\Desktop\SEED DATA-2\ADS\ADS.wim`Xclient2 ;H.Gxax(Uw{] ;H.G
xax(Uw{]r-1SPSU(Ly9K-91SPSmDpHH@.=xhH0+?&Xg ;H.Gxax(Uw{] ;H.Gxax(Uw{]client2?n6C:\Users\Admin
istrator\Desktop\SEED DATA-2\ADS\ADS.wim.........................................................
```

Figure 51 Captured Jumplist data – sample.

Capture Prefetch Files

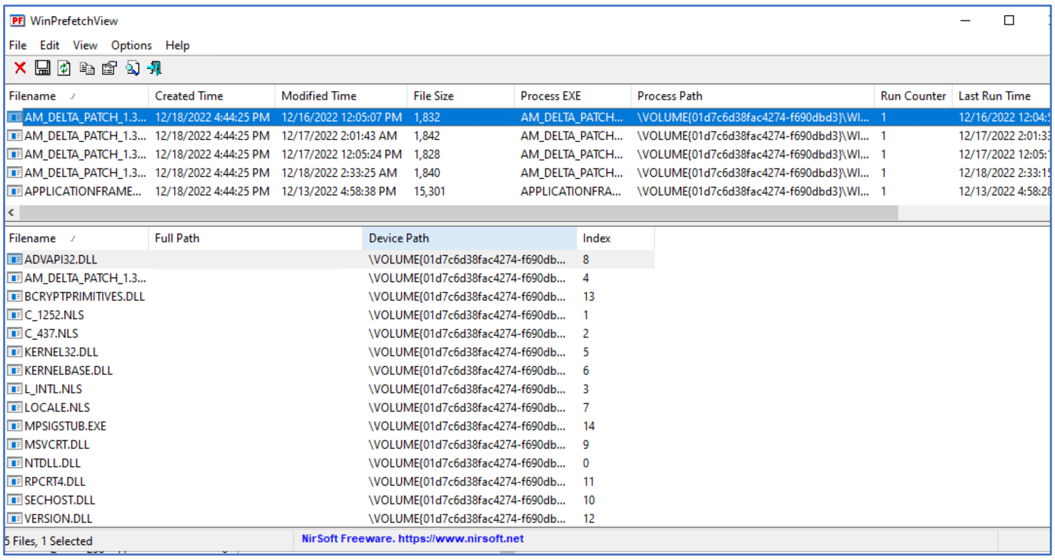
Prefetch files were captured.



●	Registry Hive: Usrclass User preferences (HKEY_CURRENT_USER\Software\Classes)	2	registry
●	Registry Hive: Windows NT+ Unknown	199	registry
●	Windows 10 Prefetch File	524	pf
●	Windows 32bit DLL	7,380	dll
●	Windows 32bit Executable File	69	exe
●	Windows 64bit DLL	8,849	dll
●	Windows 64bit Executable File	3,048	exe
●	Windows Custom Jump List	10	customDestina
●	Windows Jump List	14	automaticDesti
●	Windows Shortcut	410	lnk

AM_DELTA_PATCH_1.381.350.0.EX-FADBE539.pf	1,856	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.372.0.EX-AA24A87D.pf	1,842	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.394.0.EX-596D6BC1.pf	1,867	C:\Windows\Prefetch
AUDIODG.EXE-AB22E9A6.pf	5,904	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.556.0.EX-AFABD241.pf	1,840	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.501.0.EX-F975EA17.pf	1,842	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.509.0.EX-017263BF.pf	1,828	C:\Windows\Prefetch
AM_DELTA_PATCH_1.381.481.0.EX-D733349A.pf	1,832	C:\Windows\Prefetch
APPLICATIONFRAMEHOST.EXE-8CE9A1EE.pf	15,301	C:\Windows\Prefetch

The forensic utility WinPrefetchView from Nirsoft was used to process a sample of the captured files which it did without noting any errors (Figure 52).



Filename	Created Time	Modified Time	File Size	Process EXE	Process Path	Run Counter	Last Run Time
AM_DELTA_PATCH_1.3...	12/18/2022 4:44:25 PM	12/16/2022 12:05:07 PM	1,832	AM_DELTA_PATCH...	\VOLUME{01d7c6d38fac4274-f690dbd3}\WI...	1	12/16/2022 12:04:
AM_DELTA_PATCH_1.3...	12/18/2022 4:44:25 PM	12/17/2022 2:01:43 AM	1,842	AM_DELTA_PATCH...	\VOLUME{01d7c6d38fac4274-f690dbd3}\WI...	1	12/17/2022 2:01:3
AM_DELTA_PATCH_1.3...	12/18/2022 4:44:25 PM	12/17/2022 12:05:24 PM	1,828	AM_DELTA_PATCH...	\VOLUME{01d7c6d38fac4274-f690dbd3}\WI...	1	12/17/2022 12:05:
AM_DELTA_PATCH_1.3...	12/18/2022 4:44:25 PM	12/18/2022 2:33:25 AM	1,840	AM_DELTA_PATCH...	\VOLUME{01d7c6d38fac4274-f690dbd3}\WI...	1	12/18/2022 2:33:1
APPLICATIONFRAME...	12/18/2022 4:44:25 PM	12/13/2022 4:58:38 PM	15,301	APPLICATIONFRA...	\VOLUME{01d7c6d38fac4274-f690dbd3}\WI...	1	12/13/2022 4:58:2

Filename	Full Path	Device Path	Index
ADVAPI32.DLL		\VOLUME{01d7c6d38fac4274-f690db...	8
AM_DELTA_PATCH_1.3...		\VOLUME{01d7c6d38fac4274-f690db...	4
BCRYPTPRIMITIVES.DLL		\VOLUME{01d7c6d38fac4274-f690db...	13
C_1252.NLS		\VOLUME{01d7c6d38fac4274-f690db...	1
C_437.NLS		\VOLUME{01d7c6d38fac4274-f690db...	2
KERNEL32.DLL		\VOLUME{01d7c6d38fac4274-f690db...	5
KERNELBASE.DLL		\VOLUME{01d7c6d38fac4274-f690db...	6
L_INTL.NLS		\VOLUME{01d7c6d38fac4274-f690db...	3
LOCALE.NLS		\VOLUME{01d7c6d38fac4274-f690db...	7
MPSIGSTUB.EXE		\VOLUME{01d7c6d38fac4274-f690db...	14
MSVCRT.DLL		\VOLUME{01d7c6d38fac4274-f690db...	9
NTDLL.DLL		\VOLUME{01d7c6d38fac4274-f690db...	0
RPCRT4.DLL		\VOLUME{01d7c6d38fac4274-f690db...	11
SECHOST.DLL		\VOLUME{01d7c6d38fac4274-f690db...	10
VERSION.DLL		\VOLUME{01d7c6d38fac4274-f690db...	12

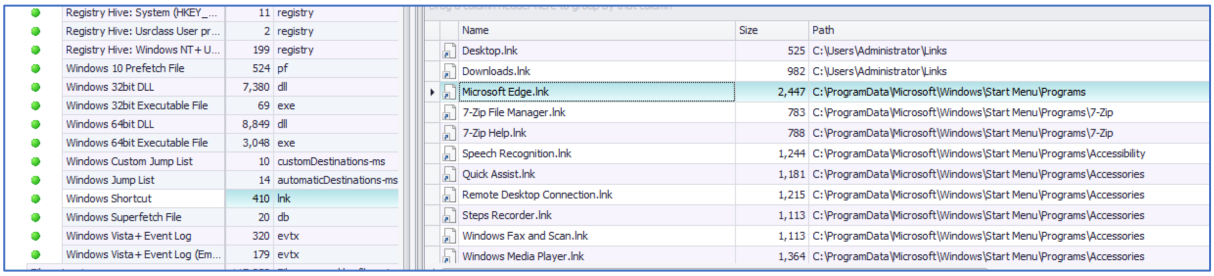
5 Files, 1 Selected

NirSoft Freeware. https://www.nirsoft.net

Figure 52 Processed sample of captured prefetch files.

Capture Windows LNK files

Windows shortcut (LNK) files were captured.

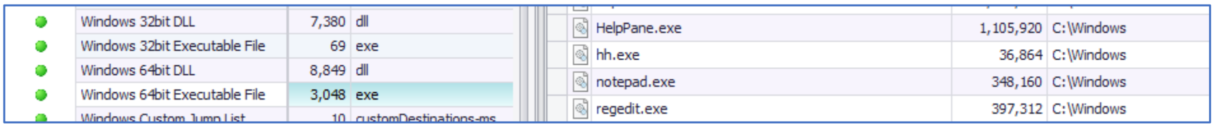


●	Registry Hive: System (HKEY_...	11	registry
●	Registry Hive: Usrclass User pr...	2	registry
●	Registry Hive: Windows NT+ U...	199	registry
●	Windows 10 Prefetch File	524	pf
●	Windows 32bit DLL	7,380	dll
●	Windows 32bit Executable File	69	exe
●	Windows 64bit DLL	8,849	dll
●	Windows 64bit Executable File	3,048	exe
●	Windows Custom Jump List	10	customDestinations-ms
●	Windows Jump List	14	automaticDestinations-ms
●	Windows Shortcut	410	lnk
●	Windows Superfetch File	20	db
●	Windows Vista+ Event Log	320	evtx
●	Windows Vista+ Event Log (Em...	179	evtx

Name	Size	Path
Desktop.lnk	525	C:\Users\Administrator\Links
Downloads.lnk	982	C:\Users\Administrator\Links
Microsoft Edge.lnk	2,447	C:\ProgramData\Microsoft\Windows\Start Menu\Programs
7-Zip File Manager.lnk	783	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\7-Zip
7-Zip Help.lnk	788	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\7-Zip
Speech Recognition.lnk	1,244	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessibility
Quick Assist.lnk	1,181	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories
Remote Desktop Connection.lnk	1,215	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories
Steps Recorder.lnk	1,113	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories
Windows Fax and Scan.lnk	1,113	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories
Windows Media Player.lnk	1,364	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories

Capture Executables and DLLs

Windows executables and DLLs were captured.

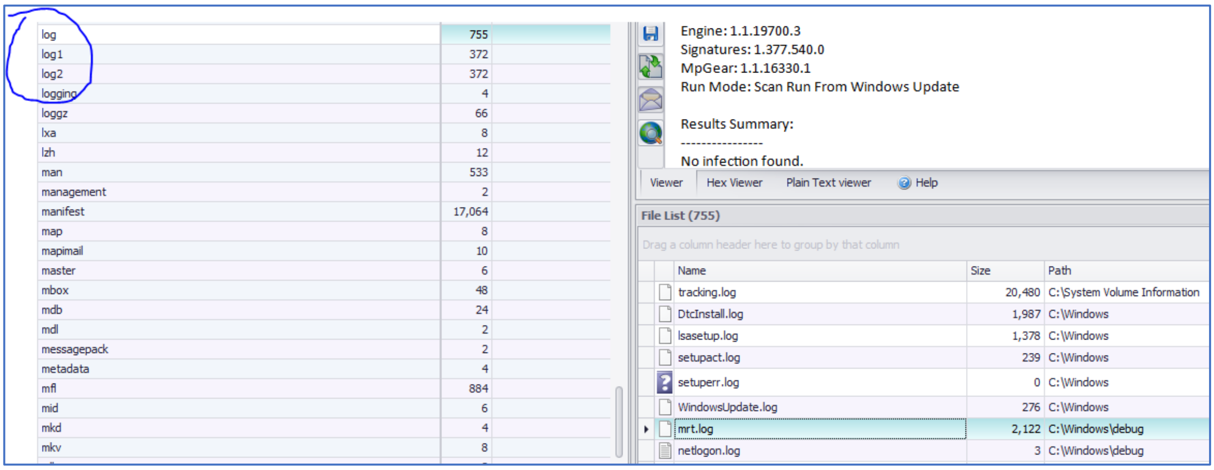


Windows 32bit DLL	7,380	dll
Windows 32bit Executable File	69	exe
Windows 64bit DLL	8,849	dll
Windows 64bit Executable File	3,048	exe
Windows Custom Jump List	10	customDestinations-ms

HelpPane.exe	1,105,920	C:\Windows
hh.exe	36,864	C:\Windows
notepad.exe	348,160	C:\Windows
regedit.exe	397,312	C:\Windows

Capture Windows System Logfiles

Windows system log files were captured.



log	755
log1	372
log2	372
logging	4
loggz	66
lxa	8
lzh	12
man	533
management	2
manifest	17,064
map	8
mapimail	10
master	6
mbox	48
mdb	24
mdl	2
messagepack	2
metadata	4
mfl	884
mid	6
mkd	4
mkv	8

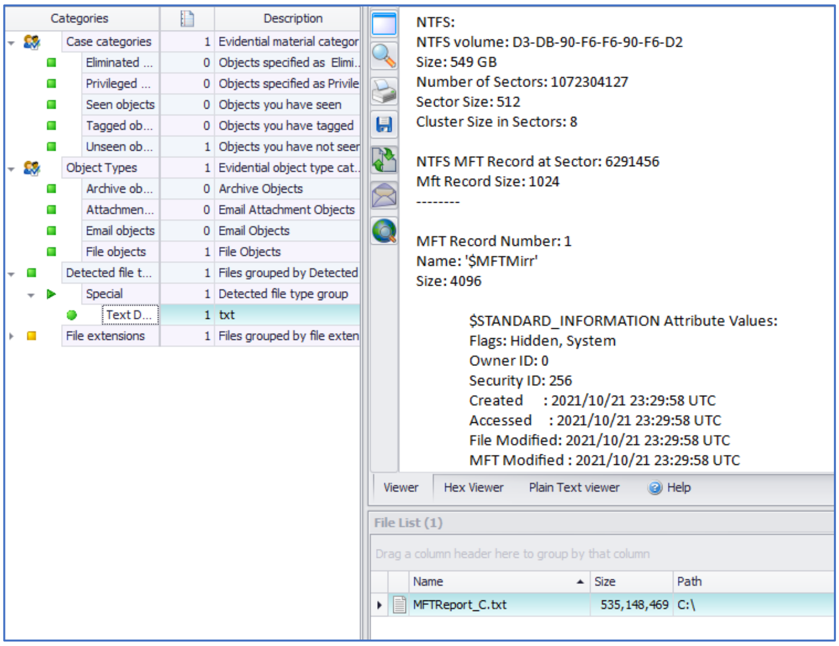
```
Engine: 1.1.19700.3
Signatures: 1.377.540.0
MpGear: 1.1.16330.1
Run Mode: Scan Run From Windows Update

Results Summary:
----------------
No infection found.
```

File List (755)

Name	Size	Path
tracking.log	20,480	C:\System Volume Information
DtcInstall.log	1,987	C:\Windows
lsasetup.log	1,378	C:\Windows
setupact.log	239	C:\Windows
setuperr.log	0	C:\Windows
WindowsUpdate.log	276	C:\Windows
mrt.log	2,122	C:\Windows\debug
netlogon.log	3	C:\Windows\debug

Capture $MFT

The tool is designed to create a text report rather than capture the $MFT file itself.



Categories		Description
Case categories	1	Evidential material categor
Eliminated ...	0	Objects specified as Elimi...
Privileged ...	0	Objects specified as Privile
Seen objects	0	Objects you have seen
Tagged ob...	0	Objects you have tagged
Unseen ob...	1	Objects you have not seer
Object Types	1	Evidential object type cat...
Archive ob...	0	Archive Objects
Attachmen...	0	Email Attachment Objects
Email objects	0	Email Objects
File objects	1	File Objects
Detected file t...	1	Files grouped by Detected
Special	1	Detected file type group
Text D...	1	txt
File extensions	1	Files grouped by file exten

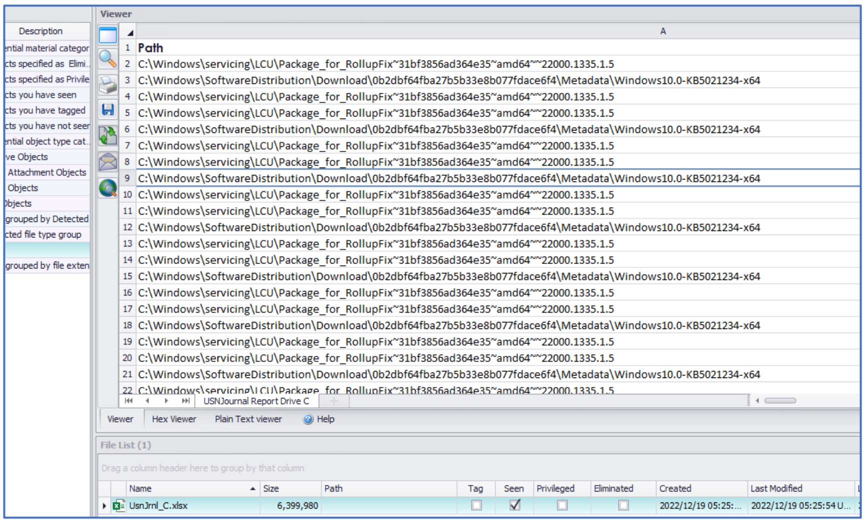
```
NTFS:
NTFS volume: D3-DB-90-F6-F6-90-F6-D2
Size: 549 GB
Number of Sectors: 1072304127
Sector Size: 512
Cluster Size in Sectors: 8

NTFS MFT Record at Sector: 6291456
Mft Record Size: 1024
--------

MFT Record Number: 1
Name: '$MFTMirr'
Size: 4096

        $STANDARD_INFORMATION Attribute Values:
        Flags: Hidden, System
        Owner ID: 0
        Security ID: 256
        Created      : 2021/10/21 23:29:58 UTC
        Accessed     : 2021/10/21 23:29:58 UTC
        File Modified: 2021/10/21 23:29:58 UTC
        MFT Modified : 2021/10/21 23:29:58 UTC
```

File List (1)

Name	Size	Path
MFTReport_C.txt	535,148,469	C:\

Capture $USNJournal

The tool is designed to create a text file report rather than capture the $USNJrnl file itself.



