

2017

Proceedings of the 15th Australian Digital Forensics Conference, 5-6 December 2017, Edith Cowan University, Perth, Australia

Craig Valli

Security Research Institute, Edith Cowan University, c.valli@ecu.edu.au

Follow this and additional works at: <https://ro.ecu.edu.au/adf>



Part of the [Information Security Commons](#)

Recommended Citation

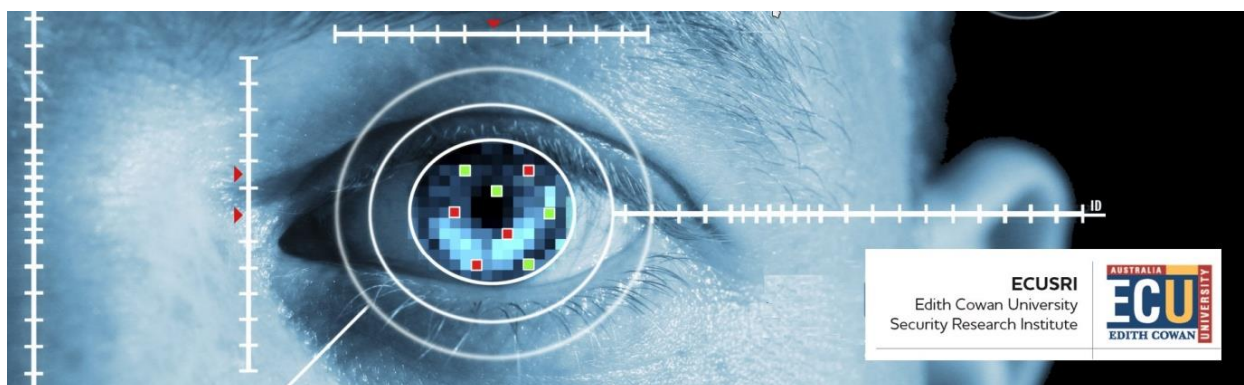
Valli, C. (2017). Proceedings of the 15th Australian Digital Forensics Conference, 5-6 December 2017, Edith Cowan University, Perth, Australia. DOI: <https://doi.org/10.4225/75/5a83a2ccee9f>

DOI: [10.4225/75/5a83a2ccee9f](https://doi.org/10.4225/75/5a83a2ccee9f)

Valli, C. (Ed.). The Proceedings of 15th Australian Digital Forensics Conference, 5-6 December 2017, Edith Cowan University, Perth, Australia.

This Conference Proceeding is posted at Research Online.

<https://ro.ecu.edu.au/adf/175>



Proceedings of the 15th Australian Digital Forensics Conference

5-6 December 2017

Edith Cowan University, Perth, Australia



Proceedings of the
15th Australian Digital Forensics Conference

Published By

Security Research Institute
Edith Cowan University

Edited By

Professor Craig Valli
Security Research Institute
Edith Cowan University

Copyright 2017, All Rights Reserved, Edith Cowan University

ISBN 978-0-6481270-9-3

CRICOS Institution Provider Code 00279B

Sponsors

ECUSRI
Edith Cowan University
Security Research Institute



Supporters



**Australian and New Zealand
FORENSIC SCIENCE SOCIETY**

Conference Foreword

This is the sixth year that the Australian Digital Forensics Conference has been held under the banner of the Security Research Institute, which is in part due to the success of the security conference program at ECU. As with previous years, the conference continues to see a quality papers with a number from local and international authors. 8 papers were submitted and following a double blind peer review process, 5 were accepted for final presentation and publication. Conferences such as these are simply not possible without willing volunteers who follow through with the commitment they have initially made, and I would like to take this opportunity to thank the conference committee for their tireless efforts in this regard. These efforts have included but not been limited to the reviewing and editing of the conference papers, and helping with the planning, organisation and execution of the conference. Particular thanks go to those international reviewers who took the time to review papers for the conference, irrespective of the fact that they are unable to attend this year.

To our sponsors and supporters a vote of thanks for both the financial and moral support provided to the conference. Finally, to the student volunteers and staff of the ECU Security Research Institute, your efforts as always are appreciated and invaluable.

Yours sincerely,

Conference Chair

Professor Craig Valli

Director, Security Research Institute

Congress Organising Committee

Congress Chair:

Professor Craig Valli

Committee Members:

Professor Gary Kessler – Embry Riddle University, Florida, USA

Professor Glenn Dardick – Embry Riddle University, Florida, USA

Professor Ali Babar – University of Adelaide, Australia

Dr Jason Smith – CERT Australia, Australia

Associate Professor Mike Johnstone – Edith Cowan University, Australia

Professor Joseph A. Cannataci – University of Malta, Malta

Professor Nathan Clarke – University of Plymouth, Plymouth UK

Professor Steven Furnell – University of Plymouth, Plymouth UK

Professor Bill Hutchinson – Edith Cowan University, Perth, Australia

Professor Andrew Jones – Khalifa University, Abu Dhabi, UAE

Professor Iain Sutherland – Glamorgan University, Wales, UK

Professor Matthew Warren – Deakin University, Melbourne, Australia

Congress Coordinator:

Ms Emma Burke

Table of Contents

A FRAMEWORK FOR FORENSIC RECONSTRUCTION OF SPONTANEOUS AD HOC NETWORKS.....	4
<i>Alastair Nisbet</i>	
A CENTRALISED PLATFORM FOR DIGITAL FORENSIC INVESTIGATIONS IN CLOUD-BASED ENVIRONMENTS.....	11
<i>Shaunak Mody, Alastair Nisbet</i>	
ISEEK, A TOOL FOR HIGH SPEED, CONCURRENT, DISTRIBUTED FORENSIC DATA ACQUISITION...	19
<i>Richard Adams, Graham Mann, Valerie Hobbs</i>	
ANALYSIS OF ATTEMPTED INTRUSIONS: INTELLIGENCE GATHERED FROM SSH HONEYPOTS.....	26
<i>Priya Rabadia, Craig Valli, Ahmed Ibrahim, Zubair Baig</i>	
BUILDING A DATASET FOR IMAGE STEGANOGRAPHY.....	36
<i>Chris Woolley, Ahmed Ibrahim, Peter Hannay</i>	

A FRAMEWORK FOR FORENSIC RECONSTRUCTION OF SPONTANEOUS AD HOC NETWORKS

Alastair Nisbet

Security & Forensics Research Group, Information Technology & Software Engineering Department
Auckland University of Technology, Auckland, New Zealand
anisbet@aut.ac.nz

Abstract

Spontaneous ad hoc networks are distinguished by rapid deployment for a specific purpose, with no forward planning or pre-design in their topology. Often these networks will spring up through necessity whenever a network is required urgently but briefly. This may be in a disaster recovery setting, military uses where often the network is unplanned but the devices are pre-installed with security settings, educational networks or networks created as a one-off for a meeting such as in a business organisation. Generally, wireless networks pose problems for forensic investigators because of the open nature of the medium, but if logging procedures and pre-planned connections are in place, past messages, including nefarious activity can often be easily traced through normal forensic practices. However, the often urgent nature of the spontaneous ad hoc communication requirements of these networks leads to the acceptance onto the network of anyone with a wireless device. Additionally, the identity of the network members, their location and the numbers within the network are all unknown. With no centre of control of the network, such as a central server or wireless access point, the ability to forensically reconstruct the network topology and trace a malicious message or other inappropriate or criminal activity would seem impossible. This research aims to demonstrate that forensic reconstruction is possible in these types of networks and the current research provides initial results for how forensic investigators can best undertake these investigations.

Keywords: spontaneous, wireless, wifi, MANET, simulation, forensic investigations

INTRODUCTION

With the introduction in 1997 of the original IEEE 802.11 wireless networking standard, wireless communication ‘came of age’. Whilst other, more basic attempts at wireless networking had been developed, the IEEE suite of standards that sprang from the original 2Mbps standard has seen steady advances in throughput, security and usability. Whilst originally envisaged as extensions to existing wired networks by utilising wireless access points, a distinct type of network that departed from using access points was quickly developed. This ad hoc mode allowed users of often mobile devices to connect directly to one another’s devices to form a mesh network that could be used in a truly peer to peer fashion. The main advantage was the ability to quickly connect to each other’s devices without the need for a preconfigured access point to act as the centre of control for the network (Kuo, Chu 2016). The rapid deployment of ad hoc networks meant that not only pre-planned networks with pre-configured devices could form this topology, but that unplanned networks could also be created on-the-fly for a very specific and often urgent need (Nelson, Steckler et al. 2011). This gives great potential to this type of truly ad hoc network, often referred to as spontaneous networks, but as yet this potential remains largely untapped. With concerns over security, especially with unplanned networks which may allow any person with a device to connect without the need for authorisation or authentication, these networks are used rarely. However, in the past few years the benefits of quick and unplanned network formation are being discussed and security implementations are proposed that have the potential to make even totally unplanned networks have robust security (Lacuesta, Lloret et al. 2013).

One area that has been left out of these discussions and proposed security protocols is that of forensic reconstructions of the networks to trace misbehaving devices. Forensics plays a vital part in seeing the potential of computers and networks utilised because it allows misbehaviour to have a consequence, that of identifying the nefarious device and user and bringing them to account for their behaviour. This research examines the issue of forensic reconstruction of spontaneous ad hoc networks and introduces unique research into how forensic investigators can use techniques which will allow them to trace the origins of misbehaviour back to the originating device. Initial results are discussed and a guide as to how these techniques will be further developed to provide practical implementations of forensic investigations are made.

STATE OF THE ART

On the 14th of September 1999, the two extensions to the original standard provided a security protocol built into the standard which was seen as a significant step forward in acceptance of wireless networking. The 802.11a and 802.11b protocols also greatly increased the throughput from a maximum of 2Mbps to 54Mbps and 11Mbps respectively (IEEE 1999). Whilst the security in the form of WEP proved to have serious flaws which would later be rectified in the form of WPA and WPA2, the increased bandwidth had an immediate and significant effect (Cam-Winget, Housley et al. 2003). Not only did it provide much increased packet throughput, often greater than what was being provided on the LAN at the time, but it meant that an ad hoc networking topology was much more usable (Wang, Wang et al. 2005). This is because in a mesh network, there are often devices communicating at the same time in pairs and within the radio range of other communicating devices. The high bandwidth meant multiple message passing could be carried out with an acceptable bandwidth, even with the interference caused to each other's radio signals. Later, bandwidth would be significantly increased in later revisions to the standard so that today we enjoy bandwidths up to 150Mbps or more in ad hoc networking mode (IEEE 2009). The increasing acceptance of mesh networking, including unplanned spontaneous networks has seen increased awareness of security issues, especially in the form of cryptography with the associated encryption key management challenges. Whilst these are being overcome as new ideas lead to improved protocols, the forensics of spontaneous networks has largely been ignored.

Digital forensics traditionally utilises a life cycle of 4 phases (NIST 2006). These phases are shown in figure 1.

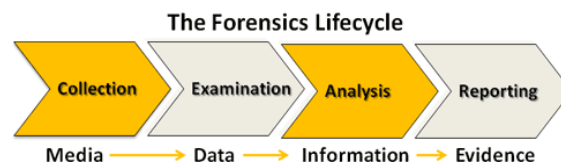


Figure 1: The Digital Forensic Life Cycle

The phases begin by seizing a device, often a computer but possibly other hardware such as a router, switch or wireless access point. In many cases these devices will have logs that will record much of the network traffic that has passed through the device. Whilst configuration of the devices may not include collecting all packets, source and destination address and often some identifying packet information such as headers may be recorded. This can be invaluable information for a forensic investigator who may be building a case against a misbehaving user such as an errant employee in a company or an outsider intruding into a company network. Serious offences may be committed with unauthorised access, use of resources or destruction of data. Many of the offences relating to computer crimes are now contained in New Zealand's Crimes Amendment Act 2003 and include harsh penalties for such behaviour. Most other countries have similar legislation with harsh penalties including long prison terms.

However, the investigation often relies on the ability to seize the offending device and perform an acquisition and analysis of the evidence on this device. Whilst monitoring network traffic may be possible in wired networks, this is not practical in a spontaneous wireless network (N. Malik, J Chandramouli et al. 2017). Network logs, if they exist on the device, may point to the errant device and provide corroborating evidence of the misbehaviour. However, in spontaneous networks where network deployment has no prior planning and often no prior notice of its formation, this first step may prove to be impossible. This is because the users of the network are usually not identified, their devices are not registered for use on the network, their location may change as they move around the network and they may come and go as mobile devices are shut down to save battery power and restarted later to join the network multiple times. Additionally, a node that may be monitoring traffic has a radio range of approximately 300 metres and therefore may be out of range of most of the network. If misbehaviour is detected, there may be the ability to eject a device from the network by revoking a digital certificate (Nisbet, 2014), but attempting to physically locate the device to perform a forensic examination may be unsuccessful (Zhang, Lee et al. 2003). This could be especially true if the user of the device is aware that their device is being sought.

What is required is a method of reconstructing the network at a particular time in its life so that the forensic reconstruction can be performed, possibly without access to the offending device. This research demonstrates through simulation that this type of reconstruction may be possible provided some of the devices in the network can be examined. Additionally, the type and amount of information that can be realistically logged onto the devices is discussed so that an initial guideline can be made as to what percentage of devices in the network are needed to be examined given these varying parameters. The following section discusses the challenges inherent

in forensic evidence gathering from spontaneous network deployment, especially when evidence may be required long after the network has served its purpose and disbanded.

FORENSIC RECONSTRUCTION

In a truly ad hoc network, a mesh network will often form where many of the devices in the network, referred to as nodes, will have one or more neighbours within direct communication distance. This not only allows neighbours to communicate directly with each other, but they can often be used as hopping points for messages to greatly increase the distance of the network communication from the usual 300 metres radio range, to in theory unlimited distances (Nisbet and Rashid 2009). One other benefit of multiple neighbours is that the path utilised to send a message can have many different possible paths from sender to receiver (Suzuki, Kaneko et al. 2006). This redundancy of routes greatly increases the efficiency of message passing and is one significant benefit of mesh networks, but when these networks have at least some nodes that are mobile, the message paths may also change rapidly. This requires a very efficient reactive routing protocol that will find a path through multiple hopping points from sender to receiver, often in both directions if a reply is given as the same route back to the sender may not exist by the time the message is received. This means that routes are changing rapidly as are neighbours and often locations of nodes. This all adds to the considerable complexity of these types of networks and the forensic challenges that stem from these temporal and spatial changes (Dewald 2015).

The current research is in the development phase of initial testing through simulations. A wireless network simulator was constructed utilising Matlab and has been modified to permit nodes in the network to record certain parameters which can then be called upon to attempt to rebuild the network. The process involves deployment of an ad hoc network where the nodes are placed at random on a grid and the network grows as time passes. This simulates the spontaneous deployment of an ad hoc network for any purpose where people have begun communicating with neighbours. Whilst the purpose of deployment of the network is unimportant, it could be for such urgent and unplanned reasons such as when a major disaster has struck knocking out communications or some similar event that requires wireless communication by masses of devices (Shimoda and Gyoda 2011). The network grows as more and more people join the networks, with some leaving the network temporarily, perhaps to save battery power on their devices, and others joining at apparent random points on the grid. During the time the simulated network is 'live', nodes will appear at random points with some nodes mobile and some shutting down temporarily or permanently.

The network simulation runs for a simulation time of 10 minutes which is sufficient with the growth to mimic perhaps several hours in real time. During this time, information is recorded on each node as to its current status, location and the source and destination of any messages that it may send, pass on or receive. One issue here is that recording of information may be desirable but with limited time to record data and limited storage available on many devices, it is necessary to store as little information as is necessary. Therefore, the simulation utilises the ability to call only some or all of the information recorded. This parameter will allow comparisons to be made as to what information is necessary and what difference not having some information will make to the success of a reconstruction. The information recorded in the nodes is shown in table 1.

Table 1: Information recorded in the nodes

Attribute	Format
Live	yes / no
Neighbours	node IDs
Neighbour Location	x and y coordinates
Server	yes / no
Network Number	integer
Position	x and y coordinates

Once the simulation has completed, a time from 1 second to the end of the simulation is selected to attempt the reconstruction. This time represents when a malicious act has occurred and the offending node needs to be identified. For example, this may be an act such as a false message about rescue in a disaster area that may have led to serious injury or death. Each node records the information shown in table 1, along with other information related to security settings. It is unlikely that all devices that were in the network can be recovered. Here, a distinction needs to be drawn as to which devices qualify. Either it can be those devices that were live at the time the misbehaviour occurred or alternatively all devices that have ever appeared in the network. A call to those people who may have participated in the network may result in devices that were not live at the time of the misbehaviour and for this reason the reconstructions have chosen to include these 'no longer live' nodes. The

reconstruction of the networks begins with a variable parameter of what percentage of nodes were able to be recovered for examination. This could range from one device to all devices. Figure 1 shows a network that has grown over a period of 60 seconds and now contains a total of 24 nodes. What is interesting to note is that rather than a single network forming, 3 separate networks have formed independently. Of these, only the biggest network has reached a capacity where digital certificates can be issued, shown by the red labels on the nodes. The nodes out of 300 metre range of another node are shown isolated with a circle representing the 300 metre radio range. The 2 small networks at the bottom of the grid have 2 members each. This snapshot of the network is taken during the live network simulation. The goal is therefore to reconstruct the network well after the network has ceased and the nodes have disbanded. This must be achieved as accurately as possible so that nodes in the network can be identified and placed in their location at any chosen time, such as when a malicious message was sent.

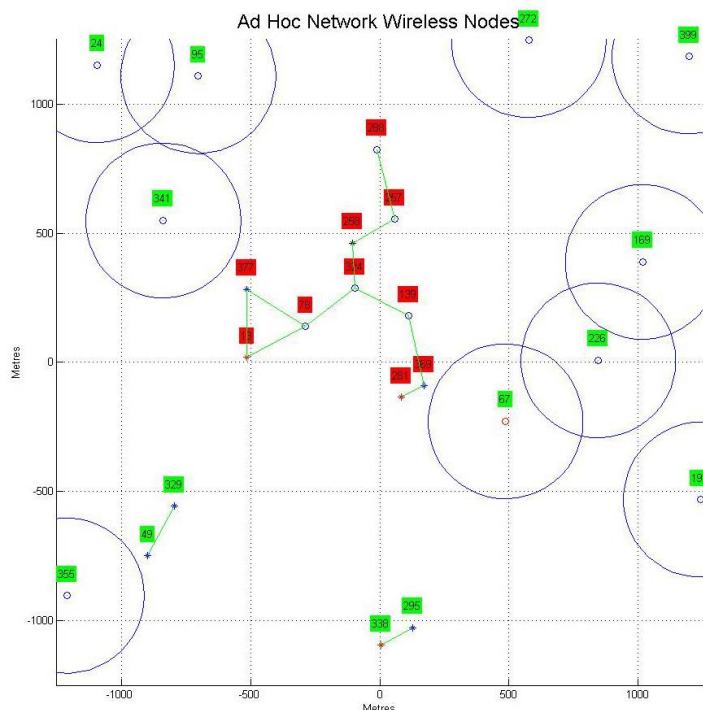


Figure 2: Spontaneous network at 60 seconds

One challenge that is immediately apparent is that whilst the goal may be to identify a misbehaving node in a network, there is no record of the members of the network until devices that participated in the network have been examined (Graffi, Mogre et al. 2007). This leads to 2 desirable records that could be stored dynamically that would greatly assist the forensic investigator to reconstruct the network and trace the origin of the malicious message. The first is to identify and collect at least 1 member of the target network. The second is to have a record of all members who were in the network at the time recorded in every device in the network. This would mean that collecting any single device would give all information about the identities of all devices in the network. Here, identities simply refer to a unique identifier for the device but this would not include who was using the device. Additionally, the unique identifier, at least for the lifetime of the network, and the location of the device at the chosen time would not disclose any private information relating to the user of the device.

