Open-Source Intelligence Investigations: Development and Application of Efficient Tools

By

Cecelia Horan

B.S., University of Kansas, 2020

Submitted to the graduate degree program in Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

_____

Chair: Hossein Saiedian, Ph.D.
Professor and Thesis Advisor

_____

Fengjun Li, Ph.D.
Committee Member

_____

Drew Davidson, Ph.D.
Committee Member

Date Defended: April 27, 2022

The thesis committee for Cecelia Horan certifies that this is the approved version of the following thesis:

Open-Source Intelligence Investigations: Development and Application of Efficient Tools

_____

Chair: Hossein Saiedian, Ph.D.
Professor and Thesis Advisor

Date Defended: April 27, 2022
Date Approved: April 27, 2022

Abstract

Open-source intelligence is a branch within cybercrime investigation that focuses on information collection and aggregation. Through this aggregation, investigators and analysts can analyze the data for connections relevant to the investigation. There are many tools that assist with information collection and aggregation. However, these often require enterprise licensing. A solution to enterprise licensed tools is using open-source tools to collect information, often by scraping websites. These tools provide useful information, but they provide a large number of disjointed reports. The framework we developed automates information collection, aggregates these reports, and generates one single graphical report. By using a graphical report, the time required for analysis is also reduced. This framework can be used for different investigations. We performed a case study regarding the performance of the framework with missing person case information. It showed a significant improvement in the time required for information collection and report analysis.

Acknowledgements

I would like to thank everyone who helped me complete this work. Thank you for being patient with my constant obsession with working on this, for the late nights, and last-minute plans. Without your help, this never would have happened. I would like to especially thank my parents, who were always willing to listen to my constant ramblings about this project. You both have been an inspiration to me. I would also like to thank my brother. We may be complete opposites, but you are still my friend for life. Finally, I would like to thank advisory board at KU and my academic advisor Hossein Saiedian. This would have been impossible without your guidance.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: CYBERCRIME AND CYBER INVESTIGATIONS

1.1. CYBERCRIME AND CYBER INVESTIGATIONS

The term cybercrime is often misused or ill-defined. It is often used as a buzzword that has little true meaning. For the purposes of this research, cybercrime is defined to mean any crime where an attacker uses technological means to attack the victim. Cybercrimes could be anything from fraud to ransomware to terrorism. This has many effects on the technology industry, as well as many other businesses and industries.

In the past five years there has been an upward trend in the amount of complaints and the total cost of cybercrime, with the overall cost being 13.3 billion dollars. As Table 1.1 shows, the total cost of cybercrime in 2020 alone amounted to 4.1 billion dollars with almost eight hundred thousand complaints made to the Internet Crime Complaint Center (Internet Crime Complaint Center, 2021). The UK alone saw a 14% increase in complaints in 2018 with 2,547 investigations launched (Computer Fraud and Security, 2018). An example of the most damaging and costliest crimes is business email compromise (BEC), costing 1.8 billion dollars. BEC is a threat that affects any business or organization that uses email in their operations, making it a dangerous threat.

Table 1.1: Five-year statistics of cybercrime effects

Year	2016	2017	2018	2019	2020
Cost	1.5B	1.4B	2.7B	3.5B	4.2B
Complaints	278,728	301,580	351,937	467,361	791,790

One thing that can be done to combat this rise in cybercrime is to investigate the crimes. This can be done by analysts, researchers, and law enforcement. Methods of cyber investigations are broken into two main categories: forensics and intelligence. Forensics have many different parts, but they depend on digital evidence from various devices. Intelligence consists of the gathering and aggregating of information in order to identify patterns or draw conclusions based on the information. Cyber investigation techniques can also be extended to almost any other type of investigation. For example, in missing persons cases, investigators often rely on cyber intelligence gathering techniques to gather information on the individual who is missing.

As Ignaczak et al. (2022) said, “Many cybersecurity activities involve the analysis of unstructured data. This type of data has increased in recent years, influenced by the web and social networks.” Open-source intelligence (OSINT) is one of these activities. It is a subset of the intelligence part of cyber investigations where investigators gather and aggregate publicly

available information for analysis. This can be used in multiple areas of the technology industry, but it is particularly helpful for cyber investigations. Investigators can use OSINT to gather information on individuals or events from sources such as social media or chat forums on the dark web. This information can give helpful leads in cases and indications to crime trends and possible future attack vectors.

Cyber investigations have defined stages where particular activities occur. These are investigation initiation, modeling, assessment, impact and risk, planning, tools, action, and outcome (Hunton, 2011). OSINT can be used in any of these stages to help further the investigation, but it is most helpful in the beginning stages, such as modeling and assessment. This is because the quantity and quality of intelligence gathered at the beginning of an investigation may have a great impact on how the rest of the investigation goes.

There are many OSINT tools available to investigators. However, many of the comprehensive tools on the market require enterprise licensing and are expensive for individual investigators. This makes them inaccessible to freelance investigators who use their time outside of work to gather information through OSINT. There are free tools available to these types of investigators, but these tools have a drawback.

The free tools that are available typically only do one function. This could be searching for a name, username, or IP address. This is helpful to investigators in the narrow domain that these tools operate in, but these tools do not cover all the areas that are critical for investigators. The disconnect between the tools makes documentation difficult and time-consuming because there will be separate reports from each of the tools that the investigators use.

1.2. IMPACT OF CYBERCRIME

There are many good motivations for conducting research into the use of OSINT in cybercrime investigations. By investigating cybercrime with the intention of finding and stopping the criminal, cybercrime rates will decrease more rapidly (Computer Fraud and Security, 2012). One way of doing this is by profiling the criminal based on the activity patterns of the crime (Nykodym, Taylor, and Vilela, 2005). Much of this information can be gathered through OSINT methods.

Cybercrime has had an enormous impact on the field of technology. Currently, there are almost five hundred thousand cybersecurity jobs open in the United States alone (Cyber Seek, 2021). Cybersecurity has driven much of the recent changes in technology. For example, secure coding was not something developers were focused on, but now it is one of the more common non-functional requirements in software. This demonstrates the impact that cybersecurity and cybercrime have had on other areas of technology.

Cybercrime has not only affected the technology industry, but it affects any industry that utilizes technology. There are many attacks that are performed on other industries. For

example, phishing attacks are directed against any organization that utilizes email, phone, or instant messaging. This puts many critical industries at risk, such as the financial and health industries because they both use these communication technologies. These kinds of attacks can cause massive amounts of damage to these industries and can lead to opening the door for other attacks, such as ransomware.

Not only does cybercrime affect businesses negatively, but charities and non-profits as well. Figure 1.1 shows the impact on UK businesses and charities. It shows that charities are more likely to be impacted by cybercrimes, especially in the case of having money stolen or systems damaged (Furnell et al., 2021). Cybercrime many different costs associated with them that the charities and business have to deal with. These include customer retention, security improvement measures, consulting fees, and data or software loss, just to name a few (Furnell et al., 2021). This is an unfortunate fact of cybercrime and demonstrates why it is critical for security professionals to combat cybercrime in the most efficient way possible to reduce cybercrime.

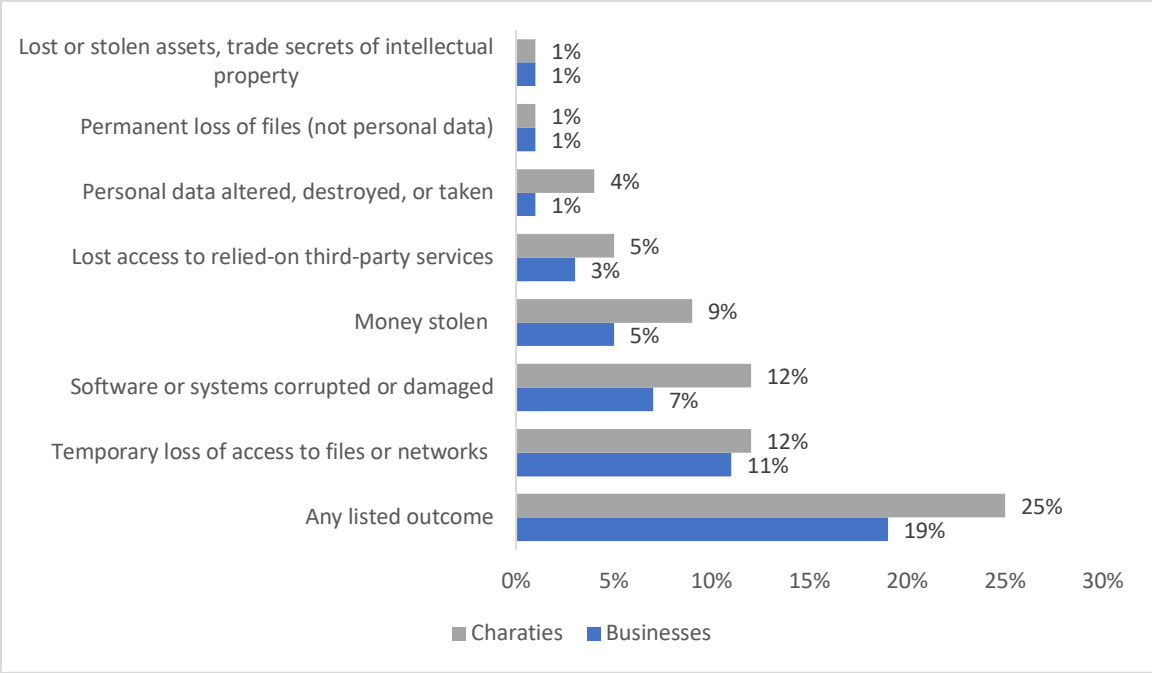


Figure 1.1: Impact of cybercrime on businesses and charities (Furnell et al., 2021)

Cybercrime investigations are not only helpful for the technology industry, but they promote positive change for the larger community. Investigating cybercrime means a lower rise in the cybercrime rate. In October of 2021, 150 individuals were arrested after an investigation of drug trafficking over the dark web (The United States Department of Justice, 2021). Part of this investigation relied on OSINT gathered from the dark web. This shows the power and impact that OSINT investigations can have in the real world for positive change.

Two other recent cases demonstrate the impact that cybercrime has at a global scale. In both of these cases, law enforcement from different countries coordinated with each other to take down botnets and ransomware operators (Computer Fraud and Security, 2021a; Computer Fraud and Security, 2021b). Neither of these events would be possible without the global coordination of law enforcement, demonstrating the wide reach that cybercrime has.

Research into this area of OSINT can bleed into the other areas where OSINT can be used. For example, penetration testers use OSINT to determine the threat exposure of the organization they are testing. This is often called reconnaissance. Researching the OSINT investigations can lead to more research into other areas of OSINT, which will benefit the cybersecurity field as a whole. By expanding the research in the cybersecurity industry, other areas of technology and other industries will benefit from better security and more accessible cyber investigations.

1.3. A CONNECTED FRAMEWORK OF OSINT TOOLS

To assist OSINT investigations, having a framework of connected open-source tools would mean that OSINT investigations will be more accessible to a wider variety of investigators. This will allow for more people to preform OSINT investigations, which will help decrease the workload on other investigators. We are looking to increase two things: efficiency and effectiveness. We believe that to increase efficiency, automation for collection can be performed to decrease time for collecting information and a graphical reporting mechanism can be used to decrease the time to analyze the information. To increase effectiveness, we believe a graphical reporting method will assist investigators in finding connections that they otherwise would not have discovered.

A framework for these free and open-source tools exists. However, this framework simply organizes the tools. It shows the kind of information the tools collect and categorizes them, but it does not provide a connection between them. This lack of connection creates difficulties for investigators. Investigators will need to run these tools separately, generating separate reports for each tool. This can lead to lengthy investigations, and an increased possibility that information will be lost when these reports are combined. It also means that the tasks investigators are performing are redundant and could benefit from automating the collection of information.

To solve the problem of these disconnected tools and redundant tasks, this research has begun to build a connection with these tools in the cloud and automate the collection and reporting of information with the tools. It is out of scope for this research to build a connection between all the existing tools. There are simply too many and to wide a variety of tools. The researchers focused on tools that collected specific kinds of information to begin research in this area. The types of information the researchers focused on is information that is useful in missing persons cases.

The types of information this connection of tools searches for are information that is valuable in a missing persons case, which includes names, usernames, locations, and events. Much of this information comes from social media posts and their metadata. The intent is to gather information that shows connections between the missing person and anything that could help investigators locate them. By using multiple tools that overlap on the information they gather, investigators will be able to validate information and look for inconsistencies in what is gathered.

The information collected will then be sent to AWS Neptune, a graphical relational database. This database shows investigators a pictorial representation of the relationships between the missing person found in the data collected by all the tools. By having a visual representation of the relationships found between the information collected, investigators can find patterns and connections easier and faster.

Further research and development can add more tools from the existing framework to this connection of tools. It can be expanded to cover other areas where OSINT is used, such as reconnaissance in penetration testing. This opens the door to further development and research in the OSINT field, which will benefit many other areas of cybersecurity, such as penetration testing as mentioned earlier.

1.4. METHOD FOR BUILDING THE FRAMEWORK

To build this framework, the researchers preformed four main steps: a literature review, system design, implementation, and a case study to test the results. Each of these steps gave the researchers further insight into the needs of the field and how this research can be valuable to investigators.

A literature review was performed on existing literature regarding cybercrime investigation methodologies. This determined the landscape of the field and where research gaps exist. To gain a good idea of current literature and knowledge in the area of cybercrime investigation and cloud architecture, we reviewed research papers from such journals as *IEEE Transactions on Cloud Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Network and Service Management, IEEE Security and Privacy, Computers and Security, ACM Computing Surveys*, and *IEEE Internet of Things Journal* among others. We found that OSINT investigations were underrepresented in the current research literature and that this research would help in the field and in other areas of technology. It also showed vast sources where information can be gathered for OSINT investigations. This gave the researchers a baseline to start from when searching for tools to gather information and what kind of information to search for.

The system was designed based on the basic requirements and constraints of the tools used during implementation. There were several constraints of the systems being used that the researchers accounted for. These are explained in detailed in Chapter 3 of this research. By

defining the constraints, the researchers were able to determine the requirements of the system. Multiple tools will be evaluated based on their conformity to these restraints. The tools that conform to the requirements and constraints then become part of the formal architecture of the system.

After the design was completed, the researchers implemented the system based on the specifications. This implementation involved the configuring and deploying of the cloud connection in AWS. Any changed in the design during implementation were thoroughly documented so the current design is accurate to the system. Figure 1.2 shows a high-level overview of the proposed design of the connection. This is a very high-level design representation, but it shows the relationship between the existing tools and the AWS services that the researchers implemented.

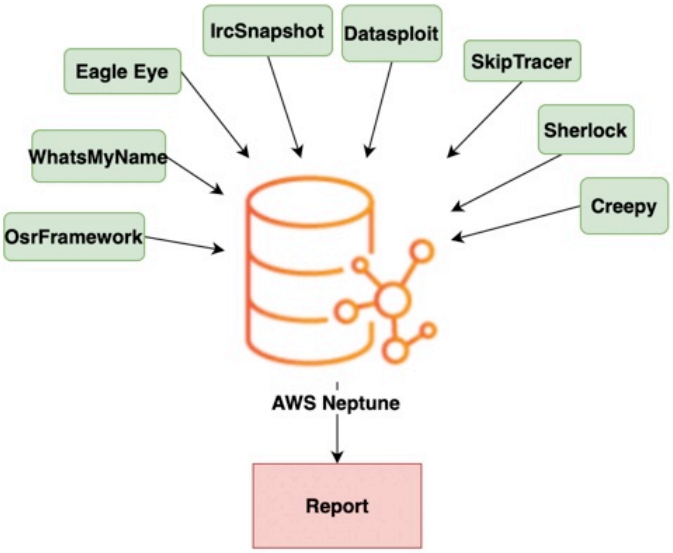


Figure 1.2: Overview of solution for tool connection

In order to test the results of this research, the researchers preformed a case study on missing persons case information. This case study shows the comparison of this tool against manual collection methods and crowdsourcing information gathering. By drawing this comparison, the researchers are able to accurately show how this tool changes OSINT investigations. The details of this case study are discussed in Section 1.6 of this chapter.