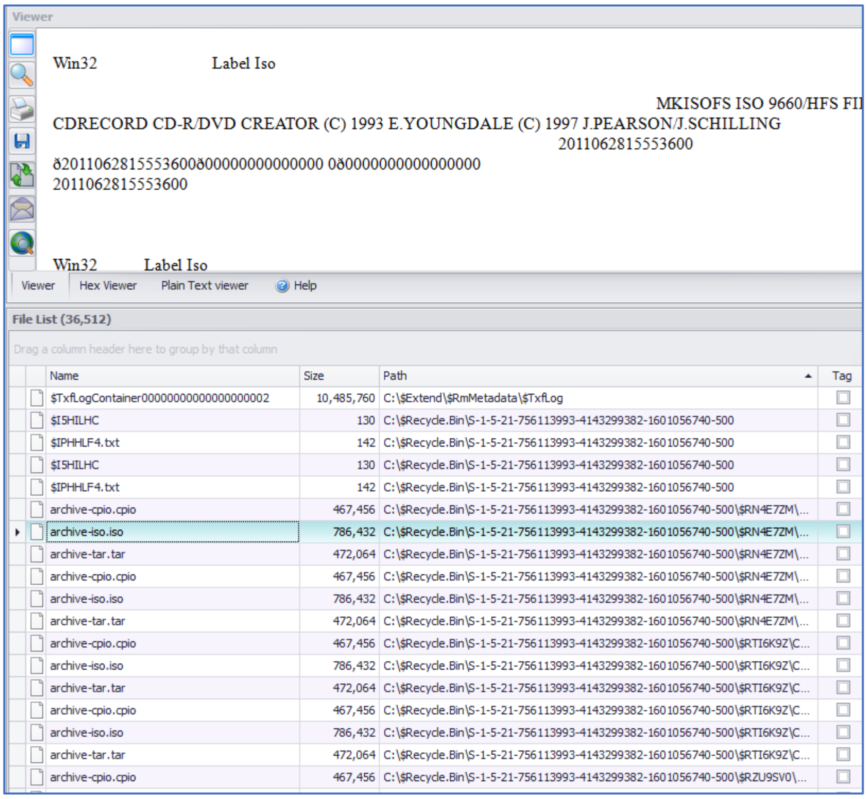
	A
1	**Path**
2	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
3	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
4	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
5	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
6	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
7	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
8	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
9	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
10	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
11	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
12	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
13	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
14	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
15	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
16	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
17	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
18	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
19	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
20	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5
21	C:\Windows\SoftwareDistribution\Download\0b2dbf64fba27b5b33e8b077fdace6f4\Metadata\Windows10.0-KB5021234-x64
22	C:\Windows\servicing\LCU\Package_for_RollupFix~31bf3856ad364e35~amd64~~22000.1335.1.5

USNJournal Report Drive C
Viewer
Hex Viewer
Plain Text viewer
Help
File List (1)
Drag a column header here to group by that column

Name	Size	Path	Tag	Seen	Privileged	Eliminated	Created	Last Modified
UsnJrnl_C.xlsx	6,399,980		□	☑	□	□	2022/12/19 05:25:...	2022/12/19 05:25:54 U...

Capture Recycle Bin Artefacts

Recycle Bin artefacts were captured.

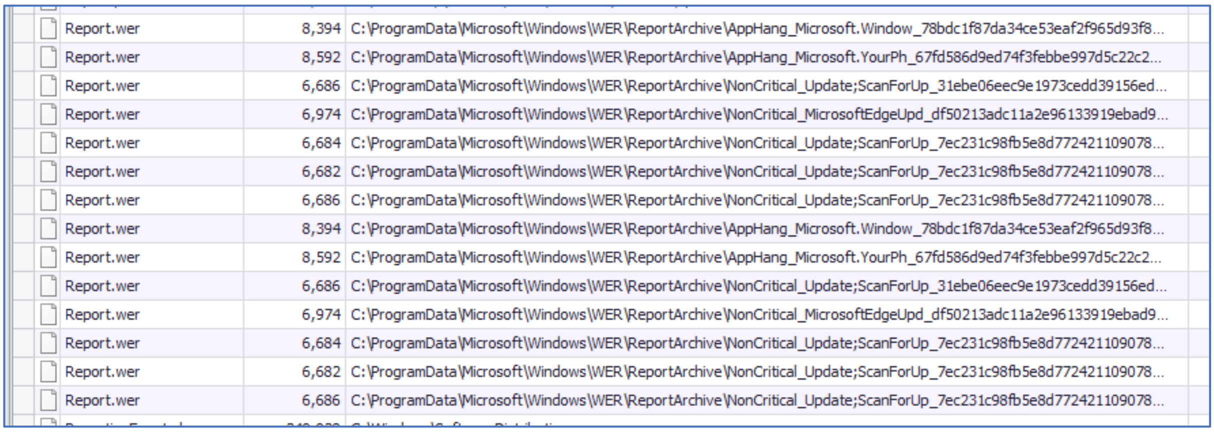


```
Win32                 Label Iso

                                                              MKISOFS ISO 9660/HFS FI
CDRECORD CD-R/DVD CREATOR (C) 1993 E.YOUNGDALE (C) 1997 J.PEARSON/J.SCHILLING
                                                   2011062815553600
ð2011062815553600ð0000000000000000 0ð0000000000000000
2011062815553600


Win32         Label Iso
```

Name	Size	Path	Tag
$TxfLogContainer00000000000000000002	10,485,760	C:\$Extend\$RmMetadata\$TxfLog	□
$I5HILHC	130	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500	□
$IPHHLF4.txt	142	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500	□
$I5HILHC	130	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500	□
$IPHHLF4.txt	142	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500	□
archive-cpio.cpio	467,456	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-iso.iso	786,432	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-tar.tar	472,064	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-cpio.cpio	467,456	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-iso.iso	786,432	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-tar.tar	472,064	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RN4E7ZM\...	□
archive-cpio.cpio	467,456	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-iso.iso	786,432	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-tar.tar	472,064	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-cpio.cpio	467,456	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-iso.iso	786,432	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-tar.tar	472,064	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RTI6K9Z\C...	□
archive-cpio.cpio	467,456	C:\$Recycle.Bin\S-1-5-21-756113993-4143299382-1601056740-500\$RZU9SV0\...	□

Capture Windows Error Reporting (WER) Artefacts.

WER artefacts were captured.

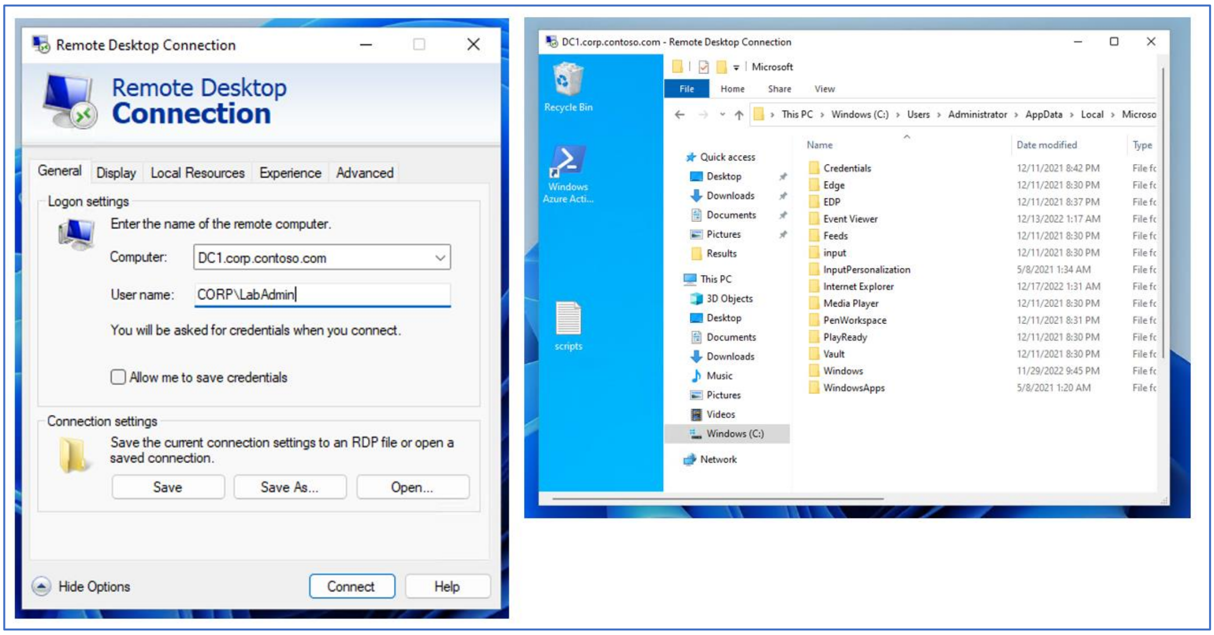


Report.wer	8,394	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\AppHang_Microsoft.Window_78bdc1f87da34ce53eaf2f965d93f8...
Report.wer	8,592	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\AppHang_Microsoft.YourPh_67fd586d9ed74f3febbe997d5c22c2...
Report.wer	6,686	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_31ebe06eec9e1973cedd39156ed...
Report.wer	6,974	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_MicrosoftEdgeUpd_df50213adc11a2e96133919ebad9...
Report.wer	6,684	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...
Report.wer	6,682	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...
Report.wer	6,686	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...
Report.wer	8,394	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\AppHang_Microsoft.Window_78bdc1f87da34ce53eaf2f965d93f8...
Report.wer	8,592	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\AppHang_Microsoft.YourPh_67fd586d9ed74f3febbe997d5c22c2...
Report.wer	6,686	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_31ebe06eec9e1973cedd39156ed...
Report.wer	6,974	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_MicrosoftEdgeUpd_df50213adc11a2e96133919ebad9...
Report.wer	6,684	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...
Report.wer	6,682	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...
Report.wer	6,686	C:\ProgramData\Microsoft\Windows\WER\ReportArchive\NonCritical_Update;ScanForUp_7ec231c98fb5e8d772421109078...

Capture RDP Cache

The virtual machines created as part of the Windows 11 and O365 Deployment Lab do not have RDP artefacts on installation.

An RDP connection was made to the domain controller server (DC1) to create the required artefacts.



Remote Desktop Connection

General | Display | Local Resources | Experience | Advanced

Logon settings

Enter the name of the remote computer.

Computer: DC1.corp.contoso.com

User name: CORP\LabAdmin

You will be asked for credentials when you connect.

□ Allow me to save credentials

Connection settings

Save the current connection settings to an RDP file or open a saved connection.

Save | Save As... | Open...

Hide Options | Connect | Help

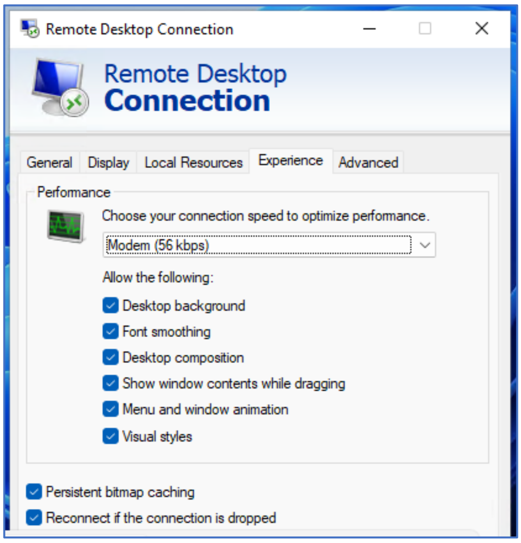
DC1.corp.contoso.com - Remote Desktop Connection

This PC › Windows (C:) › Users › Administrator › AppData › Local › Microso

Name	Date modified	Type
Credentials	12/11/2021 8:42 PM	File fc
Edge	12/11/2021 8:30 PM	File fc
EDP	12/11/2021 8:37 PM	File fc
Event Viewer	12/13/2022 1:17 AM	File fc
Feeds	12/11/2021 8:30 PM	File fc
input	12/11/2021 8:30 PM	File fc
InputPersonalization	5/8/2021 1:34 AM	File fc
Internet Explorer	12/17/2022 1:31 AM	File fc
Media Player	12/11/2021 8:30 PM	File fc
PenWorkspace	12/11/2021 8:31 PM	File fc
PlayReady	12/11/2021 8:30 PM	File fc
Vault	12/11/2021 8:30 PM	File fc
Windows	11/29/2022 9:45 PM	File fc
WindowsApps	5/8/2021 1:20 AM	File fc

Figure 53 Establishing an RDP connection to the domain controller DC1.

To force the cache files to be created on a relatively fast connection the connection speed settings were changed to ‘Modem (56kps).



Choose your connection speed to optimize performance.

Modem (56 kbps)

Allow the following:

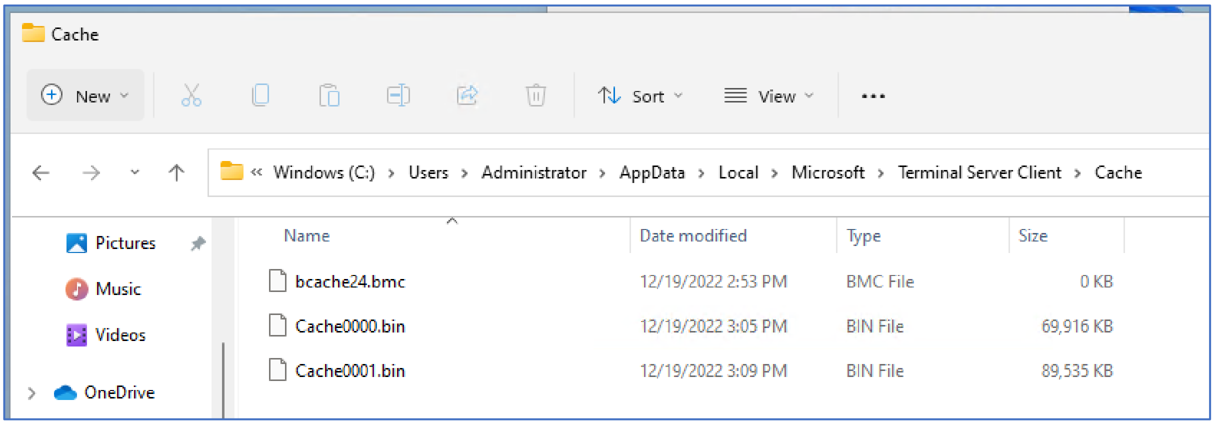
- ☑ Desktop background
- ☑ Font smoothing
- ☑ Desktop composition
- ☑ Show window contents while dragging
- ☑ Menu and window animation
- ☑ Visual styles

☑ Persistent bitmap caching

☑ Reconnect if the connection is dropped

Figure 54 Configuring slow RDP connection to force cache creation.

These settings quickly generated cache artefacts on the client machine (HYD-CLIENT2 in this case) once the connection had been established and a few windows had been opened (Figure 55)

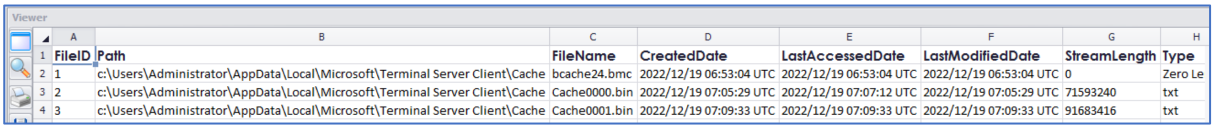


« Windows (C:) › Users › Administrator › AppData › Local › Microsoft › Terminal Server Client › Cache

Name	Date modified	Type	Size
bcache24.bmc	12/19/2022 2:53 PM	BMC File	0 KB
Cache0000.bin	12/19/2022 3:05 PM	BIN File	69,916 KB
Cache0001.bin	12/19/2022 3:09 PM	BIN File	89,535 KB

Figure 55 RDP cache artefacts created on client machine within a few minutes.

Running the tool only created a list of the files without capturing them.



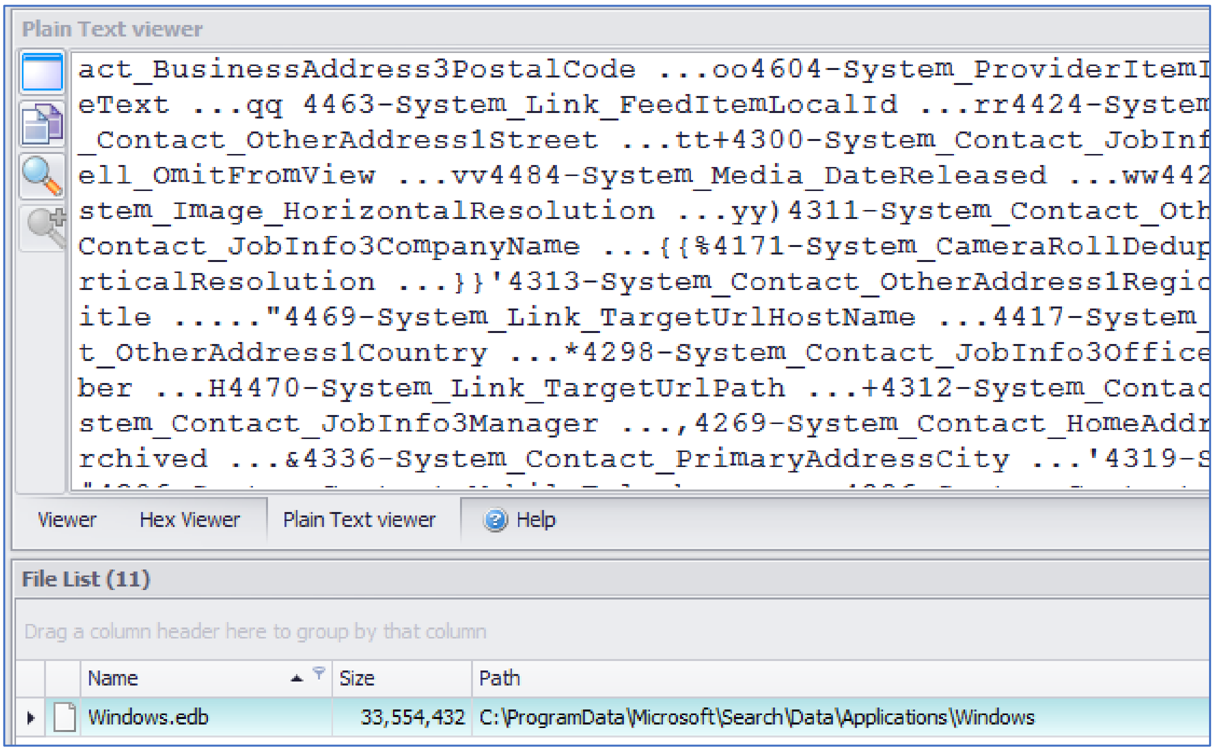
FileID	Path	FileName	CreatedDate	LastAccessedDate	LastModifiedDate	StreamLength	Type
1	c:\Users\Administrator\AppData\Local\Microsoft\Terminal Server Client\Cache	bcache24.bmc	2022/12/19 06:53:04 UTC	2022/12/19 06:53:04 UTC	2022/12/19 06:53:04 UTC	0	Zero Le
2	c:\Users\Administrator\AppData\Local\Microsoft\Terminal Server Client\Cache	Cache0000.bin	2022/12/19 07:05:29 UTC	2022/12/19 07:07:12 UTC	2022/12/19 07:05:29 UTC	71593240	txt
3	c:\Users\Administrator\AppData\Local\Microsoft\Terminal Server Client\Cache	Cache0001.bin	2022/12/19 07:09:33 UTC	2022/12/19 07:09:33 UTC	2022/12/19 07:09:33 UTC	91683416	txt

The developer was notified.