This is an example of what would be most desirable and could be written into a protocol with forensic readiness in mind when designing the protocol. Without this preparedness, the investigator would need to collect as many devices as possible, possibly through advertising for network participants, and then analyse the devices initially to determine if they had belonged to the relevant network at the time of the offence. With the ability to store relevant forensic evidence within each of the network members' devices, the network can be reconstructed by looking through the evidence of each collected node and placing nodes in the network space as they were at the chosen time. An example of this is 2 views of the network reconstructed in this manner and shown in figure 3.

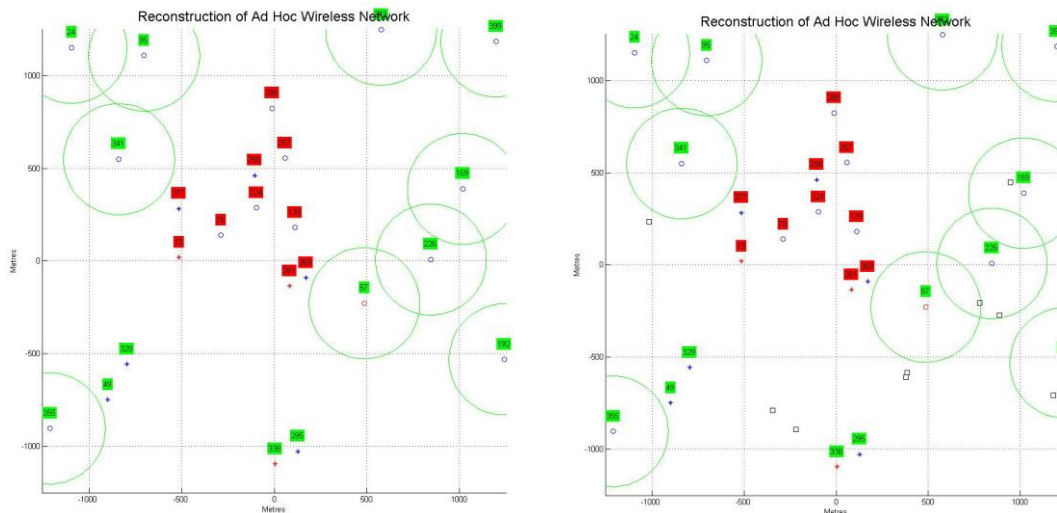


Figure 3: Reconstruction of the network at 60 seconds

A comparison with figure 2 shows that the reconstruction is successful. The target time is 60 seconds after the formation of the network, and this could represent a specific time when a malicious message was sent leading to serious consequences. The left side of this figure shows only those nodes that were live at the tick of the clock of 60 seconds. The right figure shows all nodes live at 60 seconds and all nodes that were once live but are no longer live and are identified as black squares. This is an option that can be chosen and may be useful to determine any other nodes in the vicinity of the network that may still contain some evidence of the network members or malicious behaviour even though they had closed down by the time a malicious message was sent.

The example shows what can be done if either all nodes have been successfully gathered and their evidence examined, or some isolated nodes are gathered and at least some nodes in the main network in the middle are gathered and examined. Figure 4 shows a reconstruction of the network at 10, 30 and 50 seconds and demonstrates how the reconstruction time can be entered by the forensic investigator to display the network at any given time during its lifetime. The update period for the simulation is 1 second, so at any time during the life of the network, the reconstruction can be made and is accurate to within the 1 second update period.

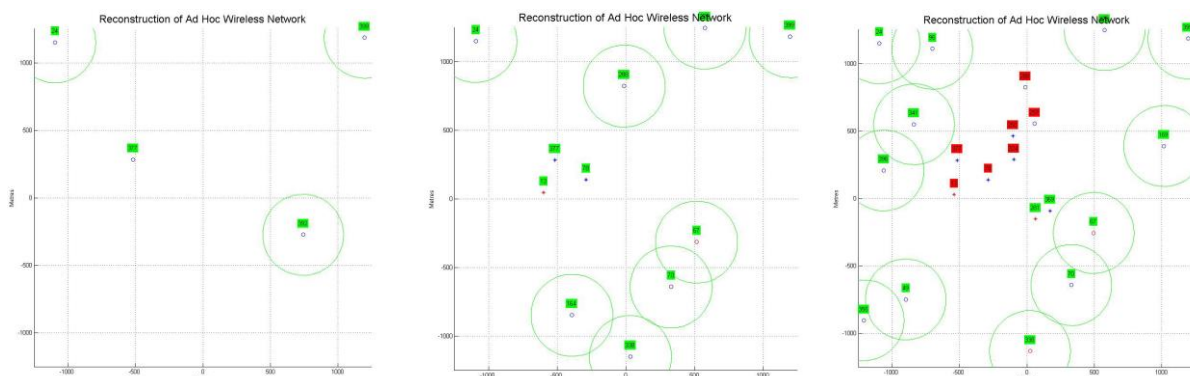


Figure 4: Reconstruction at 10, 30 and 50 seconds

Whilst the goal for the forensic investigator is to collect 100% of the devices that existed in the network, it is more likely that only a percentage of nodes can be collected and the reconstruction is therefore attempted with this limited number of nodes. Here, the amount and type of information stored in the nodes makes a significant difference in the success of the reconstruction. Figure 5 shows a similar simulation run to 60 seconds representing approximately an hour or more for the network to form. In this figure, the reconstruction is run at the 60 second mark but a limited percentage of nodes were able to be collected. Nodes that had been present in the network previously but were not live at the target time are present as black squares.

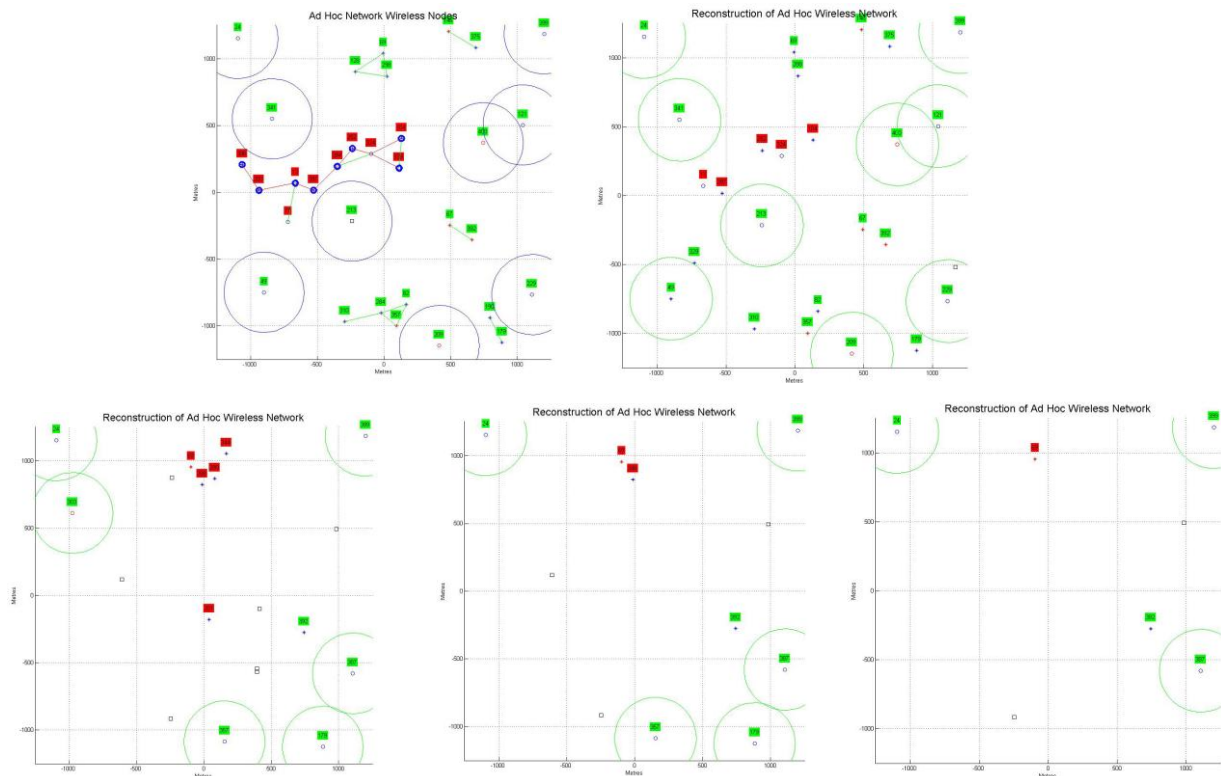


Figure 5: Network at 60 seconds, reconstruction with 80%, 50%, 30%, 20% nodes

It is apparent that relying solely on the node to be recovered to place it at the correct location in the network is problematic. Even with 80% of the total nodes recovered, there is no guarantee that this will equate to 80% of the nodes in the network of interest. If the target of the investigation is the network at centre left with 10 nodes, then the reconstruction with 80% of nodes recovered has in fact recovered only 50% of the nodes in this network although 80% are recovered overall. This is unlucky but a consequence of chance in recovering fewer than all the nodes.

Rather than rely solely on each node's information about itself, neighbouring nodes may have stored information that can be used to identify and place nodes on the grid that have not been successfully collected. This would mean that for the 5 nodes recovered from the target network, their logs which are recovered may contain sufficient information to place the remaining 5 nodes in their respective locations. If an investigation is tracing the origin of a malicious message, then the information required may be the path the message has taken which can then be traced from receiver to the nefarious sender. Acquiring the nodes utilised as hopping points for the message may not provide any more useful information, so that information about these nodes may be sufficient. This does however require that nodes collect and store the information in case a forensic investigation is undertaken later. The forensic readiness of the network could lead to a much more successful outcome by simply querying neighbours attributes on a regular basis and storing the information. At this stage of development, this has not been designed into the forensic protocol but will rather be the next stage of development for the forensic reconstruction simulation software.

CONCLUSION

Wherever a network is present, security and forensic readiness should be a major consideration. Preparing for the worst means that should some serious issue occurs that leads to the requirement for a forensic investigation of a spontaneous ad hoc network, the forensic investigator can have the best chance of the successful outcome of identifying the perpetrator by conducting a forensic investigation. It may be that serious misbehaviour in the network has led to serious injury or death with the only hope of identifying the perpetrator being the number of devices that can be recovered and the information that they contain. The development of protocols that allow for forensic readiness is something that has been largely ignored with the focus in the past more on security issues in these networks. The ability to reconstruct a spontaneous ad hoc network, possibly days or weeks after it has been disbanded is something that is urgently required by forensic investigators. The further development in this

research, it is hoped, will lead to better guidelines as to what information and what percentage of devices will likely be needed for successful forensic investigations. Further development will greatly enhance the protocol and provide results that can act as these guidelines.

REFERENCES

- Cam-Winget, N., R. Housley, et al. (2003). "Security flaws in 802.11 data link protocols" *Commun. ACM* 46 (5): 35-39
- Dewald, A. (2015). Characteristic Evidence, Counter Evidence and Reconstruction Problems in Forensic Computing. Ninth International Conference on IT Security Incident Management & IT Forensics (IMF), 2015 Megdeburg: 77-82.
- Graffi, K., P. S. Mogre, et al. (2007). Detection of Colluding Misbehaving Nodes in Mobile Ad Hoc and Wireless Mesh Networks. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE.
- IEEE (1999). "IEEE Std 802.11b-1999." Retrieved 15th June 2004, 2004.
- IEEE (2009). "IEEE Std 802.11n." Retrieved 8th December 2009, 2009.
- Lacuesta, R., J. Lloret, et al. (2013). "A Secure Protocol for Spontaneous Wireless Ad Hoc Networks Creation." *IEEE Transactions on Parallel and Distributed Systems* 24(4): 629-641.
- Kuo, W., Chu, S. "Energy Efficiency Optimization for Mobile Ads Hoc Networks". *IEEE Access*, vol 4, pp928-940. doi: 10.1109/ACCESS.2016.2538269
- N. Malik, J Chandramouli, et al. (2017). Using network traffic to verify mobile device forensic artifacts. Consumer Communications & Networking Conference (CCNC) Las Vegas, Nevada, USA.
- Nelson, C. B., B. D. Steckler, et al. (2011). The Evolution of Hastily Formed Networks for Disaster Response: Technologies, Case Studies, and Future Trends. Global Humanitarian Technology Conference (GHTC), 2011 IEEE.
- Nisbet, A. (2014). A Simulation-based Study of Server Selection Rules in MANETs Utilising Threshold Cryptography. Proceedings of the 11th Australian Information Security Management Conference, 2-4 December 2013, Perth, Australia.
- Nisbet, A. and M. A. Rashid (2009). A Scalable and Tunable Encryption Key Management Scheme for Mobile Ad Hoc Networks. International Conference on Wireless Networks 2009, Las Vegas, NV.
- NIST (2006). Guide to Integrating Forensic Techniques into Incident Response Special Publication 800-86.
- Shimoda, K. and K. Gyoda (2011). Analysis of Ad Hoc Network Performance for Disaster Communication Models. 10th International Symposium on Autonomous Decentralized Systems (ISADS), 2011
- Suzuki, H., Y. Kaneko, et al. (2006). An Ad Hoc Network in the Sky, SKYMESH, for Large-Scale Disaster Recovery. Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th.
- Wang, J.H., L.C. Wang, et al. (2005). Coverage and Capacity of a Wireless Mesh Network. International Conference on Wireless Networks, Communications and Mobile Computing, Taiwan, IEEE.
- Zhang, Y., W. Lee, et al. (2003). "Intrusion detection techniques for mobile wireless networks." *Wirel. Netw.* 9(5): 545-556.

A CENTRALISED PLATFORM FOR DIGITAL FORENSIC INVESTIGATIONS IN CLOUD-BASED ENVIRONMENTS

Shaunak Mody, Alastair Nisbet,
Security & Forensics Research Group, Information Technology & Software Engineering Department
Auckland University of Technology, Auckland, New Zealand
Shaunakmody14892@gmail.com, anisbet@aut.ac.nz

Abstract

Forensic investigations of digital media traditionally involve seizing a device and performing a forensic investigation. Often legal and physical obstructions must be overcome so that the investigator has access to the device and the right to secure it for investigation purposes. Taking a forensic image of a hard disk may need to be done in the field but analysis can usually be performed at a later time. With the rapid increase in hard disk size, the acquiring of a forensic image can take hours or days. This poses significant issues for forensic investigators when potential evidence resides in the cloud. What is highly desirable is the ability to perform the acquisition of the image and the data recovery whilst the data remains in the cloud. The comparatively small amount of recovered data can then be downloaded from the cloud. This may solve legal, time and physical obstacles with one relatively simple method. This research describes the development of cloud-based software to perform a digital forensic investigation in the cloud and describes the efficiency of the process under several different configurations utilising Amazon Web Services cloud solutions.

Keywords: cloud, forensics, network security, investigation

INTRODUCTION

As the increase in Internet speeds has seen rapid growth, the uptake of basing data in the cloud for many businesses and individuals has followed that growth. Many businesses are utilising the cloud for some, or all of their data needs, including platforms, software and infrastructure such as their corporate databases. The advantages are many with cheaper access to software that may be prohibitively expensive to purchase, the latest iterations of operating systems and other software for no extra fee and the ability for employees to access the same data from anywhere in the world. One drawback of cloud-based data for organisations is that they store all their corporate data in one place controlled by a third party. This single point for a company's data may also be attractive for the nefarious hacker who targets the data for malicious destruction, modification or theft.

The cloud computing spending in the overall IT industry is expected to increase greatly by the end of 2017. In a recent report published by Gartner in the year 2016 and cited by Cooke, it is predicted that the worldwide spending on the cloud is expected to grow by 18% by the end of 2017. The expected investment in the cloud computing industry is approximated to reach a total of US\$247 billion dollars, up from US\$209 billion dollars in 2016. A major share of this is expected to come from IaaS and PaaS. A growth of 37% is projected in 2017 in IaaS and PaaS and SaaS which is expected to grow by 20% in 2017 (Viveca Woods, 2016) (Cooke, 2016).

However, whilst the corporations are enjoying the many benefits of cloud-based services, the forensic investigators are finding these same services provide significant challenges to forensic investigations. Firstly there may be legal hurdles to overcome to gain access to the cloud-based data. Whilst the Budapest Convention on Cybercrime has seen many nations agree to cooperate in seizing and sharing of data, many countries have resisted signing up to the agreement amongst concerns over bypassing the usual checks and balances of privacy and security provided by the law, including New Zealand. The legal process in some countries may be extremely challenging and time-consuming and when time-constraints exist, both for time-critical investigations but also to prevent deletion of evidence, this alone can foil an investigation. If the legal hurdles can be overcome, then the next challenge is to have access to the data. Usually this will reside on one or many hard drives on a cloud server residing in an often undisclosed location in the world. The investigator may have to trawl through terabytes or more of information looking for the evidence required. This 'logical' view of the data is not what is normally of interest as often the best evidence has been recently deleted. Therefore, a 'physical' view of the evidence, which is a direct bit by bit copy of the hard drive(s) is much preferred. This forensic image often allows the investigator to locate and recover deleted files. However, there is currently little ability to do this rather than attempt to copy

the entire logical image of the hard drive or drives to the investigator's computer using the Internet. This is a relatively slow process and with the very large hard drives that are common in the cloud may not be practical. What is urgently required is the ability to perform the investigation on the data whilst it remains in the cloud, and this includes the forensic acquisition of the image as well as the locating and recovery of the evidence. This research attempts to show, through development of new software in the form of a bash script that can be uploaded to the cloud, that acquisition and analysis of data in the cloud is possible and realistic.

The following section discusses the cloud environment and the options available to users.

LITERATURE REVIEW

The cloud offers a 'shared responsibility model' which describes the responsibilities of the parties involved. This shared responsibility model generally has three main parties involved, the cloud provider, the service provider and the service consumer. A cloud provider represents an entity which is responsible for providing and building the infrastructure for running cloud-based services. The cloud providers are generally the owners of the infrastructure where the services are set up. A service provider is an entity which is responsible for running services, applications and provides an interface for the end-users to use. A service consumer is the entity which utilises the services provided by the service providers (Mohamed Al Morsy, 2016).