1.5. EXPECTED RESULTS OF FRAMEWORK RESEARCH

The results of this research are expected to produce a tool that brings existing tools together into one unit that will collect and report information on a specific subject. This tool will bring together existing tools for the collection, analysis, and reporting of information related to a

case. As a result of using this tool, investigations will be more efficient and effective, and the information collected is more useful to the case.

The foundation of OSINT investigations is information. These types of investigations gather large amounts of information, which can lead to complications with the organization and recording of this information. If investigators use tools that are not unified, the information collected can be lost, or possibly changed. Lost or changed information can cause investigations to last longer than they originally would because the information would have to be corrected or collected again.

With using a unified tool, the information gathered in OSINT investigations is less likely to be lost or changed because of the centralized reporting and consolidation of the tools. Centralizing the reporting mechanism will also ensure that the information gathered is well organized and complete. This way there is a direct connection between all the tools the investigators are using and a single report. Overall, this will make OSINT investigations faster and more effective.

1.6. MISSING PERSONS INFORMATION CASE STUDY

To evaluate this connection of tools, the researchers will perform a case study involving missing persons case information. The type of information collected was specified for this case study in order to fully test the tool. This case study is described in detail, including a detailed testing methodology, in Chapter 4.

To have a set of information to compare this tool's performance against, the researchers must collect information manually. The same tools used in the cloud connection are utilized in order to have accurate testing. This shows the direct comparison of one investigator using the tools individually versus the connection of tools and will ensure the integrity of the test data. All the information gathered is manually documented to compare the documentation speeds of manual gathering and the cloud connection.

Also, to add another vector of comparison for this tool, crowdsourced information is utilized. The information will be collected through an organization, TraceLabs, that crowdsources intelligence gathering for missing persons cases. There is no way to test which tools the individuals collecting the information through this method use, but it will serve as a good way to test the connection of tools against crowdsourced information. This will give a comparison of the differences in time of a team of investigators versus one investigator using the framework.

The comparison is based on the following criteria: time to collect, time to analyze, accuracy of the information, amount of information, and time to generate a report. The time to collect and analyze the information shows the change in time for the investigative work. The accuracy of the information will show if any of the collection methods gather information that is not accurate, which could divert the entire investigation. The amount of information shows

which method is more efficient when compared to the time to collect. Finally, the time to generate the report shows the efficiency of having a centralized reporting system. Each of these comparison methods shows the increase in efficiency of investigations from connecting the tools.

Bertolino et al. (2019) proposed a framework for testing cloud tools, as depicted in Figure 1.3. There are six phases in this framework that were used in testing this collection of tools. The phases we focused on were the test design, test execution, test objective, and test evaluation. The design, execution, and evaluation are discussed in Chapter 4 of this research. The objective of this test is to compare the time for collection and analysis of the framework and compare it against other methods of collecting and analyzing information.

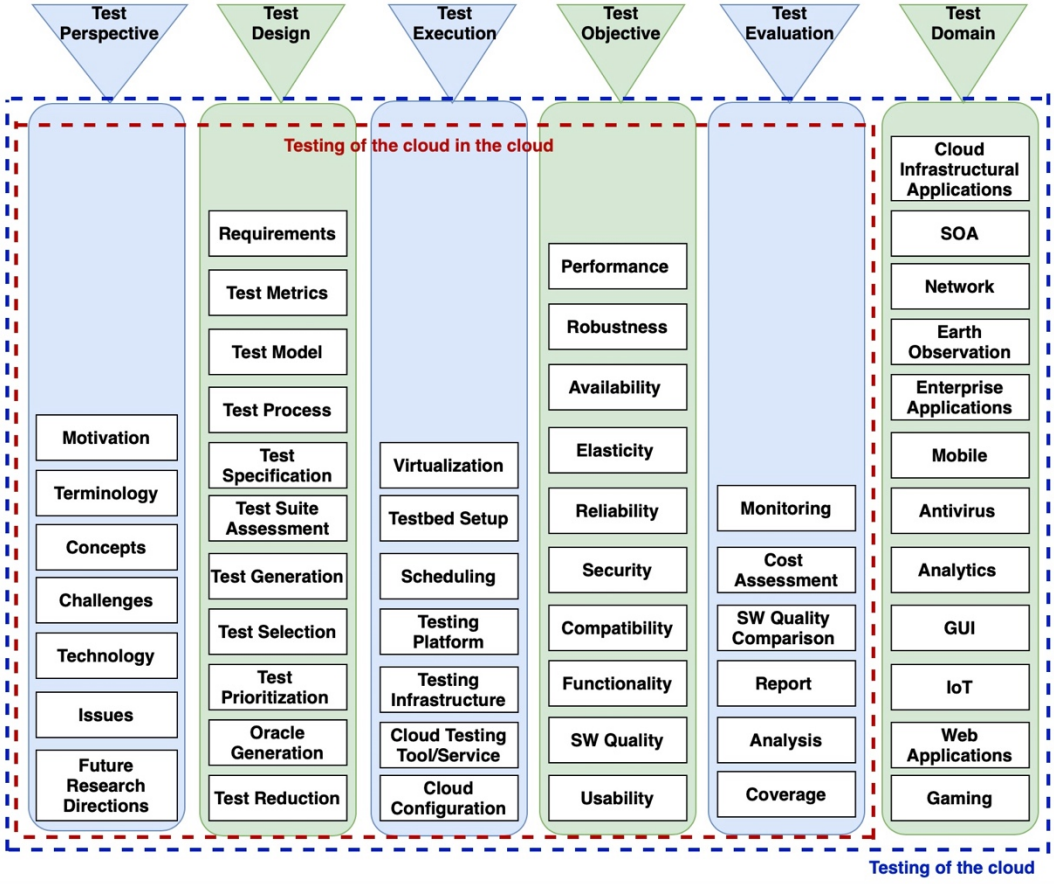


Figure 1.3: Framework for testing cloud tools (Bertolino et al., 2019)

All information collected through this case study is thoroughly documented to ensure that the comparison of the data sets is as accurate and complete as possible. Also, by using a real case for the case study, the researchers are able to determine how well this tool functions in the real world, giving investigators the best possible understanding of the advantages of using this connection of tools. To ensure the integrity of the testing in this case study, the same

case will be used for all three of the collected data sets. This will ensure that no collection method is given an unfair advantage. Also, the same tools will be used for the manual collection that are utilized in the connection of tools to ensure that neither method has an advantage over the other.

1.7. THE BROADER IMPACT OF CYBERCRIME RESEARCH

Cybercrime has a broader impact on society and the economy. Investigating cybercrime will have a positive impact that will spread to other areas indirectly. Furthermore, research into OSINT investigations will increase this impact because this research focuses on making the investigative tools more accessible. This will allow more individuals to perform investigations, which will increase the impact on these areas.

By investigating cybercrime, investigators will be able to affect change and reduce the rise in cybercrime. A lower crime rate is good for society because it means that the risk of cyber-attacks against individuals is not rising so quickly. Lowering crime rate will also help with improving the overall quality of life for the community. “Effective crime prevention can both maintain and reinforce the social cohesion of communities and assist them to act collectively to improve their quality of life,” (*National Crime Prevention Framework*, n.d.).

With the reduction in cybercrime crime rate, this will have a positive economic impact for many people. This includes, not only organizations, but individuals as well. A positive economic impact lifts the entire community. Also, a lower crime rate means more opportunities for business, which translates to economic growth in many industries (*National Crime Prevention Framework*, n.d.). With the reduced risk from cyber threats, businesses and organizations are able to use their resources to make a positive impact for their employees and society as a whole.

CHAPTER 2: CYBERCRIME, CYBER INVESTIGATION, AND CLOUD ARCHITECTURE LITERATURE REVIEW

2.1. INTRODUCTION TO CYBERCRIME AND CYBER INVESTIGATION LITERARURE REVIEW

Ever since the creation of the internet, people have been finding ways to conduct illegal activities using it as a tool. In order to counteract these actors, tools and methods have been developed to track these criminals. It is critical for security and law enforcement professionals to understand these tools and how they are developing, so they can better preform in their job roles. Internet crime is something that affects anyone who uses a computer, thus making it critically important to counteract it in any way possible. Knowing and understanding the current field of cybercrime helps investigators use their skills most efficiently in cyber investigations (Horan & Saiedian, 2021).

Some of the most common technologies and methods for tracking cyber criminals are digital forensics and online investigations, which leverages open-source intelligence, or OSINT. Within these areas, there are many different technologies and techniques that can be used to gather data on the malicious actor. This data can then be aggregated to determine who committed the crime and build a case against the individual. This paper will cover a survey of these technologies and the methods associated with them.

Digital forensics is a key field to tracking cyber criminals and counteracting crime. It is mainly made up of network forensics and memory analysis. By analyzing information found on disks and through networks, investigators can learn about other potential conspirators in the crime. This could help them track down these individuals and stop them before another crime is committed.

Much of the tracking of criminals is done online. The different layers of crime on the internet can be broken up into three categories: the surface, or open, web, the deep web, and the dark web. These areas of the web contain a host of information that can be valuable to investigations, so it is important to understand the tools that allow investigators gather and use this information. For example, investigators can utilize information regarding cryptocurrency transactions on the dark web to learn about criminal activity.

The changes and developments in this field are occurring rapidly and it is important for security professionals to keep up to date. Some new developments are coming from automation and machine learning. By automating their tools, investigators can speed up their process and reach their goal sooner. AI forensics will in the future help combat the growing trend of AI crime. This paper will also cover a summary of the developing technologies in this area and how they could change investigations in the future.

2.1.1. Motivation for Literature Review

Having this information compiled in a single document allows for easy comparison of methods and the information that these methods provide. None of the methods available to investigators are able to gather all the information they require for a case, making it even more important to understand how this information is gathered and how to fill the information gaps. If investigators are able to gain a complete picture of a crime, then they will be able to take action against the criminal or potentially stop a future crime from occurring.

The forensic aspect of cyber investigations is not discussed thoroughly in the rest of the chapters of this research. However, it is critical to understand investigations wholistically. This provides investigators with a better understand of how each part of the investigation affects the other parts, and how these changes can provide improvements in effectiveness later in the investigation. For example, information gathered through open-source intelligence can lead to the collecting of physical devices, which leads to digital analysis. As investigators are searching the device for information, it is helpful for them to understand the kind of information they are looking for, and this context comes from the information gathered earlier in the investigation through OSINT.

2.1.2. Methodology of the Literature Review

This paper utilizes the basic research and survey methodologies by leveraging existing research, synthesizing the material, and compiling information. Investigators must use a variety of tools, and their knowledge of the field must stay current with any developments. When comparing these tools and methods, there must be defined criteria of comparison.

When comparing digital forensics methods, there need to be some criteria to compare against. First, the complexity of the method must be determined. This shows how easy it is to preform, how costly it is, and the time consumption of the process. The risk to the data must then be determined, showing the risk to the data when using the method.

When comparing methods used in OSINT investigations, these methods must meet some requirements. First, the methods must be faster to use than manually searching for the information. The methods will then be evaluated on their application to the field, namely types of information that can be gathered through each method, the different methods of gathering this information, and the number of different types of cases where this method can be used.

2.1.3. Contribution of the Literature Review

Understanding the tools used to find cyber criminals is an important part of information security because knowing the tools available gives security professionals a deeper understanding of their profession. By understanding these technologies, security professionals can also better understand how crime often occurs, which will help when developing a security plan to prevent crime.

This paper is intended to compile a summary of technologies and methods used to track criminals online and through forensics, as well as the newest advancements in the field. By organizing this information in one place, it will be easily accessible to anyone interested in knowing about the cutting-edge technologies in this field.

2.2. DIGITAL FORENSICS

Digital forensics is the practice of collecting and organizing information found on an electronic device for investigative purposes. It is important to know both the tools and the methods and frameworks investigators use in this field.

Digital forensics can be broken into four areas: host forensics, mobile forensics, network forensics, and cloud forensics. Each of these four areas provides investigators with different kinds of information, with very little overlap. This makes it ideal to categorize the methods into these areas because many of the techniques for gathering and analyzing the information from these sources are unique to each source.

2.2.1. Host Forensics

Host forensics is often called digital forensics because it encompasses forensics done on "normal" devices, such as desktops, servers, and other non-specialized sources of data. This method has been long established, but the tools used are constantly evolving as technology is progressing.

Investigators can also utilize the method of weighting forensic evidence with blockchain technology. This can help with certifying the validity of digital evidence with it is presented in a court. This weighting system first collects evidence in a blockchain that records when the evidence was collected and who was in possession of it at the time. This data can then be categorized by relevance to the case and a timeline of events can be created (Billard, 2018). This method allows investigators to confidently show that evidence was processed correctly and was not tampered with. It can also be helpful with IoT forensics because of the large amount of information gathered in those investigations (Zhang, Li, Wang, et al. 2021).

A challenge that investigators face with host forensics is the randomization of kernel addresses. In order to face this problem, investigators can use four approaches to derandomize this information: brute force code, patched code, unpatched code, and read only kernel data. The brute force method simply scans the entire kernel code. For the patched code option, the kernel must know where to apply patches. The signature from this gives investigators insight into the organization of randomized address locations. The unpatched code signatures come from the code that has been identified as having not been patched. Finally, for read only kernel data, static pointers can help investigators shift data to find offsets, which will lead them to the proper address (Gu, Lin, 2016).

Another thing investigators must take into account when preforming forensics is the operating system of the device in question. Each operating system performs tasks differently and stores information in different places in the system, which affects all areas of digital forensics (Barmpatsalou et al., 2018). It is critical for investigators to be familiar with the many different types of operating systems in order for them to be able to gather all relevant evidence.

2.2.2. Mobile Forensics

As technology has developed, mobile devices have become more common. This means that mobile forensics is a critical part of investigations and should be understood by anyone in the field. Mobile forensics is distinct from any other kind of forensics because of the difference in "hardware, software, power consumption, and overall mobility," (Barmpatsalou et al., 2018). Furthermore, mobile devices are presumed to have personal data, which could be critical to an investigation.

Investigation Phases. There are four investigation phases in mobile forensics investigations: preservation, acquisition, examination analysis, and reporting. These phases are depicted in Figure 2.1.

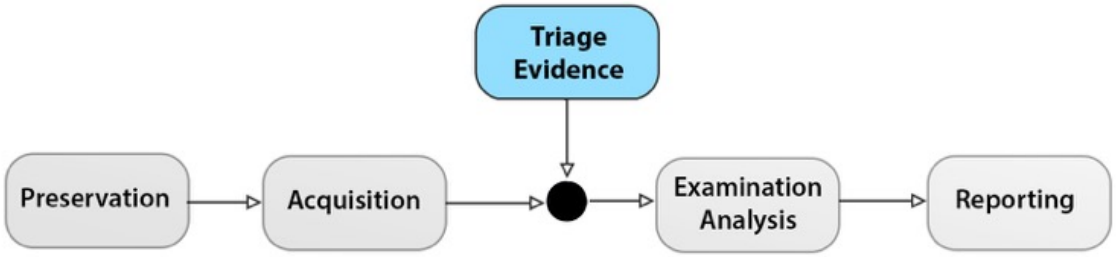


Figure 2.1: Mobile forensic investigative phases (Barmpatsalou et al., 2018)

The preservation phase is where mobile devices are taken by investigators and tracked to ensure that the data on them is not tampered with. The acquisition phase is where the data on the mobile device is copied to another device for the analysis that occurs in the examination analysis phase. Finally, the reporting phase is where all the information investigators uncover in the examination analysis phase is documented (Barmpatsalou, et al., 2018). Each of these phases must be followed properly to ensure the integrity of any investigation involving mobile devices.