A note was made to add a paragraph to the Framework to include instructions for generating the cache files.

Capture Windows Search Database

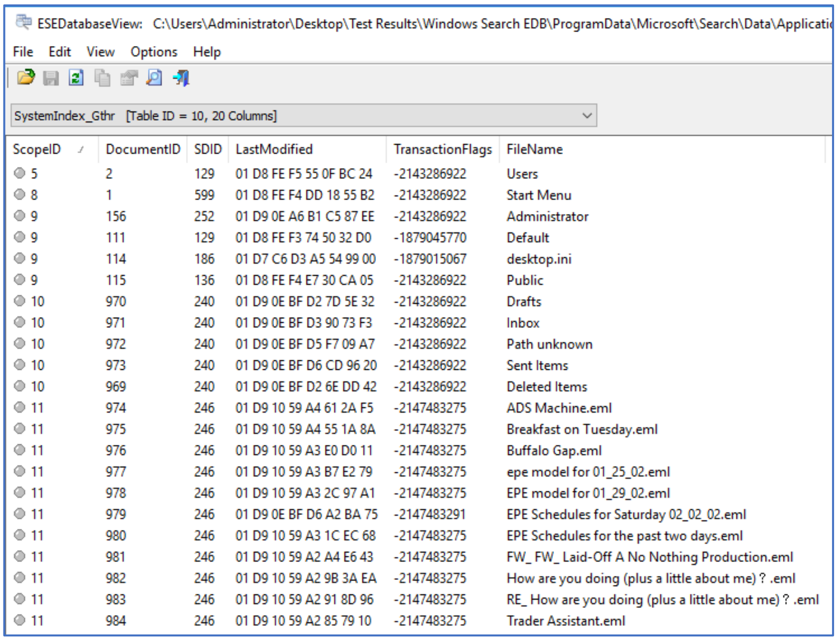
The file Windows.edb was captured. It was then viewed using the Nirsoft utility ESEDatabaseView which ran without showing any errors.



```
act_BusinessAddress3PostalCode ...oo4604-System_ProviderItemI
eText ...qq 4463-System_Link_FeedItemLocalId ...rr4424-System
_Contact_OtherAddress1Street ...tt+4300-System_Contact_JobInf
ell_OmitFromView ...vv4484-System_Media_DateReleased ...ww442
stem_Image_HorizontalResolution ...yy)4311-System_Contact_Oth
Contact_JobInfo3CompanyName ...{{%4171-System_CameraRollDedup
rticalResolution ...}}'4313-System_Contact_OtherAddress1Regic
itle ....."4469-System_Link_TargetUrlHostName ...4417-System_
t_OtherAddress1Country ...*4298-System_Contact_JobInfo3Office
ber ...H4470-System_Link_TargetUrlPath ...+4312-System_Contac
stem_Contact_JobInfo3Manager ...,4269-System_Contact_HomeAddr
rchived ...&4336-System_Contact_PrimaryAddressCity ...'4319-S
```

Name	Size	Path
Windows.edb	33,554,432	C:\ProgramData\Microsoft\Search\Data\Applications\Windows

Figure 56 Captured Windows search database.



ESEDatabaseView: C:\Users\Administrator\Desktop\Test Results\Windows Search EDB\ProgramData\Microsoft\Search\Data\Applicati

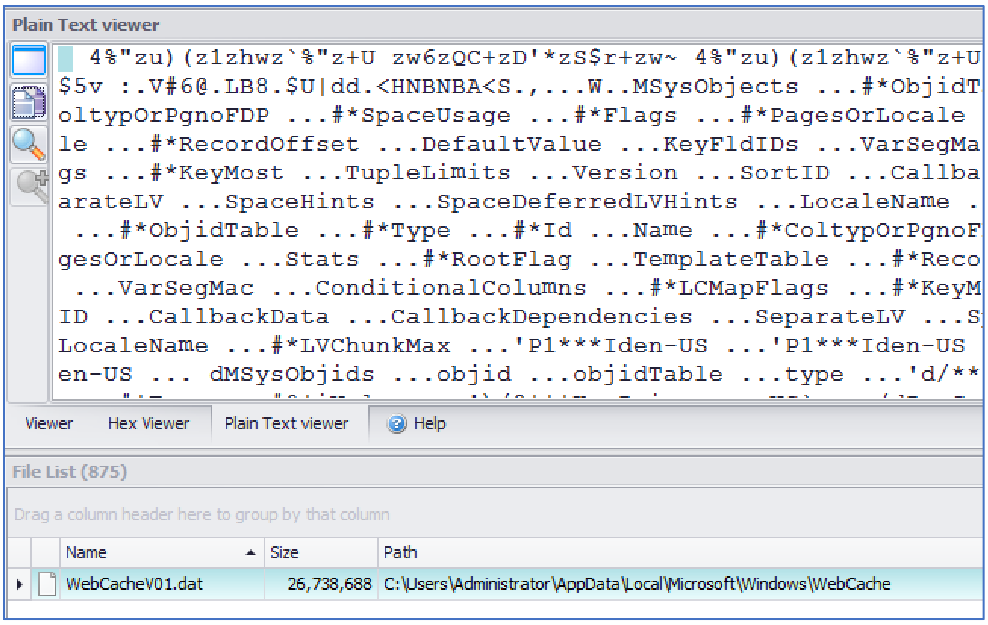
SystemIndex_Gthr [Table ID = 10, 20 Columns]

ScopeID	DocumentID	SDID	LastModified	TransactionFlags	FileName
5	2	129	01 D8 FE F5 55 0F BC 24	-2143286922	Users
8	1	599	01 D8 FE F4 DD 18 55 B2	-2143286922	Start Menu
9	156	252	01 D9 0E A6 B1 C5 87 EE	-2143286922	Administrator
9	111	129	01 D8 FE F3 74 50 32 D0	-1879045770	Default
9	114	186	01 D7 C6 D3 A5 54 99 00	-1879015067	desktop.ini
9	115	136	01 D8 FE F4 E7 30 CA 05	-2143286922	Public
10	970	240	01 D9 0E BF D2 7D 5E 32	-2143286922	Drafts
10	971	240	01 D9 0E BF D3 90 73 F3	-2143286922	Inbox
10	972	240	01 D9 0E BF D5 F7 09 A7	-2143286922	Path unknown
10	973	240	01 D9 0E BF D6 CD 96 20	-2143286922	Sent Items
10	969	240	01 D9 0E BF D2 6E DD 42	-2143286922	Deleted Items
11	974	246	01 D9 10 59 A4 61 2A F5	-2147483275	ADS Machine.eml
11	975	246	01 D9 10 59 A4 55 1A 8A	-2147483275	Breakfast on Tuesday.eml
11	976	246	01 D9 10 59 A3 E0 D0 11	-2147483275	Buffalo Gap.eml
11	977	246	01 D9 10 59 A3 B7 E2 79	-2147483275	epe model for 01_25_02.eml
11	978	246	01 D9 10 59 A3 2C 97 A1	-2147483275	EPE model for 01_29_02.eml
11	979	246	01 D9 0E BF D6 A2 BA 75	-2147483291	EPE Schedules for Saturday 02_02_02.eml
11	980	246	01 D9 10 59 A3 1C EC 68	-2147483275	EPE Schedules for the past two days.eml
11	981	246	01 D9 10 59 A2 A4 E6 43	-2147483275	FW_ FW_ Laid-Off A No Nothing Production.eml
11	982	246	01 D9 10 59 A2 9B 3A EA	-2147483275	How are you doing (plus a little about me) ? .eml
11	983	246	01 D9 10 59 A2 91 8D 96	-2147483275	RE_ How are you doing (plus a little about me) ? .eml
11	984	246	01 D9 10 59 A2 85 79 10	-2147483275	Trader Assistant.eml

Figure 57 Captured search database viewed using Nirsoft utility.

Capture Browser Cache Files

Internet Explorer (EDGE) cache files were captured.

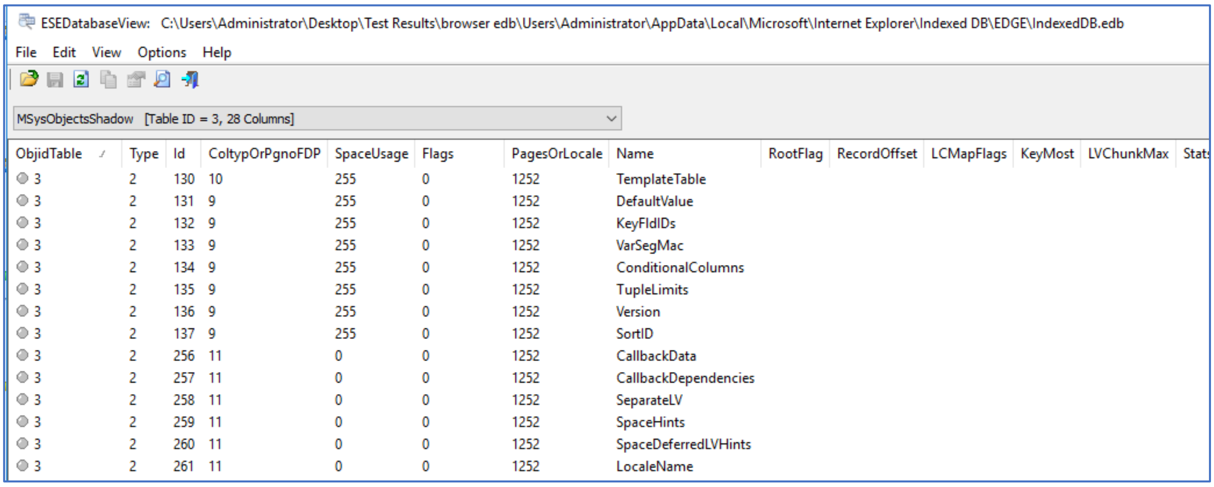


```
 4%"zu)(z1zhwz`%"z+U zw6zQC+zD'*zS$r+zw~ 4%"zu)(z1zhwz`%"z+U
$5v :.V#6@.LB8.$U|dd.<HNBNBA<S.,...W..MSysObjects ...#*ObjidT
oltypOrPgnoFDP ...#*SpaceUsage ...#*Flags ...#*PagesOrLocale
le ...#*RecordOffset ...DefaultValue ...KeyFldIDs ...VarSegMa
gs ...#*KeyMost ...TupleLimits ...Version ...SortID ...Callba
arateLV ...SpaceHints ...SpaceDeferredLVHints ...LocaleName .
 ...#*ObjidTable ...#*Type ...#*Id ...Name ...#*ColtypOrPgnoF
gesOrLocale ...Stats ...#*RootFlag ...TemplateTable ...#*Reco
 ...VarSegMac ...ConditionalColumns ...#*LCMapFlags ...#*KeyM
ID ...CallbackData ...CallbackDependencies ...SeparateLV ...S
LocaleName ...#*LVChunkMax ...'P1***Iden-US ...'P1***Iden-US
en-US ... dMSysObjids ...objid ...objidTable ...type ...'d/**
```

Name	Size	Path
WebCacheV01.dat	26,738,688	C:\Users\Administrator\AppData\Local\Microsoft\Windows\WebCache

Figure 58 Captured Internet Explorer cache file.

The captured cache file was viewed in the Nirsoft utility ESEDatabaseView which reported no errors.


ESEDatabaseView: C:\Users\Administrator\Desktop\Test Results\browser edb\Users\Administrator\AppData\Local\Microsoft\Internet Explorer\Indexed DB\EDGE\IndexedDB.edb
File
Edit
View
Options
Help
MSysObjectsShadow [Table ID = 3, 28 Columns]

ObjidTable	Type	Id	ColtypOrPgnoFDP	SpaceUsage	Flags	PagesOrLocale	Name	RootFlag	RecordOffset	LCMapFlags	KeyMost	LVChunkMax	Stat
3	2	130	10	255	0	1252	TemplateTable						
3	2	131	9	255	0	1252	DefaultValue						
3	2	132	9	255	0	1252	KeyFldIDs						
3	2	133	9	255	0	1252	VarSegMac						
3	2	134	9	255	0	1252	ConditionalColumns						
3	2	135	9	255	0	1252	TupleLimits						
3	2	136	9	255	0	1252	Version						
3	2	137	9	255	0	1252	SortID						
3	2	256	11	0	0	1252	CallbackData						
3	2	257	11	0	0	1252	CallbackDependencies						
3	2	258	11	0	0	1252	SeparateLV						
3	2	259	11	0	0	1252	SpaceHints						
3	2	260	11	0	0	1252	SpaceDeferredLVHints						
3	2	261	11	0	0	1252	LocaleName						

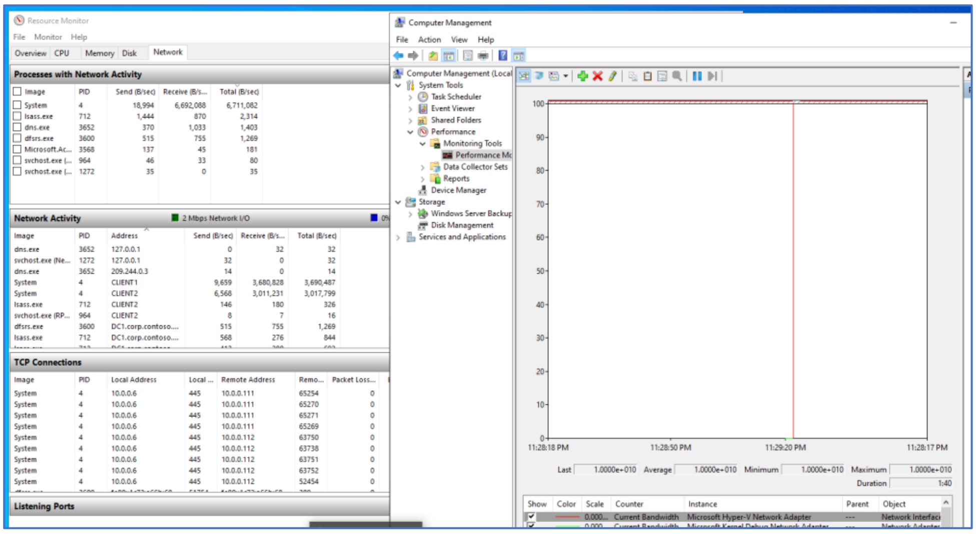
Figure 59 Captured Microsoft Edge cache.

Specific cache files would require the practitioner to install the appropriate browser at one or more of the endpoints and create some browsing data. This would be relatively trivial (assuming there was a network connection available).

A note was made to add a paragraph to the Framework instructions relating to the installation of other browsers (if required).

6.6.2 Other functional tests

1. Scalability - the ability to run multiple concurrent search/collections. This may be estimated by having the forensic tool search through the entire file system while monitoring the network load at the central collection machine.



Processes with Network Activity

Image	PID	Send (B/sec)	Receive (B/s...	Total (B/sec)
☐ System	4	18,994	6,692,088	6,711,082
☐ lsass.exe	712	1,444	870	2,314
☐ dns.exe	3652	370	1,033	1,403
☐ dfsrs.exe	3600	515	755	1,269
☐ Microsoft.Ac...	3568	137	45	181
☐ svchost.exe (...	964	46	33	80
☐ svchost.exe (...	1272	35	0	35

Network Activity 🟩 2 Mbps Network I/O

Image	PID	Address	Send (B/sec)	Receive (B/s...	Total (B/sec)
dns.exe	3652	127.0.0.1	0	32	32
svchost.exe (Ne...	1272	127.0.0.1	32	0	32
dns.exe	3652	209.244.0.3	14	0	14
System	4	CLIENT1	9,659	3,680,828	3,690,487
System	4	CLIENT2	6,568	3,011,231	3,017,799
lsass.exe	712	CLIENT2	146	180	326
svchost.exe (RP...	964	CLIENT2	8	7	16
dfsrs.exe	3600	DC1.corp.contoso....	515	755	1,269
lsass.exe	712	DC1.corp.contoso....	568	276	844

TCP Connections

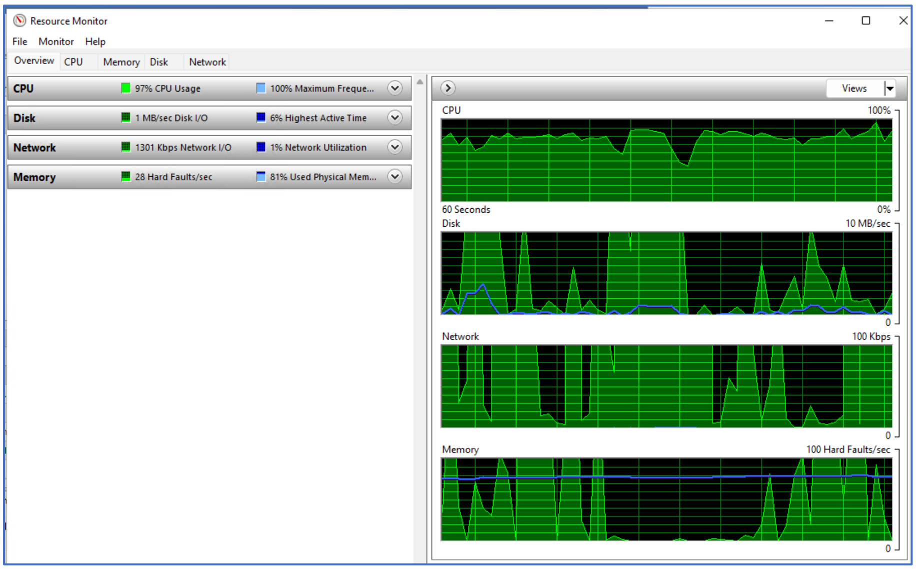
Image	PID	Local Address	Local ...	Remote Address	Remo...	Packet Loss...
System	4	10.0.0.6	445	10.0.0.111	65254	0
System	4	10.0.0.6	445	10.0.0.111	65270	0
System	4	10.0.0.6	445	10.0.0.111	65271	0
System	4	10.0.0.6	445	10.0.0.111	65269	0
System	4	10.0.0.6	445	10.0.0.112	63750	0
System	4	10.0.0.6	445	10.0.0.112	63738	0
System	4	10.0.0.6	445	10.0.0.112	63751	0
System	4	10.0.0.6	445	10.0.0.112	63752	0
System	4	10.0.0.6	445	10.0.0.112	52454	0

Last 1.0000e+010 Average 1.0000e+010 Minimum 1.0000e+010 Maximum 1.0000e+010
Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object
☑	🟥	0.000...	Current Bandwidth	Microsoft Hyper-V Network Adapter	---	Network Interfac

Figure 60 Network Bandwidth utilisation on machine DC1 while running a collection.