Forensic investigations follow accepted practices to ensure that the evidence acquired retains its integrity and can be presented in a court of law if necessary. Whilst guidelines exist on the finer points of acquiring and analysing evidence the basic investigation process includes seizing the device, examining the data recovered for evidence, analysing the evidence for probative value and finally reporting and presenting the evidence in a sound manner (NIST 2006).

The digital forensic guidelines provided by NIST, ACPO and other law enforcement agencies cannot at this stage be extended to the cloud environments as they only describe how to perform examinations on the infrastructure where they have physical access to the evidence. These guidelines do not mention the steps that need to be taken by examiners while conducting investigations in a distributed environment such as that of the cloud. This lack of a cloud-specific framework on how to perform investigations in the cloud has made it difficult for the examiners to conduct examinations in the cloud (Rani & Geethakumari, 2015). Several researchers have suggested using a combination of various existing frameworks for conducting investigations on the cloud. A widely used combination is that of NIST and the McKemmish framework where the acquisition and recovery is performed in the same phase followed by examination, analysis and reporting. This combination does aid an investigation but still does not guide an investigator on how to perform investigations in the cloud (Rani & Geethakumari, 2015).

The current most common commercial forensic tools such as EnCase and FTK do not provide the functionality to perform investigations on the cloud and require administrator permissions to access the data (Alqahtany, Clarke, Furnell, & Reich, 2016). These extra permissions that are pose a threat as these agents need to be trusted for transferring confidential data from the cloud to the local server (Rani & Geethakumari, 2015). Tools such as Oxygen Forensic Extractor have been found to be successful in examining the evidence acquired from a distributed computing environments but it has also been observed that it cannot be used for performing acquisitions on the cloud environment (Rani & Geethakumari, 2015). The trust factor required by tools that can acquire an image but then necessitate downloading the image to a local computer means they have failed to penetrate the cloud forensics industry (Dykstra & Sherman, 2012).

A framework called 'The integrated conceptual digital forensic framework for cloud computing' is a combination of the framework proposed by NIST and McKemmish. The NIST framework proposes a four step forensic approach including data collection, examination, analysis and reporting (Mell & Grance, 2014). The McKemmish framework consists of the identification of the digital evidence, the acquisition of the digital evidence, the analysis of digital evidence and the preservation of the digital evidence (McKemmish, 1999). While the NIST framework focuses on the examination and reporting of the data, the McKemmish framework focuses on identification and preservation of the evidence and is an iterative framework which focuses on the identification of the source of the data for the examination (Martini & Choo, 2012).

The California Fire Assistance Agreement (CFAA) model proposed a central command and control server responsible for performing all the forensic activities on the cloud (Alqahtany et al., 2016). This proposed framework aims at solving the restrictions of cross-border data privacy and security laws that exist in a cloud environment by providing them with a platform deployable on the cloud itself (Sibiya, Venter, & Fogwill, 2015). For a trusted third party to be able to perform forensic investigations on the cloud, they have to be given access to the cloud environments and also have to be verified by both the cloud service providers and the customer

whose cloud infrastructure they will be accessing for performing examinations which is a serious drawback. (Meera, Alluri, Powar, & Geethakumari, 2015). The majority of these guides fail to address the concern of the efficiency of the tools and frameworks (Casey, Katz, & Lewthwaite, 2013).

What is required is an efficient solution that will address the legal and technical issues involved in cloud forensics. This research attempts to do this by describing the development of custom-designed software that can be uploaded to an organisation's cloud-based data centre as a file that will be accepted onto the cloud. The file is a custom-written script that performs 2 distinct operations. Firstly, it acquires an image of the partition, hard drive or possibly multiple hard drives, and secondly it processes that image by analysing it for the desired files, including deleted files. These files may be documents, graphics or any other file that the examiner chooses to look for.

The efficiency is present because the acquisition and recovery of the evidence is done at the same stage in the cloud. The acquired image is then sent into a data stream which accepts the evidence as input and then runs a recovery and generates a directory containing extracted data. The software runs an automated phase for examination and analysis of the extracted data to improve the efficiency by reducing the time required for performing an examination, as this is a desirable feature of forensic software (Casey et al., 2013). It was expected that performing the acquisition and analysis in the cloud would be very efficient as there is no data transfer time required until the recovered files are downloaded after the forensic processes have completed.

The following section describes the experimental setup and testing of the new software.

RESEARCH DESIGN

This research focuses on the Linux-based environments as according to Amazon, more than half a million servers running on AWS are Linux (Amazon Web Services, 2017a). The software was designed and developed on the Ubuntu operating system version 14.04.01. The scripting language used for developing this software is bash scripting with 1400 lines of code and the automation platform utilises cssh. Amazon Web Services (AWS) was selected as reports published by Gartner and Forbes state that currently AWS is the leading cloud service provider in the market as discussed by Cooke (Cooke, 2016). The Cloud offers its users the option to choose an operating system from a wide range of distributions such as Microsoft Windows Servers, Linux Servers including Ubuntu Server, Red Hat Enterprise Linux, Suse Linux and CentOS. This type of flexibility makes it easy for the customer to deploy a hybrid infrastructure much more easily on the cloud than with local on-premise infrastructure.

Furthermore, once the user has selected the operating system, the cloud service provider offers an option of selecting the configuration of the server based on the purpose of the deployment. Each server has a 10GB hard disk, so that multiples of 10GB must be acquired and analysed. For example, the single server required a 10GB acquisition, 2 servers requires 20GB and 3 servers 30GB. The timings of these acquisitions and analysis can then be used as a guide for forensic investigators who can multiple the time taken to then calculate the time required for larger disks. For example, the timing of 10GB on a single server is multiplied by 100 if the hard disk in the cloud is a single server with 1TB, and so forth. The RAM and CPU configurations range from a minimum of 512 MB of RAM and 1 vCPU to a terabyte of RAM and 128 vCPU's.

The software was tested and developed on the T2 series of server models offered by Amazon Web Services. The software was developed and tested on the three of the biggest models offered by the T2 series, T2.large, T2.xlarge and T2.2xlarge. These three server models were tested against all the server models offered by T2 family of Amazon web services. The client nodes that used were as follows t2.micro, t2.small, t2.medium, t2.large, t2.xlarge, t2.2xlarge. Each model represents servers with different hardware configurations. Each server was tested against all the client nodes for benchmarking the performance of the server to identify the most optimal server.

Each experiment involved uploading the script to the cloud where the target data resides. Then the script was executed which acquired a forensic image of the selected partition. Once the image was acquired, a forensic analysis was made. This involved extensive keyword searches for both logical files and deleted files. The results were compared against each other using the time required to perform the actions as a comparison. Each experiment was run 18 times, as this gave an acceptable but manageable number of iterations to manage and average that was then used for the final result.

RESULTS

The first experiment involved testbed-1. The tested server is t2.large and is equipped with 8GB of RAM and 4 vCPUs. Six different client nodes were tested ranging from t2.micro to t2.2xlarge. The next experiment, testbed-2, utilised t2.xlarge with 16GB of RAM and 8 vCPUs. Similar ranges of client nodes were used. Finally, the third testbed, testbed-3, was T2.2xlarge with 32GB of RAM and 8vCPUs with similar ranges of client nodes. It was felt that this would give a good comparison in performance for the client nodes with the increasing ability of the server to process the information. The number of client servers present during a given iteration varied from a single server to a maximum of three servers during the experiments. Figure 1 shows the time taken to acquire the image using T2 Large.

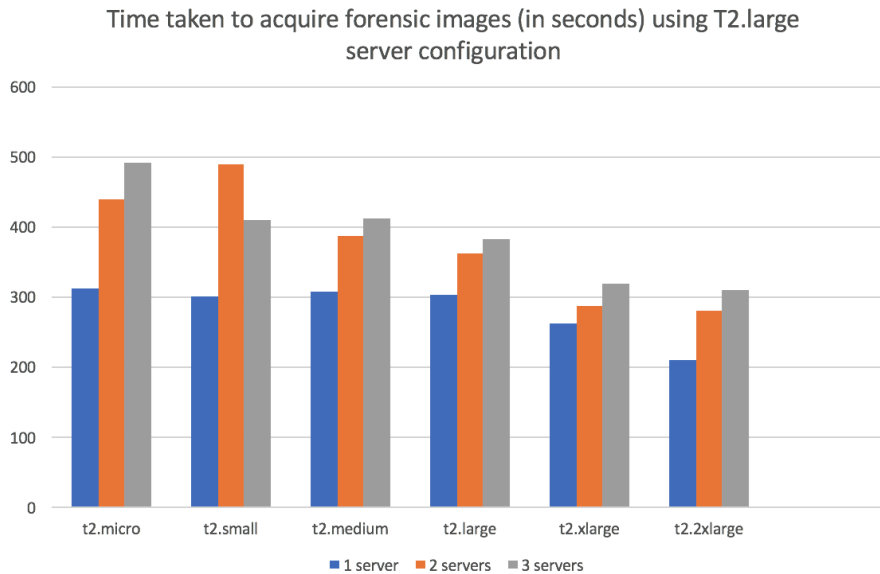


Figure 1: T2.large acquisition results

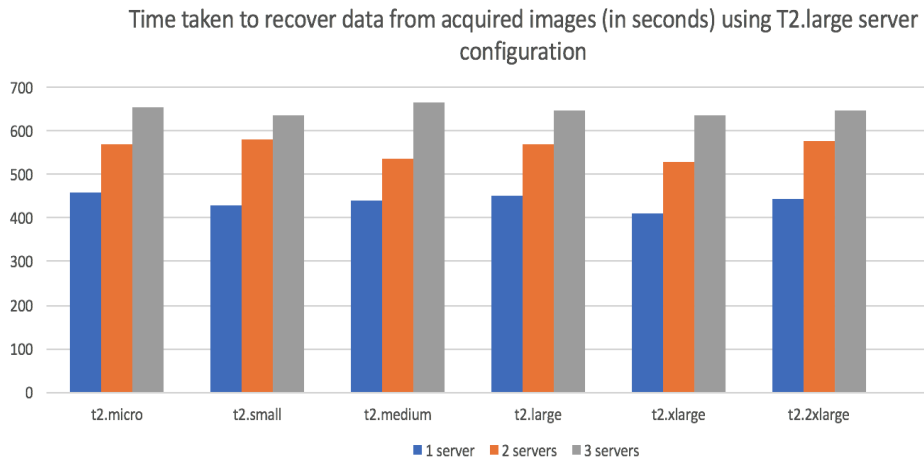


Figure 2: T2.large data recovery results

Figure 2 shows the results for a data recovery performed on this image whilst it resides in the cloud. The results from sever T2.large tested against the 6 client nodes with 1 to 3 servers indicate that the more servers available the slower the time to acquire an image. Generally, a linear decrease in time is seen across all clients as they move from micro through to t2.2xlarge. Image time remains approximately the same for all client nodes. Table 1 expands on these acquisition results by showing the T2.large server tested against the T2.large client in its 3 differing configurations.

Table 1: Server T2.large with client nodes T2

Client node configuration	No. of servers	Time taken to acquire forensic image
T2.large	1	5 minutes 3 seconds
	2	6 minutes 2 seconds
	3	6 minutes 22 seconds
T2.xlarge	1	4 minutes 23 seconds
	2	4 minutes 47 seconds
	3	5 minutes 20 seconds
T2.2xlarge	1	3 minutes 30 seconds
	2	4 minutes 40 seconds
	3	5 minutes 9 seconds

What is clear is that predicting the time taken from the number of servers cannot be done precisely but a reasonable estimate of the time required is possible. This may be useful when acquiring the image as it is desirable that the data is not be modified by users when the image is acquired. Figure 3 shows the results from acquiring an image when utilising T2.xlarge as the server.

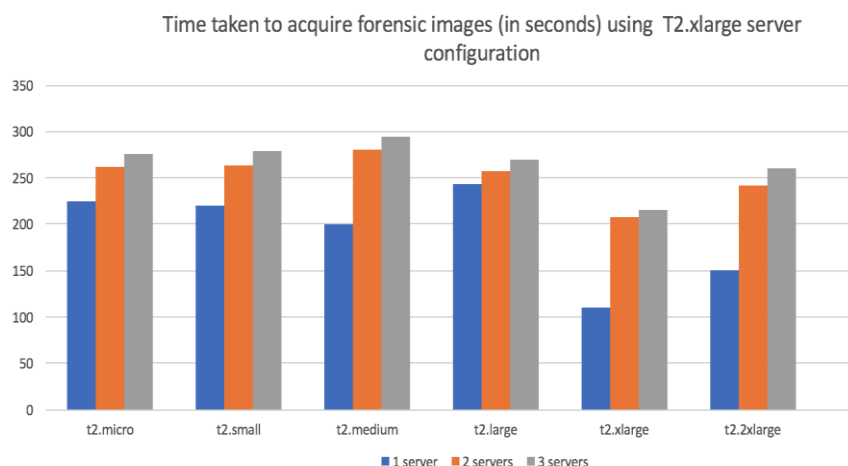


Figure 3: T2.xlarge acquisition results

The next step is to recover data from the forensic image and the time taken to complete this task with 1-3 servers is shown in figure 4.

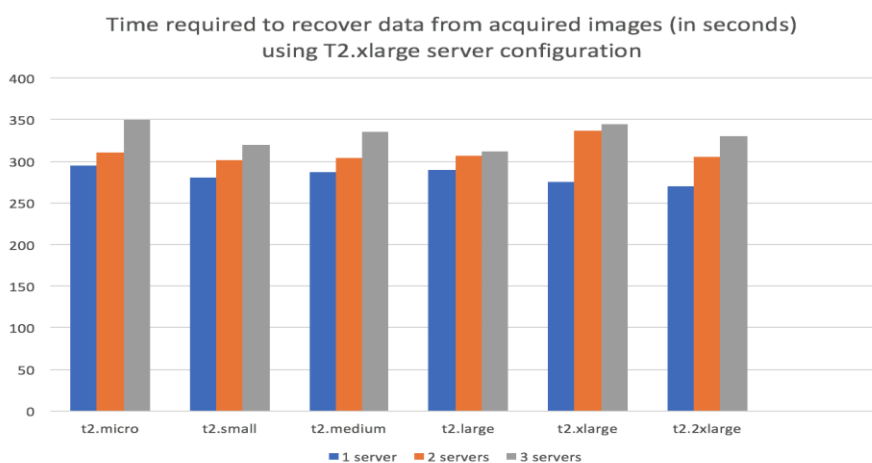


Figure 4: T2.xlarge acquisition results

Figures 3 and 4 indicate that as the servers utilised are more powerful in terms of increased RAM and virtual CPUs, the time required to acquire an image and recover data from the image decreases significantly.

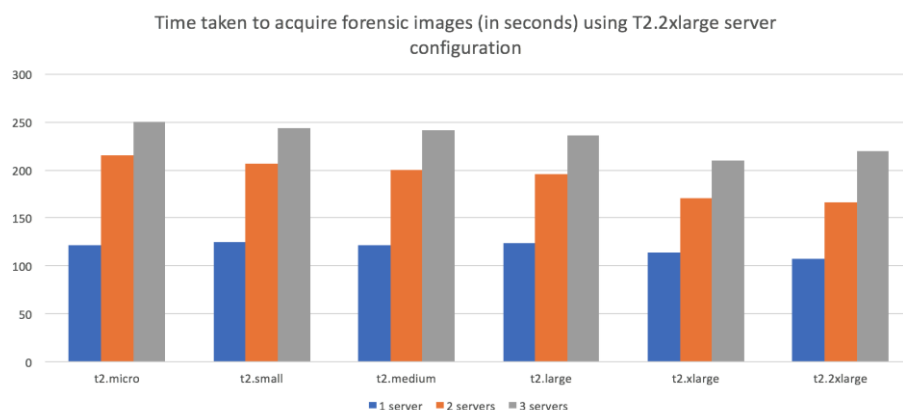


Figure 5: T2.xlarge acquisition results

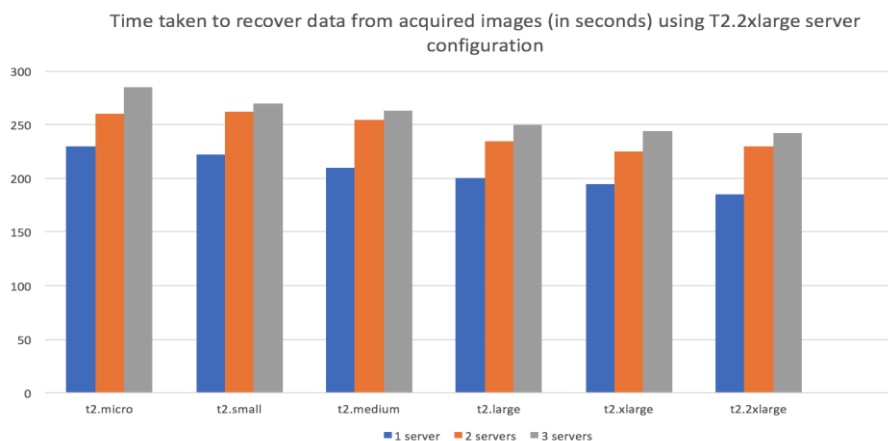


Figure 6: T2.xlarge acquisition results

Figures 5 and 6 show that an increase in RAM and vCPUs lead to increasing efficiency in acquisition and data recovery. The time decreases from using the less powerful T2.large but not significantly. The results obtained from running the experiment with central forensic server as t2.large client model showed that t2.micro took the most time for acquiring images and recovering data. The quickest client node was t2.2xlarge. The average time required to acquire images from the client nodes was between 4 minutes and 6 minutes. The results also showed that in order to acquire the evidence more efficiently the server performs best with client node running t2.2xlarge. Further, while recovering data from the acquired images it was found that t2.2xlarge performed the best when the server was configured as t2.large. While identifying the most efficient server model for this software deployment, it was found that the time difference between the most efficient node and the other two nodes is minimal. While the results of the test cases show that t2.micro was the slowest performing client node, t2.2xlarge was the most efficient client node for performing recovery and a combination of t2.2xlarge, t2.large and t2.small was identified as the most efficient for acquisition. T2.2xlarge was the identified as the most efficient model having 8 vCPU's and 32GB of RAM, the highest configuration server amongst the three.

The average time taken to acquire the image and perform recovery is shown in table 2. This shows that t2.2xlarge is the best performing server of the 3 tested. This increase in efficiency is measured by time and given as a percentage compared to the baseline.