Data Extraction. There are common collection methods, also called data extraction, used in mobile forensics. Data from mobile devices must be extracted during investigations. There are five levels of data extraction: manual, logical, hex dumps, chip-offs, and micro reads

(Chernyshev et al., 2017). Each of these options allow investigators to gather different information from different areas of the device with varying levels of complexity. Table 2.2 shows a comparison of these methods based on the criteria described in Section 2.1.2.

Manual extractions have the lowest complexity because this is where investigators interact with the device using normal methods, such as the touch screen. However, this method can be risky because investigators could accidentally damage or modify the data on the device.

Logical extraction is where investigators will extract data from the device to an external workstation using technology such as Bluetooth or a USB. This method also has a risk of inadvertent data modification. Logical extraction is a good method to begin with during an investigation because it allows investigators to analyze data from a different device.

Hex dumps require specialized tools to download the device's flash memory and allows investigators to access data remnants. It is a good way to read and analyze bits of data that may be residing between larger files. This method, however, can be difficult because it requires investigators to parse memory, which can be challenging and requires specialized training.

Chip-offs are where investigators physically remove flash memory and create a binary image of it that can help in traditional analysis. However, it presents the danger of physical damage to the device.

Finally, micro reads are the most complicated method out of these five. They use electron microscopes to analyze the logic gates in order to determine the readable data. This method is considered a "last resort" method because it is challenging and resource exhaustive.

As shown in Table 2.1, manual extraction, although easy to preform, is the least recommended because of the risk it poses to the data's integrity. The best methods are logical extraction and hex dumps. These analyze information from different places, so they give investigators a method of gathering different evidence that the other method does not access. Logical extraction and hex dumps have medium or low complexity, making them faster and more efficient to use. Finally, both of these methods pose a low risk to data integrity because they utilize a separate workstation for data manipulation.

Table 2.1: Comparison of the methods of mobile data extraction (Chernyshev et al. 2017)

Method	Complexity	Risk	Notes
Manual Extraction	Low Complexity	High Risk	Puts the integrity of the data at risk of accidental tampering.
Logical Extraction	Low Complexity	Low Risk	Utilizes an external workstation
Hex Dumps	Medium Complexity	Low Risk	Analyzes dumps of flash memory on an external device
Chip-offs	Medium Complexity	Medium Risk	Physically removes the flash memory
Micro Reads	High Complexity	High Risk	A last resort option because it is very complex and time consuming

2.2.3. Network Forensics

Network forensics is the practice of analyzing information from a host or an entire network (Caviglione et al., 2017). The forensic information can be obtained through logs or traffic captures. There are many ways this forensic data can be used, gathered, and evaluated. For example, determining the kill chain of an attack is critical for investigators who want to answer questions regarding how the attackers got into and used the network for their own purposes (Bryant & Saiedian, 2020).

Three of the layers of the TCP/IP Model can provide investigators with useful information. These layers are the application, transport, and network layers. The only layer not included in this is the network interface layer, which includes ethernet frames and the physical connections of a network. Forensic information can be gathered from the application layer through logs that hosts gather. This can be information regarding failed logons or timestamps, which could be critical information in an investigation. The transport and network layer are where firewalls are classified. Firewalls, if properly configured, can contain log data of traffic that has been dropped from the network (Caviglione et al., 2017). This can give investigators information about potentially malicious traffic that has been seen by the firewall. Figure 2.2 shows the relationship between the layers and the information that investigators can gather.

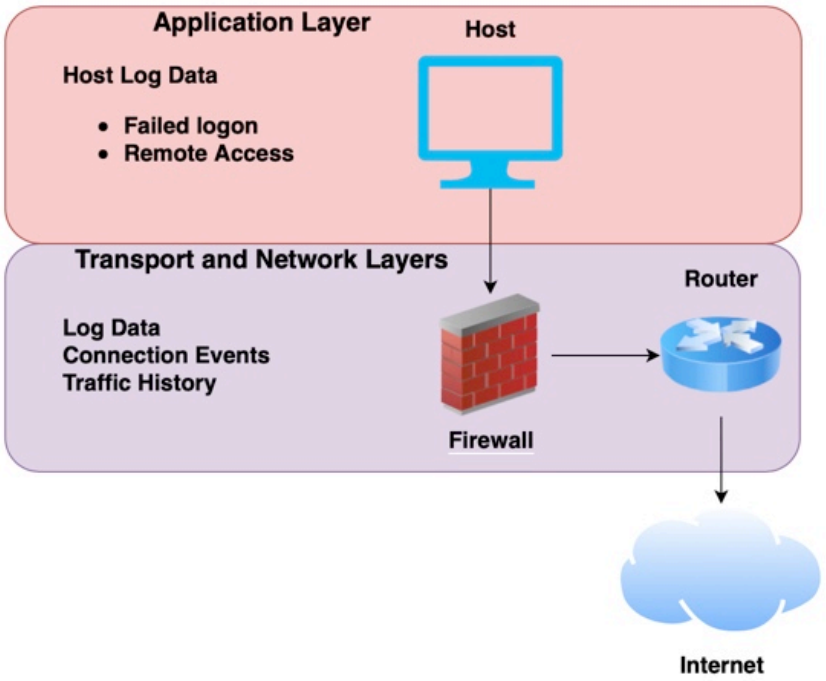


Figure 2.2: Evidence gathered from network layers

There are many ways attackers can gain access to the network, especially when more devices are being connected on the Internet of Things. A concept an investigator would have to consider is the network architecture and how these different approaches are used in different scenarios (Bryant & Saiedian, 2021). This can cause different kinds of information to present itself in different ways. Knowing the network topology can increase the probability of gathering useful information. There are many ways of securing these IoT devices, such as secured keying or a zero-trust model (Bhattacharjya & Saiedian, 2021). However, these topics are out of scope of this research.

The largest challenge facing investigators in network forensics is the amount of traffic and log data that can be present in an investigation. Although it is possible in theory, it is impossible in practice for investigators to collect and analyze every single packet in a capture of an entire network. The amount of data will not only take too much time to analyze, but it also poses investigators with a monetary cost (Caviglione et al., 2017).

Another challenge that investigators face with network forensics is the growing trend of the internet of things (IoT). These devices rely on networks to function, meaning there are more end hosts on networks that create logs and traffic. This not only increases the challenge of log and capture size, but it also complicates investigations when determining the scope of the investigation (Stoyanova et al., 2020).

2.2.4. Cloud Forensics

Cloud forensics is the practice of analyzing data from cloud services and infrastructure in order to gather information for an investigation (Manral et al. 2020). Cloud technology is becoming more popular among businesses and individuals. This means that it is a crucial area for investigators to understand.

Forensics as a Service. A new development that is changing the field of forensics, especially cloud forensics, is forensics as a service (FaaS). FaaS is a cloud-based service where an organization or individual will pay for the forensics services of another company, similar to cloud computing with providers, such as Amazon's AWS. FaaS is changing how forensics is being handled by moving it further into the cloud, which makes cloud forensics more important to understand (Barmpatsalou et al., 2018).

Methods and Frameworks. Manral et al. (2020) breaks cloud forensics into two sections, agent-based solutions and log-based solutions. Log forensics are more popular and widely used. These can be spread into four kinds of investigations: incident driven, provider driven, consumer driven, and resource driven investigations. Figure 2.3 shows the differences between these two methods.

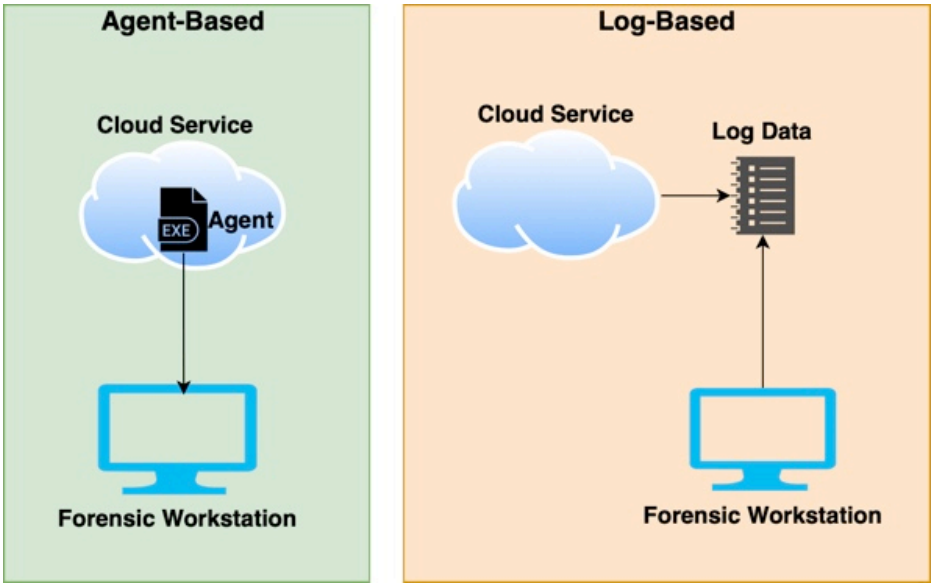


Figure 2.3: Agent-based versus log-based forensics

Agent-based cloud forensics relies on an application that collects information and sends it back to another location where it can be forensically analyzed. An example of agent-based cloud forensics is having an application, or agent, in the VM being used by the client that sends forensics reports to the investigators (Manral et al., 2020).

Log-based cloud forensics relies on logs created from events that occur in the cloud that can be then forensically analyzed. As Khan et al. (2016) discussed, cloud log forensics can be broken into three subsections: investigation, synchronization, and security. Investigation is focused on analyzing log files for vulnerabilities that could have been caused inadvertently or through malicious intent. Synchronization is focused on creating consistency across different log files from different sources. Security is focused on keeping log files safe from users that may harm the integrity of the data either inadvertently or on purpose.

Cloud Forensics and Mobile Devices. Cloud forensics and mobile devices are treated differently than other cloud and mobile forensic areas. Because of the growing trend to use cloud computing with mobile devices, investigators must account for the cloud aspects of investigations that involve mobile devices (Barmpatsalou et al., 2018). This provides a challenge to investigators because they must account for two different types of forensics during an investigation. One way proposed by Barmpatsalou et al. (2018) is continuous monitoring, where a monitoring system will track and report incidents on the device.

2.3. ONLINE INVESTIGATIONS

Online investigations are the process of gathering, structuring, and using information that can be obtained online. These can be performed by law enforcement, security professionals, or any individual. The main method for gathering information in online investigations is Open-source intelligence (OSINT). This method is the aggregation and use of the information that is gathered using other methods described in sections below. The information gathered for this type of investigation shows relationships, identities, or events that are relevant to the cyber investigation.

2.3.1. Sources of Information

There are many sources of information that investigators use in online investigations. The three main sources are the open, deep, and dark web. Each of these sources can provide investigators with valuable information. Figure 2.4 shows an illustration of these layers of the web. In the implementation of this research in Chapter 4, only the open web will be utilized when collecting information. Scraping the dark and deep web requires further research that is out of scope for this research.

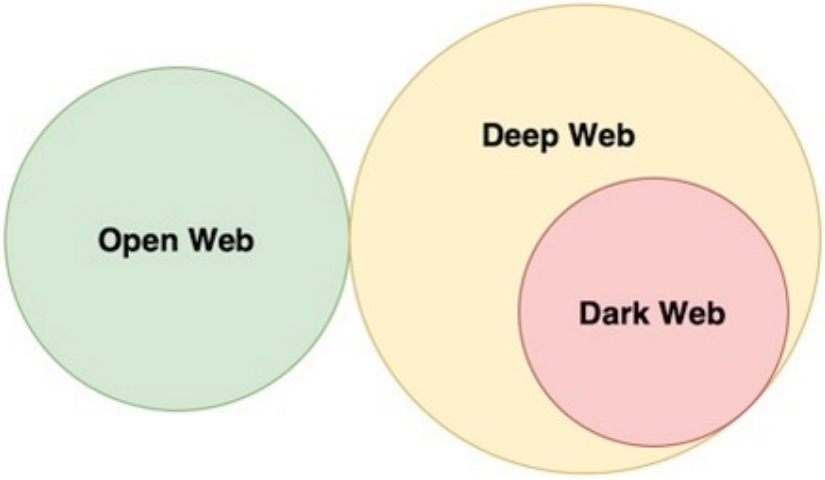


Figure 2.4: Layers of the web

Open Web. The open web is the part of the internet that is open to all users and indexed by normal search engines, such as Google. This searching method can provide investigators with helpful information. For example, there are some forums and chats that criminals may use that are available on the open web. When this information is collected and analyzed, it can assist in investigations. There is a wealth of information available on the Internet about crimes and criminal activity. This information can be mined and aggregated to help investigators learn about these activities.

Deep Web. The deep web is the part of the internet that is not indexed by normal search engines. This area of the internet is not necessarily illegal, but it can be. An example of a deep web site is any site that requires login credentials to access the content (Tavabi et al., 2019). This could be a simple news article or a streaming service that requires payment. The deep web can provide investigators with various types of helpful information. Some of this information could be chat logs that are linked to an individual's account. The information could help identify criminals and their social networks.

Dark Web. The dark web is the subset of the deep web where illegal activity occurs. It can be accessed using specialized software, such as Tor, that allows users to access servers, forums, and blogs that would be otherwise unavailable to users. Investigators can also use this specialized software to crawl the dark web for information (Tavabi et al., 2019). The dark web has information that can be useful to investigators in many ways. Nazah et al. (2020) discussed eight major information and crime types on the dark web: human trafficking, pornography, child pornography, assassination, drug selling, terrorist activity, cybercrime markets, and

currency exchange (Bitcoin). This information can be used in investigations to determine the identity, motivation, or even location, of the criminal.

2.3.2. Specialized Sources of Information

There are specialized sources of information that investigators can gather evidence from: social media and bitcoin flow. These sources can be accessed through the open or deep web, but they offer investigators with specific types of information that can be gathered as evidence. Much of the information gathered in this research comes from social media sites through the use of tools that search for information on these sites.

Social Media. Social media is one of the greatest tools that investigators have when it comes to online investigations. Information that is scraped from social media sites or profiles can assist in investigations. It has a wealth of information about organizations and personal lives, not only of regular people, but also criminals and criminal activity as well. As Nazah et al. (2020) stated, "to communicate and sell stolen identities, credit card numbers and other information, cybercriminals rely heavily on social media platforms such as Facebook, Snapchat Instagram, WhatsApp, Telegram and other social media platforms." This means that all this information, which could be critical to an investigation, is stored with these companies and can potentially be accessed by investigators.

Bitcoin Flow. Bitcoin flow is the task of analyzing records of Bitcoin associated with a crime. It is common for investigators to inspect financial statements in an investigation to determine who is involved in an organization and how a crime was financed. In recent times, many criminals have moved to cryptocurrencies to conduct the financial transactions surrounding the crime. By analyzing the flow of Bitcoin across the dark web, investigators can perform financial analysis. They can sometimes also determine the location of criminals based on transaction records and wallet addresses associated with individuals (Nazah et al., 2020). This could also be specifically helpful in financial crime cases.

2.3.3. Data Mining

Data mining is the practice of searching the web for information, organizing this information into a report, and using it in an investigation. Edwards et al. (2015) discussed five data mining methods that are used in investigations: natural language processing, information extraction, social network analysis, computer vision, and machine learning. Machine learning will be discussed in Section 2.4 of this chapter. Each of the other methods are described below. Out of the methods described by Edwards et al. (2015), natural language processing and social network analysis were the most commonly used in industry.

As described in Section 2.1.2, the methods used in data mining must meet several criteria. First, they must be more efficient for investigators to use. All four of the methods described in Table 2.2 meet the first criteria of being faster than manual searching. They are compared according to the second criteria in Table 2.2.