2. Minimal impact on end user – This may be estimated by logging into one of the endpoint machines during processing and determine the resources available to the user.



CPU	🟩 97% CPU Usage	🟦 100% Maximum Freque...
Disk	🟩 1 MB/sec Disk I/O	🟦 6% Highest Active Time
Network	🟩 1301 Kbps Network I/O	🟦 1% Network Utilization
Memory	🟩 28 Hard Faults/sec	🟦 81% Used Physical Mem...

Figure 61 Resource utilisation on CLIENT2 endpoint while running a collection.

6.7 CHAPTER CONCLUSION

This chapter has covered the Demonstration stage of the DSR methodology in which AFERA has been used to evaluate a digital forensic tool that employs remote agents. The evaluation environment was installed, the data set copied to endpoints on the virtual domain after which the functional and non-functional criteria were completed. This exercise revealed several items that needed to be addressed:

- There was an error in the data set that produced too many search results for one of the terms.
- The current method of compressing the data set for downloading and deployment was found to be failing to preserve the alternate data streams for one of the tests.
- The data for testing the capture of RDP caches was not present.
- There was no instruction to limit the search location when evaluating the tools which meant that the evaluation process was spending a lot of time working through irrelevant operating system data.

These items were addressed by making changes to the data set, making additions to the instructions for using the Framework and by changing the compressing/decompression process.

Furthermore, two bugs were found in the tool chosen for the Demonstration exercise which were reported to the developer.

Having run through an instance of using the Framework and made the necessary changes to the Framework's documentation and data set the next stage in the DSR methodology is to have experts review the artefact and provide feedback – this is covered in the next chapter.

7 EVALUATION OF THE FRAMEWORK

7.1 INTRODUCTION

A "central and critical" part of design science research is the evaluation of the design artefact (Cross, 2001; John Venable, Pries-Heje, & Baskerville, 2014) and the key aim of Information Systems (IS) design science research is 'utility' (Hevner et al., 2004; March & Smith, 1995; Siau & Rossi, 2011).

"*Evaluation is the process of determining the worth, merit, or significance of entities; and evaluations are the outcome of that process. Evaluation may be external or internal, or a mix of these; and it may be quantitative or qualitative, or a mix of these. It is strongly although not always sharply distinct from explanation*." (Scriven, 1998, p. 7)

The main purpose of evaluation in DSR is to determine how well a designed artefact or achieves its expected environmental utility. A secondary purpose is to provide evidence that the theory leads to an artefact that has utility. (John Venable, Pries-Heje, & Baskerville, 2016)

When adopting a DSR methodology the researcher needs to be mindful of the potential errors that can occur when it comes to evaluating the designed artefact. These errors take the form of Type 1 Errors (false positives) and Type 2 Errors (false negatives) (John Venable et al., 2016).

False positive results occur when the evaluation finding is that the artefact works (or the associated design theory is correct) when this is not the case. False negative results occur when the evaluation finding is that the artefact does not work (or the associated design theory is incorrect) when this is also not the case.

Sonnenberg and Brocke (2012) agree with Venable et al. (2014) and Winter (2008) in relation to design science research literature lacking in guidance for researchers on how to structure evaluation strategies for their artefacts. They propose the adoption of 'patterns' than can be used by researchers to "articulate and justify" strategies for artefact evaluation.

Sonnenberg and Brocke (2012) arrived at their patterns after conducting a survey of prior DSR literature, and state that to ensure that the design, construction, and use of an artefact meets the design objectives (to address the identified research problem) there needs to be an evaluation based on a set of criteria.

7.2 Framework for Evaluation in Design Science

The general lack of guidance for researchers using the DSR methodology prompted Venable et al. (John Venable et al., 2014) to develop the Framework for Evaluation in Design Science (FEDS) which has been adopted for this research. The framework provides a two-dimensional characterisation of the DSR evaluation. One dimension identifies the functional purpose of the evaluation, and the other identifies the paradigm of the evaluation as shown in Figure 62.

The functional purpose dimension components are:

1. Naturalistic: natural environment (case study, focus group, participant observation, ethnography, phenomenology, survey (qualitative, quantitative)).
2. Artificial: mathematical/logical proof, lab experiment, role-playing simulation, computer simulation, field experiment.

The paradigm dimension components are:

1. Formative evaluations that are used to interpret the results of observation or experience with the artefact to shape its development and improve its performance or characteristics.
2. Summative evaluations that are used to assist with supporting decisions involved in the selection of the artefact for a particular application through interpretation of observations from which are derived shared meanings about the artefact in different contexts.

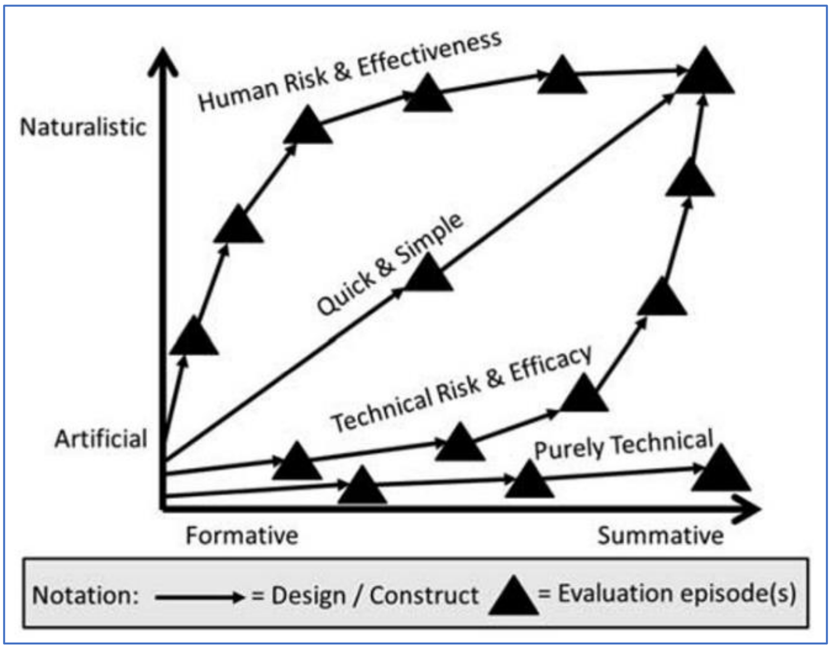


Figure 62 Framework for Evaluation in Design Science (FEDS) (Venable et al. 2014)

The framework is intended to associate the evaluation goals and the evaluation strategies by providing a classification of evaluation strategies and relating these to evaluation goals. The evaluation strategies relate to the trajectories taken for circumstances of artefacts and evaluation methods.

The authors point out that *"the increasing use of more naturalistic evaluations improves the quality of the knowledge outcomes concerning the artefact's effectiveness in real use, as the artefact increases in quality and the risks become low enough for real use by real users*" (John Venable et al., 2016, p. 4)

There are four steps in the FEDS:

1. Explicate the goals of the evaluation.
2. Choose the evaluation strategy (or strategies)
3. Determine the properties to evaluate.
4. Design the individual evaluation episode(s).

7.2.1 Step 1 Explicate the goals of the evaluation.

There are four goals for the evaluation: rigour, uncertainty and risk reduction, ethics, and efficiency. Of these, 'rigour' has been selected as being most appropriate goal as it is associated with ensuring that the artefact works in a real situation, i.e., it is effective, which fits in with the requirements for forensic tools. The other strategies have a stronger focus on risk and costs.

7.2.2 Step 2 Choose the evaluation strategy (or strategies)

There are four processes to be followed in choosing a design strategy.

The first process involves evaluating and prioritising design risks. Given the relatively simple design process the design risks for this research have been classified as minimal. This eliminates the Human Risk and Effectiveness strategy.

The second process for choosing a design strategy involves assessing how costly the evaluation would be in a real setting on real systems. Given that the proposed framework incorporates its own realistic environment (or one that can be configured as required without having to risk operational systems) the cost implications for evaluation are considered minimal. This eliminates the Technical Risk and Efficacy strategy.

The third process requires that an evaluation takes place to consider if the artefact is 'purely technical' or is intended to be deployed in the far future. The proposed artefact is not considered to be purely technical, in that it will be used by real people, and it is intended to be deployed within a short timescale. This eliminates the Purely Technical strategy.

The final process in choosing an evaluation strategy involves considering whether the construction phase is 'small and simple' or 'large and complex.' Having concluded that the construction phase for the current research is 'small and simple' and having eliminated the alternative strategies as being inappropriate via the preceding selection processes, the Quick and Simple design strategy has been selected for this research.

Relating this choice back to Figure 62 the evaluation episodes will have been the Demonstration DSRP activity followed by the Expert Practitioner Panel evaluation.

7.2.3 Step 3 Determine the properties to evaluate

The Expert Panel are requested to review the framework for ease of use, content and whether it meets its aim of providing a useful way for practitioners to evaluate remote forensic tools (see Appendix 1). This addresses the goal of 'rigour.' The Expert Panel Evaluation involves each expert completing the six items in the section 'Tasks for Reviewers' as follows:

1. Please review the documentation and determine if you believe that it is clear and simple to follow. If this is not the case, please identify the areas that need improvement.
2. Consider the non-functional requirements and determine if you believe that they are reasonable and complete. If this is not the case, please identify the areas that need improvement.
3. Consider the functional requirements and determine if you believe that they are reasonable and complete. If this is not the case, please identify the areas that need improvement.
4. Consider the use of the Windows 11 and Office 365 Deployment Lab and determine if this is a useful environment for evaluating the remote forensic tools. If this is not the case, please identify the areas that need improvement.
5. Consider if the framework meets its aim of providing a means by which practitioners can evaluate remote forensic tools. If this is not the case, please identify the areas that need improvement.
6. Consider if there are any other comments you wish to provide as feedback.

7.2.4 Step 4 Design the individual evaluation episode(s)

The design of the evaluation episode for this research involves a three-step process:

1. Identifying and analysing environment constraints

   Without knowledge of the specific resources available to the Expert Panel members the generic key constraints were identified as:

    a. Time to set up the tool testing environment.
    b. Resources required for the tool testing environment.
    c. Time to complete the required evaluation tasks.

2. Prioritising the factors associated with the key constraints.

   The DSRP Demonstration activity confirmed that the inclusion of the Windows 11 Deployment Kit addressed the key resource and time burden constraints on the Expert Panel by creating the tool evaluation environment in a self-contained package that required minimal intervention by the user and that could be run on a 'basic' specification laptop with sufficient disk space.

   The required evaluation tasks were kept to a minimum and feedback was requested in the form of a Yes/No/Comments response to basic questions regarding the Framework, thus minimising the time to complete the evaluation tasks.

3. Plan for the number of evaluation episodes

   In keeping with the Quick and Simple design strategy selected for the DSRP Evaluation Activity a single evaluation episode with an Expert Panel was chosen.

7.3 THE EXPERT PANEL

An Expert Panel was created who were then provided with the Framework documents and the questionnaire (Appendix 7) as well as the required ethics consent and information forms. As Bruce (Bruce, 2007) points out, "*Ultimately, the number of data events is less important than the trustworthiness of the reporting*". While the Panel is not intended to be a representative sample of all digital forensic practitioners it comprises of very experienced and qualified practitioners within the field who can provide authoritative feedback. While the number of experts required may be influenced by the complexity of the artefact, Cornelissen et al. (Cornelissen, Berg, Koops, & Kaymak,

2002) suggest that the recruitment of a group of experts is completed when the group is diverse and able to bring different perspectives to bear in their evaluation.

Reuzel (2001) also suggests that a heterogenous group of experts is preferrable to a homogenous group as this compensates for different points of view.

The criteria used in the selection of the Expert Panel is based on Cornelissen et al. (2002), namely:

- a person's period of learning and experience in a specific domain of knowledge
- the specific circumstances in which their experience has been gained.

Twelve people were selected for the Expert Panel based on a review of their background and experience. Their roles/job descriptions are:

- Cyber security expert – 15 years with State Police
- Founder and Director of a digital forensics consultancy, member of a State Cyber Crimes Team
- Ex-Lead in Global Investigations and Threat Intelligence for Fortune 500 company and digital forensics instructor
- Associate Director for Big Four consulting company leading a digital forensics team
- Serving Defence Force cyber warfare practitioner and instructor
- Researcher and consultant, cyber security and digital forensics
- Principal of digital forensics consultancy
- Managing Director of digital forensics consultancy and instructor
- Principal Forensic Analyst and SANS instructor
- Cyber security professional, instructor, and SANS contractor
- Developer and CEO of a digital forensics software company, instructor
- The founder of a digital forensics tool company and ex-Federal law enforcement agent.

7.3.1 Expert Panel Feedback

The feedback received from the Expert Panel has been summarised based on the six questions for which feedback was requested.

1. **Consider if the framework is clear and simple to follow.**

   100% of respondents answered 'yes' to this question.

   Comments included:

- *"Clear and concise"*
- *"The framework is easy to follow."*
- *"I felt the framework was easy simple and easy to follow."*
- *"Part 1 was easy to understand what was required as feedback for setting up and running the EDR that has been chosen."*

- *"Evaluating participant did not perceive any vague instruction in the framework."*

2. **Consider the non-functional requirements and determine if you believe that they are reasonable and complete.**

80% of respondents answered 'yes' to this question.

For those who felt there were elements that could be addressed comments included:

2.1 "Ability to recover from broken connections (whether temporary dropout or if a user shuts down for the night)"

2.2 "I think it comes under "Useability", but I'd also rate a tool on the usefulness of its error messages. "Task failed" is not as useful as "unable to communicate with target location, is port 443/8443 blocked?""

2.3 "You may have to consider network IDS, Proxy setting s etc so to get true results without triggering any network alerts but this will be network specific and not always apply".

Researcher's observations:

Referring to comment 2.1, This was added to the list of Non-functional Considerations.

Referring to comment 2.2, The feature 'Usability' in the list of Non-functional Considerations will be extended to explicitly include error messages, although how this will be determined may involve reference to a user guide rather than using the tool.

Referring to comment 2.3, The practitioner can tailor the lab configuration for environment-specific tests (including installing additional software) which should accommodate most of this type of consideration.

3. **Consider the functional requirements and determine if you believe that they are reasonable and complete.**

80% of respondents answered 'yes' to this question.

Comments included:

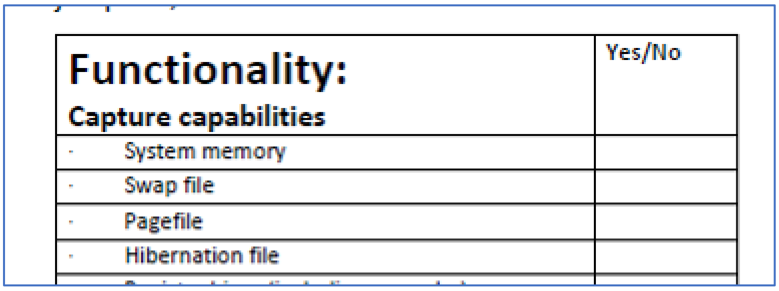
3.1 "Very good list of functional requirements which covers the most commonly expected artifacts usually sought by a practitioner."

3.2 "I felt the overall functional objectives made sense. The one area I felt could be improved around volatile data collection to complete the incident response capabilities."

3.3 "All functional requirements included in this test have been encountered either in prior DFIR engagements and other testing scenarios."

Researcher's observations:

Referring to comment 3.2, system memory capture (including pagefile etc.) is already a consideration in the features list.



Functionality: Capture capabilities	Yes/No
- System memory	
- Swap file	
- Pagefile	
- Hibernation file	

4. **Consider the use of the Windows 11 and Office 365 Deployment Lab and determine if this is a useful environment for evaluating the remote forensic tools.**

   80% of respondents answered ‘yes’ to this question.

Comments included:

4.1 “It is a worthwhile environment as it is very common, and most practitioners would be familiar with it.”

4.2 “Although noting that the majority of real use cases currently won’t be on a Win’11 environment so if the objective is to test for “real world” scenario’s a Win10 environment might be more realistic.”

4.4 “Evaluating participant does not use Office 365 as a routine work production suite but the Deployment Lab should be sufficient for testing purposes.”

Researcher’s observation:

Referring to comment 4.2, a note has been added to the User Guide that mentions that there is a Windows 10 version of the Lab.

5. **Consider if the framework meets its aim of providing a means by which practitioners can evaluate remote forensic tools for use across networked systems.**

   100% of respondents answered ‘yes’ to this question.

   Comments included:

“Hits most of the critical measurements to give an objective comparison.”

“Overall, I believe this framework would allow for consistent measurement against various forensic tools.”

“I believe the framework would meet the aim of being able to compare various remote collection tools.

“It is also reasonable to imagine that the framework will evolve with its application.”

“The framework is focused on a controlled Windows environment – and seems solid for this purpose.”

“I've completed my assessment of the materials, and they are totally fit for purpose.”

“The methodology and objectives are clear and meaningful for the normal, "standard" ICT person and should provide a good resource for them.”

“Obviously, one could nit-pick, but I see no point in narrowing applicability for niche improvements - you've got the balance right already.”

6. **Consider if there are any other comments you wish to provide as feedback.**

Detailed feedback:

6.1 “The IR process felt incomplete, and other IR practitioners may also have more input in this area. I felt like this was a solid process to validate the forensics capabilities of any product capable of searching, collecting, and conducting incident response.”

6.2 “I really like the Cloud capability in this framework as many struggles with not only the cost of IR in the cloud but conducting a consistent, streamlined IR process when dealing with cloud-related incidents.”

6.3 “One of the questions I anticipate the IR community will want to know is if the remote agents can run the entire IR collection process meaning it can collect based on the order of volatility and collect Volatile data first, then move through the IR collection process and do so on each major OS. You already identified Cloud IR capabilities.

6.7 “Depending on how the agent app is built, an antivirus (or similar specific tool) exception might be needed (this is a comment to the “setting up”, p.2)”

6.8 “I am unsure that the speed of agent deployment is an important criterion. However, ease of this process is important, especially given the variety of network topologies and products which may imply permission restrictions. I would also say that mass deployment abilities are important to have, when more than a single computer is to be acquired.”

6.10 “I would say that the agent size can also be important.”

6.13 “Would be good to integrate with other aspects of Forensics especially DFIR and capture some IR points for broader use in the community e.g., identify any malware or Breaches in the environment and more than just windows environments.”