Table 2: Percentage difference from benchmark T2.2xlarge

Client node configuration	No. of servers	Time difference for acquisition (in percentage)
T2.micro	1	149%
	2	157%
	3	159%
T2.small	1	143%
	2	146%
	3	157%
T2.medium	1	146%
	2	138%
	3	133%
T2.large	1	144%
	2	129%
	3	123%
T2.xlarge	1	125%
	2	102%
	3	103%
T2.2xlarge	1	100%
	2	100%
	3	100%

DISCUSSION

It can be seen from the performance of the script in the cloud that acquisition and recovery times are a few minutes for 10GB hard drives, and even expanding this to 30GB with 3 servers requires only a relatively small increase in time of up to 50%, something that is still very acceptable. As more servers are added, the time increases by a small amount and this follows the configuration changes for the client nodes, where more powerful nodes consequently reduce the time for acquisition and recovery but only by a relatively small amount.

Therefore, the results indicate that performing a data acquisition followed by file recovery is possible in the cloud. Further, the increased efficiency by performing these actions whilst the data remains in the cloud means that this type of forensic process could realistically be performed on very large hard drives. Whilst the time to acquire and recover files could take many days or more, this may be of little concern if the primary concern is to recover the forensic evidence. The alternative and more usual practice of gaining physical access to the hard drive(s) is likely not possible in a cloud environment. Legal issues, especially present in many countries where the data may be stored, along with Internet speeds that are simply too slow to transfer terabytes or more of data, mean that traditional recovery is often not realistic. Additionally, downloading a large amount of data in an acquisition and then recovering a small amount of data that is of interest means that most of the data is of no value to the investigator. Overall, results indicate that this process has significant benefits for forensic investigators, and whilst more development is required to expand its use to more platforms, services and configurations, the basic concept shows significant promise.

CONCLUSION

The cloud continues to see extensive growth and acceptance by organisations and individuals, especially as faster Internet speeds are offered globally. However, the challenges posed to forensic investigators by this technology are not keeping pace with the developments of cloud technologies. Currently, the legal, technical and other practical challenges make forensic investigations of cloud-based data extremely difficult and therefore often unsuccessful. These results show that these challenges can largely be overcome by deploying this type of software in the cloud alongside the forensic data sought by the investigator. The results of these experiments show that this is an effective and efficient method for collecting forensic data as no data transfer is required during the process, other than downloading the recovered files of interest once the processes have completed. Whilst this software is sufficient to perform the forensic processes in the tested environment, many different

services running different hardware and software configurations exist in the cloud. Therefore, continued development of this type of software is required so that all services on all platforms can be investigated in this manner, leading to timely and effective forensic data gathering by investigators around the world.

REFERENCES

- Alqahtany, S., Clarke, N., Furnell, S., & Reich, C. (2016). A forensic acquisition and analysis system for Iaas. *Cluster Computing*, 19 (1), 439-453.
- Amazon Web Services. (2016). AWS cloud formation. Blog.
- Amazon Web Services. (2017a). Amazon ec2 instance types. Retrieved from <https://aws.amazon.com/ec2/instance-types/>
- Casey, E., Katz, G., & Lewthwaite, J. (2013). Honing digital forensic processes. *Digital Investigation*, 10 (2), 138-47.
- Cooke, L. (2016, August). What's changed: Gartner's 2016 cloud infrastructure-as-a-service magic quadrant. Retrieved from <https://solutionsreview.com/cloud-platforms/whats-changed-gartners-2016-cloud-infrastructure-as-a-service-magic-quadrant/>
- 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, S90-S98.
- Meera, G., Alluri, B. K. R., Powar, D., & Geethakumari, G. (2015). A strategy for enabling forensic investigation in cloud iaas. *International conference on computer and communication technologies (icccct)*, 2015 ieee (pp. 1-5).
- Mell, P. & Grance, T. (2014). Nist cloud computing forensic science challenges. Draft Nistir, 8006.
- Mohamed Al Morsy, I. M., John Grundy. (2016). An analysis of the cloud computing security problem. *Computer Science & Software Engineering*, Faculty of Information & Communication Technologies Swinburne University of Technology, Hawthorn, Victoria, Australia.
- Martini, B. & Choo, K.-K. R. (2012). An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation*, 9 (2), 71-80.
- McKemmish, R. (1999). What is forensic computing? Australian Institute of Criminology, Trends and Issues in crime and criminal justice.
- NIST (2006). Guide to Integrating Forensic Techniques into Incident Response Special Publication 800-86.
- Rani, D. R. & Geethakumari, G. (2015). *A meta-analysis of cloud forensic frameworks and tools*. IEEE Conference on power, control, communication and computational technologies for sustainable growth. 11-12 December 2015. Kurnool, India. doi: 10.1109/PCCCTSG.2015.7503922
- Sibiya, G., Venter, H. S., & Fogwill, T. (2015, May). Digital forensics in the cloud: the state of the art. In 2015 IST-Africa Conference. 6-8 May 2015. Lilongwe, Malawi. doi:10.1109/ISTAFRICA.2015.7190540
- VivecaWoods, R. (2016, January). Gartner says worldwide public cloud services market is forecast to reach US\$204 billion in 2016. Retrieved from <http://www.gartner.com/newsroom/id/3188817>

ISEEK, A TOOL FOR HIGH SPEED, CONCURRENT, DISTRIBUTED FORENSIC DATA ACQUISITION

Richard Adams¹, Graham Mann², Valerie Hobbs²

¹XtremeForensics, ²Murdoch University

ra@xtreme forensics.com, g.mann@murdoch.edu.au, v.hobbs@murdoch.edu.au

Abstract

Electronic discovery (also written as e-discovery or eDiscovery) and digital forensics are processes in which electronic data is sought, located, secured, and processed with the expectation that it may be used as evidence in legal proceedings. Electronic evidence plays a fundamental role in many aspects of litigation (Stanfield, 2009). However, both eDiscovery and digital forensic approaches that rely on the creation of an index as part of their processing are struggling to cope with the huge increases in hard disk storage capacity. This paper introduces a novel technology that meets the existing and future data volume challenges faced by practitioners in these areas. The technology also addresses the concerns of those responsible for maintaining corporate networks as it does not require installation of 'agents' nor does it have any significant impact on network bandwidth during the search and collection process, even when this involves many computers. The technology is the embodiment of a patented process that opens the way for the development of new functionality, such as the detection of malware, compliance with corporate Information Technology (IT) policies and IT auditing. The technology introduced in this paper has been incorporated into a commercial tool called ISEEK that has already been successfully deployed in a variety of environments.

Keywords: digital forensics; eDiscovery; data acquisition tools; hybrid forensics

INTRODUCTION

Electronic evidence plays a fundamental part in many areas of litigation. In the digital forensic arena, the traditional tools that rely on creating bit-by-bit copy images of devices and then creating an index of their contents are now struggling to cope with the huge increase in hard disk storage capacity seen in recent years (Jusas, Birvinskas, & Gahramanov, 2017). This issue is also present in eDiscovery situations where practitioners typically deal with corporate servers and large numbers of computers (Sondhi, & Arora, 2016). It has therefore become clear that innovation is urgently required if two fundamental aspects of litigation, digital forensics and eDiscovery, are not to impede the legal process through their inability to handle modern volumes of data¹.

Both digital forensics and eDiscovery begin with the forensic acquisition of data that may be used as evidence. For that data to be 'reliable', and therefore admissible, Steel (2006) provides three conditions:

1. The data acquired were from the indicated source
2. The data acquired were collected using proven tools and techniques
3. The data have not been altered since the time of acquisition.

To cope with increasing data volumes, digital forensic practitioners are increasingly resorting to creating 'logical containers' (holding a collection of files and directories) rather than bit-by-bit forensic images. This is very like the collection activity associated with eDiscovery, where the need to process large amounts of data, typically across a network connection, has earned this practice a reputation for being slow, cumbersome and expensive (Sondhi, & Arora, 2014).

Time is always a critical factor in digital forensics and eDiscovery, not only in relation to the court process itself but also in relation to the extent of the disruption caused to those entities involved in collecting data as part of litigation (Adams, Hobbs & Mann, 2013).

In the remainder of this paper, we detail further problems for the collection of electronic data. We discuss current approaches in eDiscovery and digital forensics and identify some of their fundamental limitations. We then propose a new Hybrid Forensics approach to address these problems together with a practical tool called ISEEK. We include some test results from ISEEK deployment in a Windows domain environment and finally summarize

the state of development of ISEEK and comment on other potential uses based on feedback from its deployment in the field.

PROBLEMS WITH TRADITIONAL FORENSIC DATA ACQUISITION

Our experience leads us to believe that, due to the slowness of the process, the creation of bit-by-bit images is practical in only a limited number of cases and that the situation is getting worse given the growth in disk storage capacity (Quick & Choo, 2014), (Franke et al., 2017). The current eDiscovery approach, while also suffering due to the growth in data volumes, is far from being robust. It requires significant human involvement and relies on the creation of indexes that have the potential to miss evidence by, amongst other issues, failing to recognise foreign languages, excluding ‘noise words’ and by introducing word length restrictions. These issues are covered in more detail under the following headings of Acquisition Speed and Indexing.

Acquisition Speed

In one of our experiments, a series of tests was conducted using virtual machines and virtual disks configured in an ‘ideal’ situation (i.e. not having to compensate for hardware factors), using a representative selection of forensic imaging tools. These tools were compared for their relative speed to acquire a forensic image of the 160GB source diskⁱⁱ. The results were split into two sections, one section for those tools that could boot into a write-blocked environment (listed in Table 1) and another for those that required some form of separate write-blocking to prevent alteration of the source data (listed in Table 2).

The results shown in Tables 1 and 2 display a wide range of completion times for creation of the forensic images. These results suggest that, assuming the same speeds were maintained, even the fastest tool would take almost two hours to acquire a forensic image of a 1TB disk (Table 1 - IXImager) while the slowest tool would take over 6 hours (Table 2 – EnCase Forensic Imager) even when ignoring issues such as the time to boot a machine or having to remove the disks to attach them to a write-blocking device.

1. Collection Test – Tools Not Requiring Write-Blocker or Dongle

Tool	Time	Image Size	Image Type
IXImager	17 min.	78.6 GB	ASB
Adepto	56 min.	149 GB	RAW
EnCase LineN	1 hr 3 min.	149 GB	EO1
Raptor	1 hr 9min	68.3 GB	EO1

2. Collection Test – Tools Requiring Write-Blocker or Dongle

Tool	Time	Image Size	Image Type
X-Ways Forensic	27 min.	74.4 GB	EO1
FTK Imager	50 min.	149 GB	EO1
EnCase Forensic Imager	1 hr 14 min.	149 GB	EO1

Indexing

In digital forensics, after data acquisition, the next stage is typically to index all the data contained in the forensic image to speed up subsequent searching. When the indexing process was originally developed, the storage capacity of a typical hard disk drive was around 100GB, but now disk drives of 8TB are not uncommon. Unfortunately, the speed of disk storage devices has not kept up with increasing storage capacity meaning that the indexing of a forensic image might take days, even with high-performance processors. In addition, the massive

size increase in the subsequent index files themselves now means that a robust database management system is required to handle them.

Index engines do not recognize (and therefore process) all file types that they come across, and because they tend to determine file type using the file extension they could also be fooled by a malicious user who has changed the file extensions on files they wished to hide.

In addition to having to recognize the file type, because indexes are based around collections of characters (typically words), adding items to an index is only meaningful when it is possible to identify strings of symbols as discrete words or 'related sequences of characters' within a block of source data. That entails not only being able to properly decode all the file types in the data to be indexed, but also managing to identify the words or related sequences of characters contained in those files. This causes problems when faced with foreign languages that do not contain word breaks, i.e. where words do not necessarily have white space characters between them.

To increase speed and reduce index size, indexing algorithms typically ignore white space and 'noise' charactersⁱⁱⁱ, so the process to retrieve responsive documents may become more complicated in situations where the search term includes either of these features. For example, a sentence such as "Mary had a little lamb" will not exist as a single index entry but is broken up into the separate words requiring the user to find them individually or else put together expressions, for instance by seeking for the word "Mary" within so many words of "lamb". Tools may also place limitations on the length of the words they process to manage the size of the index. Typically, a lower limit of 4 characters is imposed, but this excludes many words that can place a sentence in context. With upper limits set to some arbitrary level, key terms might be excluded such as foreign names, chemical and drug designators. In addition, some words that in English are considered 'noise' words and are therefore excluded from the index may be 'significant' words in another language.

SPECIFIC ISSUES FOR EDISCOVERY

eDiscovery tools typically connect to the 'live' data stored on devices accessible via a network, then create a central index to identify where the data of interest resides. This could be carried out manually with a digital forensics tool, but in an eDiscovery context there are likely to be large numbers of custodians and the requirement for a forensic practitioner to physically set up every instance of data indexing is impractical at the outset, both from the perspective of the amount of time it would take and the logistics of managing a significant number of separate collections.

Many of the leading eDiscovery tools came from existing digital forensics tools that were modified. For instance, Guidance Software added 'agents' to their forensic tool. These agents are small applications that have to be installed across networked systems and that serve as an interface between the central indexing machine and the disks attached to individual computers on a network. These agents also provide a connection to a management system that controls multiple indexing and collection processes. This concept of using agents has been replicated by other developers.

The idea of having agents installed on custodian computers that generate an index and collect data to a central secure point seems logical. However, in practice the fundamental flaws with this approach have become apparent. Notwithstanding all the limitations of an index approach mentioned earlier, the administrators of networks are now reluctant to adopt a process that requires them to install software across these networks, especially when they know that they will cause a large volume of traffic to be generated and could potentially interfere with normal business operations. In addition, given the pressures placed on litigants to meet strict court deadlines, the time required to create an index of data across a large number of systems has become a significant issue for in-house legal teams.

HYBRID FORENSICS APPROACH

The technology exists to overcome the difficulties discussed in previous sections. Rather than imposing restrictions and limits on the search and collection process, it is possible to provide more functionality with greater speed and greatly reduced processing costs.

Traditional approaches to both digital forensics and eDiscovery have focused on a central processing point and have also relied heavily on indexing. With advances in virtualization technology it has been possible to develop an application that runs inside its own virtual environment situated entirely in memory^{iv}. This enables the

application to be distributed across an unlimited number of target machines for true parallel processing as each instance is self-contained with no central dependencies. Another advance in technology has provided the ability to 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 ability to process email on the custodian machine is a significant benefit given the key role email now plays in litigation.

Combining these two developments provides the ability to carry out parallel processing across a large domain while significantly reducing the volume of data being transported across the network compared to that involved in the 'remote agent and indexing' model.

The Hybrid Forensics approach combines the concept of remote collection of data from multiple sources concurrently (as in eDiscovery) with the collection of the types of data that are generally only important in a digital forensics investigation, e.g. registry information. The key to implementing the Hybrid Forensics approach is an independent collection tool with the ability to undertake literal string searches at a disk level (rather than an operating system level) with the code running entirely in memory on each custodian. This provides five significant benefits:

1. Deployment is fast, easy and doesn't require the participation of custodians
2. Only responsive data is ever moved across the network, thus greatly reducing the impact on the host organisation
3. The search process is much more effective and will find responsive material missed by the index approach
4. The speed of collection is greatly increased as all processing and collection is carried out in parallel rather than individually or in small batches
5. Remote collections on non-networked machines becomes possible.

THE HYBRID FORENSICS APPROACH APPLIED TO DIGITAL FORENSICS

The Hybrid Forensics approach directly addresses three key problems: that of dealing with large data storage devices, acquiring data from multiple systems concurrently and remotely acquiring data with minimum resources at the endpoint.

By design, a bit-by-bit digital forensic image captures the entire contents of a data storage device including deleted and unused space. While it is possible to compress these data, the image is still likely to be too large to be transmitted across a network, especially if more than one image is involved or the data are from a file server or NAS device. The process also requires either that the device is removed from the source computer or that the computer is booted into a digital forensic environment to create the image. Both processes require the hands-on involvement of a digital forensic practitioner.

In some cases, the data of interest can be obtained from a selection of well-defined data types depending on the nature of the investigation, whether these are email (both application-specific and webmail), user files, certain system files (including the registry on Microsoft Windows machines) or deleted files. Hybrid Forensics caters for the collection of all these artefacts. The collected data can be sent to an encrypted container on a device physically attached to the target system while it is still in use. Alternatively, the data can be sent to an encrypted container located on a network share or even to the cloud.

The Hybrid Forensics process can be repeated across as many systems as necessary. These processes run in parallel utilizing the resources of the host systems. Remote collections can be undertaken by:

1. using a deployment agent such as EasyDeploy to run PSEXEC instances on networked systems that will load and execute the hybrid tool
2. sending a disk containing the hybrid tool plus its configuration file to one or more users at the remote site where it can be replicated and deployed as necessary by a system administrator or consultant with the appropriate access. The data will be sent to a specified target location.
3. sending the hybrid tool plus its configuration file to a system administrator at the remote site who can deploy the tool from a network share and login script or by using PSEXEC.
4. a system administrator using an RDP session to connect to the remote systems to deploy and run the tool manually.
5. sending a webpage link to selected users for them to download the executable and config files together with instructions for running the tool.

THE HYBRID FORENSICS APPROACH IN PRACTICE

Following the award of a patent for the Hybrid Forensics process^v an application and its associated configuration and extraction tools have been developed. The suite of tools includes ISEEK-Designer (which creates an encrypted configuration file containing the search/collection containers) and ISEEK-Explorer (which opens the encrypted containers in which are stored the audit results and collected data for viewing and further processing). The deployed search and collection tool itself has been named ISEEK.

The collection process for digital forensic acquisition and that for electronic discovery now appear very similar because with the new methodology the only difference is how the search and collection tool is configured. The key differences between the two applications are:

- For digital forensics, the collected artefacts tend to be complete directories, system files and entire email containers. For eDiscovery, only a limited number of specific files or emails will be collected.
- For digital forensics deleted files are likely to be recovered; these are rarely required for eDiscovery.
- For digital forensics, there will always be data collected from each system, whereas for eDiscovery there may be no items meeting the conditions for collection.

In both digital forensics and eDiscovery collections the configuration files, collected data and logs are encrypted so no aspect of the process is revealed to any unauthorised person who may come into possession of these files.

The ISEEK tool has already been deployed in several instances. In one case, several server farms were searched for data relating to a significant lawsuit in the United States. The entire process was completed with 5 hours whereas a previous attempt using conventional tools was cancelled after several days with no outcome. This case involved searching for terms that were unsuitable for an indexing approach as they included several foreign language terms (including Japanese) coupled with strings of characters that would typically be excluded in an index.

Another case involved a subpoena relating to the emails of 17 bank employees. Following estimates of 3 months to complete the work of identifying relevant emails across approximately 4 TB of data using the existing technology, ISEEK was deployed by two bank employees and within 48 hours they had collected and processed 27,000 relevant emails.

ISEEK is currently being deployed by a large US government contractor, a US military defence agency and a multi-national aerospace company.

TESTING

An experiment to demonstrate the effectiveness of the new process and technology was carried out using 9 custodian systems running a combination of Windows 10 Pro, Windows 10 Enterprise, Windows Server 2012 and Windows Server 2016 in a Windows domain.