As shown in Table 2.2, natural language processing has more applications in the field than any other method, with fifteen relevant uses in cases. This is largely from the subcategories of this method that can be applied to the same types of cases but give investigators different types of information. For example, authorship profiling can be applied to cases of terrorism and extremism in order to determine attributes of the author, such as militancy, which gives investigators an indication of whether the author is a threat. Sentiment analysis can also be used in cases of terrorism and extremism because it gives investigators the ability to identify the emotions that the author of a text is experiencing, which indicates if the individual is, or is not, part of the criminal organization. The fact that two subcategories can be used on the same case increases the number of uses for natural language processing.

All the methods are applicable to cases in many different ways, as seen in the "methods of obtaining information" column. This shows that there are various ways investigators can gather and analyze the information used in these methods. However, as with the number of applicable cases, natural language processing is clearly the most applicable method for data mining and analysis.

Natural Language Processing. Natural language processing is the relationship between human languages and computing. There are four main categories within natural language processing: authorship analysis, author profiling, sentiment analysis, and text classification. Authorship analysis is the process of determining who the author is of a particular text. Author profiling is the process of analyzing a piece of text in order to determine the characteristics of the author. Sentiment analysis is the identification of the motivation behind a piece of text. Finally, text classification is determining where a piece of text falls in various predetermined categories (Edwards et al., 2015).

Authorship attribution is commonly used in online identification. This method allows investigators to determine the identity of the individual behind a piece of text. Authorship attribution can be done by clustering structural, linguistic, or orthographic features that appear in the text. This can help identify previous texts that match the clusters, helping investigators identify the author.

Author profiling is most often related to crimes against children. Author profiling uses common characteristics of language used by certain demographics to determine some characteristics of the author, such as age. Using this method, investigators can detect the ages of the authors, which can identify any mined chat data where one author is a child and the other is an adult. This can be an identifier of a potential crime.

Table 2.2: Comparison of data mining methods (Edwards et al. 2015)

Method	Types of Information	Number of Cases	Methods of Obtaining Information	Notes
Natural Language Processing	1 (text)	18	85	Contains four subcategories, each of which can be used in investigations
Social Network Analysis	1 (text)	5	22	Looks for relationships and patterns in user activity
Information Extraction	4 (URLs, technical sophistication, text, webpages)	6	39	Utilizes web crawling technology to look for crime trademarks
Computer Vision	3 (video, image, audio)	6	21	Searches images, video, and audio for criminal content

Sentiment analysis can be used for terrorism and extremism cases and harassment cases. In both of these case types, this method is used to detect emotion, the direction of that emotion, and the strength of the emotion. It is able to do this by categorizing the lexicon of the text and determining the tone of the writing.

Finally, text classification can be used in cases of crimes against children. Investigators can determine and then define the key characteristics of child abuse media. Then, they will be able to classify media according to these definitions. One of the most common ways to identify and define these media types is by commonly used keywords.

Social Network Analysis. Social network analysis is the use of technologies to learn about the network among criminal organizations and platforms. This method uses tools to scrape information about a criminal or terrorist organization and their connections from an online source (Nazah et al., 2020).

Investigators have various tools and methods they can use in analyzing social networks. One of the most useful methods is scraping data from blog posts and forums that are known to

have criminal activity. Another method used by investigators is mining data from graphical information, such as YouTube's social graph. This graph reveals connections between extremist videos and communities.

In social network analysis, it can also be important to employ natural language processing techniques, such as emotion detection (Zhang, Li, Ying et al., 2020). This can help investigators determine the relationship between individuals and groups, helping them know who is associated with the group and their activities.

There are two main areas where social network analysis is used in investigations: terrorism and extremism and criminal organization. Both of these crime types rely on the identification of organizations and attributing crimes to organizations.

Social network analysis can be used in terrorism and extremism cases when analyzing the activity found on dark web forums of individuals involved in these organizations. This will give investigators the ability to know who is in these organizations, who the key people are, and potentially give them insight into the past and future activity of the organization (Nazah et al., 2020). A method applied to this area focuses on the process of online radicalization, searching forums based in various locations to determine who the recruiters of an organization are.

When this method is applied to investigations into criminal organizations, the tools and methods are similar to those for terrorism and extremism cases. However, the sources of information are more likely to be news articles and other text-based pages. Investigators can search for terms, names, and geolocation data to learn about the social network that exists between criminals and criminal organizations.

Information Extraction. Information extraction is the process of automatically extracting and organizing information. This takes the methods previously described and organizes the information scraped from the web into a report. Information extraction is designed to reduce the time load on investigators.

The tools used for this method must be able to account for different interfaces, such as online databases. One of the common tools used for this method is web crawlers that will search for any page that is associated with a site and reports on the common topics on these sites (Edwards et al., 2015). This will give investigators the ability to learn about potential future targets and criminal events.

Information extraction is commonly used in cases of terrorism and extremism. Forums and websites that are associated with these activities are common sources of information for investigators. Investigators can scrape the forums and websites to learn what the common themes of these groups are.

Computer Vision. Computer vision is the practice of using images and videos that can be found online to gain information on a target or crime. This includes not only images, but also text and

audio found within a video. This method can provide information such as the identities and affiliations of users online, which can be used in investigations (Edwards et al., 2015). Chapter 3 discusses some available tools in AWS for implementing computer vision.

There are many different techniques that can be implemented in the practice of computer vision. One of the most used techniques where computer vision can be applied is in the identification of individuals through images, video, or audio. When identifying individuals, there are multiple methods available to investigators. One method of doing this is using facial recognition software on avatars that are generated from a photograph. This method is found to be accurate, but only if the user uses their own image to craft the avatar.

Another common usage of computer vision is spam filtering. Many spam emails present their messages as images to avoid spam filters. However, by using computer vision, these images can be inspected for unwanted content. This same concept can be applied to other types of content, specifically crimes against children, threats and harassment, and terrorism. Information gathered by investigators can be inspected for content that would be considered a crime, such as child pornography or threats made in a video.

The main criminal investigations where computer vision can be used are crimes against children, threats and harassment, and terrorism. When used in investigations involving crimes against children, computer vision can be used to detect images or videos that are suspected to contain child abuse content. In cases of threats and harassment or terrorism, computer vision can be used to identify the faces that appear in content that fall into these categories. transaction records and wallet addresses associated with individuals (Nazah et al., 2020).

2.4. NEW TECHNOLOGIES

There are several new technologies and areas of development in this field. Two of the fastest growing areas are automation and machine learning. Automation is the process of preforming a task automatically, without any human intervention. Machine learning, also known as artificial intelligence, is the use of algorithms that can be taught to recognize patterns or objects.

2.4.1. Automation

One area that has experienced development regarding automation is the ability for investigators to detect indicators of automatically. Liao et al. (2016) discussed a tool, iACE, that can be used to collect intelligence automatically from multiple sources and compare the relationships of the information gathered. This can be extremely helpful to investigators preforming online investigations. It will reduce the amount of time searching for relevant information and it will help generate reports with information relationships.

2.4.2. Machine Learning (AI)

Machine Learning as an Investigative Tool. Machine learning has revolutionized criminal tracking and investigations. It can be applied to investigations in various ways. Investigators can use machine learning to teach their systems how to recognize elements of crimes from sources like social media or surveillance footage (Raaijmakers, 2019).

This method can be applied to online identification. It uses machine learning to teach the system how to recognize what criminal organization may be behind a crime by what trademarks are seen in the crime. For example, different scamming organizations have different methods of operation. As Edwards et al. (2015) discussed, investigators are able to detect which organizations are behind scams using machine learning techniques.

Another use of machine learning can be in coordination with computer vision (Zhang, Li, Zhang et al., 2018). It can be applied to the process of detecting the identity of individuals in videos or images, helping investigators determine the identity of individuals associated with crimes.

Machine learning is most often used in terrorism and extremism cases and harassment cases. In terrorism cases, investigators are able to identify terrorists and terrorist activities by the online footprints of these organizations. For example, information acquired from a data mine of Twitter can be analyzed using machine learning to detect links within unstructured data (Edwards et al., 2015). In harassment cases, machine learning can be used to detect text-based threats. Edwards et al. (2015) discussed that threats in emails can be detected using a decision-tree algorithm.

Machine learning is also being developed in the field of digital forensics. AI can be used to analyze forensic material, such as network traffic and taught to look for patterns that indicate malicious activity (Du et al., 2020). This will make investigators task of analysis faster and less costly.

Machine Learning as a Criminal Tool. Machine learning can also be applied to the criminal aspect of investigations. Figure 2.5 shows a taxonomy for AI crimes. This taxonomy shows how AI can be used by criminals as a tool, but also as a target of their crimes. If criminals can harm a victim's AI systems, it could cause a lot of damage to the victim and their systems. Also, criminals can essentially teach their AI systems to attack the victim's systems, which causes the attack to be faster and more sophisticated than attacks done by individuals (Jeong, 2020). Because this method can be used by criminals, it is critical for investigators to understand this approach and know how to handle it during investigations.

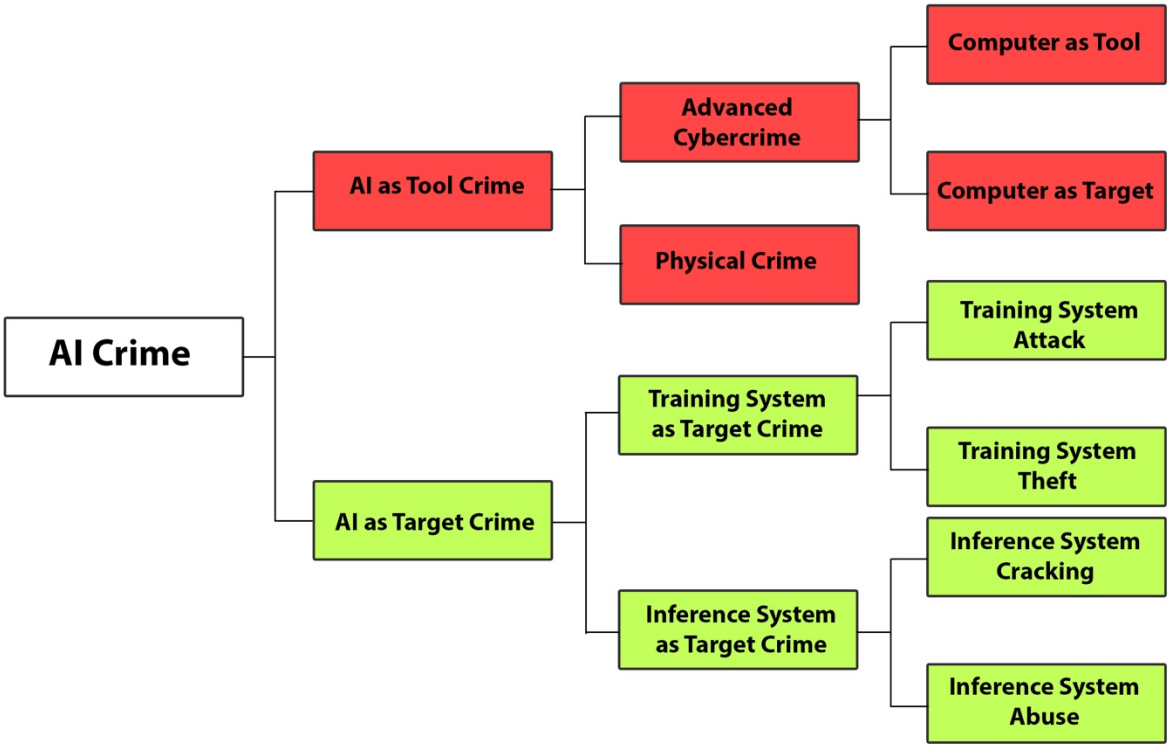


Figure 2.5: AI Crime Taxonomy (Jeong, 2020)

2.5. CLOUD ARCHITECTURE

This section describes outside research that influenced the decisions made in the architecture of this system. To do this, the researchers preformed a brief literature review of existing cloud architectures that involved AWS.

In order to design this system in the most effective way possible, the researchers reviewed other similar systems that used AWS in their architecture. Barua and Mondal's architecture for big data cloud systems can be seen in Figure 2.6. Their research proposed this architecture for mining data into the cloud. The purpose of their architecture is similar to that in this research. This is discussed further later in this chapter.

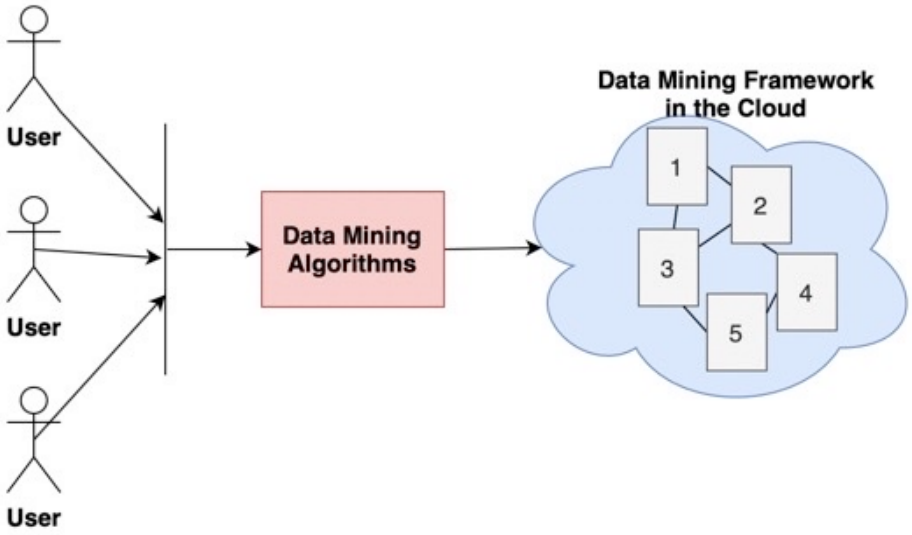


Figure 2.6: Big data cloud architecture (Barua and Mondal, 2019)

Some of the common decisions found in this research include disk and memory characteristics, CPU and RAM allocation, transfer port bits, and DMA. In Turan, Roy, and Verbauwhede's architecture, the researchers were concerned with CPU, memory allocation, and transfer port bits because their architecture was focused on encryption, and these were important elements that needed to be included. On the other hand, Kumbhare et al.'s (2015) research was only interested in CPU and network behavior because their research focused on cloud provisioning. For the scope of this research, only the memory and CPU will be discussed in Chapter 4 of this document.

Bermudez et al. (2014) used a layered architecture in their monitoring architecture. The elements they employed from AWS were S3 buckets, EC2 instances, and the AWS CloudFront monitoring tool. These elements were built into a layered architecture, the layers being measurement, analysis, and management. Each layer employed an AWS service that preformed the activities needed in the layer.

Barika et al. (2019) discusses the importance of determining the workflow of big data cloud projects. The data flow they identified was determining the stream of data, data aggregation, clustering, removing outliers, and pattern identification and matching. This data flow is helpful in this research project because it is similar to the data flow of OSINT investigations.

2.6. ARCHITECTURAL MISMATCH

Architectural mismatch is a concept in software architecture that discusses the importance of creating an architecture where all the elements match each other. This concept applies to cloud architecture. For a cloud architecture to match, the elements and components must be able to

communicate with each other in the same way that software architecture components must be able to communicate with other software components of the system. This communication, done between connectors, is how the architecture completes the task it is designed to complete by the architect.

Architectural mismatch can be identified as interface incompatibility and assumption mismatch. This is where a component is developed with certain assumptions regarding how it interacts with other components. When this component is removed from the original architecture and placed in another architecture, the assumptions of that component and the other components of the new architecture might not be the same. Assumptions that are not documented are called implicit assumptions, and these are difficult to identify and analyze. This mismatch in assumptions makes reusing components difficult.