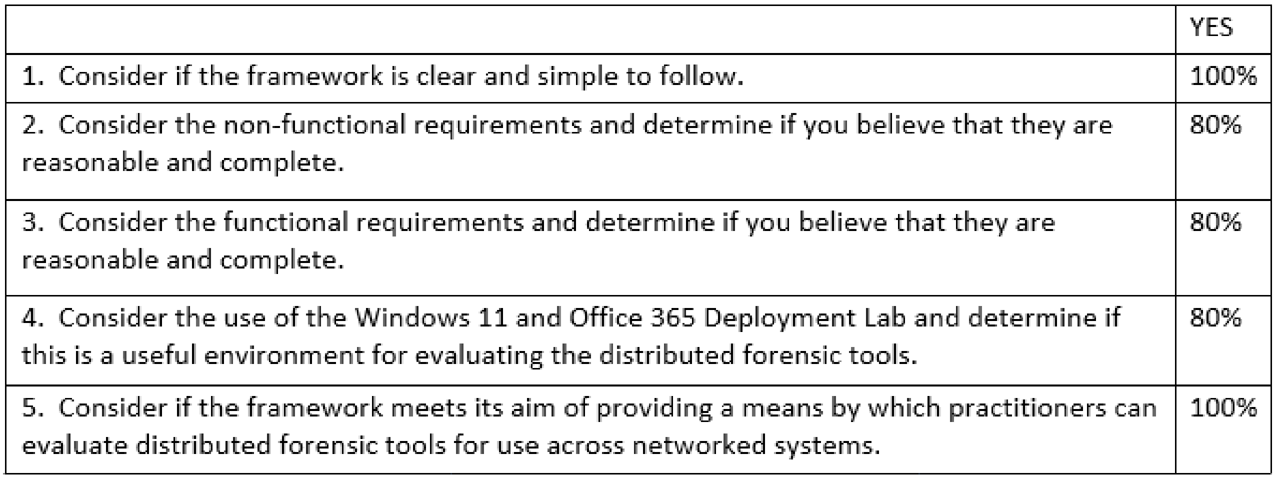
Researcher’s observations:

General - The scope of this research and framework is the field of digital forensics, although there is some overlap of features, and the Framework could be extended to cover other areas.

Referring to comment 6.10, the size of the installed agent will be added to the list of non-functional considerations as it will impact the memory capture process to a degree determined by how much of the memory it uses for itself.

7.4 CHAPTER CONCLUSION

The Expert Panel feedback for the 5 questions involving a yes/no option were:



	YES
1. Consider if the framework is clear and simple to follow.	100%
2. Consider the non-functional requirements and determine if you believe that they are reasonable and complete.	80%
3. Consider the functional requirements and determine if you believe that they are reasonable and complete.	80%
4. Consider the use of the Windows 11 and Office 365 Deployment Lab and determine if this is a useful environment for evaluating the distributed forensic tools.	80%
5. Consider if the framework meets its aim of providing a means by which practitioners can evaluate distributed forensic tools for use across networked systems.	100%

Following on from the Demonstration activity covered in Chapter 6 which led to some minor adjustments to the Framework, the Evaluation activity covered in this chapter has verified that the Framework is a suitable means by which practitioners can evaluate remote forensic tools. The suggestions for improvements that have been highlighted under 'Researcher's comments' has been implemented in the final version of AFERA that is included in Appendices 4, 5 and 6.

8 CONCLUSION

8.1 INTRODUCTION

This chapter summarises how the DSRP has been applied to this research to develop an artefact – AFERA. It also includes discussion of how the results of this research will be made available to the wider community of digital forensic professionals in accordance with the communications stage of the DSRP.

8.2 RESEARCH SUMMARY

The problem addressed in this research is that there has been no framework for evaluating digital forensic tools that use remote agents as part of their processing. This means that practitioners have not had a consistent way to validate the outputs or compare tools which they rely upon and that are fundamental to their work.

This research has been structured around the DSRP model with the aim "*to develop a framework that will enable practitioners to evaluate the performance of remote agent-based forensic tools designed for use in DF investigations.*" This aim was addressed through meeting the research objectives (Section 1.6.3):

1. Identify the necessary elements for evaluating remote agent-based tools for use in DF investigations.
2. Identify or create one or more datasets for use in the evaluation of remote agent-based tools.
3. Develop an artefact for evaluating remote agent-based tools for use in DF investigations.
4. Identify an appropriate evaluation environment for use with the new artefact.
5. Demonstrate the utility of the new artefact, data set and evaluation environment.

Objective 1 was met in Chapter 4 which was the Design and Development stage of the DSRP model and involved the identification of the required features of a remote forensic tool by drawing on information provided by two authoritative sources (NIST and SANS) as well as a panel of expert practitioners.

The development of AFERA has shown that many test scenarios can be refined to produce a small number of functional criteria that can be used as the basis for tool evaluation without reducing the utility of the method.

Objective 2 was met in Chapter 5 through the development of a dataset.

This research has shown that by building on the refined functional criteria developed during the development of AFERA it is possible to produce a relatively small dataset and still evaluate critical functionality.

Objective 3 was met in Chapter 5 with the development of AFERA (the DSR artefact) which consists of an evaluation environment, a set of instructions for using AFERA, a set of criteria and an associated data set.

The evaluation criteria were created together with instructions on how some of the required features may be assessed using the evaluation environment and the data set that has been built to contain the various elements in a form that could be easily deployed to multiple endpoint systems.

Objective 4 was met in Chapter 5 which identified an appropriate environment for forensic practitioners to use when evaluating forensic tools that may be deployed across a network. This environment is relatively easy to create and will remain relevant to systems likely to be encountered by forensic practitioners given that it is maintained by Microsoft to the latest Windows build and is updated every 6 months.

While it is likely that Microsoft will maintain their Lab Deployment Kits and make them readily available this research has shown that a search of online resources has the potential to identify timesaving generic tools and techniques that are not associated with digital forensic testing.

Objective 5 was met in Chapters 6 and 7. Chapter 6 covered the Demonstration stage of the DSR methodology in which AFERA was used to evaluate a digital forensic tool that employs remote agents. The evaluation environment was installed, the data set copied to endpoints on the virtual domain after which the functional criteria were evaluated.

Chapter 7 showed that seeking feedback from experts in the field can provide independent verification of an artefact’s utility.

8.2.1 Limitations

The in-house test of the Framework for the 'Demonstration' stage of the DSRP used only a single tool. However, this activity was only intended as a 'proof of concept' to show if there are any serious issues with the initial design of the Framework such that modifications can be made prior to involving the Expert Panel in the final independent evaluation.

Although the Expert Panel included experts from a broad range of digital forensics practice areas and in different jurisdictions it cannot be claimed to be representative of the overall population of digital forensic practitioners. However, these experts were chosen because they are both practitioners and, in several cases, authorities in the field. In addition, two of the practitioners develop (and therefore test) digital forensic software and are thus able to provide valuable insights.

As identified in some of the Expert Panel feedback, this research has focussed on the Microsoft Windows operating system and so it does not aid those working in Linux or Apple OS environments. The research has also focussed on a single test domain to keep the framework simple and easy to create, however the virtual machines can be replicated and configured to produce multiple domains if required.

8.3 FUTURE WORK

The design of AFERA is such that it can be updated to keep track with changes in the Windows operating system. It is also extensible such that additional seed data can be included as applications are updated or others become of interested in the evaluation process.

AFERA could be extended to accommodate more Incident Response functionality through additions to the seed data and the running of executable code on the endpoints to simulate a compromised machine.

The limitations in relation to both the types of operating system targeted and the network environment are key areas that could be addressed through future research. For instance, while there isn't a direct equivalent to Microsoft's Lab Deployment Kit (LDK) specifically designed for Linux environments, there are various tools and approaches available in the Linux ecosystem that can serve a similar purpose. Some of these tools include:

1. **Virtualization Platforms:** In addition to Windows Hyper-V, Linux supports a range of virtualization platforms such as KVM, VirtualBox, and VMware that allow you to create and manage virtual machines for testing and development purposes.

2. **Docker and Containers:** These technologies provide lightweight and isolated environments for deploying applications.

3. **Vagrant:** This is a tool for managing virtualized development environments. While often associated with creating development environments for web development, it can also be used for other scenarios.

4. **Ansible:** This is an automation tool that can be used to deploy and manage lab environments.

5. **Proxmox Virtual Environment (Proxmox VE):** This is an open-source virtualization management platform that combines two virtualization technologies: KVM (Kernel-based Virtual Machine) for virtual machines and LXC (Linux Containers) for lightweight container-based virtualization.

6. **Cloud Services:** Cloud platforms like Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure provide options to create and manage virtual machines and development environments in the cloud.

While these tools may not offer an exact equivalent to Microsoft's Lab Deployment Kit, they provide a range of options for creating, deploying, and managing lab environments on Linux that could form the basis for digital forensic tool evaluation.

In relation to the Apple environment, many of the generic resources identified for the Linux environment would work for Apple computers. In addition, there are:

1. **VMware Fusion:** This is a virtualization software that allows you to create and run virtual machines on macOS (it also supports other guest operating systems, including Windows and Linux).
2. **Parallels Desktop:** This is another popular virtualization solution for macOS. It enables you to run virtual machines alongside your native macOS environment.
3. **Xcode:** This is Apple's integrated development environment that includes tools for creating and managing development environments. You can create virtual machines for testing and development purposes.

4. **macOS Server:** While macOS Server has been streamlined in recent years, it still provides some management capabilities for lab environments, including network services and profile management.

8.4 COMMUNICATION

The final stage of the DSRP is the 'Communication' stage. To address this element, several journal papers are being developed for publication in 2023 and the final thesis will be available electronically from the Curtin University Library website. This information will be made available to the members of appropriate user groups and forums.

Proposed titles for research papers:

- The Advanced Framework for Evaluating Remote Agents (AFERA):
  A resource for Digital Forensic Practitioners
- The use of the Windows 11 O365 Deployment Lab for Testing Digital Forensic Tools
- Creation of a Simple Data Set for Testing Digital Forensic Tools
- A Design Science Approach to Digital Forensics Theory Development

9 APPENDICES

9.1 APPENDIX 1 - THE FRAMEWORK GUIDE DOCUMENT (INITIAL VERSION)

Using the Framework

Introduction

This framework is intended to assist forensic practitioners evaluate digital forensic tools that employ remote agents to collect data from endpoints. The Framework does not provide a score as different environments will necessarily place different emphasis on a tool's capabilities and functions. Instead, it provides two lists of features, functional and non-functional, to be considered when evaluating a tool as well as the means to undertake a few tests for some of the key features, such as the ability to locate certain character strings.

Having reviewed the non-Functional Considerations and the outcome of the tests for the Functional Considerations for each of the forensic tools you are evaluating you will be able to determine if any are likely to meet your requirements as well as form a basis for future tool comparison.

Given that the time and the resources involved in setting up a consistent test environment makes it difficult to undertake a thorough evaluation in the workplace, the Framework incorporates the use of the Microsoft Windows 11 and Office 365 Deployment Lab Kit which automatically creates a small domain consisting of different server types and workstations which can be run on a mid-range laptop with 16GB of RAM. The Lab Kit does not have to be used if a suitable test environment is already available. Once installed, the virtual domain can be used 'as is' or can be further configured to suit your purposes.

To carry out functional tests a collection of seed data has been created that contains a range of different file types and character strings while being relatively small such that it is a trivial task to copy it to the endpoints, whether these are part of the Lab Kit or another test environment.

The forensic tool evaluation itself consists of two parts. The first part is where you consider the non-functional capabilities of the tool you are evaluating. There is no structured format for recording the results for non-functional capabilities of a tool, the intention is to provide a structure for assessment based on your priorities.

The second part of the tool evaluation consists of several functional tests using the provided seed data. This will confirm the tool's remote agent's ability to capture certain forensic artefacts from remote systems and locate data within different filetypes using a range of search methods.

For the functional tests the 7 MB 'seed' data can be downloaded from here:

https://drive.google.com/file/d/1JCk_ezXo1V-gNPeuj63PFCmDKxY6ehkH/view?usp=share_link

Once downloaded the seed data can be extracted from the Zip file to any folder location on the endpoints on which you will deploy the remote forensic agents for the evaluation.

There are a few artefacts that may be required to be installed in addition to the seed data that are specific to your requirements, such as browsers.

In summary, the evaluation tasks consist of the following stages:

- Install the evaluation environment (if required).
- Copy the seed data to the required endpoints.
- Install and configure the tool(s) to be evaluated.
- Work through the criteria for non-functional considerations.
- Work through the criteria for functional considerations.
- Evaluate the tools under review by referring to the results obtained by using the two sets of criteria.

Setting up the evaluation environment (if required)

1. Download the Windows 11 and Office 365 Deployment Lab Kit and lab guides from here (note the download is 30GB): https://info.microsoft.com/ww-landing-windows-11-office-365-lab-kit.html?culture=en-us&country=us

2. Ensure the lab machine you will use for the evaluation meets the following system requirements:

    - Hyper-V role installed.
    - Administrative rights on the device
    - 150 GB of free disk space (300 GB recommended)
    - High-throughput disk subsystem
    - 16 GB of available memory (32 GB recommended)
    - High-end processor for faster processing

3. Read the lab guides for installation and install the Deployment Lab on your evaluation machine.
4. Extract the seed data to the required number of endpoints.
5. Make any changes that are needed to the servers and/or workstations to better reflect your own environment (if appropriate).

9.2 APPENDIX 2 - THE NON-FUNCTIONAL CRITERIA (INITIAL VERSION)

Forensic Tool (and version)

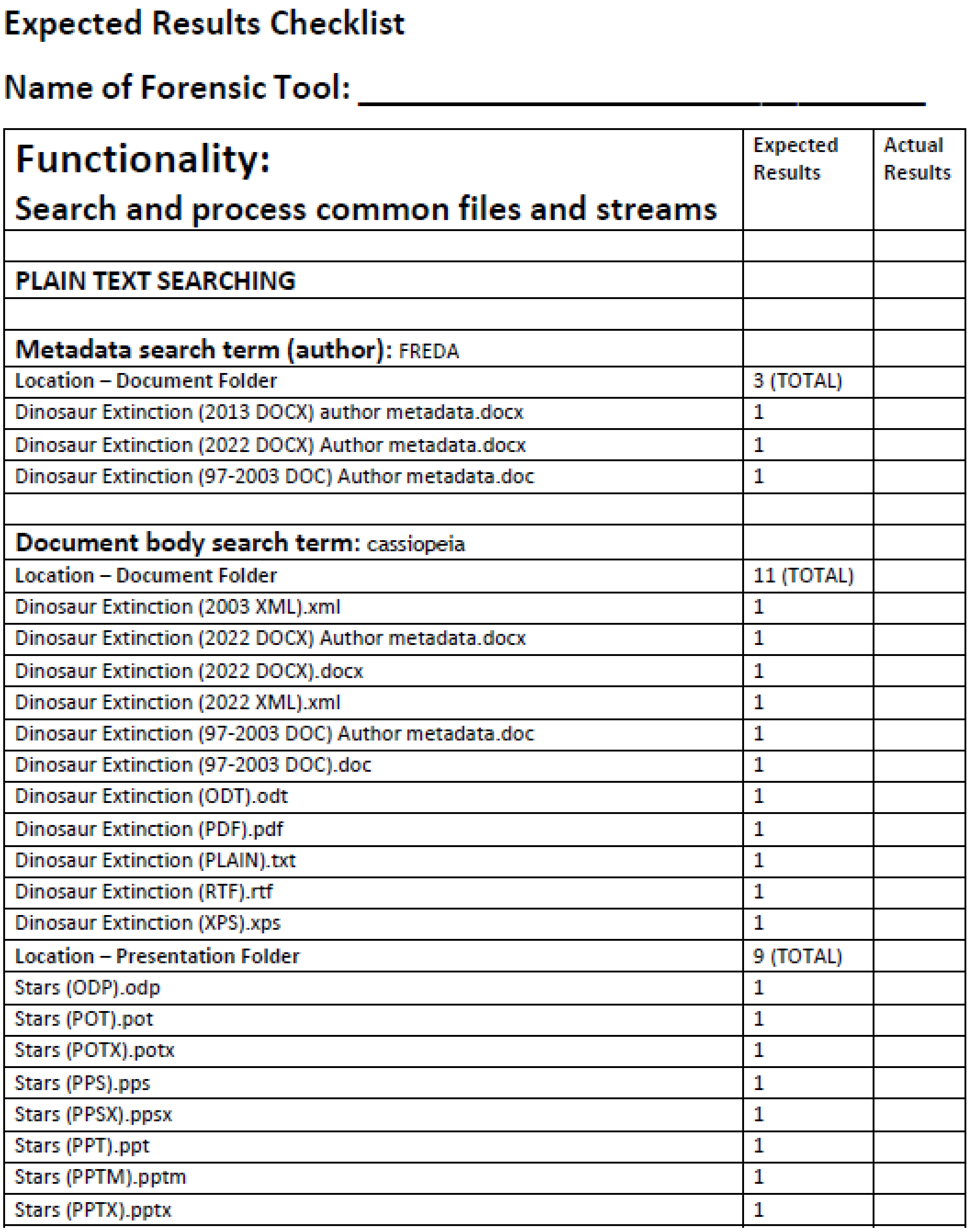
____

Part 1 - Evaluating Non-functional Considerations

The Non-functional items should be considered but there is no score associated with these items as their relevance and importance will differ from one organisation to another.

1. Accuracy and Reliability – this is determined by the results of the combined functional tests.
2. Scalability - the ability to run multiple concurrent search/collections. This may be estimated by having the forensic tool search through the entire file system while monitoring the network load at the central collection machine.
3. Minimal impact on end user – for the Framework this will be inferred by the resource requirements at the endpoints running the remote clients.
4. Speed – the time to complete the activity.
5. Dependencies - consideration should be given in relation to the agent's dependencies, i.e., any other software that must be installed for the agent to perform.
6. Ease of use - in terms of General Characteristics, the time it takes for a practitioner to learn how to use and deploy the remote agents.
7. Customisation – the ability to change the configuration of the tool, e.g., such that unnecessary functions are not performed.
8. Auditability – the ability to audit the agent's processing.
9. Options for exporting the results of processing.
10. Cost – the cost of licencing the tool and any other pre-requisites.
11. Ability to process machines not on the enterprise network.
12. Invisible to end user
13. Can be run in the cloud.
14. Data sovereignty – e.g., if data is required to be stored in the cloud, where is the data physically located with respect to the applicable jurisdiction.
15. Minimal impact on network bandwidth – for the Framework this will be inferred by the network utilisation of the machine acting as the 'server' for the remote deployed agents.