Using the deployment utility, nine instances of ISEEK were started on the custodian systems in 48 seconds. ISEEK was configured to locate and collect (to a network share) files and emails containing two search terms that were in the "c:\users" path. The terms were: "Fuld & Company" and "489,628 Dth/d". Both terms are contained in files and attachments to emails within the Enron email data set.

Each custodian system had a mixture of large and small files of various types, including PST, ZIP and HTML. Three of the custodian systems were 'seeded' with a PST from the Enron email data set. Each target system had either 2GB RAM (workstations) or 4GB RAM (servers).

The results of the test are shown in Table 3.

3. Results of ISEEK deployment to nine custodian systems in a Windows domain

Machine	Searched data	Responsive data	Number of files searched	Responsive Files	Responsive Emails	Time to complete
WS-1	527 MB	0	3,578	0	0	00:00:45
TRID	14 GB	0	3,772	0	0	00:02:53
WIN-2	9 GB	22 MB	84,259	57	10	00:03:21
WIN-1	13 GB	45 MB	239,017	114	20	00:03:53
ROD	25 GB	0	3,996	0	0	00:05:06
DESK-5	15 GB	0	411,187	0	0	00:08:28
TIG	21 GB	0	6,372	0	0	00:10:18
XF	33 GB	0	9,345	0	0	00:16:38
DESK-4	26 GB	12 MB	548,780	20	4	00:26:12

For comparison purposes, a digital forensics tool that employs an index engine was used to create an index of the same data searched by ISEEK on the custodian system WIN-1. From Table 3 the entire process took just under 4 mins for ISEEK to complete. However, it took 51 mins for the forensics tool to index the same data on the remote system from an i7 8-core system with 12.5GB RAM and creating the index on a local solid-state drive.

In addition, having created an index, the forensics tool was unable to locate the search terms in the same form as that provided to ISEEK, which had completed the whole process on all nine custodian systems in under 30 minutes (with two of those systems containing responsive items processed in under 4 minutes).

Network utilisation peaked at 32Mbps during the process (which included the RDP traffic for monitoring the activities).

Further development is underway to create an integrated deployment application and refine the configuration options by grouping some of them under specific headings, such as the creation of a Forensics tab.

For eDiscovery scenarios, a bulk extraction utility creating XML metadata output together with the collected files for ingesting into a review platform is being refined as well as an API allowing direct access to the encrypted containers for a review platform. A pilot project has already been successful involving the direct import of ISEEK data into the Ringtail review platform.

CONCLUSION

ISEEK has been developed to the stage where it has been used in various environments. The virtualization technology employed has opened the way for the development of further uses with ISEEK, such as processing Windows registry hives for artifacts relevant to security and malware investigations. Conversations with large consulting firms have also identified a potential role for ISEEK in IT compliance engagements, ranging from simply checking licence details of installed software to identifying the presence of confidential documents being stored outside of authorized locations.

Users have identified the key benefits of the Hybrid Technology used in ISEEK as being that:

- the tool does not need to be installed
- the tool does not impact the network infrastructure
- 'live' email can be searched and collected without requiring the users to stop working
- the tool can run without the need for any user assistance (or knowledge of the process)
- the process of search and collection is much faster than using alternative methods.

REFERENCES

- 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): 25-48. doi: <https://doi.org/10.15394/jdfsl.2013.1154>.
- Franke, K. & Årnes, A. (2017). Challenges in digital forensics. In A. Årnes (Ed.), *Digital Forensics* (pp. 313-317). Chichester, England: John Wiley & Sons.
- Jusas, V., Birvinskas, D., & Gahramanov, E. (2017). Methods and tools of digital triage in forensic context: Survey and future directions. *Symmetry*, 9(4), 49. doi:10.3390/sym9040049
- Sondhi, S., & Arora, R. (2014). *Applying lessons from e-Discovery to process Big Data using HPC*. Paper presented at the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment, Atlanta, GA, USA. July 13-18, 2014. New York, NY, USA: ACM.
- Sondhi, S., & Arora, R. (2016). Big Data processing in the eDiscovery domain. In R. Arora (Ed.), *Conquering big data with high performance computing* (pp. 287-307). Cham: Springer International Publishing.
- Stanfield, A. (2009). *Computer forensics, electronic discovery and electronic evidence*. London, England: Reed International Books.
- Steel, C (2006). *Windows forensics: The field guide for conducting corporate computer investigations*. Indianapolis, IN: Wiley Publishing.
- 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>

ⁱ <https://www.legaltechnology.com/wp-content/uploads/2013/02/Corporate-Litigation-and-eDisclosure-Current-Trends-and-Future-Challenges.pdf>.

ⁱⁱ Available at <https://www.slideshare.net/RichardAdams3/forensic-imaging-tools-draft-v1-24228558>

ⁱⁱⁱ https://help.kcure.com/9.2/Content/Recipes/Searching__Filtering__and_Sorting/Using_Stop_Words_and_Making_Some_Characters_Searchable_in_a_dtSearch.htm

^{iv} For further information on the technology visit <https://turbo.net/>

^v <http://www.google.com/patents/US8392706>

ANALYSIS OF ATTEMPTED INTRUSIONS: INTELLIGENCE GATHERED FROM SSH HONEYPOTS

Priya Rabadia, Craig Valli, Ahmed Ibrahim, Zubair Baig
Security Research Institute, Edith Cowan University, Perth, Western Australia
p.rabadia@ecu.edu.au, c.valli@ecu.edu.au, ahmed.ibrahim@ecu.edu.au, z.baig@ecu.edu.au

Abstract

Honeypots are a defensive cyber security countermeasure used to gather data on intruder activities. By analysing the data collected by honeypots, mitigation strategies for cyberattacks launched against cyber-enabled infrastructures can be developed. In this paper, intelligence gathered from six Secure Shell (SSH) honeypots is presented. The paper is part of an ongoing investigation into analysing malicious activities captured by the honeypots. This paper focuses on the time of day attempted intrusions have occurred. The honeypot data has been gathered from 18th July 2012 until 13th January 2016; a period of 1,247 days. All six honeypots have the same hardware and software configurations, located on the same IPv4/24 subnet. Preliminary analysis of the data from all six hosts has been combined to show the number of attempted intrusions recorded by each honeypot and the top 20 countries attacking IP addresses have originated from. However, there is a variation in the number of attempted intrusions recorded on each of the six hosts. Findings from the research conducted suggest, there is a pattern of organised attempted intrusions from attacking IP addresses originating from China and Hong Kong during an 8am to 6pm working day. An additional investigation into the possible use of organised attacking workforces was conducted.

Keywords: Cybersecurity, SSH, Secure Shell, Honeypots

INTRODUCTION

Honeypots are decoy systems used to gather data on attempts made to gain unauthorised access to IT systems. There are three main types of honeypots that can be deployed. Firstly, a low-interaction honeypot is a system with minimal functionality and interaction with the actual honeypot. The configuration process is simple, with minimal maintenance required to sustain the honeypot. Secondly, a high-interaction honeypot. This system emulates a fully functional ‘real’ system; with an extensive configuration process. Due to the configuration of the system the maintenance and interaction required is demanding (Zemene & Avadhani, 2015). The final type of honeypot is a medium-interaction honeypot; it emulates some functionalities of a ‘real’ system. The configuration process is simpler than a high-interaction honeypot but the maintenance required is more demanding than a low-interaction honeypot (Zemene & Avadhani, 2015). An example of a medium-interaction honeypot is a Secure Shell (SSH) honeypot named Kippo (Desaster, 2013).

Kippo SSH is an application specific honeypot that imitates some functions that are exhibited by a ‘real’ SSH system to the attacker. SSH is designed to securely transmit data using a point to point encrypted tunnel. Kippo honeypots are designed to collect various data from attacks propagated against the SSH service (Rabadia & Valli, 2014). An open-source, python 2.7 based event-driven program called Twisted libraries (TwistedMatrixLabs, 2013) is deployed by the Kippo honeypot to imitate and project a legitimate SSH session to the attacker. Data for this study has been acquired from identically configured Kippo SSH honeypots, using Ubuntu 11 Long Term Support (LTS) servers as their base operating system. All the honeypots were located on inexpensive Virtual Private Servers (VPS) The six honeypots are referred to as: Bobtail, Bronx, Dugite, Goanna, Magpie and Mopoke. Three of the honeypots were based in the United States (Bobtail, Magpie and Mopoke) with the other three located in the Netherlands (Bronx, Dugite and Goanna.)

Preliminary analysis was conducted on the combined data from all six honeypots: the number of attempted intrusions recorded by each honeypot and the top 20 countries attacking Internet Protocol (IP) addresses have originated from. The focus of this research was on the time of day (24-hours) attempted intrusions had occurred. This paper is part of an ongoing investigation into data collected from Kippo SSH honeypots, with work conducted over the past five years 2012, 2013, 2014 and 2015 (Rabadia & Valli, 2014; Valli, 2012; Valli, Rabadia, & Woodward, 2013, 2015). An attempted intrusion is an unauthorised attempt to gain access or control of a honeypot.

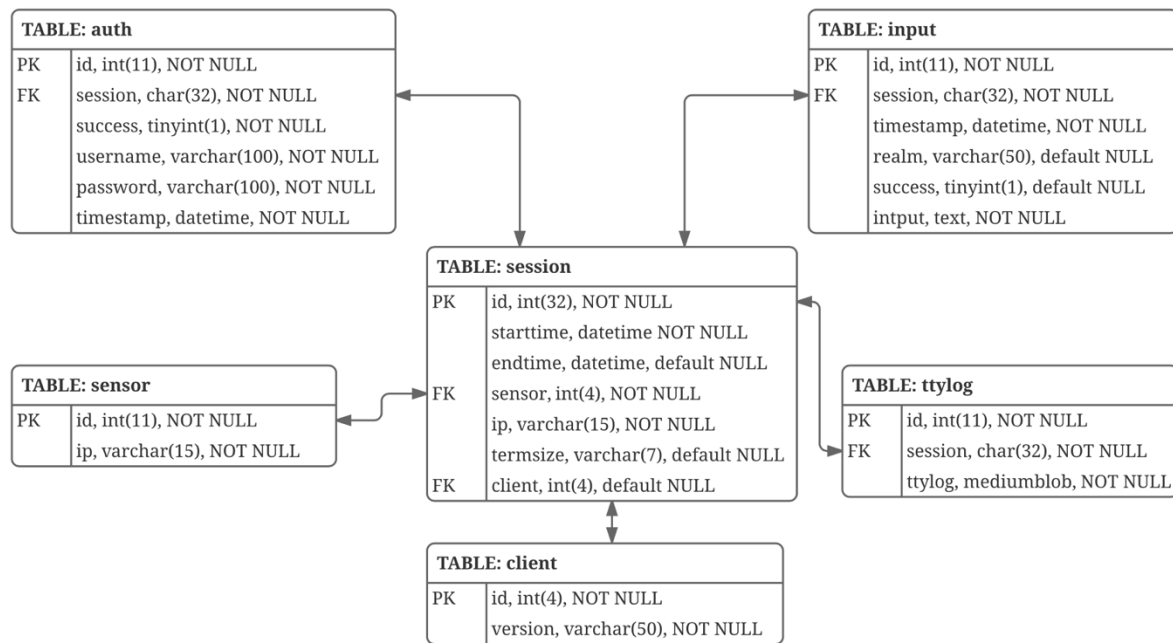


Figure 1 - MySQL database structure for Kippo honeypot, adapted from Valli, (2012)

Overview of the Honeypot setup

Kippo honeypots are designed to collect data from attacker interaction with the emulated SSH service. The emulated SSH service is provided by an open-source, Python based event-driven program called Twisted (BruteforceLabs, 2011; TwistedMatrixLabs, 2013). Twisted provides the libraries that are utilised and deployed by Kippo honeypot to imitate a valid “encrypted” SSH session to an entity (BruteforceLabs, 2011). The Kippo honeypot also emulates a fake file system to present to the attacker, along with, false system reporting Kippo allows interaction with artefacts such as `/proc/cpuinfo` or `.bash_history` log file. The level of deception in the default setting is limited however, this functionality however, is able to be expanded and modified at will. For this experiment key elements were modified such as `/proc` entries and different bash file entries to deceive attackers.

The Kippo SSH honeypots are written in Python with a simple installation process. Source code was obtained from the kippo.googlecode.com Wiki (Code.google.com, 2012). The setup for these particular systems used in the data collection was conducted as specified by the Bruteforce Lab Guide (BruteforceLabs, 2011). This deviates from the original Kippo SSH documentation and uses the authbind daemon instead of `twistd` as the initial connecting daemon for the service. The configuration lets authbind handle the binding of the `twistd` as a non-root user to a low numbered TCP port and passes this to the Kippo daemon. This configuration has been found to be more consistent, reliable and secure during the conduct of the research project.

During the installation process a local MySQL database was configured securely to record all the interactions with the Kippo honeypots. Figure 1, reproduced from Valli (2012), shows the MySQL database structure used in the Kippo honeypots to record all the interactions.

After recording to the local MySQL database, these data were then transmitted to a centralised PostgreSQL SQL server that was running a Debian-Linux operating system (Valli et al., 2013). Communication is achieved using a Python extension that uses a PostgreSQL driver to connect to the SURFIDS system IDS logging server (IDS, 2013; SURFcrtIDS, 2013). The centralised logging server utilises the SURFIDS system for storing the data from the honeypots into an aggregated PostgreSQL database. The database has functions and tables specifically for the Kippo honeypots data. Since 2013 however, Elasticsearch has also been used to consume data from the honeypots in addition to recording with SQL.

In addition, on the honeypots that run Kippo the researchers also operate Dionaea (TheHoneyNetProject, 2011) and Glastopf (MushMushFoundation, 2011) which also report to the database instances, however these data are not used in this analysis for this research.

PRELIMINARY ANALYSIS

Analysis conducted on the total number of attempted intrusions on each of the six honeypots and the top 20 countries attacking IP addresses have originated from is presented. An attempted intrusion is an unauthorised attempt to gain access or control of a honeypot. The data used in this study was collated from the sessions tables of the honeypot dataset, shown in Figure 1. The data used in this investigation has been collected from 18th July 2012 until 13th January 2016; this is a period of 1,247 days. Table 1 shows the number of attempted intrusions and the geolocations of the honeypots. A total of 5,554,680 attempted intrusions have been recorded, with the host known as Mopoke recording the most at 1,399,203 attacks. Host Goanna has received the least number of attacks at 520,250 as shown in Table 1.

Honeypot	Attempted Intrusions	Geolocation
Bobtail	934,966	United States
Bronx	1,319,381	Netherlands
Dugite	1,064,723	Netherlands
Goanna	520,250	Netherlands
Maggie	316,157	United States
Mopoke	1,399,203	United States
Total	5,554,680	

Table 1: Number of attempted intrusions and the geolocation of the honeypots.

Using the combined data from all six honeypots, the country of each attacking IP address session had been compiled using IP to ANS Mapping (TeamCymru, 2016) an open-source application that uses netcat. The top 20 countries attacking origin IPs was compiled and represented in Table 2. From 5,554,680 total attempted intrusions recorded; 31,596 unique attacking IP addresses have been identified attempting to gain unauthorized access to the six honeypots. Each host records the geolocation of the connections origin; however, this may not indicate the actual origin of the intruder attacking the hosts if the attacker is using a proxy. Data gathered from all six honeypots show ~64.47% of the recorded attacks originate from Chinese IP addresses followed by Romanian, Hong Kong, United States and France respectively as represented in Table 2.

	Countries	Attempted Intrusions
1	China	3,581,114
2	Romania	708,793
3	Hong Kong	427,209
4	United States	360,606
5	France	99,161
6	South Korea	90,983
7	Germany	34,253
8	Taiwan	19,906
9	Brazil	19,260
10	Ukraine	19,113
11	Russia	18,644
12	India	14,930
13	Indonesia	14,925
14	Czech Republic	13,999
15	Turkey	13,904
16	Japan	12,488
17	United Kingdom	12,066
18	Mexico	11,897
19	Canada	10,975
20	Italy	8,961

Table 2: Top 20 originating countries of attacking IP addresses

INTELLIGENCE GATHERED

In this section, the time of day attempted intrusions occurred for each of the six honeypots is investigated, and the combined dataset is presented. RStudio Version 1.0.136 (RStudio, 2016) was used to examine the honeypot datasets. The start time of the attacking IP sessions was used as opposed to the end time, as the research investigates the number of attempted intrusions occurring during a 24-hour day. Next, the date part of the timestamps was stripped, leaving only the time segment. Each time segment was rounded to the nearest 15-minute interval. As hourly analysis would result in 24 data points and analysis conducted using 15 minute intervals resulted in 96 data points utilised. Allowing for a substantial data trend to be analysed, furthermore a reduced time interval could result in the data trend being hidden. The frequency of each 15-minute interval was compiled and depicted in Figure 2 to Figure 7. Once individual graphs for all six honeypots had been generated. The data from all six honeypots were combined and a graph was generated as shown in Figure 8.

Records of the interaction for each honeypot were stored in a combined MySQL database, the structure of which is shown in Figure 1. As the nature of the research requires an analysis of the timestamps; maintaining the timestamp throughout the data collection and data analysis phases of the research was essential for the purpose of validity. To ensure the integrity of the timestamp data recorded on all six hosts are consistent, the Network Time Protocol (NTP) is used to sync all six hosts to GMT +8 time.

Attempted Intrusion Detected on the Honeypots

Figure 2 shows the time of day (24-hours format) using GMT+8 time as datum, for attempted intrusions that had occurred for the host known as Bobtail located in the United States. The highest recorded number of hits is at 04:15 with 11,495 attempted intrusions and the lowest number of hits recorded is at 14:45 with 9,122 hits. There is a significant decline in the number of attempted intrusions occurring after 12:00 until 00:15, with hits under 9,500. Whereas between 00:15 to 11:45 the number of intrusion attempts are above 10,000. This steep decline is clearly shown in Figure 2.

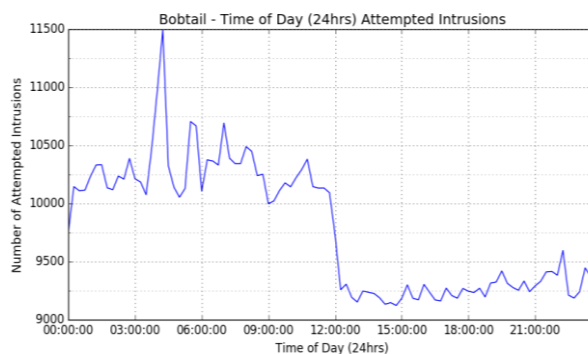


Figure 2: Time of day (24-hours) attempted intrusions occurred, for Bobtail.

Unlike Bobtail, the host known as Bronx shown in Figure 3, does not have a steep decline in the number of attempted intrusions recorded. Instead Bronx has a steady decline in the number of attempted intrusions throughout the day (24-hours). The highest recorded number of attempted intrusions is at 8:00 with 14,752 recorded and the lowest recorded at 21:45 with 12,899 hits. There is a steady decline in the number of hits recorded after 08:00 until 22:00, where the number attempted intrusions steadily increase again.