Reusing components is an important concept in software architecture development. However, the issues of architectural mismatch make it difficult or impossible to reuse components. The assumptions of a component are not always stated by the developer, as described above, and sometimes the components do not come from the same source, which adds another layer of complexity. For example, open-source software come from different developers who have different backgrounds. They are not required to provide extensive documentation on the assumptions of the architecture of the component. This leads the assumptions of the component to become implicit, which could lead to architectural mismatch in future reuse of the component.

There are four types of architectural mismatch: component, connector, global, and construction process. Component level mismatch is where the assumptions regarding the components of a system do not align regarding the infrastructure, control model, or data model. Connector mismatch is where the assumptions of the connectors between the components do not align regarding the protocols or data model of the system. Global mismatch is where the system's components and how they support the global output of the system and whether a component should be included or not. Finally, mismatch with the construction process is where assumptions of the construction process itself do not match, such as the instantiation process (Garlan, Allen, & Ockerbloom, 1995). Each of these types of architectural mismatch occur because there were some assumptions made in the construction or the construction process of the elements of a system that were not documented or did not align with the assumption of other elements.

There are several solutions that software architects can utilize to avoid software architectural mismatch. These solutions utilize techniques that either prevent mismatch from the start of the design process or add patches to the mismatched elements to create a match between them. The first solution architects can employ is making assumptions explicit. This ensures that the assumptions of the components are documented for future reuse. The second solution is to use orthogonal subcomponents. By using these, architects can create

subcomponents that can function separately or in a different architecture. The third solution is to use bridging techniques such as using mediators or wrappers. These act as transformers between the components that do not share architectural assumptions. Finally, architects can avoid architectural mismatch by developing sources of architectural design guidelines. This will ensure that all systems built with these guidelines follow the same pattern and have the same assumptions (Garlan, Allen, & Ockerbloom, 1995).

2.7. OPEN ISSUES WITHIN OSINT AND MOTIVATION OF THIS RESEARCH

This literature review covers methodologies used in digital forensics and online investigations. One cannot be used without the other. The create an entire investigative program when used together. This literature review showed the current research in both fields and where more research is needed.

By studying cybercrime investigation methodologies, the researchers were able to determine the landscape of current research and identify gaps where more research is required. This research showed that there is a large amount of research in the field of digital forensics, both historical and current. However, there is not as much research into online investigations, specifically with open-source tools and effectively using them.

Open-source intelligence and online investigations are not a new method, but investigators are always applying new technologies to these methods. Multiple open-source methods in this field can be applied to online investigations in order to gain as much intelligence on threat actors as possible. Each of these methods provide some information, but none of them provide all the information necessary for an investigation. By building a connection between these disconnected parts, the researchers investigate the possibility of making OSINT investigations using open-source tools faster and more effective.

Open-source intelligence is a growing field and there are many opportunities for further research. Automation and machine learning provide potential areas of further research as these technologies become more sophisticated. Open-source intelligence techniques were also found to be underrepresented in the research field, opening another area for future research. This research into OSINT is being explored in the later chapters of this research.

CHAPTER 3: THE ARCHITECTURE AND IMPLEMENTATION FOR OUR CLOUD FRAMEWORK

3.1. AN INTRODUCTION TO THE FRAMEWORK FOR OUR CLOUD ARCHITECTURE

The system we designed focused on a social network analysis of a missing individual. This is done by discovering social media accounts belonging to the individual and collecting information from this account, such as followers, following, interests, and known locations associated with this individual. The architecture for a system like this must have a way to collect, store, and represent this data in a graphical fashion in order to depict the social network associated with the individual.

There are many challenges when designing the architecture of a new system. One of the best tools available to architects is documentation. It is critical to document everything about a system for future maintenance or changes. Cloud architecture provide particular challenges as they are completely remote from the architect and the entire network must be taken into account when building the system. For any architecture, the requirements, constraints, and steps for building the architecture must be defined.

When designing the architecture of a new system, the designers must first define the requirements and constraints of the system. The requirements are the basic functions, both functional and non-functional, that this system must accomplish for the project to be a success. The constraints are communication, storage, and other requirements of the elements that the system uses. These must be defined because the system designers must work around these when implementing the system.

The next step the architects is to make the final architectural decisions based on the requirements and constraints defined earlier. This includes how the elements interact with each other, as well as how the user interacts with the elements and the final product. A well-documented architecture ensures that future architects and designers can have a good understanding of the system.

Finally, the steps for building the architecture are defined, which includes defining the method of implementation and deployment. This makes the implementation phase simpler and less likely to deviate from the defined system architecture. It is critical that the implementation stays true to the defined architecture, unless there are necessary changes. Several conditions make changes necessary, including tools being unsupported or incompatible with the system. This will cause a necessary change within the structure or use of the system. These changes should then be well documented. This provides future developers for the system to understand the implementation, which makes system maintenance simpler and easier.

The constraints and requirements of the cloud architecture used in this research are described in this chapter. Each of the elements are also defined and described in terms of what kind of information they collect and the value they provide to the framework. The steps to build the architecture are also discussed. These are in relation to the elements and how they will interact with each other based on the requirements and constrains defined earlier in the chapter.

We also discuss the implementation of our architecture and the outcomes, challenges, and changes of that architecture. The researchers found several changes needed to be made to the original plan when implementing the architecture. This chapter discusses these changes, the cause of them, and the final decisions of the system.

This architecture required specific configurations of the tools and new automation to improve the efficiency of the proposed framework. This automation was built to automate the process of sending the information collected by the tools to the AWS S3 bucket. By doing this, the investigator will now need to enter only one command to run the tools and save the information into the correct location.

The outcomes of the implementation of the framework provided some challenges to the researchers. During the implementation, the researchers found that some of the tools that had not been supported for several years were not compatible with current technologies. There were several reasons for these challenges, including various differences between Linux distributions and unsupported elements of the tools. However, the outcome of this implementation also showed that other open-source tools worked as expected.

From these challenges, the researchers modified the proposed architecture to match the existing framework. There were several changes to the architecture that occurred during the implementation that were an outcome of the challenges encountered. These are documented later in this chapter. This new architecture is used in all the tests during the case study.

3.2. PROPOSED FRAMEWORK ARCHITECTURE FOR OSINT TOOLS IN THE CLOUD

This section describes the connection architecture for this framework and the implementation plan of this architecture. The architecture depicts the best way for the elements to be connected based on the requirements and constraints defined earlier. The implementation plan details any items that must be taken into consideration when building the framework.

3.2.1. Cloud Architecture

At a high level, the final architecture for this system is the deployment of an EC2 Amazon Linux virtual machine instance, which will then send information to an S3 bucket. This bucket will receive the formatted data and store it for the graphical relational database, AWS Neptune, to access and generate reports from. This architecture differs from Barua and Mondal (2019) in

that the mining algorithms are hosted in the cloud instead of locally. However, the idea of having the data collection framework in the cloud is similar to their research.

Adhikari et al. (2019) describe the differences between Infrastructure as a Service, Platform as a Service, and Software as a Service. Figure 3.1 illustrates the differences between these services. It is important to distinguish where this research falls under in order to better design the architecture. This framework for OSINT tools falls under both IaaS and SaaS. The infrastructure can be seen with the virtual machine and storage instances in AWS and the software service is the software available to the investigator on the EC2 instance.

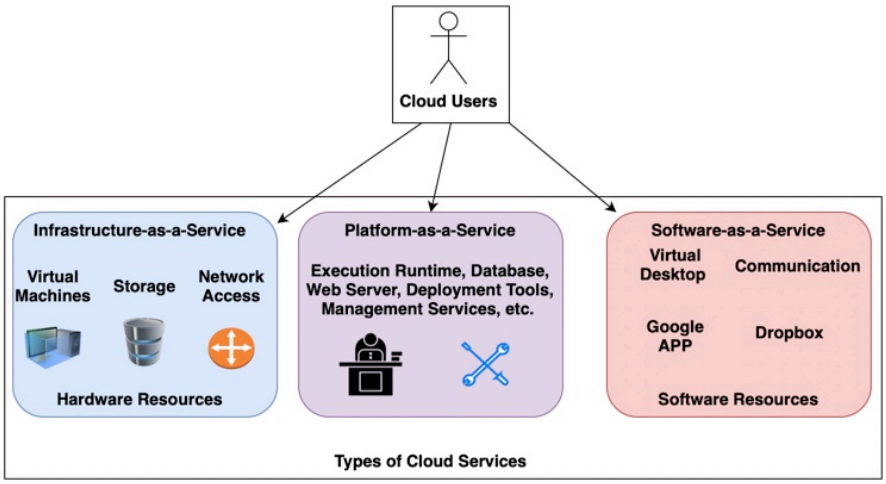


Figure 3.1: Cloud workflows (Adhikari et al., 2019)

Figure 3.2 shows a depiction of this system's architecture. There are some formatting requirements in the EC2 instance that must be followed before the information is sent on to the S3 bucket. All files sent to the S3 bucket must be in CSV format in order for this information to be read by AWS Neptune. This will ensure that AWS Neptune has access to all the information collected from all the tools. Much like Bermudez et al. (2014) research, this architecture the researchers developed is also layered. Each AWS service acts as a layer and only interfaces with the adjacent services in the architecture.

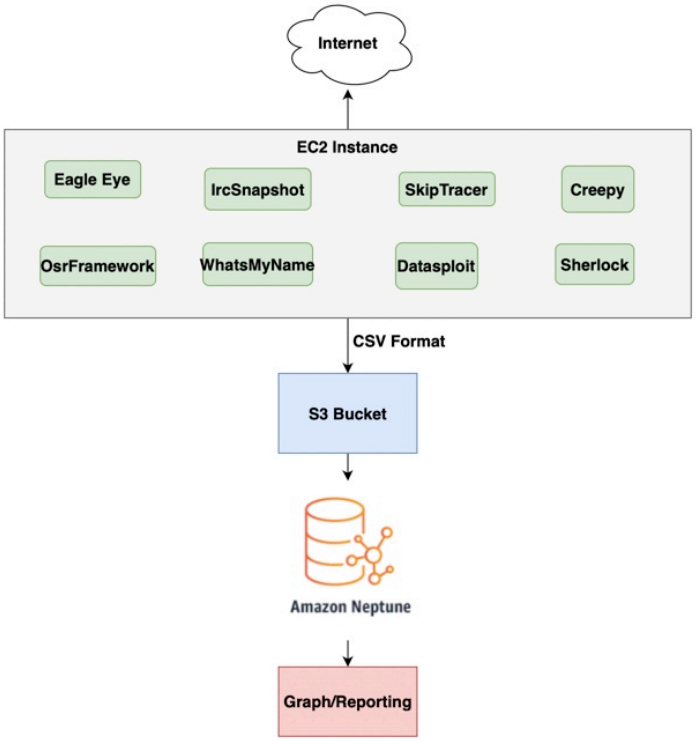


Figure 3.2: Architecture for the framework

Figure 3.3 shows a use case of how investigators will interact with the system. There are two parts of the system that investigators have access to: the EC2 instance and AWS Neptune. There is no need for users to have access to the S3 bucket. Investigators can access and run all the tools that are downloaded to the virtual machine on the EC2 instance. They can then send the collected information to AWS Neptune by using a connected Jupyter Notebook which will generate a graphical relation that they can analyze.

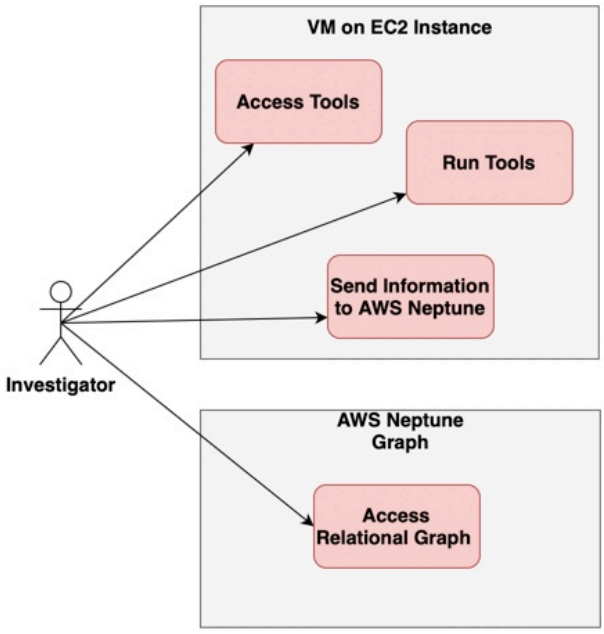


Figure 3.3: Use case for the system

3.2.2. Steps for Building the Architecture

This section defines the steps the researchers must take when building the architecture in the implementation phase. These steps define how this architecture should be implemented and any details that must be covered in the different areas of implementation. There are three main areas that the researchers identified: building and configuring the EC2 instance, configuring the S3 bucket, and configuring AWS Neptune. Each of these steps is based on each layer of the architecture described earlier.

Any time the permissions of user access or entity access must be configured or modified for any element of this framework, that can be done via the IAM role portal in the AWS web console. There, users can configure user roles, groups, or polices regarding entity access. AWS provides preconfigured roles, groups, and policies that users can edit to match what is needed by the architecture, or they can create these from scratch. We found it easier to simply create the roles and policies from scratch in order to define exactly what was needed from the role or policy without the chance of creating a misconfiguration. Misconfiguration of IAM roles or policies could lead to security and privacy issues within the framework architecture.

EC2 Instance. The EC2 instance is a Linux virtual machine running on AWS's resources. The selected tools are installed locally on this instance to make it easier to run these tools with as little human effort as possible. These tools will reach out to the Internet to collect information from social media sites. When the instance is created, a permissions file will be generated that

must be stored and used any time the user logs into the EC2 instance. The command to connect to the instance is shown below.

```
ssh –i <location and name of pem file> <username>@<public DNS name found in
AWS web console>
```

The tools and EC2 instance must be configured to collect the data in the required format. This includes ensuring that the information is being saved in CSV format. This will also require any of the tools that do not save the information in CSV format to be reformatted before the data is sent. To do this, either the tools have specific commands to save the data in CSV format, or the data files will be reformatted after collection. The tasks of collection and reformatting are automated by the researchers using a pipe and filter script when running the tools that require reformatting.

Finally, the information must be sent to the S3 bucket. For this to be properly configured, first the proper write access needs to be granted to the user. This is accomplished through the IAM portal in the AWS web console. Finally, the account must be configured so it maps to the correct EC2 instance, and a copy command must be run from the EC2 instance to the S3 bucket.

S3 Bucket. The S3 bucket is a simple storage location for large amounts of data. It must be configured with the proper access privileges to receive the data that is being sent from the EC2 instance. This access is defined as write only into the S3 bucket. These configurations are done in the IAM portal in the AWS web console where a new rule can be created and defined with the correct S3 and EC2 information.

Also, it must be configured to send the data to AWS Neptune according to the system's requirements. For this to be properly configured, the MIME type must be application/json. There is a specific command that needs to be run in order to upload the data. This command has several options, such as role, source, and format. These determine where the data will go and what permissions the user has.

Finally, the data within the S3 bucket must be configured based on the requirements of the Neptune database. There are several open-source tools found on GitHub provided by AWS that assist with ensuring the proper format is achieved before the data is loaded into the database. This ensures that Gremlin, the querying language used, can properly index and find data that is loaded into the database. This ensures that all the data collected as part of the investigation is retrievable in the database.

AWS Neptune. After the AWS Neptune instance is created, it must be configured with the proper permissions to receive the information coming from the S3 bucket. This then generates the graphical representation of the data. The connection between the S3 bucket and the EC2

instance is made through the use of Jupyter notebooks in AWS's SageMaker tool. Jupyter Notebook is a tool that runs various commands on the associated machine or database. There are also APIs available to extract the data, but that is out of scope for this research.