9.3 Appendix 3 – The Functional criteria (Initial Version)



Expected Results Checklist

Name of Forensic Tool: ______________________________

Functionality: **Search and process common files and streams**	Expected Results	Actual Results
PLAIN TEXT SEARCHING		
Metadata search term (author): FREDA		
Location – Document Folder	3 (TOTAL)	
Dinosaur Extinction (2013 DOCX) author metadata.docx	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Dinosaur Extinction (97-2003 DOC) Author metadata.doc	1	
Document body search term: cassiopeia		
Location – Document Folder	11 (TOTAL)	
Dinosaur Extinction (2003 XML).xml	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Dinosaur Extinction (2022 DOCX).docx	1	
Dinosaur Extinction (2022 XML).xml	1	
Dinosaur Extinction (97-2003 DOC) Author metadata.doc	1	
Dinosaur Extinction (97-2003 DOC).doc	1	
Dinosaur Extinction (ODT).odt	1	
Dinosaur Extinction (PDF).pdf	1	
Dinosaur Extinction (PLAIN).txt	1	
Dinosaur Extinction (RTF).rtf	1	
Dinosaur Extinction (XPS).xps	1	
Location – Presentation Folder	9 (TOTAL)	
Stars (ODP).odp	1	
Stars (POT).pot	1	
Stars (POTX).potx	1	
Stars (PPS).pps	1	
Stars (PPSX).ppsx	1	
Stars (PPT).ppt	1	
Stars (PPTM).pptm	1	
Stars (PPTX).pptx	1	

Stars (XML Presentation).xml	1	
Location – Spreadsheet Folder	12 (TOTAL)	
Stars(97-2003 XLS).xls	1	
Stars(CSV).csv	1	
Stars(CSV-UTF8).csv	1	
Stars(DIF).dif	1	
Stars(Excel 5 XLS).xls	1	
Stars(ODS).ods	1	
Stars(Tab Delimited).txt	1	
Stars(XLSB).xlsb	1	
Stars(XLSM).xlsm	1	
Stars(XLSX).xlsx	1	
Stars(XLTX).xltx	1	
Stars(XML 2003).xml	1	
Location – Publication Folder	3 (TOTAL)	
Publication1(PUB).pub	1	
Publication1(PUB2000).pub	1	
Publication1(PUB98).pub	1	
Location – Database Folder	4 (TOTAL)	
Database1(ACCDB).accdb	1	
Database1(Access 2000 mdb).mdb	1	
Database1(Access 2002-2003 mdb).mdb	1	
places.sqlite	1	
Location – Container Folder	12 (TOTAL)	
archive-7z.7z	1	
archive-cab.CAB	1	
archive-cpio.cpio	1	
archive-dmg.dmg	1	
archive-iso.iso	1	
archive-LZH.lzh	1	
archive-rar.rar	1	
archive-tar.tar	1	
archive-tar_bzip2.tar.bz2	1	
archive-tar_gzip.tar.gz	1	
archive-wim.wim	1	
archive-zip.zip		
Location – ADS Folder (note: Alternate data stream)	1 (TOTAL)	
empty4.txt:ads1.txt	1	

Functionality: **Foreign Language Support**	Expected Results	Actual Results
Foreign Language Search Term1: 中国 東京		
Unicode Chinese/Japanese ideograms	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX).docx	1	
Foreign Language Search Term2: 서울		
Unicode CJK Korean Hangul	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Foreign Language Search Term3: スバル みつびし		
Unicode CJK Japanese phonetic Kana	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX).docx	1	
Foreign Language Search Term4: Сибирь		
Unicode Cyrillic (Russian)	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX) author metadata.docx	1	
Foreign Language Search Term5: الكسكس		
Unicode RTL (Arabic)	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	

Functionality: **Other Search Features**	Expected Results	Actual Results
Normalized Form D search: Mäuse	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX) author metadata.docx	1	
Normalized Form C search: Mäuse	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
WILDCARD SEARCH		
Term: *glycerin	2 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX).docx	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
WHOLE WORD SEARCH		
Term: glycerin	NONE	
BEGINNING OF WORD SEARCH		
Term: Nitro	2 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX).docx	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
END OF WORD SEARCH		
Term: glycerin	2 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2013 DOCX).docx	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
REGEX SEARCH		
Expression: \d{3}-\d{2}-\d{4}	2 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX).docx	1	
Dinosaur Extinction (ODT).odt	1	
HEX SEQUENCE SEARCH		
Sequence: 6F6C6576616E	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (RTF).rtf	1	
CASE SENSITIVE SEARCH		
Term: Nitroglycerin	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	

Functionality: **Search embedded documents**	Expected Result	Actual Result
Search term: Embedded Level1	1 (TOTAL)	
Location: \EMBEDDED\Level 1\ Embedded Level 1.docx	1	
Search term: Embedded Level2	1 (TOTAL)	
Location: \EMBEDDED\Level 2\ Embedded Level 2.docx	1	
Search term: Embedded Level3	1 (TOTAL)	
Location: \EMBEDDED\Level 3\ Embedded Level 3.docx	1	

Functionality: **Search email**	Expected Result	Actual Result
Search term: Buffalo Gap		
Location: PST	5 (TOTAL)	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Deleted Items Subject: RE: Tuesday Morning	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Deleted Items Subject: RE: Buffalo Gap	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Breakfast on Tuesday	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Tuesday Morning	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Buffalo Gap	1	
Location: EML	5 (TOTAL)	
\EMAIL\PST\EML Output\Deleted Items\RE_ Buffalo Gap.eml Subject: RE: Buffalo Gap	1	
\EMAIL\PST\EML Output\Deleted Items\RE_ Tuesday Morning.eml Subject: RE: Buffalo Gap	1	
\EMAIL\PST\EML Output\Sent Items\Breakfast on Tuesday.eml Subject: Breakfast on Tuesday	1	
\EMAIL\PST\EML Output\Sent Items\Buffalo Gap.eml Subject: Buffalo Gap	1	
\EMAIL\PST\EML Output\Sent Items\Tuesday Morning.eml Subject: Tuesday Morning	1	
Location: MSG	1 (TOTAL)	
\EMAIL\PST\MSG Output\NFC Strings - FW Schedule Crawler HourAhead Failure.msg Subject: NFC Strings - FW: Schedule Crawler: HourAhead Failure	1	
Location: MBOX	1 (TOTAL)	
\EMAIL\PST\MBOX\cindyloh3333@gmail.com.mbox Subject: Accounts to close	1	

Functionality: **Identify encrypted files**	Expected Results	Actual Results
Location – Encrypted Folder	**2 (TOTAL)**	
Dinosaur Extinction1 (PDF).pdf	1	
Dinosaur Extinction (2022 DOCX).docx	1	

The following functional test is for you to determine if the tool is able to capture the files identified in the list. For some of the items it will be necessary for you to cause the files to be created on the endpoint machine(s), e.g., jumplists, browser cache files.

Functionality: **Capture capabilities**	Yes/No
- System memory	
- Swap file	
- Pagefile	
- Hibernation file	
- Registry hives (including amcache)	
- EVT & EVTX logs	
- SRUDB	
- Jumplist files	
- Prefetch files	
- Windows LNK files	
- Executables and DLLs	
- Windows system logfiles	
- NTFS $MFT	
- NTFS $USNJournal	
- Recycle bin	
- Windows Error Reporting (WER) artefacts	
- RDP Cache	
- Windows EDB (Search)	
- Web-Based Enterprise Management (WBEM)	
- Browser cache files	

Other functional tests

1. Scalability - the ability to run multiple concurrent search/collections. This may be estimated by having the forensic tool search through the entire file system while monitoring the network load at the central collection machine.
2. Minimal impact on end user – log into one of the endpoint machines during processing and determine the resources available to the user.
3. Minimal impact on network bandwidth – observe the network bandwidth being used during processing/collection either directly or by observing the network resource load on the machine being used to collect the results of the testing process.

NOTES:

9.4 APPENDIX 4 – THE NON-FUNCTIONAL CRITERIA (FINAL VERSION)

Forensic Tool (and version) ______________________________

Part 1 - Evaluating Non-functional Considerations

The Non-functional items should be considered but there is no score associated with these items as their relevance and importance will differ from one organisation to another.

1. Speed – the time to complete the activity (including deploying agents and configuration).
2. Dependencies - consideration should be given in relation to the agent's dependencies, i.e., any other software that must be installed for the agent to perform.
3. Ease of use - in terms of General Characteristics, the time it takes for a practitioner to learn how to use and deploy the remote agents.
4. Customisation – the ability to change the configuration of the tool, e.g., such that unnecessary functions are not performed.
5. Auditability – the ability to audit the agent's processing.
6. Options for exporting the results of processing.
7. Cost – the cost of licencing the tool and any other pre-requisites.
8. Ability to process machines not on the enterprise network.
9. Invisible to end user
10. Can be run in the cloud.
11. Data sovereignty – e.g., if data is required to be stored in the cloud, where is the data physically located with respect to the applicable jurisdiction.
12. File size – the installed file size of the remote agent (smaller is better)
13. Ability to recover from communication dropouts.
14. Usefulness of error messages – this may require examining the User Guide for the tool.

9.5 APPENDIX 5 - THE FUNCTIONAL CRITERIA (FINAL VERSION)

Part 2 - Evaluating Functional Considerations

The tool functionality can be assessed using the Expected Results checklist once the ‘seed’ data has been copied to the endpoints. The results can be checked for each file/email type. NOTE: for the foreign language and normalised data searches you should cut and paste the search terms from an electronic copy of the Expected Results Checklist.

Expected Results Checklist

Name of Forensic Tool: ____________

Functionality: **Search and process common files and streams**	**Expected Results**	**Actual Results**
PLAIN TEXT SEARCHING		
Metadata search term (author): FREDA		
Location – Document Folder	3 (TOTAL)	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Dinosaur Extinction (97-2003 DOC) Author metadata.doc	1	
Document body search term: cassiopeia		
Location – Document Folder	11 (TOTAL)	
Dinosaur Extinction (2003 XML).xml	1	
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Dinosaur Extinction (2022 DOCX).docx	1	
Dinosaur Extinction (2022 XML).xml	1	
Dinosaur Extinction (97-2003 DOC) Author metadata.doc	1	
Dinosaur Extinction (97-2003 DOC).doc	1	
Dinosaur Extinction (ODT).odt	1	
Dinosaur Extinction (PDF).pdf	1	
Dinosaur Extinction (PLAIN).txt	1	
Dinosaur Extinction (RTF).rtf	1	
Dinosaur Extinction (XPS).xps	1	
Location – Presentation Folder	9 (TOTAL)	
Stars (ODP).odp	1	
Stars (POT).pot	1	
Stars (POTX).potx	1	
Stars (PPS).pps	1	
Stars (PPSX).ppsx	1	
Stars (PPT).ppt	1	

Stars (PPTM).pptm	1	
Stars (PPTX).pptx	1	
Stars (XML Presentation).xml	1	
Location – Spreadsheet Folder	12 (TOTAL)	
Stars(97-2003 XLS).xls	1	
Stars(CSV).csv	1	
Stars(CSV-UTF8).csv	1	
Stars(DIF).dif	1	
Stars(Excel 5 XLS).xls	1	
Stars(ODS).ods	1	
Stars(Tab Delimited).txt	1	
Stars(XLSB).xlsb	1	
Stars(XLSM).xlsm	1	
Stars(XLSX).xlsx	1	
Stars(XLTX).xltx	1	
Stars(XML 2003).xml	1	
Location – Publication Folder	3 (TOTAL)	
Publication1(PUB).pub	1	
Publication1(PUB2000).pub	1	
Publication1(PUB98).pub	1	
Location – Database Folder	4 (TOTAL)	
Database1(ACCDB).accdb	1	
Database1(Access 2000 mdb).mdb	1	
Database1(Access 2002-2003 mdb).mdb	1	
places.sqlite	1	
Location – Container Folder	12 (TOTAL)	
archive-7z.7z	1	
archive-cab.CAB	1	
archive-cpio.cpio	1	
archive-dmg.dmg	1	
archive-iso.iso	1	
archive-LZH.lzh	1	
archive-rar.rar	1	
archive-tar.tar	1	
archive-tar_bzip2.tar.bz2	1	
archive-tar_gzip.tar.gz	1	
archive-wim.wim	1	
archive-zip.zip	1	
Location – ADS Folder (note: Alternate data stream)	1 (TOTAL)	
empty4.txt:ads1.txt	1	

Functionality: **Foreign Language Support**	**Expected Results**	**Actual Results**
Foreign Language Search Term1: 中国　東京		
Unicode Chinese/Japanese ideograms	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX).docx	1	
Dinosaur Extinction (2022 PDF)	1	
Foreign Language Search Term2: 서울		
Unicode CJK Korean Hangul	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
Foreign Language Search Term3: スバル　みつびし		
Unicode CJK Japanese phonetic Kana	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (97-2003 DOC) Author metadata.doc	1	
Foreign Language Search Term4: Сибирь		
Unicode Cyrillic (Russian)	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (97-2003 DOC).doc	1	
Foreign Language Search Term5: الكسكس		
Unicode RTL (Arabic)	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	

Functionality: Other Search Features	**Expected Results**	**Actual Results**
Normalized Form D search: Mäuse	4 (TOTAL)	
Location – Document Folder		
NFD Strings - FW_ Schedule Crawler_ HourAhead Failure.eml	1	
albert_meyers_000_1_1.pst	1	
NFD Strings - FW Schedule Crawler HourAhead Failure.msg	1	
Dinosaur Extinction (ODT).odt	1	
Normalized Form C search: Mäuse	4 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
NFC Strings - FW Schedule Crawler HourAhead Failure.msg	1	
albert_meyers_000_1_1.pst	1	
NFC Strings - FW_ Schedule Crawler_ HourAhead Failure.eml	1	
WILDCARD SEARCH		
Term: *glycerin	2 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
WHOLE WORD SEARCH		
Term: glycerin	NONE	
BEGINNING OF WORD SEARCH		
Term: Nitro	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
END OF WORD SEARCH		
Term: glycerin	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	
REGEX SEARCH		
Expression: \d{3}-\d{2}-\d{4}	3 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX).docx	1	
Dinosaur Extinction (2022 PDF).pdf	1	
Dinosaur Extinction (ODT).odt	1	
HEX SEQUENCE SEARCH		
Sequence: 6F6C6576616E	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (RTF).rtf	1	
CASE SENSITIVE SEARCH		
Term: Nitroglycerin	1 (TOTAL)	
Location – Document Folder		
Dinosaur Extinction (2022 DOCX) Author metadata.docx	1	

Functionality: **Search embedded documents**	Expected Result	Actual Result
Search term: Embedded Level1	1 (TOTAL)	
Location: \EMBEDDED\Level 1\ Embedded Level 1.docx	1	
Search term: Embedded Level2	1 (TOTAL)	
Location: \EMBEDDED\Level 2\ Embedded Level 2.docx	1	
Search term: Embedded Level3	1 (TOTAL)	
Location: \EMBEDDED\Level 3\ Embedded Level 3.docx	1	

Functionality: **Search email**	Expected Result	Actual Result
Search term: Buffalo Gap		
Location: PST	5 (TOTAL)	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Deleted Items Subject: RE: Tuesday Morning	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Deleted Items Subject: RE: Buffalo Gap	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Breakfast on Tuesday	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Tuesday Morning	1	
\EMAIL\PST\albert_meyers_000_1_1.pst\Top of Personal Folders\Sent Items Subject: Buffalo Gap	1	
Location: EML	5 (TOTAL)	
\EMAIL\PST\EML Output\Deleted Items\RE_ Buffalo Gap.eml Subject: RE: Buffalo Gap	1	
\EMAIL\PST\EML Output\Deleted Items\RE_ Tuesday Morning.eml Subject: RE: Buffalo Gap	1	
\EMAIL\PST\EML Output\Sent Items\Breakfast on Tuesday.eml Subject: Breakfast on Tuesday	1	
\EMAIL\PST\EML Output\Sent Items\Buffalo Gap.eml Subject: Buffalo Gap	1	
\EMAIL\PST\EML Output\Sent Items\Tuesday Morning.eml Subject: Tuesday Morning	1	
Location: MSG	1 (TOTAL)	

\EMAIL\PST\MSG Output\NFC Strings - FW Schedule Crawler HourAhead Failure.msg Subject: NFC Strings - FW: Schedule Crawler: HourAhead Failure	1	
Location: MBOX	1 (TOTAL)	
\EMAIL\PST\MBOX\cindyloh3333@gmail.com.mbox Subject: Accounts to close	1	

Functionality: **Identify encrypted files**	Expected Results	Actual Results
Location – Encrypted Folder	**2 (TOTAL)**	
Dinosaur Extinction1 (PDF).pdf	1	
Dinosaur Extinction (2022 DOCX).docx	1	

The following functional test is for you to determine if the tool can capture the files identified in the list. For some of the items it will be necessary for you to cause the files to be created on the endpoint machine(s), e.g., jumplists, browser cache files.

Functionality: **Capture capabilities**	**Yes/No**
· System memory	
· Swap file	
· Pagefile	
· Hibernation file	
· Registry hives (including amcache)	
· EVT & EVTX logs	
· SRUDB	
· Jumplist files	
· Prefetch files	
· Windows LNK files	
· Executables and DLLs	
· Windows system logfiles	
· NTFS $MFT	
· NTFS $USNJournal	
· Recycle bin	
· Windows Error Reporting (WER) artefacts	
· RDP Cache	
· Windows EDB (Search)	
· Web-Based Enterprise Management (WBEM)	
· Browser cache files	

Additional non-functional tests

1. Scalability - the ability to run multiple concurrent search/collections. This may be estimated by having the forensic tool search through the entire file system while monitoring the network load at the central collection machine.
2. Minimal impact on end user – log into one of the endpoint machines during processing and determine the resources available to the user.

3. Minimal impact on network bandwidth – observe the network bandwidth being used during processing/collection either directly or by observing the network resource load on the machine being used to collect the results of the testing process.

NOTES:

9.6 Appendix 6- Using the Framework (final version)

Introduction

This framework is intended to assist forensic practitioners evaluate digital forensic tools that employ remote agents to collect data from endpoints. The Framework does not provide a score as different environments will necessarily place different emphasis on a tool's capabilities and functions. Instead, it provides two lists of features, functional and non-functional, to be considered when evaluating a tool as well as the means to undertake a few tests for some of the key features, such as the ability to locate certain character strings.

Having reviewed the non-Functional Considerations and the outcome of the tests for the Functional Considerations for each of the forensic tools you are evaluating you will be able to determine if any are likely to meet your requirements as well as form a basis for future tool comparison.

Given that the time and the resources involved in setting up a consistent test environment makes it difficult to undertake a thorough evaluation in the workplace, the Framework incorporates the use of the Microsoft Windows 11[33] and Office 365 Deployment Lab Kit which automatically creates a small domain consisting of different server types and workstations which can be run on a mid-range laptop with 16GB of RAM. The Lab Kit does not have to be used if a suitable test environment is already available. Once installed, the virtual domain can be used 'as is' or can be further configured to suit your purposes.