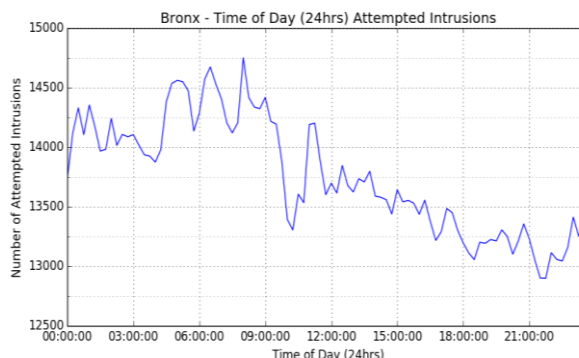


Figure 3: Time of day (24 hours) attempted intrusions occurred, for Bronx.

Similar to Bobtail (Figure 2), Dugite has a steep decline (Figure 4) in the number of attempted intrusions. The highest recorded attack is 12,222 at 7:30 and 14:00. With the lowest recorded hit at 17:00 with 10,186 attempted intrusions recorded. The steep decline in the number of attempted intrusions were recorded between 14:30 and 00:00, with around 10,500 and below. Whereas between 00:15 and 14:30 the number of hits were above 11,000. This is similar to the findings in from Bobtail.

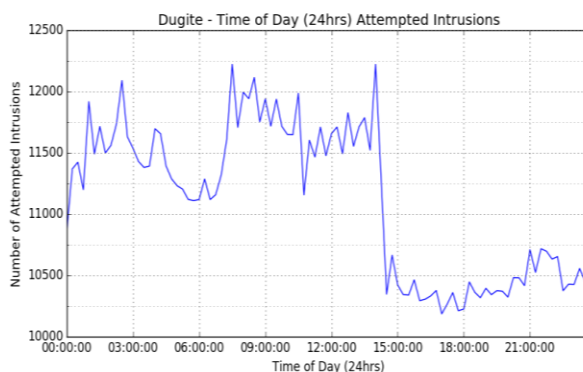


Figure 4: Time of day (24 hours) attempted intrusions occurred, for Dugite.

The host known as Goanna shown in Figure 5 has a steep decline in the number attempted intrusions similar to Figure 2 and Figure 4. The decline in the number of attempted intrusions starts 11:15 and continuous until 00:00. The number of attempted intrusions during this time are under 3,000. Most attempts occur between 00:15 and 11:00 with over 3,600 recorded attempts. The highest recorded number of attempted intrusions is 4,050 at 02:00, and the lowest number recorded is 2,718 at 17:15.

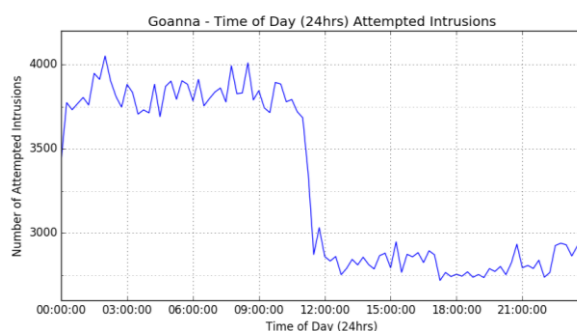


Figure 5: Time of day (24-hours) attempted intrusions occurred, for honeypot known as Goanna.

Figure 6 is from the Magpie host, it is similar to Bobtail, Dugite and Goanna as there is a steep decline in the number of attempted intrusions recorded. However, the number of attempted intrusions is mostly consistent throughout the day at around 5,500. There is a short decline from 20:45 until 23:45 with 4,700 recorded. The highest recorded attempted intrusions are at 8:45 with 6,371 hits recorded and the lowest at 23:00 with 4,529 hits recorded.

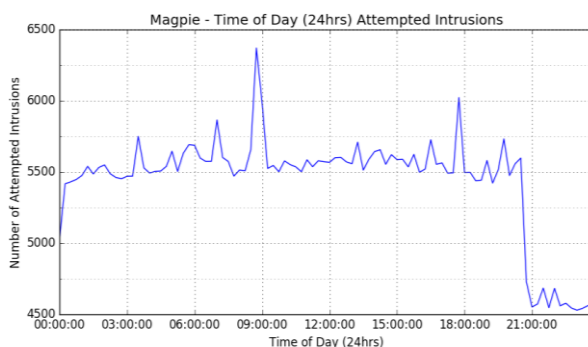


Figure 6: Time of day (24 hours) attempted intrusions occurred, for Magpie.

Host Mopoke shown in Figure 7, has the highest recorded number of attempted intrusions with 15,625 at 07:00, and the lowest number of hits recorded with 13,714 at 21:45. Mopoke, has similarities to Magpie, as there is a steep decline in attempted intrusions for a short time period compared to the other honeypots. The decline in the number of attempted intrusions recorded is from 19:30 until 00:00, with majority of the hits recorded below 14,000. While between 00:15 and 19:15 the number of hits are recorded above 14,500, showing Mopoke and Magpie had similar trends in the number of attempted intrusions.

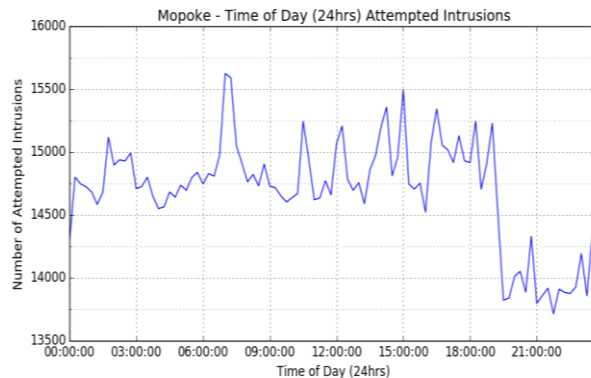


Figure 7: Time of day (24 hours) attempted intrusions occurred, for Mopoke.

Figure 8 is the combined data from all six honeypots; the mean line shows there is a steady decline in the number of attempted intrusions recorded. Whereas, the individual hosts aside from Bronx had a clear decline in the number of attempted intrusions recorded throughout the day (24-hours). The highest recorded number is at 8:00 with 60,219 and the lowest at 14:00 with 56,188 recorded. The mean line illustrates the number of attempted intrusions recorded increase during the hours between 00:15 and 09:00 and thereafter decreased.

Summary

Findings from the data above suggest the number of attempted intrusions fluctuates as the day progresses. With Bobtail, Dugite and Goanna having similarities in a steep decline in the number of attacks from use times only. Whereas, Magpie and Mopoke only had a steep decline in the number of hits for a short period of time. Bronx and the mean from combined dataset show a steady decline in the number of attempted intrusions throughout the day.

Table 3 summaries the findings from all six honeypots including the combined dataset. As shown in Table 3, all means between 00:00-11:59 are higher than the means between 12:00-23:59 and the mean hits for a day. Additionally, all the peak numbers have occurred early in the timeline, while the minimum numbers have occurred later in the timeline.

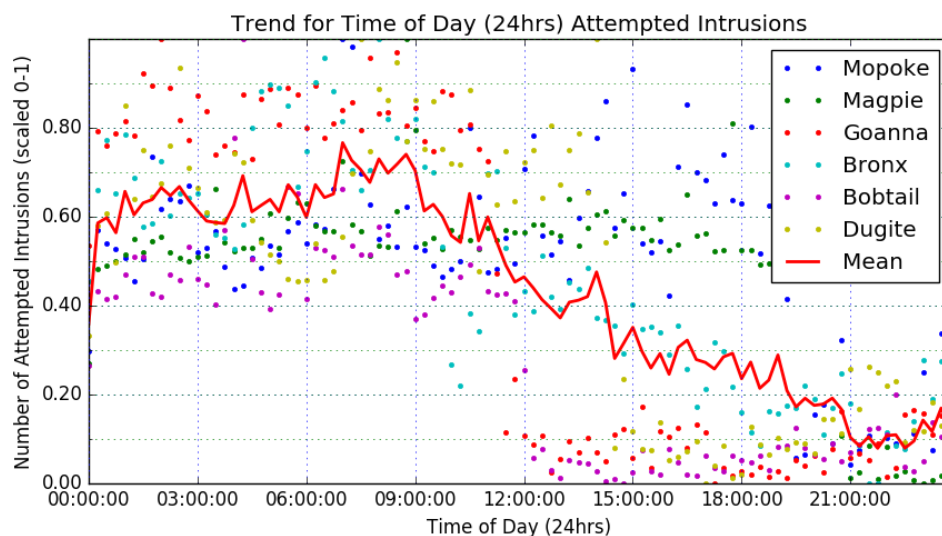


Figure 8: Time of day (24 hours) attempted intrusions occurred, for the combined hosts.

Table 3: Summary of the findings

Honeypot	Geolocation	Mean hits for a day	Mean between 00:00-11:59	Mean between 12:00-23:59	Peak Time	Peak Hits	Minimum Time	Minimum Hits
Bobtail	United States	9,780	10,287.29	9,272.71	04:15	11,495	14:45	9,122
Bronx	Netherlands	13,756.18	14,145.42	13,366.94	08:00	14,752	21:45	12,899
Dugite	Netherlands	11,115.53	11,547.75	10,683.31	07:30	12,222	17:00	10,186
Goanna	Netherlands	3,295.292	3,768.25	2,822.33	02:00	4,050	17:15	2,718
Magpie	United States	5,435.98	5,563.583	5,308.38	08:45	6,371	23:00	4,529
Mopoke	United States	14,689.52	14,800.25	14,578.79	07:00	15,625	21:45	13,714
Combined		58,072.5	58,407.23	57,737.77	08:00	60,219	14:00	56,188

Table 4: Summary of honeypot datasets excluding attacking IP addresses from China and Hong Kong

Honeypot	Geolocation	Total number of hits	Mean hits for a day	Mean between 00:00-11:59	Mean between 12:00-23:59	Peak Time	Peak Hits	Minimum Time	Minimum Hits
Bobtail	United States	341,777	3,560.18	3,502.94	3,617.42	11:45	5,521	05:15	1,962
Bronx	Netherlands	343,386	3,576.94	3,689.42	3,464.46	02:00	5,347	06:00	2,172
Dugite	Netherlands	205,860	2,144.38	2,178.15	2,110.6	11:45	3,325	07:45	283
Goanna	Netherlands	15,402	160.44	166.73	154.15	08:45	845	09:45	0
Magpie	United States	197,113	2,053.26	1,729.4	2,377.13	20:00	3,917	12:45	240
Mopoke	United States	463,093	4,823.89	4,535.54	5,143.76	12:00	8,120	07:15	2,812
Combined		1,566,631	16,319.07	16,111.29	16,526.85	23:45	18,858	01:30	12,485

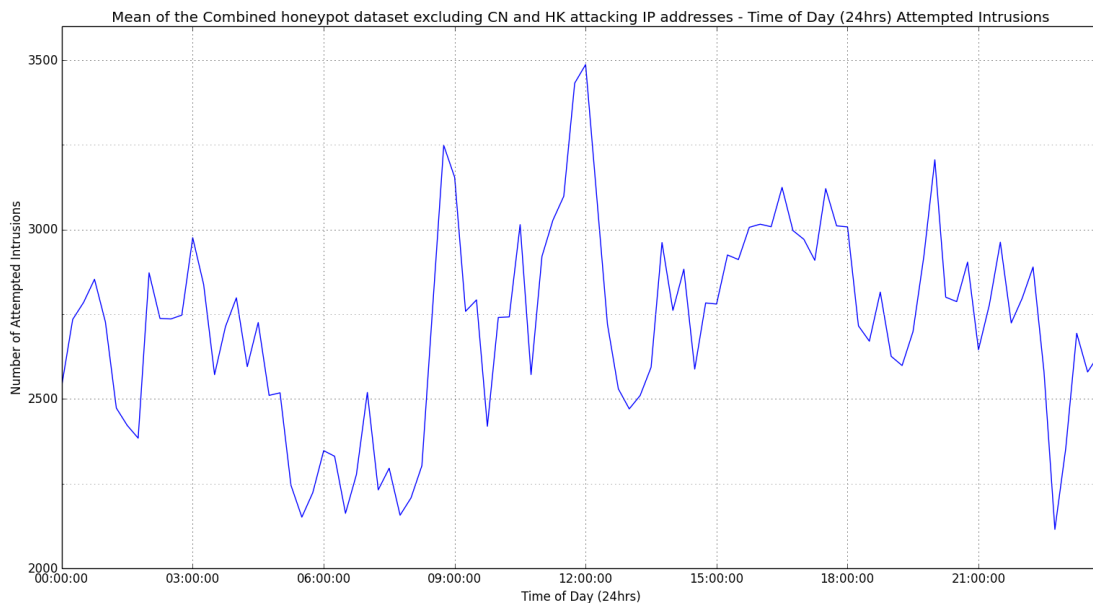


Figure 9: Combined dataset of mean time of day attempted intrusions have occurred excluding attacking IP addresses originating from China and Hong Kong.

DISCUSSION

Preliminary analysis was conducted on the data collected from all six honeypots; the number of attempted intrusions recorded by each of the honeypots was presented. Mopoke recorded the most activity at 1,399,203. Among the top 20 countries where attacks originated (Table 2), IP address from China comprised about 65% of the attacks.

The mean from the complete datasets for all the honeypots show the mean between 00:00-11:59 is higher than the mean between 12:00-23:59. From the findings it is suggested that the geolocation of the honeypots may not

determine when attempted intrusions are carried out, as Bobtail (United States), Dugite (Netherlands) and Goanna (Netherlands) showed similar activities. However, Magpie and Mopoke were both located in the United States and similar activities had been identified. This mean indicates the global or distributed nature of these observed attacks.

The observation made from the data represented in the previous section suggest the number of attempted intrusion decrease as the day progresses. The honeypot datum is GMT+8 time. As shown in Table 2 attacking IP address originating from China and Hong Kong are first and third respectively. To verify whether there is a linkage between the decline in activity as the day progresses and attacking IP addresses originating from China and Hong Kong additional experiments had been conducted. IP addresses from China and Hong Kong had been removed from the individual honeypot datasets as well as the combined dataset.

Table 4 represents data from the honeypot datasets with attacking IP addresses originating from China and Hong Kong excluded. The honeypots located in the United States and the combined dataset have a higher mean for the hours between 12:00-23:59 as opposed to the mean for the hours between 00:00-11:59 and the mean for the whole day, unlike Table 3. Whereas, the honeypots located in the Netherlands have a higher mean for the hours between 00:00-11:59. The peaks are spread over the day, Table 3 shows peak occurring between 02:00-08:45, while Table 4 shows times between 02:00-20:00 and 02:00-23:45 when including the combined dataset. In Table 3, all minimum hits have occurred between 12:00-23:59 but Table 4 shows all minimum hits have occurred between 00:00-11:59 aside from Magpie with the minimum number of hits occurring at 12:45, suggesting the use of organised attacks.

Table 5: Summary of honeypot datasets from attacking IP addresses from China

Honeypot	Geolocation	Mean hits for a day	Mean between 00:00-07:59 & 17:00-23:59	Mean between 08:00-16:59	Mean between 00:00-08:29 & 17:30-23:59	Mean between 08:30-17:29	Mean between 00:00-08:59 & 18:00-23:59	Mean between 09:00-17:59	Mean between 00:00-07:59 & 18:00-23:59	Mean between 08:00-17:59
Bobtail	United States	5,731.99	5,819.28	5,586.5	5,882.92	5,480.44	5,898.52	5,454.44	5,879.93	5,524.88
Bronx	Netherlands	9,225.43	8,900.83	9,766.42	8,905.15	9,759.22	8,896.6	9,773.47	8,853.48	9,746.15
Dugite	Netherlands	7,733.6	7,745.08	7,714.44	7,823.62	7,583.56	7,880.12	7,489.39	7,831.86	7,596.03
Goanna	Netherlands	3,295.29	3,346.82	3,209.42	3,381.3	3,151.94	3,419.5	3,088.28	3,387.77	3,165.83
Magpie	United States	2,339.97	2,205.67	2,563.81	2,242.12	2,503.06	2,253.97	2,483.31	2,214.82	2,515.18
Mopoke	United States	9,137.96	9,239.28	8,969.08	9,243.05	8,962.81	9,236.08	8,974.42	9,241.25	8,993.35
Combined		37,303.27	37,256.13	37,381.83	37,243.62	37,402.69	37,235.85	37,415.64	36,961.95	37,781.13

The trends observed with the datasets including the attacking IP addresses from China and Hong Kong are not present when excluding these attacking IP addresses. Figure 9, depicts the mean for time of day and the number of attempted intrusions made for the combined dataset excluding attacking IP addresses originating from China and Hong Kong. Unlike figure 8, figure 9 does not have a distinct trend of a steady decline as the day progresses. There are many clear peaks and troughs throughout the day in figure 9 as opposed to figure 8. The trend of the number of attempted intrusions made decreases as the day progresses is present in the datasets including the Chinese and Hong Kong attacking IP addresses however it is not depicted in the datasets with these attacking IP addresses excluded.

Further investigation into the suggestion of organised attacker workforces being deployed by countries such as China has been conducted. The mean of the number of attempted intrusions occurring during the “average working day” ranging of hours between 08:00-17:59 were compared to the mean for the hours outside this range. By comparing the means of the number of attempted intrusions occurring between the “average working day” 08:00-17:59 and outside these specific hours of 00:00-07:59 then 18:00-23:59, it could be suggested organised attacks are being deployed within the working day. Table 5, shows the results from the analysis conducted. An “average working day” consisting of hours between 08:00-16:59, 08:30-17:29, 09:00-17:59 and 08:00-17:59. From the information shown in Table 5, only Bronx, Magpie and the combined dataset suggest presence of organised attacks being deployed. However, none of the means of the “average working day” for Bobtail, Dugite, Goanna and Mopoke are higher the means for the remainder of the day. Further investigation is needed

as Bronx and Magpie suggest the possible presence of organised attacks using workforce being deployed by countries such as China.

CONCLUSION

The focus of this research was analysing the time of day attempted intrusions have occurred, the datum for this was GMT +8 time. The findings from this research show, the complete datasets from all six honeypots have a decline in the number of attacks as the day progressed through GMT +8 time. Three trends had been identified:

- A steep decline in the number of hits in the afternoon with the lower number of hits sustained between 00:00-11:59 GMT +8 time.
- A steep decline in the number of attacks, the decline in activity was for a shorter time period and much later in the day.
- A steady decline throughout the day.