Neptune has a data loading and querying language option called Gremlin. In order to use this format, the files used to load the data have to follow specific guidelines, which will be discussed later in this chapter. Gremlin also gives the option to load single data elements (vectors), mass data files, and make connections between data elements (edges). These elements and connections are stored in the database and can be queried using Gremlin functions. The most important functions show the connections between data elements, such as the outE() and inV() functions that show the connections between vectors and edges. However, the most useful script that can be created with Gremlin is one that finds connections between vectors by traversing the entire database to find a connection. This method is depicted in Chapter 4 of this research.

3.3. CLOUD PLATFORM ARCHITECTURE REQUIREMENTS AND CONSTRAINTS

In order to successfully build this tool, designers must determine the requirements of the system and the constraints of all the components. These two items determine how the architecture must be designed because the requirements determine the functionality of the system, and the constraints define how the elements of the system must work in order for them to be functional.

3.3.1. Requirements

There are several requirements for this system that the architects must follow. These define the basic functionalities of the system including how the users interact with the system, the functionalities of the system, and the output of the system. Each of the elements of the system and the structure of the system must follow these requirements in the architecture of this framework.

Information on Individuals. A critical requirement is that the tools used must search for information on individuals that is relevant to missing persons cases. This includes names, usernames, emails, locations, and events. All the tools selected must search for at least one of these types of information. By gathering and aggregating this information, investigators are able to find connections relating to the missing person that could be helpful in the case.

Centralized. The connection of tools must be centralized. This means that all the tools send their reports to one location, which will then send all the collected information to AWS Neptune for graphical analysis. By centralizing the collection of information, the amount of

work required of investigators to analyze and report on their findings is reduced, allowing for more time that investigators can use to collect more information.

AWS Neptune Communication. The system must send all the information collected to AWS Neptune. The use of AWS Neptune is part of the novelty of the system, making it critical that all the elements can utilize this service. This tool allows for faster analysis of relationships in the gathered information, making investigations faster and more efficient.

Single Report Generated. This tool must generate a single report or graphical analysis for the investigator. In this case, this report is in the form of a graphical report from AWS Neptune. There are also APIs available for pulling data from AWS Neptune, but these are out of scope for this research. Having a single, automatically generated report that contains all the information will make investigations faster and will reduce the likelihood that information is lost.

Free Tools. All the tools used in this connection must be free and open source. The purpose of this connection is to make OSINT more accessible to more investigators. This means that the tools must have a low cost for the researchers and developers. Having a lower cost will allow more investigators to utilize the tool.

Internet Access. In order to gather information from online sources, this connection of tools must have access to the internet, and it must allow inbound traffic and downloads. The tools must search online forums and resources and download the information that is collected. The internet access of these tools must be properly configured to allow for this without making the tool insecure and vulnerable to online attacks.

3.3.2. Elements and Their Constraints

Architects must consider the constraints of the elements in a system before implementation. This is because these constraints often determine how the system can be implemented. Each element has its own constraints that are defined in this section. This will also help evaluate each of the intended elements and if they can be used in the system.

This section includes all the tools that are proposed to be used in this system as described in earlier sections. They will be evaluated in this chapter and tested for functionality in Chapter 4. It will be determined if they can be used in the system based on the requirements and constraints of the system. This section will also provide a description of the elements of the system and the value and information they provide to investigators.

Many of these tools overlap in the information they gather. This is designed on purpose. Some tools only search specific sources for the data types, which makes it critical to use as

many tools as possible to find all the relevant data from all online sources. It also helps verify the information gathered by the tools.

AWS Neptune. AWS Neptune is a graphical relational database available through Amazon Web Services. This tool gives investigators the ability to input data and determine relationships through the use of a connector graph. This is helpful in investigations because it can speed up the process of determining connections between the data gathered in an investigation. By doing this, the entire investigation process can become more efficient, which will make the investigator's work more effective.

AWS Neptune has requirement constrictions that this tool must conform to. Figure 3.4 shows the architecture of AWS Neptune as defined by Amazon. It shows that AWS Neptune requires information to be sent into it from an S3 bucket containing data in CSV format. Since AWS Neptune is such a critical component to this entire connection, all other tools must be able to output their information to CSV format in order for them to be integrated with the system at large.

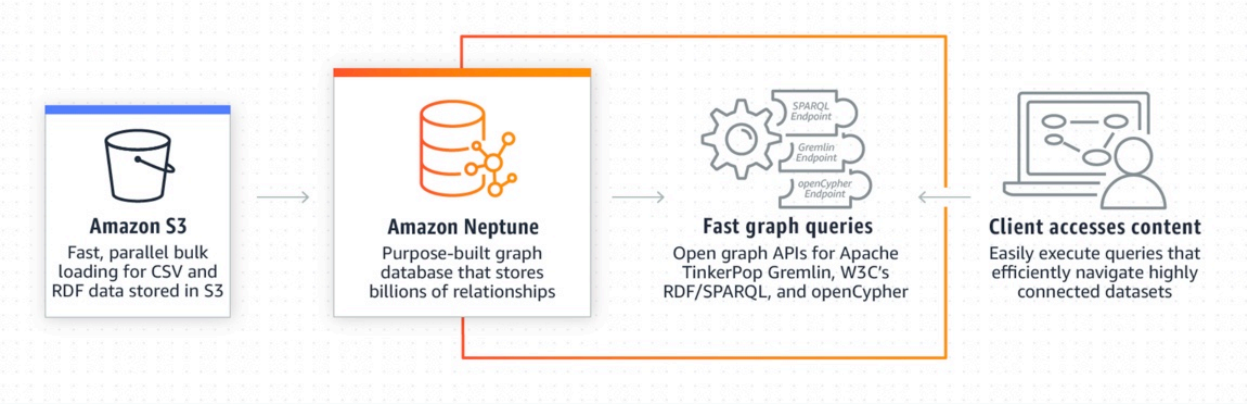


Figure 3.4: AWS Neptune architecture (Amazon, n.d.)

Creepy. Creepy is a tool that collects geolocation data from various online sources. It collects both location and date information from the sources. Creepy was chosen because the information it collects can be helpful in the missing persons case study. For example, social media post dates and locations can show where an individual was during, or around, the time in question, or it can show patterns of behavior.

Creepy can be locally downloaded and it requires installing a specific version of python and other libraries. These are listed here:

- Python-qt4
- Pytz
- Flickerapi

- Python-instagram
- Yapsy
- Tweepy
- Google-api-python-client
- Python-dateutil

This causes no issues in the implementation as these can easily be installed to the Linux machine. After it is downloaded, the command to run this tool is “python CreepyMain.py”. Creepy shows the information gathered on a map, but it can also export the location information it gathers in CSV format.

One concern of using Creepy is it has not been maintained since 2016, so some sites may not be compatible, such as Twitter, with it as it is gathering data. This will be explored further in the implementation chapter of this research. There may be a need for configuring the tool to work with the different social media APIs.

Sherlock. Sherlock is an OSINT tool that searches for a specified username to find other accounts on public websites by the same name. Searches can be performed on multiple usernames at the same time, increasing efficiency. This tool can be used after other tools are used to determine the relevant usernames to the missing person’s case. This information can be helpful to investigators because it can help identify potentially relevant information on otherwise unknown accounts.

The Sherlock tool can be locally downloaded to the Linux virtual machine. It can also export the collected data in CSV format. After it is downloaded and the docker image is built, the command to run this tool is “python3 sherlock” with the name of the user or any other options the investigator wants to use. There are a number of options available to the investigators with this tool. These depend on the information that needs to be collected and should be chosen as the tool is used.

Datasploit. Datasploit is a tool that crawls the Internet for phone, email, and username information. Not all of the options in Datasploit are relevant to a missing person case; however, some of the functionality can be utilized by investigators to gather useful information. This tool can be helpful in an investigation because it can help verify the same types of information gathered from other sources, as well as potentially gathering new information.

This tool can be locally downloaded to the virtual machine; however, it does not save its information in CSV format. It saves the information as a text file. This output will have to be converted to CSV format before it can be sent to the S3 bucket. After it is downloaded, the command to run this tool is “python datasploit.py.” There are many options for this tool that investigators can use depending on the information the investigator is looking for.

WhatsMyName. WhatsMyName is a tool that performs "user and username enumeration on various websites." This means that it will search various social media sites, and other websites that have public usernames, for a known username and then searches for other variations of this username. The information it gathers can help investigators because it provides investigators with a way of searching for other possible accounts linked with the case. Users often cannot always use the username they always use. This means it is critical for investigators to be able to search for other possibilities quickly.

WhatsMyName has two scripts that can be locally downloaded to the Linux virtual machine. Both of these gather the same kind of information, but both will be used because they check for the information in different ways. There are a number of options available to the investigators with both of these scripts. These depend on the information that needs to be collected and should be chosen as the tool is used. These scripts export the collected data in CSV format.

OsrFramework. OsrFramework is a tool that provides investigators with many OSINT tools such as DNS lookup and username search. Many of this tool's options are out of scope for this research. However, there are some options that can be helpful to investigators in the missing person case study. Like WhatsMyName, OsrFramework is another tool that can look up usernames and search for possible enumerations. It can also find information about emails. If an email is known to be associated with a case, this tool can help search for more information connected to that email address.

The default export option for this tool is as a csv file. It can also be locally downloaded and run. After it is downloaded, the command to run this tool is "python osrf." There are a number of options available to the investigators with this tool. These depend on the information that needs to be collected and should be chosen as the tool is used.

IrcSnapshot. IrcSnapshot is a tool that gathers information from IRC servers. These are servers built for instant messaging and has chat log data freely available in public discission forums. This tool can check for specific chat channels or for specific usernames and scrape information from these. The chat data that IrcSnapshot provides investigators can be useful in locating a missing person by determine their intentions or any individuals they were communicating with before they went missing.

IrcSnapshot can be locally downloaded and run from the virtual machine. This tool does not use CSV format in its exports, but rather a JSON file. The output of this tool must be converted to CSV format before it is sent to the S3 bucket. There are a number of options available to the investigators with this tool. These depend on the information that needs to be collected and should be chosen as the tool is used.

Eagle Eye. Eagle Eye is a python tool that searches for a person based on an image of that individual by using facial detection and recognition. This can be helpful in identifying social media accounts that potentially belong to the individual but are under a different name. This could give an investigator a lead into the activity of the missing person and what locations they may have frequented.

Eagle Eye requires the use of Firefox and docker, which can be downloaded to the virtual machine. The Eagle Eye program can also be downloaded and run locally. It is unknown at this time if the output is in CSV, or if it will need to be reformatted. There are a number of options available to the investigators with this tool. These depend on the information that needs to be collected and should be chosen as the tool is used.

SkipTracer. SkipTracer scrapes websites for personal information, such as addresses, phone numbers, and even license plate numbers. This tool has a multitude of modules that can gather valuable information and verify information found with other tools described earlier in this section. The kind of information SkipTracer gathers can be invaluable in an investigation to find a person. It can help determine the information of the people the missing person is associated with, which could lead to the uncovering of more helpful information.

This tool can be locally downloaded and run from the virtual machine. However, this tool generates its reports as a docx file, which must be converted to CSV before it is sent to the S3 bucket. There are a number of options available to the investigators with this tool. These depend on the information that needs to be collected and should be chosen as the tool is used.

A concern found with SkipTracer is it is not actively supported at this time. However, it is being migrated to python3, so this unsupported status may change. If it is not maintained for a long period of time, its compatibility with online sources may deteriorate and the investigator will not be able to gather information from these sources. While it is still useful, the researchers added this tool to the connection, but the usefulness of the tools should always be monitored for unsupported tools.

3.4. SCRIPTS AND CONFIGURATION SPECIFICATIONS

The researchers noted several configuration specifications and scripts were necessary to make this tool more efficient. The AWS configuration specifications that the researchers documented in the implementation of this solution were necessary for ensuring the tool would work properly and have proper access. In order to automate the process of collecting information with this framework, the researchers produced three scripts, which are run on the EC2 instance, to run the tools, format the outputs, and send the outputs to the S3 bucket. These scripts use existing functions to reduce the amount of rework required of the researchers. The scripts use a pipe and filter architecture to make this reuse work.

Some of the AWS configuration specifications included access management configurations. The researchers had to ensure that the EC2 instance had inbound traffic rules that allowed for the current source IP being used. Because of outside network configurations with DHCP, this would change every time the network changed, or the researchers logged back into the network. After these permissions were configured, we needed to access the box to run the commands. In order to reach the EC2 instance, we had to ssh into the box via the public DNS found in the AWS web console, which allowed us to access the command line. By using the command line for all the tools and scripts, the process was more efficient and easier to automate.

The first script runs all the tools used in this framework. Originally, we planned to use the alias generator to create a list of potential aliases as the first element of the script. However, we found that this tool only works if it is run by itself without any of the flags used. This can be helpful in creating a separate list, but it does add an extra step to the collection of information. The second and third functions checks for usernames using Sherlock and WhatsMyName. Finally, it utilizes IrcSnapshot to search for the individual's name or username on public irc chats. For each of the tools that outputs to a .txt file, these outputs are being converted with a script we created. Below shows the first script we tested, the changes from the alias generator, and the converting script.

```
#original script
$ python3 alias_generator.py -n <name>  -s1 <surname1> -s2 <surname2> -c
<city> -C <country> -y <birth_year> -o <output_file> | txt_to_csv.py |
python3 sherlock.py <username> --csv | python3 web_accounts_list_checker.py -
u <username> -o | txt_to_csv.py | python ircsnapshot.py -r <name> |
txt_to_csv.py

#current script
$ alias_generator (follow the prompts from the program)
$ txt_to_csv.py | python3 web_accounts_list_checker.py -u <username> -o |
txt_to_csv.py | python ircsnapshot.py -r <name> | txt_to_csv.py | python3
sherlock.py <username> --csv


#txt_to_csv.py
#Command:
$ python3 txt_to_csv.py

#code
from signal import signal, SIGPIPE, SIG_DFL
signal(SIGPIPE,SIG_DFL)
import pandas as pd

textfile = parser.add_argument("-tfile", "--testfile", help="Text File Name")
csvfile = parser.add_argument("-csvdest", "--csvdestination",
help="Destination Location for CSV File")
```

```
read_file = pd.read_csv (r'textfile')
read_file.to_csv (r'csvfile', index=None)
```

The second script moves all the output files into one directory and then moves that directory to the S3 bucket. The recursive tag ensures that all the files in the directory are moved to the S3 bucket. The users of this script must ensure they are using the correct file names and the correct path to the S3 bucket. This script ensures that all the files collected from the script described above are send to a single location for investigators to aggregate the data. This script is shown below:

```
#script to send to AWS S3 bucket
mkdir data | mv <ouputput files> data | aws s3 cp data s3://thesis-
s3bucket/collected-information-s3/ --recursive
```

It is necessary for us to define the workflow of the scripts withing the EC2 instance. Figure 3.5 shows the workflow of the framework as it is collecting the data and sending it to the S3 bucket. Each tool either saves the data in csv format or txt format. Those in txt format are converted to CSV format. All of the files are then moved to a single directory which is uploaded to the S3 bucket in one mass export. This consolidation of information helps in keeping all the data gathered organized and easily accessible.