To carry out functional tests a collection of seed data has been created that contains a range of different file types and character strings while being relatively small such that it is a trivial task to copy it to the endpoints, whether these are part of the Lab Kit or another test environment.

The forensic tool evaluation itself consists of two parts. The first part is where you consider the non-functional capabilities of the tool you are evaluating. There is no structured format for recording the results for non-functional capabilities of a tool, the intention is to provide a structure for assessment based on your priorities.

The second part of the tool evaluation consists of several functional tests using the provided seed data. This will confirm the tool's remote agent's ability to capture certain forensic artefacts from remote systems and locate data within different filetypes using a range of search methods.

For the functional tests the 286 MB 'seed' data can be downloaded from here:

https://drive.google.com/file/d/1zVGgBanqi5ZPvDmGhS94RfFuaUlXcVUs/view?usp=share_link

[33] The Windows 10-based kit is also available if more appropriate in your environment.

To preserve the Alternate Data Streams for one of the functional tests the download includes a 'WIM' file compressed into a Zip file. Once downloaded and extracted using a suitable compression tool (such as 7Zip), the WIM file seed data can itself be extracted to any folder location on the endpoints on which you will deploy the remote forensic agents for evaluation.

There are several artefacts that may be required to be installed or created in addition to the seed data that are specific to your requirements, such as browser artefacts.

In summary, the evaluation tasks consist of the following stages:

- Install the evaluation environment (if required).
- Copy the seed data to the required endpoints.
- Install and configure the tool(s) to be evaluated.
- Work through the criteria for non-functional considerations.
- Work through the criteria for functional considerations.
- Evaluate the tools under review by referring to the results obtained by using the two sets of criteria.

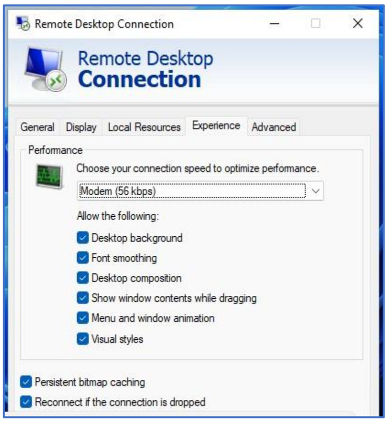
Setting up the evaluation environment

1. Download the Windows 10/11 and Office 365 Deployment Lab Kit and lab guides from here (note the download is 30GB): https://info.microsoft.com/ww-landing-windows-11-office-365-lab-kit.html?culture=enus&country=us

2. Ensure the lab machine you will use for the evaluation meets the following system requirements:

   - Hyper-V role installed.
   - Administrative rights on the device
   - 150 GB of free disk space (300 GB recommended)
   - High-throughput disk subsystem
   - 16 GB of available memory (32 GB recommended)
   - High-end processor for faster processing

3. Read the lab guides for installation and install the Deployment Lab on your evaluation machine.
4. Extract the seed data to the required number of endpoints.
5. Make any changes that are needed to the servers and/or workstations to better reflect your own environment (if appropriate).
6. For testing the RDP cache capture functionality create an RDP connection to one of the other Lab virtual machines but set the connection speed to Modem (56kps) as shown below:



Performance

Choose your connection speed to optimize performance.

Modem (56 kbps)

Allow the following:

- ☑ Desktop background
- ☑ Font smoothing
- ☑ Desktop composition
- ☑ Show window contents while dragging
- ☑ Menu and window animation
- ☑ Visual styles

☑ Persistent bitmap caching

☑ Reconnect if the connection is dropped

7. Use the inclusion/exclusion capabilities of the tool being tested to limit its data processing to the seed data folder.

8. If browser cache data other than that associated with Microsoft Internet Explorer (Edge) are required, this can be created through installation of the required browser on the endpoint machines followed by some internet browsing activity (if possible). Alternatively, data can be seeded from another machine.

9.7 APPENDIX 7 – EXPERT PANEL - INFORMATION FOR REVIEW AND FEEDBACK

The framework that you have been asked to review consists of the following stages:

- Install the evaluation environment.
- Copy the seed data to the required endpoints.
- Install and configure the tool(s) to be evaluated.
- Work through the criteria for non-functional considerations.
- Work through the criteria for functional considerations.
- The results obtained by working through the two sets of criteria are used by the practitioners to evaluate a particular tool.

As a reviewer you are asked to review the framework for ease of use, content and whether it meets its aim of providing a useful way for practitioners to evaluate remote forensic tools. Your feedback involves completing the six items in the section 'Tasks for Reviewers.'

Tasks for reviewers

1. Please review the documentation and determine if you believe that it is clear and simple to follow. If this is not the case, please identify the areas that need improvement.

Task	Outcome Yes/No	Comments
Consider if the framework is clear and simple to follow		

2. Consider the non-functional considerations and determine if you believe that they are reasonable and complete. If this is not the case, please identify the areas that need improvement.

Task	Outcome Yes/No	Comments
Consider the non-functional requirements and determine if you believe that they are reasonable and complete.		

3. Consider the functional considerations and determine if you believe that they are reasonable and complete. If this is not the case, please identify the areas that need improvement.

Task	Outcome Yes/No	Comments
Consider the functional requirements and determine if you believe that they are reasonable and complete.		

4. Consider the use of the Windows 11 and Office 365 Deployment Lab and determine if this is a useful environment for evaluating the remote forensic tools. If this is not the case, please identify the areas that need improvement.

Task	Outcome Yes/No	Comments
Consider the use of the Windows 11 and Office 365 Deployment Lab and determine if this is a useful environment for evaluating the remote forensic tools.		

5. Consider if the framework meets its aim of providing a means by which practitioners can evaluate remote forensic tools. If this is not the case, please identify the areas that need improvement.

Task	Outcome Yes/No	Comments
Consider if the framework meets its aim of providing a means by which practitioners can evaluate remote forensic tools for use across networked systems.		

6. Consider if there are any other comments you wish to provide as feedback.

Task	Outcome Yes/No	Comments
Consider if there are any other comments you wish to provide as feedback.		

Bibliography

Abt, S., & Baier, H. (2014, 11-11 Sept. 2014). *Are We Missing Labels? A Study of the Availability of Ground-Truth in Network Security Research.* Paper presented at the 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS).

Acker, G. v. (2007). FCCU Live Forensic CDROM. Retrieved from https://www.ia.nato.int/niapc/Product/FCCU-Live-Forensic-CDROM-version-11._214

Adams, R. (2012). The Emergence of Cloud Storage Highlights the Need for a New Digital Forensic Process Model. In *Cybercrime and Cloud Forensics: Applications for Investigative Processes*: IGI Global.

Adams, R. (2013). *The Advanced Data Acquisition Model (ADAM): A Process Model for Digital Forensic Practice.* (Doctor of Philosophy). Murdoch University, Retrieved from http://researchrepository.murdoch.edu.au/14422/2/02Whole.pdf

Adams, R., Hobbs, V., & Mann, G. (2013). The Advanced Data Acquisition Model (ADAM): A Process Model for Digital Forensic Practice. *Journal of Digital Forensics, Security and Law, 8*(4), 23.

Adams, R., Mann, G., & Hobbs, V. (2017). *ISEEK, a tool for high speed, concurrent, distributed forensic data acquisition.* Paper presented at the AUSTRALIAN DIGITAL FORENSICS CONFERENCE, Perth, Western Australia.

ADF Solutions. (2020). Triage-Investigator v5.1.1 User Guide. Retrieved from https://www.adfsolutions.com/hubfs/User%20Guides/Triage-Investigator%20v5.1.1%20User%20Guide.pdf

Applegate, L. M. (1999). Rigor and Relevance in MIS Research-Introduction. *MIS Quarterly, 23*(1), 1-2. Retrieved from http://www.jstor.org/stable/249402

Archer, L. B. (1984). *Systematic Method for Designers*. London: John Wiley.

Argy, P. (2006). Electronic Evidence, Document Retention and Privacy

Retrieved from http://www.mallesons.com/publications/2006/Mar/8367966w.htm

Armstrong, C., & Armstrong, H. (2010). *Modeling Forensic Evidence Systems Using Design Science.* Paper presented at the IFIP WG 8.2/8.6 International Working Conference, Perth, Western Australia.

Arshad, H., Jantan, A. B., & Abiodun, O. I. (2018). Digital forensics: review of issues in scientific validation of digital evidence. *Journal of Information Processing Systems, 14*(2), 346-376.

Association of Chief Police Officers. (2012). Good Practice Guide for Computer Based Evidence. Retrieved from http://www.acpo.police.uk/asp/policies/Data/gpg_computer_based_evidence_v3.pdf

Attoe, R. (2016). Chapter 6 - Digital forensics in an eDiscovery world. In J. Sammons (Ed.), *Digital Forensics* (pp. 85-98). Boston: Syngress.

Avison, D., Baskerville, R., & Myers, M. (2001). Controlling action research projects. *Information Technology & People, 14*(1), 28-45. doi:10.1108/09593840110384762

Baggili, I., & Breitinger, F. (2015). *Data Sources for Advancing Cyber Forensics: What the Social World Has to Offer*.

Baggili, I., Mislan, R., & Rogers, M. (2007). Mobile Phone Forensics Tool Testing: A Database Driven Approach. *International Journal of Digital Evidence, 6*(2).

Balci, O. (2001). A methodology for certification of modeling and simulation applications. *ACM Trans. Model. Comput. Simul., 11*(4), 352–377. doi:10.1145/508366.508369

Barrett, D., & Kipper, G. (2010). Virtualisation and Forensics. In D. Barrett & G. Kipper (Eds.), *Virtualization and Forensics* (pp. xv-xvi). Boston: Syngress.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology, 11*(3), 235-246. doi:10.1080/026839696345289

Becker, P., Tebes, G., Peppino, D., & Olsina, L. (2019). Applying an Improving Strategy that embeds Functional and Non-Functional Requirements Concepts. *Journal of Computer Science and Technology, 19*(2), e15. doi:10.24215/16666038.19.e15

Beebe, N. (2009). Digital forensic research: The good, the bad and the unaddressed. In (Vol. 306, pp. 17-36).

Bruce, C. (2007). Questions Arising about Emergence, Data Collection, and Its Interaction with Analysis in a Grounded Theory Study *International Journal of Qualitative Methods, 6*(1), 51-68.

Casey, E. (2013). Triage in digital forensics. *Digital Investigation, 10*(2), 85-86. doi:https://doi.org/10.1016/j.diin.2013.08.001

Casey, E. (2016). Editorial - A sea change in digital forensics and incident response. *Digital Investigation, 17*, A1-A2. doi:10.1016/j.diin.2016.05.002

Casey, E. (2017). The broadening horizons of digital investigation. *Digital Investigation, 21*, 1-2. doi:https://doi.org/10.1016/j.diin.2017.05.002

Casey, E., & Stanley, A. (2004). Tool review – remote forensic preservation and examination tools. *Digital Investigation, 1*(4), 284-297. doi:10.1016/j.diin.2004.11.003

Casey, E., & Stellatos, G. J. (2008). The impact of full disk encryption on digital forensics. *SIGOPS Oper. Syst. Rev., 42*(3), 93-98. doi:10.1145/1368506.1368519

Cheng, E. (2007). Independent Judicial Research in the Daubert Age. *Duke Law Journal, 56*, 1263 - 1318.

Chernyshev, M., Zeadally, S., Baig, Z., & Woodward, A. (2017). Mobile Forensics: Advances, Challenges, and Research Opportunities. *IEEE Security & Privacy, 15*(6), 42-51. doi:10.1109/MSP.2017.4251107

Cohen, M. (2011). Putting the Science in Digital Forensics. *Journal of Digital Forensics, Security and Law, 6*(1), 7-14.

Cohen, M., Bilby, C., & Caronni, G. (2011). Distributed forensics and incident response in the enterprise. *Digital Investigation, 8*, S101-S110. doi:https://doi.org/10.1016/j.diin.2011.05.012

Cole, R., Purao, S., Rossi, M., & Sein, M. (2005). *Being Proactive: Where Action Research Meets Design Research*.

Confais, B., Arslan, S., & Parrein, B. (2022). *SToN: A New Fundamental Trade-off for Distributed Data Storage Systems*.

Cornelissen, A. M. G., Berg, J., Koops, W. J., & Kaymak, U. (2002). Eliciting Expert Knowledge for Fuzzy Evaluation of Agricultural Production Systems. *Agriculture, Ecosystems & Environment, 95*, 1-18. doi:10.1016/S0167-8809(02)00174-3

Creswell, J. (2005). *Educational research: Planning, conducting, and evaluating qualitative research* Upper Saddle River, NJ: Merrill Prentice Hall Pearson Education.

Cross, N. (2001). Designerly Ways of Knowing: Design Discipline versus Design Science. *Design Issues, 17*(3), 49-55. Retrieved from http://www.jstor.org/stable/1511801

Dell Inc. (2009). Dell Digital forensics - solution blueprint. Retrieved from https://i.dell.com/sites/csdocuments/Corporate_press-Releases_Documents/en/uk/digital-forensics-uk.pdf

Dell, P. (2018). On the dual-stacking transition to IPv6: A forlorn hope? *Telecommunications Policy, 42*(7), 575-581. doi:10.1016/j.telpol.2018.04.005

Du, X., Hargreaves, C., Sheppard, J., & Scanlon, M. (2021). TraceGen: User activity emulation for digital forensic test image generation. *Forensic Science International: Digital Investigation, 38*, 301133. doi:https://doi.org/10.1016/j.fsidi.2021.301133

Duboc, L., Rosenblum, D., & Wicks, T. (2007, 2007). *A framework for characterization and analysis of software system scalability.* Paper presented at the Foundations of Software Engineering.

Dykstra, J., & Sherman, A. T. (2012). Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digital Investigation, 9*(S), S90-S98. doi:10.1016/j.diin.2012.05.001

Easterbrook, S., Singer, J., Storey, M., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. In Shull F., Singer J., & S. D.I.K. (Eds.), *Guide to Advanced Empirical Software Engineering*. London: Springer.

Edmond, G. (2010). Impartiality, efficiency or reliability? A critical response to expert evidence law and procedure in Australia. *Australian Journal of Forensic Sciences*(42), 83-99.

Edwards, B. (2019). AD ENTERPRISE - NETWORK INVESTIGATION AND POST-BREACH ANALYSIS. In: AccessData Group Inc.

Eekels, J., & Roozenburg, N. F. M. (1991). A methodological comparison of the structures of scientific research and engineering design: their similarities and differences. *Design Studies, 12*(4), 197-203. doi:10.1016/0142-694X(91)90031-Q

Elisan, C. (2015). *Advanced Malware Analysis*: McGraw-Hill Companies.

Flandrin, F., Buchanan, W. J., Macfarlane, R., Ramsay, B., & Smales, A. (2014). Evaluating Digital Forensic Tools (DFTs).

Franke, K., Årnes, A., Flaglien, A., Sunde, I. M., Dilijonaite, A., Hamm, J., . . . Axelsson, S. (2017). Challenges in Digital Forensics. In *Digital Forensics* (pp. 313-317): John Wiley & Sons, Ltd.

Gao, Y., Richard, G. G., & Roussev, V. (2004). *Bluepipe: A Scalable Architecture for On-the-Spot Digital Forensics* (Vol. 3).

Garfinkel, S., Farrell, P., Roussev, V., & Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. *Digital Investigation, 6*(S), S2-S11. doi:10.1016/j.diin.2009.06.016

Göbel, T., Maltan, S., Türr, J., Baier, H., & Mann, F. (2022). ForTrace - A holistic forensic data set synthesis framework. *Digit. Investig., 40*, 301344.

Göbel, T., Schäfer, T., Hachenberger, J., Türr, J., & Baier, H. (2020, 2020). *A Novel Approach for Generating Synthetic Datasets for Digital Forensics*, Cham.

Goodison, S. E., Davis, R. C., & Jackson, B. A. (2015). *Digital Evidence and the U.S. Criminal Justice System: Identifying Technology and Other Needs to More Effectively Acquire and Utilize Digital Evidence*: RAND Corporation.

Grajeda, C., Breitinger, F., & Baggili, I. (2017). Availability of datasets for digital forensics – And what is missing. *Digital Investigation, 22*, S94-S105. doi:https://doi.org/10.1016/j.diin.2017.06.004

Guo, Y., Slay, J., & Beckett, J. (2009). Validation and verification of computer forensic software tools—Searching Function. *Digital Investigation, 6*(S), S12-S22. doi:10.1016/j.diin.2009.06.015

Haamann, T., & Basten, D. (2019). The role of information technology in bridging the knowing-doing gap: an exploratory case study on knowledge application. *Journal of Knowledge Management, 23*(4), 705-741.

Han, L., Harries, J., & Brown, P. (2013). Building a Virtual Constructivist Learning Environment for Learning Computing Security and Forensics. *Innovations in teaching and learning in information and computer sciences, 12*(1), 49-61. doi:10.11120/ital.2013.00006

Harichandran, V. S., Breitinger, F., Baggili, I., & Marrington, A. (2016). A cyber forensics needs analysis survey: Revisiting the domain's needs a decade later. *Computers & Security, 57*, 1-13. doi:10.1016/j.cose.2015.10.007

Hevner, A., & Chatterjee, S. (2010). *Design Research in Information Systems*. New York: Springer.

Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management information systems quarterly, 28*(1), 75-106.

Hitchcock, B., Le-Khac, N.-A., & Scanlon, M. (2016). Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists. *Digital Investigation, 16*, S75-S85. doi:https://doi.org/10.1016/j.diin.2016.01.010

Hong, I., Yu, H., Lee, S., & Lee, K. (2013). A new triage model conforming to the needs of selective search and seizure of electronic evidence. *Digital Investigation, 10*(2), 175-192. doi:https://doi.org/10.1016/j.diin.2013.01.003

Horsman, G. (2018). Framework for Reliable Experimental Design (FRED): A research framework to ensure the dependable interpretation of digital data for digital forensics. *Computers & Security, 73*, 294-306. doi:10.1016/j.cose.2017.11.009

Horsman, G. (2019). Tool testing and reliability issues in the field of digital forensics. *Digital Investigation, 28*, 163-175. doi:https://doi.org/10.1016/j.diin.2019.01.009

Horsman, G. (2020). ACPO principles for digital evidence: Time for an update? *Forensic Science International: Reports*, 100076. doi:https://doi.org/10.1016/j.fsir.2020.100076

Horsman, G., Laing, C., & Vickers, P. (2014). A case-based reasoning method for locating evidence during digital forensic device triage. *Decision Support Systems, 61*(C), 69-78. doi:10.1016/j.dss.2014.01.007

Horsman, G., & Lyle, J. (2021). Dataset construction challenges for digital forensics. *Forensic Science International: Digital Investigation, 38*, 301264. doi:https://doi.org/10.1016/j.fsidi.2021.301264

Hosseinian - Far, A., Daneshkhah, A., Hill, R., Montasari, R., Parkinson, S., & Peltola, P. (2020). Digital Forensics: Challenges and Opportunities for Future Studies. *International Journal of Organizational and Collective Intelligence (IJOCI), 10*(2), 37-53. doi:10.4018/IJOCI.2020040103

Hughes, N., & Karabiyik, U. (2020). Towards reliable digital forensics investigations through measurement science. *WIREs. Forensic science, 2*(4), e1367-n/a. doi:10.1002/wfs2.1367

Hutton, G., & Johnston, D. (2000). *Evidence and Procedure* (2nd ed.): Blackstone Press Limited.