Investigating further into the trends identified, all attacking IP addresses originating from China and Hong Kong were excluded. The observed trends in the complete dataset were not present in the dataset with the excluded attacking IP addresses. The honeypots located in the United States have a higher mean for the hours between 12:00-23:59, whereas the honeypots located in the Netherlands have a higher mean between 00:00-11:59. Unlike the complete datasets, all honeypots including the combined dataset had higher mean hits recorded between the hours of 00:00-11:59. Further investigation is needed to determine the significance of geolocation in the attempted honeypot intrusions. Upon the findings suggesting there is a linkage between time of day and attacking IP addresses originating from China and Hong Kong, additional investigation was conducted into the use of organised attacking work forces by countries such as China. The mean for the “average working day” was compared to the mean of hours outside the specify time. Analysing the results shown in Table 5, Bronx, Magpie and the combined dataset suggested the uses of organised attacking workforces being deployed as all four shifts had a high mean then the remainder on the day. However, the observation was not present in the remaining honeypots. Further research needs to be conducted as Bronx, Magpie and the combined dataset suggest the use of organised attacks during a working day by counties such as China.

Future work

In future work, further analysis will be conducted on the Kippo SSH honeypot datasets. The time of day attacks have occurred for each of the top attacking countries including China and Hong Kong will be investigated which could yield an explanation in the decline of attacks as the day progresses. Also, data on the attacking IP addresses will be explored further, to identify if a distinct pattern for attacking IP addresses originating in the same geographical location is present. Analysis will also be conducted into suggestions of organised attacks during “business hours” workforces being deployed. Further investigation is needed to determine the significance of geolocation in attempted honeypot intrusions, this can be achieved by deploying honeypots in different geographical locations.

REFERENCES

- BruteforceLabs. (2011). Installing Kippo SSH Honeypot on Ubuntu. Retrieved from <http://bruteforce.gr/installing-kippo-ssh-honeypot-on-ubuntu.html>
- Code.google.com. (2012). Kippo shows up in Metasploit. *SSH Honeypot* Retrieved from <https://code.google.com/p/kippo/issues/detail?id=48>
- Desaster. (2013). Kippo - SSH Honeypot. Retrieved from <https://github.com/desaster/kippo>
- IDS, S. (2013). SURFcert IDS. Retrieved from <http://ids.surfnet.nl/wiki/doku.php>
- MushMushFoundation. (2011). Glastopf. Retrieved from <http://mushmush.org/>
- NetworkWorkingGroup. (2006). RFC: The Secure Shell (SSH) Transport Layer Protocol.
- Rabadia, P., & Valli, C. (2014). *Finding evidence of wordlists being deployed against SSH Honeypots - implications and impacts*. Paper presented at the 12th Australian Digital Forensics Conference, Perth, W.A.
- RStudio. (2015). RStudio. Retrieved from <https://www.rstudio.com/>
- SURFcertIDS. (2013). SURFcert IDS. Retrieved from <http://ids.surfnet.nl/wiki/doku.php>

- TeamCymru. (2016). IP TO ASN MAPPING. Retrieved from <http://www.team-cymru.org/IP-ASN-mapping.html>
- TheHoneynetProject. (2011). Dionaea - catches bugs. Retrieved from <https://www.honeynet.org/project/Dionaea>
- TwistedMatrixLabs. (2013). What is Twisted? Retrieved from <http://twistedmatrix.com/trac/>
- Valli, C. (2012). *SSH: Somewhat Secure Host*. Paper presented at the Cyberspace Safety and Security, Melbourne Australia.
- Valli, C., Rabadia, P., & Woodward, A. (2013). *Patterns and Patter - An Investigation into SSH Activity Using Kippo Honeypots*. Paper presented at the Australian Digital Forensics Conference, Edith Cowan University.
- Valli, C., Rabadia, P., & Woodward, A. (2015). *Profile of Prolonged, Persistent SSH Attack on a Kippo Based Honeynet*. Paper presented at the Conference on Digital Forensics, Security and Law, Virginia, US.
- Zemene, M. S., & Avadhani, P. S. (2015, 10-13 Aug. 2015). *Implementing high interaction honeypot to study SSH attacks*. Paper presented at the Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on.

BUILDING A DATASET FOR IMAGE STEGANOGRAPHY

Chris Woolley¹, Ahmed Ibrahim², Peter Hannay²

²Security Research Institute, ¹School of Science, Edith Cowan University, Perth, Western Australia
cwoolle1@our.ecu.edu.au, ahmed.ibrahim@ecu.edu.au, p.hannay@ecu.edu.au

Abstract

Image steganography and steganalysis techniques discussed in the literature rely on using a dataset(s) created based on cover images obtained from the public domain, through the acquisition of images from Internet sources, or manually. This issue often leads to challenges in validating, benchmarking, and reproducing reported techniques in a consistent manner. It is our view that the steganography/steganalysis research community would benefit from the availability of common datasets, thus promoting transparency and academic integrity. In this research, we have considered four aspects: image acquisition, pre-processing, steganographic techniques, and embedding rate in building a dataset for image steganography.

Keywords: Dataset, Image Steganography, Steganalysis, Embedding Rate

INTRODUCTION

Common datasets are widely used in different domains such as Image Processing, Artificial Intelligence, Cyber Security, etc. Some popular examples include FVC2002 (Maio, Maltoni, Cappelli, Wayman, & Jain, 2002) used in biometrics, PhysioBank Databases (Goldberger, et al., 2000) consisting of physiological datasets, and UCI KDD datasets (Hettich and Bay, 1999) used in data mining. Common datasets that are publicly available allow researchers to validate, benchmark, and reproduce techniques previously reported or proposed, thus promoting transparency and integrity of academic research.

We have carefully chosen the title of this paper as a “dataset for image steganography” in order to scope our work to only images. Thus, we would like to define a steganographic dataset as follows:

“A steganographic dataset is a collection of media objects consisting of cover objects and corresponding steganographic objects, with variations in parameters related to the objects and steganographic techniques used, applied consistently across all objects”.

In our definition, objects may include digital file types such as text, image, audio, and video. Therefore, this paper only focuses on image steganography dataset. Additional terms such as cover objects, steganographic objects, and relevant parameters are discussed in detail in later sections of this paper.

To the best of our knowledge, there are no such datasets for image steganography. Thus datasets used in existing literature have been constructed typically by the authors using public domain images. However, we argue that using public domain images introduces the risk of the integrity of such images being compromised, as absence of steganographic content is not known a priori.

The next section provides a detailed background on concepts that influenced our decisions in building the dataset. This section is followed by the method, which outlines the experimental design and further details on activities that were carried out in each step of the experiment. The paper is then concluded with final remarks and avenues for future direction.

BACKGROUND

Steganography first emerged as a form of covert communication thousands of years ago employing techniques such as using invisible ink to hide a message in plain sight or masking secret messages within inconspicuous text (Pieprzyk, Hardjono, & Seberry, 2003). Different forms of secret communication have evolved through the centuries into two specific sciences, cryptography and steganography. Cryptography deals with rendering a secret message unreadable to anyone other than the intended recipient. In contrast, steganography hides the existence of the secret message within cover objects such as text, image, audio, or video files. An unsuspecting party would not be aware of the presence of a secret message even if they came to possess the file. However, if the presence of

the secret message were discovered, then the goal of steganography would be defeated. This form of discovery, attack, or detection, is formally known as steganalysis.

This section informs the reader with background information about factors in the literature that influenced the criteria and methodology for creating the dataset for image steganography. We identified four key factors as: image acquisition, pre-processing, steganographic techniques, and embedding capacity.

Image Acquisition

Three aspects of image acquisition that stood out from the literature were its source, type, and quantity.

Image acquisition

Three sources were predominantly used throughout the literature were:

1. public domain image datasets,
2. downloading images from public domain Internet sources, and
3. manually acquiring images.

The image datasets available in the public domain cited in the literature were NRCS (Pevny et al., 2010; Huang et al., 2010; Ker, 2005), Greenspun (Aycibas et al., 2005; Farid, 2002; Fridrich, 2004; Lyu & Farid, 2002), and ImageNet (Zeng et al., 2016) Freefoto (Lyu & Farid, 2004) and the use of stock photo or clip-art compilations (Huang, et al., 2010). However, Ker (2004) purchased the rights to the galleries used within his research.

In order to download images from Internet sources, Kharrazi et al. (2006) used web crawlers to scrape publicly available images. Others used established steganographic sources including the BOWS2 (Pevny et al., 2010) and BOSSBase (Kodovsky & Fridrich, 2013) databases.

According to Holtyak et al., (2005) and Pevny et al. (2010), both the original source and quality of the images can affect the effectiveness of steganalysis. Thus, some researchers developed their own image datasets, which allowed them to have control over the images used for their specific needs, for example, using multiple image capture sources to more closely reflect real-world application (Holtyak et al., 2005; Huang et al., 2010; Pevny et al., 2010).

Types of Image

The types of images used can be classified as either *natural* or *unnatural* images. Farid and Lyu (2003) defined natural images as photographs collected during standard or digital photography, allowing them to mimic real-world application of steganography. Within the same paper, they defined unnatural images as artificial images that have been created digitally. Additionally, Fridrich and Goljan (2002) discussed the suitability of types of images and proposed that cover images with low colour counts or unnatural images (e.g. digital art) should be avoided. This view is seemingly supported as there is an identified gap in the literature for images of these types.

Number of images

The number of images used varies drastically between the types of research conducted, with most literature exploring steganalysis techniques using quantities between 1,000 and 10,000 cover images (Aycibas, Kharrazi, Memon, and Sankur, 2005; Farid, 2002; Farid and Lyu, 2003; Fridrich, 2004; Holtyak, Fridrich & Voloshynovskyy, 2005; Huang, Shi & Huang, 2010; Ker, 2004; Ker, 2005; Kodovsky & Fridrich, 2013; Kodovsky, Pevny & Fridrich, 2010; Lyu & Farid, 2002; Pevny, Bas & Fridrich, 2010; Solanki, Sarkar & Manjunath, 2007). However, Zeng et al. (2016) argued that these numbers were not sufficient to represent real-world application and used 14,000,000 unique images of varying complexity from the ImageNet database instead. Other large quantities used in literature range from 40,000 (Lyu & Farid, 2014) up to one million (Kharrazi, Sencar, & Memon, 2006) unique cover images.

Conversely, literature that explores early steganalysis techniques (Fridrich and Goljan, 2001; Fridrich, Goljan, and Du, 2001) or new steganographic models are shown to use smaller cover image sets comprising of 10 or less images (Solanki, Sarkar & Manjunath, 2007; Zhang, Jiang, Zha, Zhang, & Zhao, 2013)

While these examples show that the number of images used vary between experiments, it is important to note that the literature does not discuss how the number of cover images may impact operational or acquisition costs, or if these factors have influenced the author's decisions.

Pre-processing

Pre-processing is used during the development of each image dataset to ensure consistency in the characteristics of the images used. This section explores how JPEG compression quality, resolution cropping, image complexity, and colour palettes are changed or observed during dataset creation.

JPEG Compression and Quality

One of the largest factors that affects the reliability of steganography within the JPEG domain is the original compression quality of the cover image. (Holotyak et al., 2005; Ker, 2005; Lyu & Farid, 2004; Pevny et al., 2010). When each JPEG image file is modified or saved, it undergoes *lossy* compression where it is compressed by stripping the image of imperceptible information, including fine details and colour gradients. In his book, Miano (1999) describes the four steps taken during JPEG compression as being:

1. downsampling Red Green Blue (RGB) colour palette to its luminance and chrominance components (YcbCr),
2. Discrete Cosine Transformation (DCT),
3. quantisation, and
4. Huffman Coding.

This process can make it easier for both a passive observer and steganalysis tools to detect differences within the image. Ker (2004) establishes the criticality of the JPEG compression process as he demonstrated higher susceptibility to steganalysis associated with higher compression rates. Holotyak, Fridrich, and Voloshynovskyy (2005) explained that this is because the compression process removed areas of high-frequency noise that would otherwise be used to efficiently and effectively embed the secret message. Because of this, experiments conducted on compressed JPEG images commonly used compression rates between 70% to 90% (Holotyak et al., 2005; Ker, 2005; Lyu & Farid, 2004; Pevny et al., 2010) and examples where compression quality were below 60% are harder to find (Ker, 2004; Ker, 2005; Kharrazi et al., 2006).

Image Cropping

Each dataset within the literature cropped the cover image file during pre-processing, with the two most common resolutions used being 512x512 (Holotyak et al., 2005; Ker 2004; Walia, Jain, and Naydeep, 2010) and 256x256 (Lyu and Farid, 2004) pixels. While other resolutions were uncommon, they did include 640x418 (Ker, 2005), 640x400 (Farid, 2002; Ker, 2005), 780x540 (Fridrich, 2004) 900x600 (Ker, 2004), 384x256 (Fridrich & Goljan, 2002), 1024x744 (Fridrich & Goljan, 2002), and 2100x1500 (Pevny et al., 2010) pixels.

As the resolutions used increase, so does the file size of the resulting image. One could intuitively assume that larger files can be used to embed larger payloads. However, Ker (2004) has shown that the increase in acceptable payload size and resolution are not proportional.

While not typically discussed within the literature, it is important to note that the cropping of images is likely due to the increased computational time and storage requirements that would be associated with larger images.

Image Complexity

While uncommon, some researchers (Liu, Sung, Xu, & Ribeiro, 2006; Lyu & Farid, 2004; Fridrich, 2004) took advantage of the cropping process to ensure that the areas used were consistent regarding image complexity and steganographic performance.

Liu, Sung, Xu, and Ribeiro (2006) also demonstrated that image complexity could affect the performance of steganalysis by specifically cropping both coloured and grayscale datasets and ranking the resulting images before performing steganalysis. This result showed that steganalysis was generally easier in images with lower complexity and that standard colour images and grayscale images with low complexity were comparable for most steganalysis techniques.

Colour Palettes

Using grayscale cover images is preferred for steganography as they are harder to detect during steganalysis compared to its colour counterparts (Liu, Sung, Chen, and Xu, 2008). According to Holotyak, Fridrich, and

Voloshynovskyy (2005), this is because it is more difficult to detect the steganography in grayscale images as the same inter colour channel relationships found in colour images do not exist in grayscale images. This is the preferred view in most literature with notable exceptions such as Fridrich (1999), Fridrich and Goljan (2002), Liu, Sung, Xu, and Ribeiro (2006), Lyu and Farid (2004), Rabie (2015), Wu and Noonan (2012) and Yu, Chang, and Lin (2007).

However, it is important to note that while grayscale is preferred in literature, real-world steganographic applications would include colour images and there is a lack of literature addressing colour images. Out of the 29 datasets examined, only the 7 papers identified above discussed or compared steganography in colour images.

Steganographic Techniques

Among the literature review for this paper, 29 discussed separate steganographic techniques that were used to build our dataset and their occurrence frequency has been listed Table 1, which clearly indicates the popularity of the individual techniques.

Table 1: Popularity of Steganographic Techniques

Steganographic Techniques	Frequency
LSB	10
Outguess	9
F5	7
Jsteg, Un-named	4
Ezstego, YASS, Steghide	3
Model-Based, s-tools, Hide4PGP, Steganos, MME3, PQ, NsF5	2
MME2, MSS, HUGO, LSBR, BCHopt, Edged-based, MB1, PQ, Pqe, Lqsteg, J-UNIWARD, UERD, UED, HQIH, JPHide, MM2, MM3	1

It is important to note that while this data shows that most sources used multiple algorithms or tools to create their dataset, when they did so it was often to compare algorithms that shared similar properties. Additionally, it was not uncommon for a single source to use a single tool (Holotyak, Fridrich & Voloshynovskyy, 2005; Ker, 2004; Ker, 2005; Kodovsky, Pevny & Fridrich, 2010; Rabie, 2015; Yu, Chang & Lin, 2007; Zhang et al., 2013)

Embedding Rate

The literature shows that the approach used to embed the payloads often varied and explored multiple variables for determining the capacity used. Common methods of embedding data involved using fixed payloads based on the size of the image (Aycibas et al., 2005; Farid, 2002; Kodovsky & Fridrich, 2013; Liu et al., 2008) or by embedding the information based on the Bits Per Non-Zero DCT Coefficient (BPC) of the images (Fridrich, 2004). In most cases, the payload size when using these methods was a multiple of 5 and ranged between 0.05 and 1.00. Notable exceptions to this were Ker (2004), who used a rate of 0.03 and Zhang et al. (2013) who included rates of 0.43, 0.61 and 0.83.

Other methods included embedded payload images with set resolutions instead of random data to create their steganographic images (Farid, 2002; Fridrich & Goljan, 2002; Lyu & Farid, 2002; Lyu & Farid, 2004; Monoharan et al., 2015; Rabie, 2015; Wu & Noonan, 2012; Yu et al., 2007). In these cases, the secondary image resolutions ranged from 32x32 to 256x256 pixels.

METHOD

As outlined in the previous section, the methodology for this research was structured based on the four aspects of image acquisition, pre-processing, steganographic techniques, and embedding capacity. Additionally, we also included a step to validate our processes by included hashing, logging, and validation of the dataset. A conceptual design of the experimental procedure is illustrated in Figure 1.

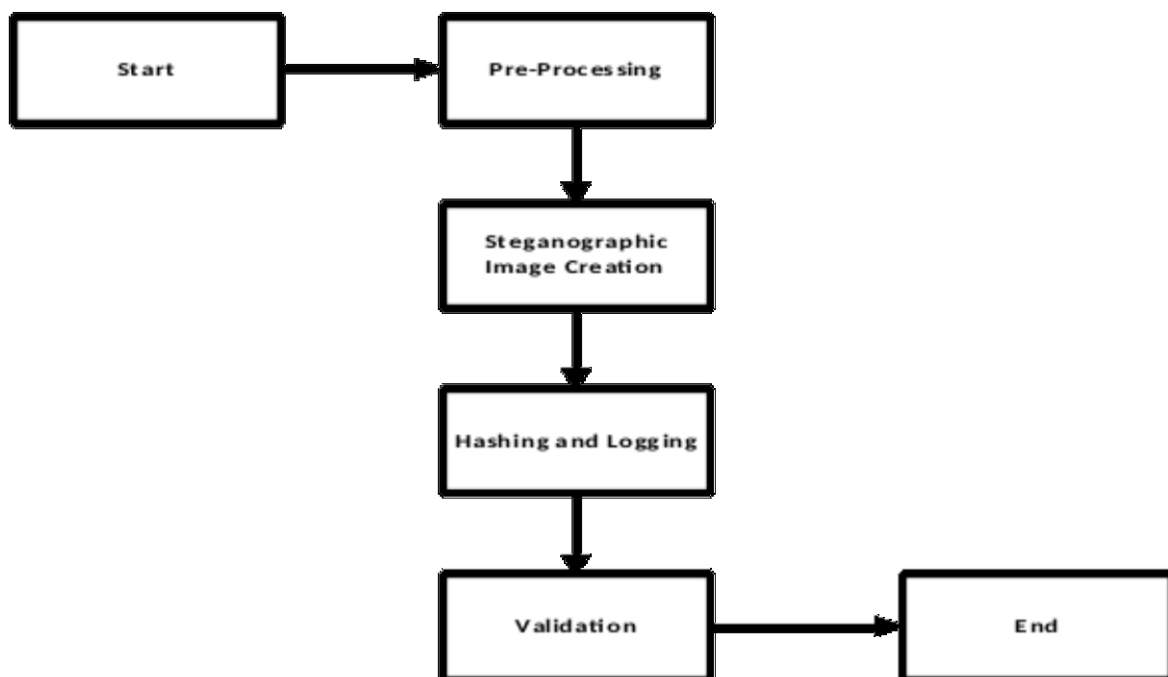


Figure 1: Conceptual Design of Experimental Procedure

Image acquisition

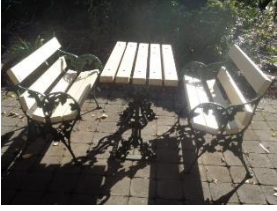









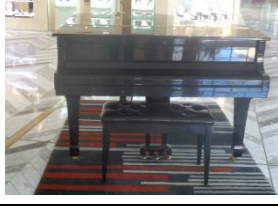
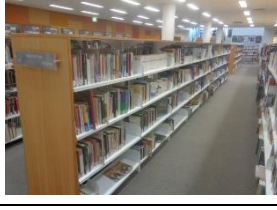
For the purpose of this research, the 1,760 images were acquired using a SONY DSC-W830 Cybershot camera (using base settings) for a duration of two months across a range of locations including public art installations, legal graffiti and street art zones, and public parks and buildings. Out of the final 1,000 images, 844 were taken in outdoor locations while 166 were taken inside public buildings. Further details of the image characteristics are listed in Table 2 while Table 3 provides visual examples of the types of images chosen. Lastly, during the image acquisition process, 760 images were discarded during the collection phase for reasons that included:

1. **Incorrect Settings:** leading to higher resolution and image sizes that would be unsustainable over the course of the cover image and its resulting steganographic image sets.
2. **Blurry/unusable images:** which would be unable to serve as an image for this purpose.
3. **Photography policies:** for public places such as the Australian War Memorial (Australian War Memorial, 2009) that we felt would not be compatible with the restrictions placed by the ethics committee approval.