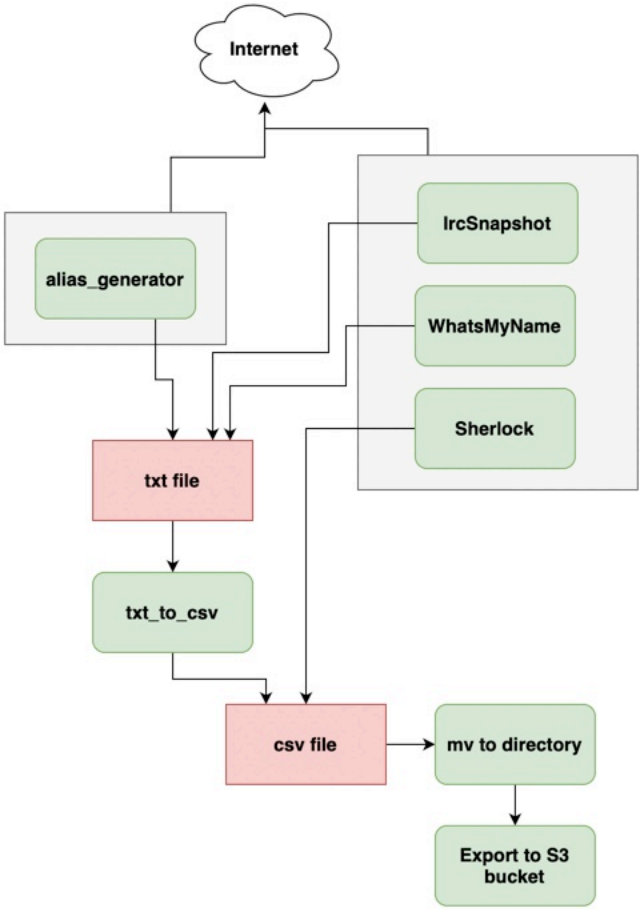


Figure 3.5: Workflow of the architecture and scripts within the EC2 instance

These scripts automate the first and second layer of the architecture (collecting in the EC2 instance and sending to the S3 bucket) as shown in earlier in this chapter. This automation improves the efficiency of using these tools in the architecture created through this research. The next step is then ensuring AWS Neptune has access to the S3 bucket. This is configured in the IAM portal in the AWS web console. To properly create this new role, the trusted entities in this role are S3 and RDS (relational database service), with an access of read only to ensure the entities have access to see the data, but not modify any data.

To load the data to AWS Neptune, a Jupyter Notebook must be created and attached to the database cluster. This allows the user to use commands to load, retrieve, and query data in the database. To bulk load the data, the %1oad command must be used. This will prompt the user to enter the source of the data (the S3 bucket), the load ARN that determines the permissions attached to the specified bucket, and any other load settings the user can modify. For this to successfully occur, the csv file must be in the proper format. One way to determine this is by using the amazon-neptune-tools GitHub repository which has a script that allows users to check if the file is in the proper format.

3.5. OUTCOMES OF THE IMPLEMENTATION AND UNEXPECTED CHALLENGES

There were several outcomes and challenges the researchers faced when implementing this solution. Several tools evaluated in Chapter 3 had to be removed from the framework for compatibility reasons. Only the applications that were able to be run in the command line script described in the section above will be used in all parts of the case study. These changes from the original plan are described in Section 4.4 of this chapter.

Some of the tools, as noted in Chapter 3, were depreciated, or had other compatibility issues and did not work with current configurations. This included Creepy, Datasploit, Eagle Eye, and Skiptracer. As was mentioned in Chapter 3, the creator of Creepy has not kept the application up to date and it is not compatible with current social media APIs. Datasploit, like Creepy, has not been update in several years and is not compatible with current technologies. Eagle Eye was a promising tool; however, it was not compatible with the EC2 setup in AWS. Skiptracer had several errors that occurred from python versioning errors. These tools were removed from consideration. Also, some modules of OsrFramework did not work and were currently under development by the contributors. These modules were also removed from consideration but may be added in later iterations of this project as it is developed in the future.

These issues are unfortunately common in open-source tools and is something investigators who wish to use these tools must acknowledge. Open-source tools rely on crowdsourcing the development of these tools, and this often leads to misconfigurations or depreciated tools as other external sources are updated. The investigators or analysts can modify these tools to try and resolve these issues, but this takes time and effort away from the investigation with no guarantee of success.

Also, we wanted to avoid architectural mismatch by using tools that could not be compatible with the architecture's requirements and constraints. Architectural mismatch is something that should be avoided because of the long-term effects on the system. It focuses on how modules interact with components and connectors, which are the bedrock of the architecture. If tools cannot work cohesively with the rest of the elements, then the connections break down over time as the system is updated. There are many different kinds and causes of architectural mismatch, such as the nature of the components and connectors, the global architecture, or the construction process of the system. Each of these can lead to mismatches within the architecture that must be avoided.

The mismatch we discovered with these tools was with the nature of the components themselves not matching the architecture of the system. These tools did not match the architecture's requirements and constraints based on their output, modules, Linux commands, or incompatibility with other elements of the underlying system. The underlying infrastructure of our system did not match the underlying infrastructure of these modules. The components

themselves did not match the component architecture, even if the tool's output matched the global architecture.

We were unable to use any of the solutions discussed in Chapter 2 for architectural mismatch. The solutions that discussed following a design plan or component construction were not applicable because we were trying to reuse existing components and new development on the existing tools was out of scope of this research. We made architectural assumptions explicit when we defined the requirements and constraints of the system. The tools we chose matched the global architecture of the system, but the underlying requirements of the tools did not match the underlying requirements of the system. This is something we did not know and had no way of knowing before testing. Finally, bridging techniques were not practical for the solution because a new and different bridging technique would be required for each element that did not match the architecture of our system.

The tools that were used in our architecture were Sherlock, WhatsMyName, IrcSnapshot, and one module of OsrFramework. These tools, used in combination with each other, can help the investigator determine the digital footprint of the individual in question. They provide an overview of potential online usernames and accounts for the individual, and this data is linked together within AWS Neptune. These components' architecture matched the architecture of the system, and their output matched the global architecture.

3.6. CHANGES TO THE PROPOSED ARCHITECTURE

This section discusses changes to the architecture of the framework that was proposed in the section above. These changes caused us to need to modify the architecture plans in order to accurately represent the architecture. The changes mainly involved incompatible elements within the open-source tools, which caused us to discontinue the use of these elements in this framework in order to avoid architectural mismatch. These changes have changed the final architecture, which are discussed in this section.

Although there were some changes required of the architecture, the basic principles surrounding the decisions of the architecture are the same. Figure 3.6 shows that the framework architecture was greatly not altered. It still uses a layered approach, and the layers still use the same technologies. The only changes needed were in the tools used to collect information. This lack of change to the underlying architecture is a good thing because it shows that they layered approach with AWS resources is the most efficient. It allows elements or layers of the implementation to change as needed without affecting the other elements or layers.

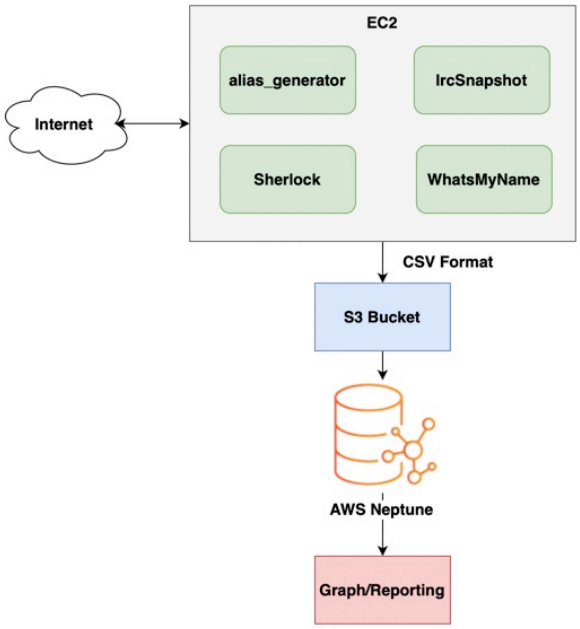


Figure 3.6: Final framework architecture

The workflow that this architecture uses is described in Figure 3.7. It shows each step of the process and which phase of the architecture where this process occurs. The data collection phase occurs in the EC2 instance. Within this phase, the framework of tools is used to collect data artifacts. These artifacts are then formatted and moved to a centralized location. This is where phase two begins and the S3 bucket is utilized. The centralized directory is exported to a repository in the S3 bucket, and that repository is then exported to the Neptune instance. The final phase of the architecture begins with the creation of connections between elements that were uploaded from the S3 bucket. Finally, investigators can query the database to find connections between elements of the data and generate a console and graphical report from this information.

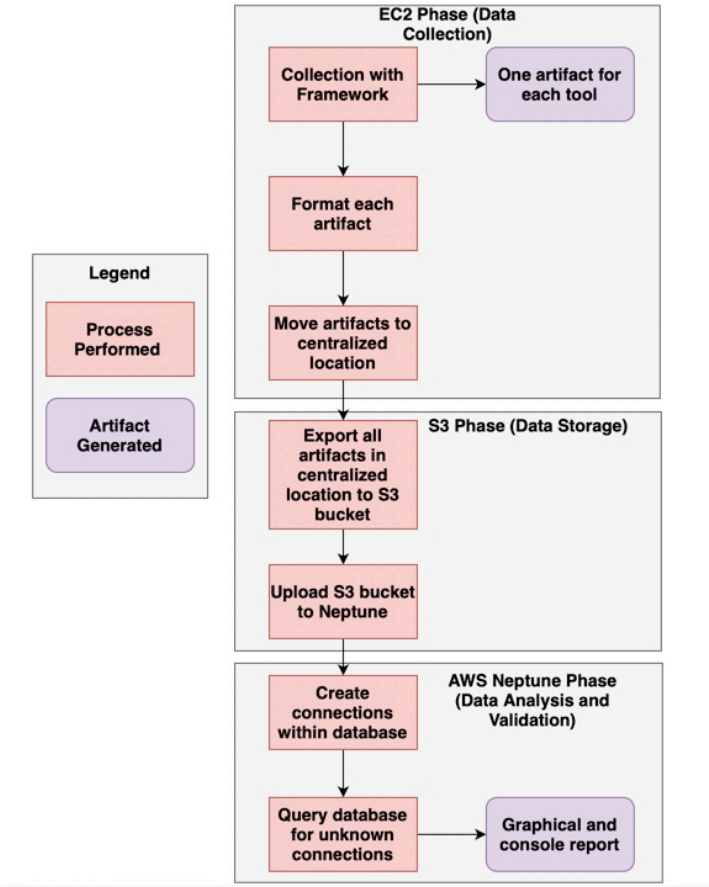


Figure 3.7: Workflow of the architecture

3.7. FUTURE OPPORTUNITIES FOR EXPANDING THE ARCHITECTURE

There are many opportunities for future expansion of this tool that were found in the literature review, including quality attributes such as performance and efficient use of resources. This section will briefly describe these future opportunities to expand this tool and improve its functionality and usability. Each of the areas of expansion described here will ultimately improve the efficiency and effectiveness of the investigative process. However, they are out of scope of this research, which is OSINT efficiency and effectiveness.

The performance of this tools is out of scope for this research. However, measuring this is a good opportunity for future building of the tool. Sun et al. (2016) discusses a tool in AWS, AWS CloudWatch, that can be used to track performance metrics. Because this tool is native to the AWS environment, it would be a good option to integrate it into the overall architecture to gather and track these metrics.

From the metrics that can be gathered by AWS CloudWatch, future research and development can be done into the quality of service, specifically in the efficient use of the elasticity of the cloud. Elasticity is the automatic allocation of cloud resources based on

demand. It is critical to ensure this is efficient for performance and cost. Qu, Calheiros, and Buyya (2018) discuss a method of automatic scaling of cloud resources. This research would be critical to any future development into this area.

Finally, to expand this tool into other areas of online investigations, cloud vision and voice sentiment analysis services can be added to the architecture. Cloud vision is similar to computer vision, which is discussed earlier in this research. AWS has an image detecting service which can be leveraged to implement this service. Li et al. (2019) discusses how this can be implemented in a cloud architecture. This can help investigators automatically identify individuals in images, which can help speed up the investigative process. Voice sentiment analysis is a service provided by AWS Comprehend, as described by Satyanarayana, Bhuvana, and Balamurugan (2020). As with the cloud vision service, it can be easily integrated with the existing architecture because it is an AWS service.

CHAPTER 4: CASE STUDY ON MISSING PERSONS CASE INFORMATION

4.1. TESTING THE INFORAMTION GATHERED

This case study is designed to test the framework of tools in order to determine if the framework is more efficient than manual collection, and if so, how much more efficient is it. This will be done by measuring the metrics defined in this section. By using this defined way of testing the framework, the researchers will be able to assess the usefulness of the framework and if it can be used in practice. The second metric we test this framework for is its efficiency. This is the amount of useful information that is gathered and identified by the framework during the collection tests.

There are several requirements for the tests performed in this case study. The tests must demonstrate the effectiveness and efficiency of each particular method, and each method should be tested equally. To do this, three sets of data will be collected as a part of this case study: time, volume, and ease of analysis. Each of these areas is critical to any cyber investigation that uses open-source intelligence, as they lend to the effectiveness of the information gathered through OSINT methods.

In each phase of the test, the time for collection will be measured. Before the collection of information begins, the time will be noted. Then, the information will be collected without stopping. Once this is complete, the end time will be noted, and the difference will demonstrate the amount of time for the method being tested. If there is any break in the testing time, it will be noted, along with its duration. The noted time will be catalogued for comparison against the other methods tested in this case study.

For each phase, the amount of information collected will also be measured. With each method that provides a final document, this final document can be compared for volume based on a line-by-line output. Duplicate information will be left in the final documents because some of the tools used collect overlapping data. This helps test the validity of the information. If there is not a line-by-line comparison metric available, then file size is compared to determine the amount of data collected by the method.

These two sets of data are compared side by side to determine the differences between the collection methods. This shows the differences between the methods, and where the strengths and weaknesses of the methods lay. The better method will have a smaller time-to-collect and a larger volume of information collected. The comparison of the data in this case study is made later in this chapter.

Measuring the ease of analysis is more difficult than the other methods because there is no way to use numbers in its measure. The researchers simply make note of the analysis

methods for each method being tested and make a direct comparison. The details of how the data can be analyzed by the investigator are noted and directly compared by using several different categories. These categories are as follows: initial analysis and detailed analysis. Initial analysis is based on what information can be gathered at first glance of the data. Detailed analysis is what information can be gained after an hour or more of reviewing the data, which also includes the simplicity of this task. The simplicity of the task includes the time it takes the tester to come to the same conclusion for each of the methods. These analysis measures in relation to this case study are discussed in detail later in this chapter.

This case study covers information on an individual that is largely unverifiable because it is still an open missing person case. Therefore, we used information on a public figure to collect information and create a graph to compare the case study information against. This will show the usefulness of this framework. The case study shows the efficiency of the framework while this test shows the effectiveness of the reporting method.

4.2. INFORMATION GATHERED FOR THE CASE STUDY

This section covers all the information gathered via the three methods described above. Each of these methods covers a different approach to gathering information. The goal of comparing these approaches is to determine how OSINT is most efficient and effective when using open-source tools. The details of the case from the missing person poster are described below. This information is what we used to search for social media accounts.

The first step in an online investigation is to gather general artifacts. These are pieces of information that might not be associated with the case and need analysis by the investigator. For example, in this case study, we gathered account information and potential account nicknames before searching to find what account belonged to the individual in question. This analysis then led us to the verification stage where we determine if an account belonged to the individual in question. Finally, after the account has been verified to belonging to the individual, the account information scraping can take place. This is where information from the individual's social media account is gathered and placed into the AWS Neptune database. Figure 4.1 shows this process.

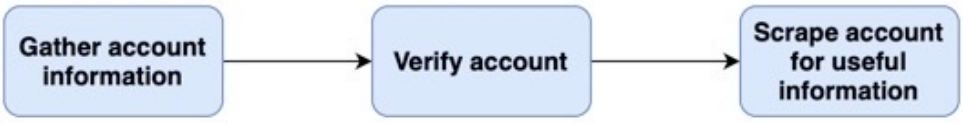


Figure 4.1: Artifact gathering workflow

The process of gathering artifacts for an investigation is as critical as the gathering itself. Having a defined and strict process ensures that the information is useful, and that the verification of information does not leave gaps or have faulty verification. This methodology

also helps with the validity of the case if it is ever used in a court case, as described in Chapter 2 of this research. By validating and verifying the information collected, investigators will find it easier to prove their findings in the investigation. Being able to prove this is an important part of a case, especially if it ever needs to be proven in court.