Irons, A., & Lallie, H. S. (2014). Digital Forensics to Intelligent Forensics. *Future Internet, 6*(3), 584-596. doi:10.3390/fi6030584

Jahanbin, A., Ghafarian, A., Hosseini Seno, S. A., & Nikookar, S. (2013). A computer forensics approach based on autonomous intelligent multi-agent system. *International Journal of Database Theory and Application, 6*.

Jogalekar, P., & Woodside, M. (2000). Evaluating the scalability of distributed systems. *IEEE transactions on parallel and distributed systems, 11*(6), 589-603. doi:10.1109/71.862209

Jones, K. J., & Vidalis, S. (2019). Rethinking Digital Forensics. *Annals of Emerging Technologies in Computing, 3*, 41-53. doi:10.33166/AETiC.2019.02.005

Jusas, V., Birvinskas, D., & Gahramanov, E. (2017). Methods and Tools of Digital Triage in Forensic Context: Survey and Future Directions. *Symmetry, 9*(4), 49. Retrieved from http://www.mdpi.com/2073-8994/9/4/49

Kebande, V. R., & Venter, H. S. (2018). On digital forensic readiness in the cloud using a distributed agent-based solution: issues and challenges. *Australian Journal of Forensic Sciences, 50*(2), 209-238. doi:10.1080/00450618.2016.1194473

Kendrick, P., Criado, N., Hussain, A., & Randles, M. (2018). A self-organising multi-agent system for decentralised forensic investigations. *Expert Systems with Applications, 102*, 12-26. doi:https://doi.org/10.1016/j.eswa.2018.02.023

Kennedy, I. (2017). *A Framework for the Systematic Evaluation of Malware Forensic Tools.* ProQuest Dissertations Publishing,

Kessler, G. C., & Carlton, G., H. (2017). Exploring Myths in Digital Forensics: Separating Science From Ritual. *International Journal of Interdisciplinary Telecommunications and Networking (IJITN), 9*(4), 1-9. doi:10.4018/IJITN.2017100101

Koopmans, M. (2010). *The art of triage with (g)PXE*. University College Dublin. Computer Science and Mathematics.

Koopmans, M., & James, J. I. (2013). Automated network triage. *Digital Investigation, 10*(2), 129-137. doi:https://doi.org/10.1016/j.diin.2013.03.002

Lankton, N. K., & Luft, J. (2014). Making and Evaluating Participant Choice in Experimental Research on Information Technology: A Framework and Assessment. *Communications of the Association for Information Systems, 35*(1), 11.

Lee, A. S. (1989). A Scientific Methodology for MIS Case Studies. *MIS Quarterly, 13*(1), 33-50. doi:10.2307/248698

Lillis, D., Becker, B. A., O'Sullivan, T., & Scanlon, M. (2016). CURRENT CHALLENGES AND FUTURE RESEARCH AREAS FOR DIGITAL FORENSIC INVESTIGATION. *Proceedings of the Conference on Digital Forensics, Security and Law*, 9.

Malin, C. H., Casey, E., & Aquilina, J. M. (2008). *Malware forensics: investigating and analyzing malicious code*: Syngress.

Marcella, A. J., & Menendez, D. (2008). *Cyber Forensics: A Field Manual for Collecting, Examining and Preserving Evidence of Computer Crimes* (Second ed.): Auerbach Publications.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251-266. doi:https://doi.org/10.1016/0167-9236(94)00041-2

Marshall, A. M. (2022). The unwanted effects of imprecise language in forensic science standards. *Forensic Science International: Digital Investigation, 40*, 301349. doi:https://doi.org/10.1016/j.fsidi.2022.301349

Mason, S. (2007). *Electronic Evidence: Disclosure, Discovery & Admissibility*: LexisNexis Butterworths.

Mason, S. (2014). Electronic evidence: A proposal to reform the presumption of reliability and hearsay. *Computer Law & Security Review: The International Journal of Technology Law and Practice, 30*(1), 80-84. doi:10.1016/j.clsr.2013.12.005

Maximov, V., & Karasik, A. (2014). The application of virtualisation technologies to organisation of individualised training of students on computerised audiences. *International Journal of Continuing Engineering Education and Life Long Learning, 24*(3-4), 343-361. doi:10.1504/IJCEELL.2014.063104

McKay, J., & Marshall, P. (2005, 30Nov - 2 Dec). *A Review of Design Science in Information Systems.* Paper presented at the 16th Australian Conference on Information Systems, Sydney.

McKemmish, R. (1999). What is Forensic Computing? Retrieved from http://www.aic.gov.au/publications/tandi/ti118.pdf

Mercuri, R. (2005). Challenges in forensic computing. *Commun. ACM, 48*(12), 17-21. doi:10.1145/1101779.1101796

Moch, C., & Freiling, F. (2009). *The Forensic Image Generator Generator (Forensig2)*.

Mohamed, A. F. A. L., Marrington, A., Iqbal, F., & Baggili, I. (2014a). Testing the forensic soundness of forensic examination environments on bootable media. *Digital Investigation, 11*, S22-S29. doi:https://doi.org/10.1016/j.diin.2014.05.015

Mohamed, A. F. A. L., Marrington, A., Iqbal, F., & Baggili, I. (2014b). Testing the forensic soundness of forensic examination environments on bootable media. *Digital Investigation, 11*(2), S22-S29. doi:10.1016/j.diin.2014.05.015

Mohiddin, S., Yalavarthi, S., & Kondragunta, V. (2019). An Analytical Comparative Approach of Cloud Forensic Tools During Cyber Attacks in Cloud: SocProS 2017, Volume 2. In (pp. 509-517).

Mohite, M., & Ardhapurkar, S. (2015). Overcast: Developing Digital Forensic tool in cloud computing environment. In M. Mohite (Ed.), (pp. 1-4).

Moser, A., & Cohen, M. I. (2013). Hunting in the enterprise: Forensic triage and incident response. *Digital Investigation, 10*(2), 89-98. doi:https://doi.org/10.1016/j.diin.2013.03.003

Mrdovic, S., Huseinovic, A., & Zajko, E. (2009). *Combining static and live digital forensic analysis in virtual environment*.

Murff, K., Gardenier, H., & Gardenier, M. (2011). DIGITAL FORENSICS AND THE LAW. *Proceedings of the Conference on Digital Forensics, Security and Law*, 13-44.

National Institute of Standards and Technology. (2020). Computer forensics Tool Testing Programme (CFTT). Retrieved from https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt

Nichols, R. K. (Ed.) (2021). *Disruptive Technologies with Applications in Airline & Marine and Defense Industries*. Manhattan: New Prairie Press.

Nikkel, B. J. (2014). Fostering incident response and digital forensics research. *Digital Investigation, 11*(4), 249-251. doi:10.1016/j.diin.2014.09.004

Nunamaker, J. F., Chen, M., & Purdin, T. D. M. (1990). Systems development in information systems research. *Journal of Management Information Systems, 7*(3), 89-106.

O'Connor, O. (2004). Deploying forensic tools via PXE. *Digital Investigation, 1*(3), 173-176. doi:10.1016/j.diin.2004.07.005

Overill, R. E., Silomon, J. A. M., & Roscoe, K. A. (2013). Triage template pipelines in digital forensic investigations. *Digital Investigation, 10*(2), 168-174. doi:https://doi.org/10.1016/j.diin.2013.03.001

Park, J. (2018). TREDE and VMPOP: Cultivating multi-purpose datasets for digital forensics – A Windows registry corpus as an example. *Digital Investigation, 26*, 3-18. doi:https://doi.org/10.1016/j.diin.2018.04.025

Peffers, K. (2011, 8 July 2011). [Modification to the DSRM].

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). *The Design Science research process: a model for producing and presenting information systems research.* Paper presented at the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77. doi:10.2753/MIS0742-1222240302

Peisert, S., Bishop, M., & Marzullo, K. (2008). *Computer Forensics In Forensis.* Paper presented at the Third International Workshop on Systematic Approaches to Digital Forensic Engineering, Oakland, California, USA.

Penrose, P., Macfarlane, R., & Buchanan, W. J. (2013). Approaches to the classification of high entropy file fragments. *Digital Investigation, 10*(4), 372-384. doi:https://doi.org/10.1016/j.diin.2013.08.004

Pollitt, M. M. (2013). Triage: A practical solution or admission of failure. *Digital Investigation, 10*(2), 87-88. doi:https://doi.org/10.1016/j.diin.2013.01.002

Quick, D., & Choo, K.-K. R. (2014). Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation, 11*(4), 273-294. doi:https://doi.org/10.1016/j.diin.2014.09.002

Raghavan, S. (2013). Digital forensic research: current state of the art. *CSI Transactions on ICT, 1*(1), 91-114. doi:10.1007/s40012-012-0008-7

Reuzel, R. P. B. (2001). *Health technology assessment and interactive evaluation: different perspectives*: [Sl: sn].

Richard, G. G., & Roussev, V. (2006). Next-generation digital forensics. *Communications of the ACM, 49*(2), 76-80.

Robotti, K. (2009). R.I.P. - Recovery Is Possible. Retrieved from https://www.linux.com/training-tutorials/rip-recovery-possible/

Rogers, M., Goldman, J., Mislan, R., Wedge, T., & Debrota, S. (2006). Computer Forensics Field Triage Process Model. *Proceedings of the Conference on Digital Forensics, Security and Law*, 27-40.

Rossi, M., & Sein, M. K. (2003). *Design research workshop: A proactive research approach*. Paper presented at the Twenty-Sixth Information Systems Research Seminar, Haikko, Finland.

Roussev, V., Ahmed, I., Barreto, A., McCulley, S., & Shanmughan, V. (2016). Cloud forensics–Tool development studies & future outlook. *Digital Investigation, 18*, 79-95. doi:10.1016/j.diin.2016.05.001

Roussev, V., & Golden, R. G. I. (2004). *Breaking the performance wall: The case for distributed digital forensics.* Paper presented at the Proceedings of the 2004 digital forensics research workshop.

Roussev, V., & Richard, G. G. (2004, 04/12). *Breaking the performance wall: The case for distributed digital forensics.* Paper presented at the Digital Forensics Research Workshop, Baltimore, MD.

Ruan, K., & Global, I. G. I. (2013). *Cybercrime and cloud forensics : applications for investigation processes / Keyun Ruan, editor*. Hershey, Pa.: Hershey, Pa. : IGI Global (701 E. Chocolate Avenue, Hershey, Pennsylvania, 17033, USA).

Rygg, K., Brataas, G., Millstein, G., & Molle, T. (2013, 2013). *Scalability testing of MS lync services: towards optimal provisioning of virtualised hardware.* Paper presented at the International Conference on Performance Engineering.

Sahaym, A., Vithayathil, J., Sarker, S., Sarker, S., & Bjørn-Andersen, N. (2023). Value destruction in information technology ecosystems: A mixed-method investigation with interpretive case study and analytical modeling. *Information Systems Research, 34*(2), 508-531.

Sankardas, R., Yan, W., & Lavenia, K. N. (2019). Experience of Incorporating NIST Standards in a Digital Forensics Curricula. In (pp. 1-6).

SC Magazine. (2016). EnCase Endpoint Investigator. In (Vol. 27, pp. 40). New York: New York: Haymarket Media, Inc.

Scanlon, M. (2017). Enabling the Remote Acquisition of Digital Forensic Evidence through Secure Data Transmission and Verification.

Scanlon, M., Du, X., & Lillis, D. (2017). EviPlant: An efficient digital forensic challenge creation, manipulation and distribution solution. *Digit. Investig., 20 Supplement*, S29-S36.

Scanlon, M., & Kechadi, M. T. (2010). Online acquisition of digital forensic evidence. In (Vol. 31, pp. 122-131).

Scientific Working Group on Digital Evidence. (2014). SWGDE Best Practices for Computer Forensics. Retrieved from https://www.swgde.org/documents/Current%20Documents/SWGDE%20Best%20Practices%20for%20Computer%20Forensics

Scientific Working Group on Digital Evidence. (2018). *SWGDE Minimum Requirements for Testing Tools used in Digital and Multimedia Forensics*. Retrieved from https://www.swgde.org/documents/Current%20Documents/SWGDE%20Minimum%20Requirements%20for%20Testing%20Tools%20used%20in%20Digital%20and%20Multimedia%20Forensics

Scriven, M. (1998). The new science of evaluation. *Scandinavian Journal of Social Welfare, 7*(2), 79-86. doi:10.1111/j.1468-2397.1998.tb00206.x

Shaw, A., & Browne, A. (2013). A practical and robust approach to coping with large volumes of data submitted for digital forensic examination. *Digital Investigation, 10*(2), 116-128. doi:https://doi.org/10.1016/j.diin.2013.04.003

Shiaeles, S., Chryssanthou, A., & Katos, V. (2013). On-scene triage open source forensic tool chests: Are they effective? *Digital Investigation, 10*(2), 99-115. doi:https://doi.org/10.1016/j.diin.2013.04.002

Siau, K., & Rossi, M. (2011). Evaluation techniques for systems analysis and design modelling methods – a review and comparative analysis. *Information systems journal, 21*(3), 249-268. doi:https://doi.org/10.1111/j.1365-2575.2007.00255.x

Slay, J., & Beckett, J. (2007). *Digital Forensics: Validation and Verification in a Dynamic Work Environment.* Paper presented at the Proceedings of the 40th Hawaii International Conference on System Sciences - 2007.

Smith, R. G., Grabosky, P. N., & Gregor Urbas. (2004). *Cyber Criminals on Trial*: Cambridge University Press.

Sondhi, S., & Arora, R. (2014). *Applying Lessons from e-Discovery to Process Big Data using HPC*. Paper presented at the Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, Atlanta, GA, USA.

Sonnenberg, C., & Brocke, J. v. (2012). *Evaluation Patterns for Design Science Research Artefacts*. Paper presented at the Practical Aspects of Design Science, European Design Science Symposium, EDSS 2011, Leixlip, Ireland.

Stanfield, A. (2009). *Computer forensics, electronic discovery & electronic evidence*: Reed International Books.

Steel, C. (2006). *Windows Forensics: The Field Guide for Conducting Corporate Computer Investigations*: Wiley Publishing.

Stephen, G. (2012). DIGITAL SEARCHES AND THE FOURTH AMENDMENT: THE INTERPLAY BETWEEN THE PLAIN VIEW DOCTRINE AND SEARCH-PROTOCOL WARRANT RESTRICTIONS. *American Criminal Law Review, 49*, 301-2021.

Stephenson, P. (2003). A Comprehensive Approach to digital Incident Investigation. *Information Security Technical Report, 8*(2), 42-54.

Stone, M., Kosack, E., & Aravopoulou, E. (2020). Relevance of academic research in information technology and information management. *The Bottom Line, 33*(3), 273-295.

Supreme Court of the United States. (2003). *Daubert v Merrell Dow Pharmaceuticals Inc 509 US at 579, 113 S.Ct. 2786, 125 L.Ed.2d 469 (1993)*.

Taipale, J. (2019). Predefined criteria and interpretative flexibility in legal courts' evaluation of expertise. *Public Understanding of Science, 28*(8), 883-896. doi:10.1177/0963662519881338

Takeda, H., Veerkamp, P., Tomiyama, T., & Yoshikawa, H. (1990). Modeling design processes. *AI Mag., 11*(4), 37-48.

Urquhart, C., & Fernández, W. (2013). Using grounded theory method in information systems: The researcher as blank slate and other myths. *Journal of Information Technology, 28*, 224-236.

Venable, J. (2006). *The Role of Theory and Theorising in Design Science Research.* Paper presented at the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA.

Venable, J., Pries-Heje, J., & Baskerville, R. (2014). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems, 25*(1). doi:10.1057/ejis.2014.36

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a Framework for Evaluation in Design Science Research. *European Journal of Information Systems, 25*(1), 77-89. doi:10.1057/ejis.2014.36

Venable, J., Pries-Heje, J., & Baskerville, R. (2017). Choosing a Design Science Research Methodology. In.

Visti, H., Tohill, S., & Douglas, P. (2015). Automatic Creation of Computer Forensic Test Images. In (Vol. 8915, pp. 163-175). Cham: Springer International Publishing.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS. *Information Systems Research, 3*(1), 36-59. doi:10.1287/isre.3.1.36

Wilkinson, R. (2019). Digital Forensics: A Comparison of Data Reduction Techniques. In R. Hollington & T. Huynh (Eds.): ProQuest Dissertations Publishing.

Winter, R. (2008). Design science research in Europe. *European Journal of Information Systems, 17*(5), 470-475. doi:10.1057/ejis.2008.44

Woods, L. (2019). Digital Privacy and Article 12 of the Universal Declaration of Human Rights. *Political Quarterly, 90*(3), 422-429. doi:10.1111/1467-923X.12740

Yadav, G., Mangla, S. K., Luthra, S., & Rai, D. P. (2019). Developing a sustainable smart city framework for developing economies: An Indian context. *Sustainable Cities and Society, 47*, 101462. doi:https://doi.org/10.1016/j.scs.2019.101462

Yang, S. J., Choi, J. H., Kim, K. B., Bhatia, R., Saltaformaggio, B., & Xu, D. (2017). Live acquisition of main memory data from Android smartphones and smartwatches. *Digital Investigation, 23*, 50-62. doi:10.1016/j.diin.2017.09.003

Yannikos, Graner, L., Steinebach, M., & Winter, C. (2014). *Data Corpora for Digital Forensics Education and Research*, Berlin, Heidelberg.

Yannikos, & Winter. (2013, 2-6 Sept. 2013). *Model-Based Generation of Synthetic Disk Images for Digital Forensic Tool Testing.* Paper presented at the 2013 International Conference on Availability, Reliability and Security.