Table 2: Cover Image Characteristics

Dimensions	640x480
Width	640 pixels
Height	480 pixels
Horizontal resolution	350 dpi
Vertical resolution	350 dpi
Bit depth	24
Compression	Uncompressed
Resolution unit	2
Colour representation	sRGB
Compressed bits/pixel	4

Table 3: Cover Image Examples

Cover Images	Sample Images		
Outdoor Settings			
Public Art Installations			
Legal Graffiti Zones			
Indoor Settings			

Pre-Processing

There were three main decisions to be made during pre-processing. These were:

1. What cropping and offset would be used?
2. What colour model to use?
3. What image types would be used?

Cropping and Offsetting

As the literature review showed, the two most common resolutions used for cropping steganographic cover images were 256x256 and 512x512 pixels. As the original cover images were only 640x480, we made the decision to crop the cover images to 256x256 pixels, which offers the added benefit of lower processing time due to fewer pixels. Furthermore, since all images were acquired manually, we did not see the need to offset during the cropping process.

Colour Palettes

As the literature identified a gap in the literature regarding colour-based steganographic datasets, we decided to use the coloured versions of the cover images wherever possible. It is important to note though, that some of the steganographic algorithms used required the use of grayscale images or converted the coloured cover images to grayscale during the embedding process.

As the processed cover image set does contain a set of grayscale images for the JPEG format, future experiments could use these files to create a compare between colour and grayscale images for JPEG steganography, which would assist with filling other identified gaps within the literature.

Image File Types

The image file types used during the creation of this dataset were JPEG and PGM. Other popular image file types such as TIFF, GIF, and BMP were not used in this dataset as they were not relevant to the steganographic techniques used. Table 4 shows the final properties for each set of cover images used.

Table 4: Post-Process Characteristics by Image Type

Property	Colour JPEG	Grayscale JPEG	Colour PGM
Dimensions	256x256	256x256	256x256
Width	256 pixels	256 pixels	256 pixels
Height	256 pixels	256 pixels	256 pixels
Horizontal resolution	350 dpi	350 dpi	350 dpi
Vertical resolution	350 dpi	350 dpi	350 dpi
Bit depth	24	8	24
Compression	Uncompressed	Uncompressed	Uncompressed
Resolution unit	2	2	2
Colour representation	sRGB	sRGB	sRGB
Compressed bits/pixel	4	4	4

The pre-processing operation is shown in Figure 2.

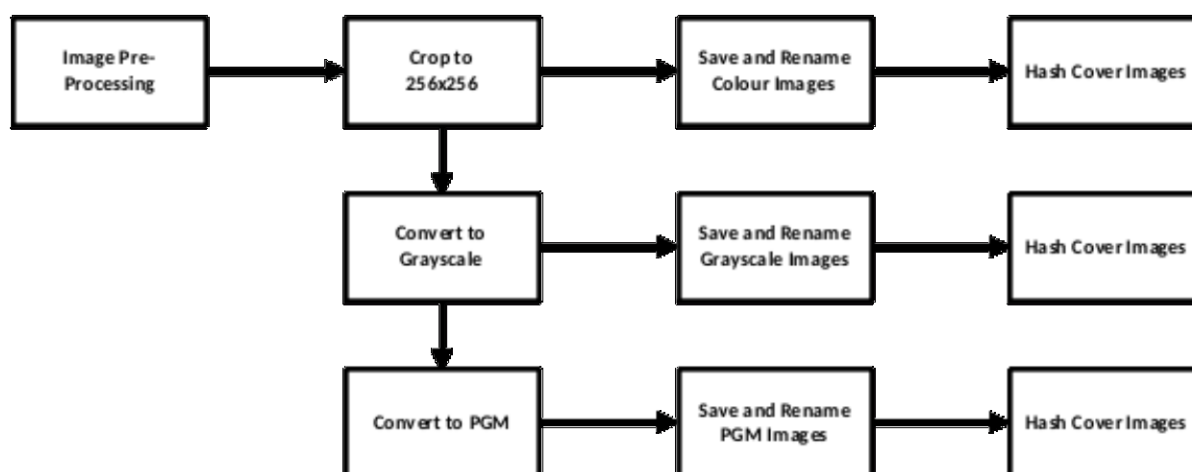


Figure 2: Pre-Processing Operation

Steganographic Techniques

The details of the steganographic techniques used during this research are summarised in Table 5 and Figure 3 shows the steganographic embedding process followed.

Table 5: Steganographic Techniques for Building the Dataset

Author	Algorithm or Tool	Domain	Input file type
Provos, N. (2001)	Outguess 0.2	JPEG	JPG
Westfield, A. (2001)	F5	JPEG	JPG
Hetzel, S. (2003)	Steghide 0.5.1	JPEG	JPG
DDE Lab. (2013b)	J-UNIWARD	JPEG	JPG
DDE Lab. (2013c)	S-UNIWARD	Spatial	PGM
DDE Lab. (2014)	SI-UNIWARD	Side-Informed	PGM
DDE Lab. (2013a)	nsF5	JPEG	JPG
DDE Lab. (2012a)	HUGO	Spatial	PGM
DDE Lab. (2012b)	WOW	Spatial	PGM
Kodovsky, J. (2007)	PQ	Spatial	JPG

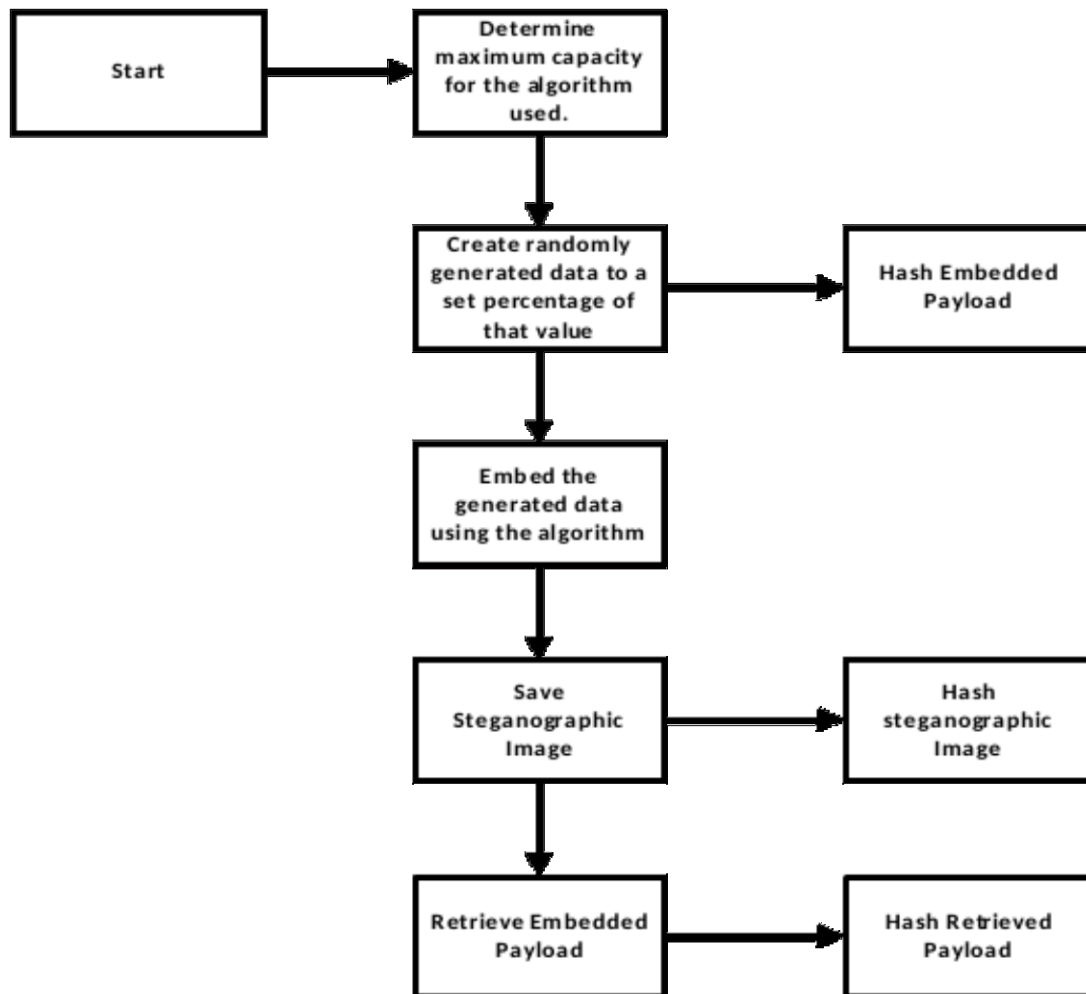


Figure 3: Steganographic Embedding Process

Embedding Rate

The embedding rate originally used within the dataset is 0.20 bits per available capacity (BPAC) for each cover image. As the BPAC will vary between each algorithm, this number must be calculated independently for each tool.

For Outguess and F5, this required embedding a small test file into each image to pull the estimated capacity from the tool's output (i.e., *-verbose* parameter). For Steghide, this required first viewing the maximum capacity (i.e., using *-info* parameter). Once these expected capacities were known, we were then able to calculate the required size of the test file based on the embedding rate selected.

For J-UNIWARD, S-UNIWARD, SI-UNIWARD, nsF5, HUGO, WOW, and PQ, the maximum embedding capacity is automatically calculated during the embedding process, and the only step required was using a variable to call the required embedding rate when executing these algorithms from the command line.

Verification and Dataset

Throughout the course of the experiment, various events were logged (date, time, embedding rate, and the path to artefact) and all individual artefacts created were hashed (MD5) for validation purpose. This allowed us to verify that the Cover images and Steganographic images were not the same after the embedding process and confirm that the exact payload could be retrieved when the tools used possessed this capability. At the end of the experiment, the final dataset we curated consisted of 14,000 images in total, which includes the original cover images, pre-processed cover images, and the final Steganographic images.

CONCLUSION AND FUTURE DIRECTION

This paper has presented the process and criteria we used to build a dataset for image steganography using. During our research, we discovered that there were different approaches used to construct the datasets used in literature in relation to steganography and steganalysis.

In our approach, we considered image acquisition, pre-processing, steganographic techniques, and embedding rate. During image acquisition, we chose to acquire manually using a digital camera, and used pre-processing to further format images to JPG and PNG, and also to crop the images to a specific resolution. Finally, we used multiple steganographic techniques and selected embedding rates determined consistent with each steganographic technique used.

One of the limitations we encountered was during the image acquisition process. Due to onsite photography policy, several images captured had to be excluded from the dataset. As a result, only a small portion of the dataset (16%) represents indoor images. We feel that there has to be a balance between indoor and outdoor images at least, in order to have a true representation of real-world applications.

REFERENCES

- Australian War Memorial. (2009). *Factsheet – Onsite Photography*. Retrieved from <https://www.awm.gov.au/sites/default/files/media/factsheet%20-%20onsite%20photography%20feb%2009.pdf>
- Avcibaş, İ., Kharrazi, M., Memon, N., & Sankur, B. (2005). Image steganalysis with binary similarity measures. *EURASIP Journal on Advances in Signal Processing*, 2005(17), 679350.
- DDE Lab, Birmingham University. (2012a). HUGO bounding dist. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- DDE Lab, Birmingham University. (2012b). WOW. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- DDE Lab, Birmingham University. (2013a). nsF5. [Computer Software]. Retrieved from <http://dde.binghamton.edu/download/nsf5simulator>
- DDE Lab, Birmingham University. (2013b). S-UNIWARD. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- DDE Lab, Birmingham University. (2013c). J-UNIWARD. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- DDE Lab, Birmingham University. (2014). SI-UNIWARD. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- Farid, H. (2002). Detecting hidden messages using higher-order statistical models. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (Vol. 2, pp. II-II). IEEE.
- Farid, H., & Lyu, S. (2003, June). Higher-order wavelet statistics and their application to digital forensics. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on* (Vol. 8, pp. 94-94). IEEE.
- Fridrich, J. (1999, April). A new steganographic method for palette-based images. In *PICS* (pp. 285-289).
- Fridrich, J. (2004, May). Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *International Workshop on Information Hiding* (pp. 67-81). Springer Berlin Heidelberg.
- Fridrich, J., & Goljan, M. (2002). Practical steganalysis of digital images: State of the art. In *Electronic Imaging 2002* (pp. 1-13). International Society for Optics and Photonics.
- Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color, and gray-scale images. *IEEE multimedia*, 8(4), 22-28.
- Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23), e215-e220.
- Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [<http://kdd.ics.uci.edu>]. Irvine, CA: University of California, Department of Information and Computer Science
- Hetzl, S. (2003). Steghide 0.5.1. [Computer Software]. Retrieved from 'sudo apt-get install steghide'
- Holotyak, T., Fridrich, J., & Voloshynovskyy, S. (2005). Blind statistical steganalysis of additive steganography using wavelet higher order statistics.
- Huang, F., Shi, Y. Q., & Huang, J. (2010). New JPEG steganographic scheme with high security performance. In *International Workshop on Digital Watermarking* (pp. 189-201). Springer Berlin Heidelberg.
- Ker, A. D. (2004). Improved detection of LSB steganography in grayscale images. In *International workshop on information hiding* (pp. 97-115). Springer Berlin Heidelberg.

- Ker, A. D. (2005). Steganalysis of LSB matching in grayscale images. *IEEE signal processing letters*, 12(6), 441-444.
- Kharrazi, M., Sencar, H. T., & Memon, N. (2006). Performance study of common image steganography and steganalysis techniques. *Journal of Electronic Imaging*, 15(4), 041104-041104.
- Kodovsky, J. (2007). PQ. [Computer Software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- Kodovský, J., & Fridrich, J. (2013, March). Quantitative steganalysis using rich models. In *IS&T/SPIE Electronic Imaging* (pp. 86650O-86650O). International Society for Optics and Photonics.
- Kodovský, J., Pevný, T., & Fridrich, J. (2010, February). Modern steganalysis can detect YASS. In *IS&T/SPIE Electronic Imaging* (pp. 754102-754102). International Society for Optics and Photonics.
- Liu, Q., Sung, A. H., Chen, Z., & Xu, J. (2008). Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images. *Pattern Recognition*, 41(1), 56-66.
- Liu, Q., Sung, A. H., Xu, J., & Ribeiro, B. M. (2006, August). Image complexity and feature extraction for steganalysis of LSB matching steganography. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (Vol. 2, pp. 267-270). IEEE.
- Lyu, S., & Farid, H. (2002, October). Detecting hidden messages using higher-order statistics and support vector machines. In *International Workshop on Information Hiding* (pp. 340-354). Springer Berlin Heidelberg.
- Lyu, S., & Farid, H. (2004, June). Steganalysis using color wavelet statistics and one-class support vector machines. In *Electronic Imaging 2004* (pp. 35-45). International Society for Optics and Photonics.
- Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., & Jain, A. K. (2002). FVC2002: Second fingerprint verification competition. In *Pattern recognition, 2002. Proceedings. 16th international conference on* (Vol. 3, pp. 811-814). IEEE.
- Mathworks. (2017). MATLAB R2017a. [Computer Software]. Retrieved from https://au.mathworks.com/products.html?s_tid=gn_ps
- Miano, J. (1999). *Compressed image file formats: Jpeg, png, gif, xbm, bmp*. Addison-Wesley Professional.
- Pevny, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security*, 5(2), 215-224.
- Pieprzyk, J., Hardjono, T., & Seberry, J. (2003). *Fundamentals of computer security*. Berlin Heidelberg: Springer Verlag.
- Provos, N. (2001). Outguess 0.2 [computer software]. Retrieved from "sudo apt-get install outguess"
- Rabie, T. (2015). Lossless quality steganographic color image compression. *Int J Adv Comput Sci Appl*, 6(4), 114-123.
- Solanki, K., Sarkar, A., & Manjunath, B. S. (2007, June). YASS: Yet another steganographic scheme that resists blind steganalysis. In *International Workshop on Information Hiding* (pp. 16-31). Springer Berlin Heidelberg.
- Walia, E., Jain, P., & Navdeep, N. (2010). An analysis of LSB & DCT based steganography. *Global Journal of Computer Science and Technology*, 10(1).
- Westfield, A. (2001). F5 [computer software]. Retrieved from http://dde.binghamton.edu/download/stego_algorithms/
- Wu, Y., & Noonan, J. P. (2012). Image Steganography Scheme using Chaos and Fractals with the Wavelet Transform. *International Journal of Innovation, Management and Technology*, 3(3), 285.
- Yu, Y. H., Chang, C. C., & Lin, I. C. (2007). A new steganographic method for color and grayscale image hiding. *Computer Vision and Image Understanding*, 107(3), 183-194.
- Zeng, J., Tan, S., Li, B., & Huang, J. (2016). Large-scale JPEG steganalysis using hybrid deep-learning framework. *arXiv preprint arXiv:1611.03233*.
- Zhang, Y., Jiang, J., Zha, Y., Zhang, H., & Zhao, S. (2013). Research on embedding capacity and efficiency of information hiding based on digital images. *International Journal of Intelligence Science*, 3(02), 77.