4.2.1. Missing Persons Case

Before we describe the information gathered regarding the case, we must discuss the missing person case used in this case study. The case we selected is that of "Jane Doe", a 22-year-old female last seen in Florida. We changed this individual's name to preserve her privacy and out of respect for her family. This case is one of the most recently posted on the FBI's missing person page. The information we will use as a basis for our information gathering will come only from this posting. The relevant information from this posting is the following:

- DOB: January 28, 2000
- Gender: Female
- Name: "Jane Doe"
- License plate: L20NAZ
- Last seen: December 20, 2021, in Davie, Florida
- Has ties to New Jersey; Baltimore, Maryland; and Korea

Only this information will be used in the testing. No other information from news sources or social media sites will be used as input for the collection tools. This is to ensure that none of the test data is influenced or changed through the use of more data than is used in the testing of other methods.

4.2.2. Information Gathered with the Framework of Tools

The first test we preformed was collecting data relating to this case using the framework we created and described in Chapter 3. This test shows the efficiency metrics for our framework that can be compared to other methods.

Table 4.1 shows the amount of information collected, number of sources, and the total time to collect that information. This collection test identified 107 possible usernames from 350 sources. Some of the information gathered by the tools was overlapping. The duplicated information was removed from final consideration on all of the methods tested in this research. The total amount of time for collection using the framework is three minutes.

Table 4.1: Collection metrics of the framework

Amount of information	Sources checked	Time to collect	Framework used
107 possible usernames or accounts	350 sources	3 min	Yes

Reviewing the report generated by this method in the initial analysis proved to be an easy and fast way of reviewing the data collected. Figure 4.2 shows the initial graphical output of the report. Anywhere where the individual's name is used has been redacted for privacy. An initial look at the AWS Neptune graph was valuable because it visually showed the number of connections between the individual and possible accounts. This graph is not included in this research because it contains personal information about the individual in question.

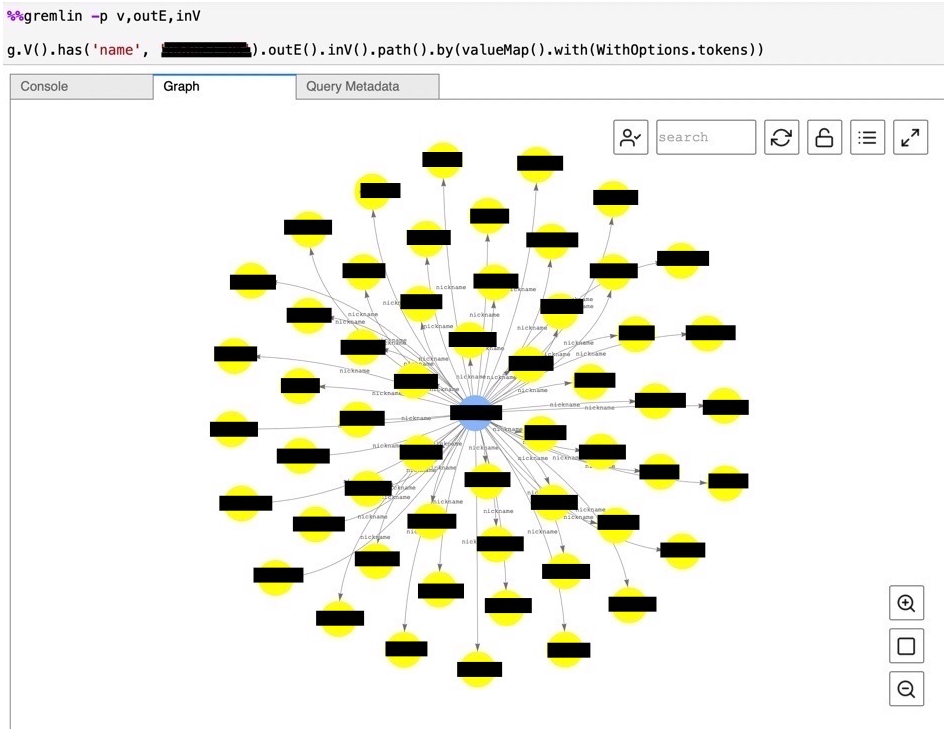


Figure 4.2: Report from a framework query for possible accounts

A detailed analysis of one hour of studying the collected information showed little more information than what was gained in the initial analysis. The value in this phase came from the reduction in useful time spent preforming a detailed analysis. All the information that could be gleaned from the collected information was gathered in five minutes of analyzing the information.

While preforming the test of this method, we encountered some challenges. The formatting of the csv documents, not only into csv from txt with our script, but also into the proper csv format that Neptune requires was challenging at first. The loader into Neptune requires a specific format for the csv file and ensuring all these requirements were covered was tedious. However, after the first time doing this, the process became much easier and more efficient.

4.2.3 Information Gathered Manually with Tools

The second test was constructed by gathering information using the tools without using any centralized reporting. This demonstrates how effective these tools are on their own, which can be used for comparison against the other methods tested.

Table 4.2 shows the tools used, the amount of information, sources, and total time to collect with this method. This phase of the case study identified 107 possible accounts and usernames from 350 sources. The total amount of time for collection using the tools separately is thirteen minutes.

Table 4.2: Collection metrics for tools used separately

Tool used	Amount of information	Sources checked	Time to collect
Alias Generator	54 nicknames generated	NA	3 min
IrcSnapshot	Nothing found	Irc chat servers	4 min
Sherlock	34 possible accounts detected	34 sites checked	3 min
WhatsMyName	19 possible accounts detected	316 sites checked	3 minutes

Reviewing the data collected was more difficult than in the previous test as there was no graph to reference during analysis. Initial analysis did not provide much useful data. It only showed the possible sites that this individual was using or other potential usernames that were being used. This volume of information was too overwhelming at first glance to gain much information from the initial analysis.

A detailed analysis of one hour of studying the collected information showed not only potential accounts, but also potential deviations in usernames that could be used in further searching with the tools. However, there was no indication on if any of these potential accounts or usernames were associated with the individual in question. After several minutes of

searching, further analysis resulted in no advancement in the investigation. Only the first quarter of the detailed analysis proved to be useful.

One of the challenging parts of using this method was initially reviewing the data collected. At first glance, it seemed overwhelming, and it took some time to grasp the connections and usefulness between the information. Another challenge we faced was the dead end of detailed analysis after some time analyzing the information collected. This method provided useful information that investigators could pivot from, but this analysis could only provide high level information.

4.2.4. Information Gathered Manually without Tools

The third test does not use any tools to gather the information. Rather we used regular searches on search engines and social media sites to gather information. This shows how effective these tools can be, whether they are used as a collection or not.

Table 4.3 depicts the sources of collection, information collected, amount of information, and total time of collection using this method. The total time for collection without using any tools was one hour and fifteen minutes. This collection resulted in no additional information gathered beyond a correction of the initial report that the license plate is a New Jersey plate instead of a Florida plate.

Table 4.3: Collection metrics for no tools used

Source	Information collected	Amount of information	Time to collect
Initial Google search	Local news; Facebook; LinkedIn; Twitter; Instagram	Starting point for the rest of the categories	8 min
Twitter	Tweet from police department regarding license plate	NJ tag instead of FL tag	15 min
	Same information as FBI poster		
	Several possible accounts, but none were confirmed to be hers		
LinkedIn	Same information as FBI poster	No additional information	7 min
	Several possible accounts, but none were confirmed to be hers		
Facebook	Several possible accounts, but none were confirmed to be hers	No additional information	20 min
Instagram	Several possible accounts, but none were confirmed to be hers	No additional information	15 min
Local News	FBI poster information - Many local news outlets with the same posting	No additional information	10 min

Unlike the other methods discussed earlier, this method did not provide any useful data that could be reviewed by the investigator. Preforming the beginning stages of an OSINT investigation without any tools is too time consuming to be practically used in any investigation. Manual searches might be helpful if more information is known, but when preforming general searches with minimal information, it is not helpful.

This method provided investigators with several different challenges that were not present in the other tests performed earlier. One of these challenges was with searching social media platforms. Facebook and Instagram were difficult to search without an account, whereas Twitter was relatively easy. Also, after a search was done for accounts on these sites, each possible account had to be checked individually, which was time-consuming and provided no useful results.

4.2.5. Crowdsourced Information

Due to structural changes within TraceLabs as we were conducting this research, this aspect of information collection cannot be used for comparison. TraceLabs currently only preforms Capture the Flag (CTF) events for information collection and no longer has open cases available

for the community to work on. This aspect of OSINT gathering can be a vector for research in the future.

4.3. RESULTS OF THE CASE STUDY

The data gathered earlier in this chapter shows the efficiency of each method tested. In this section, we discuss the comparison of this data. Table 4.1 shows this comparison of data gathering metrics as defined earlier in this chapter. Furthermore, the analysis of the information gathered is also measured in this Table. The analysis is based on a weighting system from one to five, with one being not helpful at all and five being extremely helpful. Both the framework and searching manually with tools had the same result for amount of data and sources. This is unsurprising because the same web scraping tools are used.

Initial analysis for the framework is rated as four, while detailed analysis is rated as a five. The initial analysis gave us most of the analysis information, while the detailed analysis provided us with more specific information. This specific information included things such as pivot points for the investigation. For example, it showed potential accounts that could belong to this individual that investigators could look into further.

For searching manually with the tools, the initial analysis is rated as a two because it did provide some useful information, but it did not have the same impact at first glance. The detailed analysis is rated as three because it provided some useful information, but became tedious, repetitive, and did not provide any new information after the first quarter of the analysis.

Manual searching without using any tools provided no helpful information, so both the initial analysis and detailed analysis are rated as a one. It gave investigators only one piece of useful data after a long collection time. As Table 4.4 shows, manual searching without any tools is not only time consuming, but it does not provide much useful information to investigators.

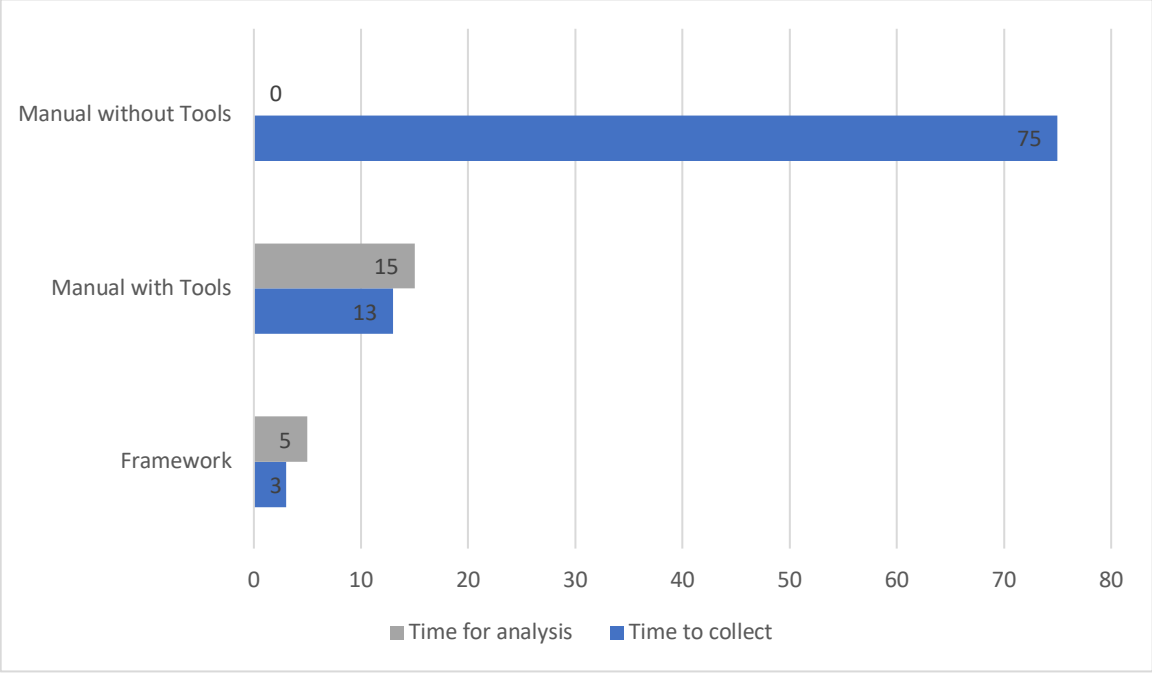
The final metric is how much time of the hour-long detailed analysis resulted in useful information. A lower number means a more effective reporting method because this shows that analyzing the information takes less time. However, it should be noted that the manual without tools collection bar shows zero useful analysis minutes because there was no useful information collected with this method.

Table 4.4: Data comparison

Collection Type	Time to collect	Amount of data	Sources	Initial analysis	Detailed analysis	Useful Detailed Analysis Time
Framework	3 minutes	107	350	5	4	5 min
Manual with Tools	13 minutes	107	350	2	3	15 min
Manual without Tools	1 hour 15 minutes	1 item (NJ plate instead of FL)	6	1	1	0 min

Table 4.5 depicts the distribution of time for each of the methods tested. The gray bar shows the amount of time that provided useful information during detailed analysis, while the blue bar shows the time for collection. This diagram shows that using tools greatly reduces the time for collection and analysis. It also shows that the framework reduces that time even further, which continues to improve the efficiency of the investigation.

Table 4.5: Time for collection and detailed analysis



Tables 4.6 and 4.7 show the amount of data collected and the number of sources from each collection method. They demonstrate how the framework and manual searching with tools resulted in the same number for both of these metrics. This is because they use the same tools for collection. Manual searching without tools, however, has a much lower number than either of the two other methods.

Table 4.6: Amount of data collected

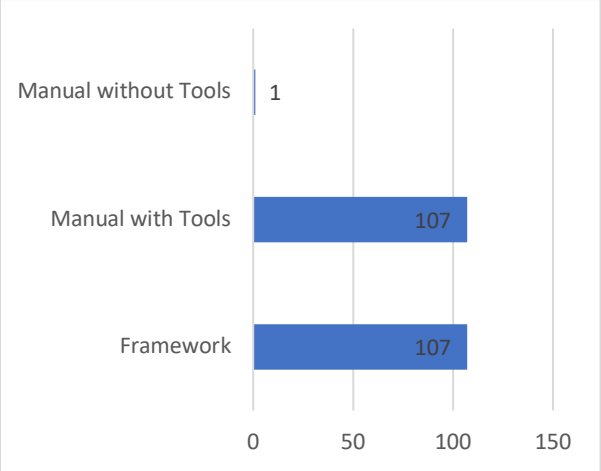


Table 4.7: Number of sources

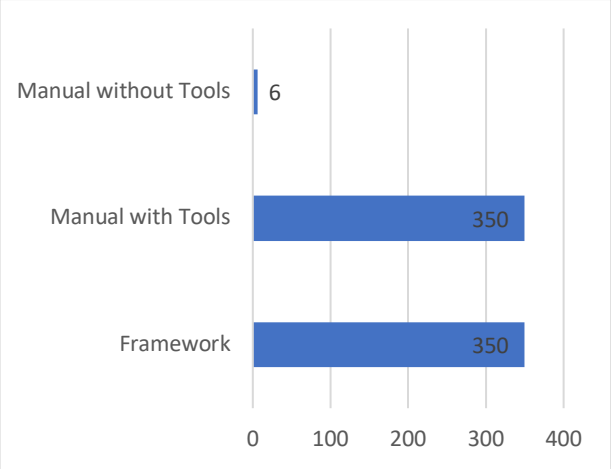
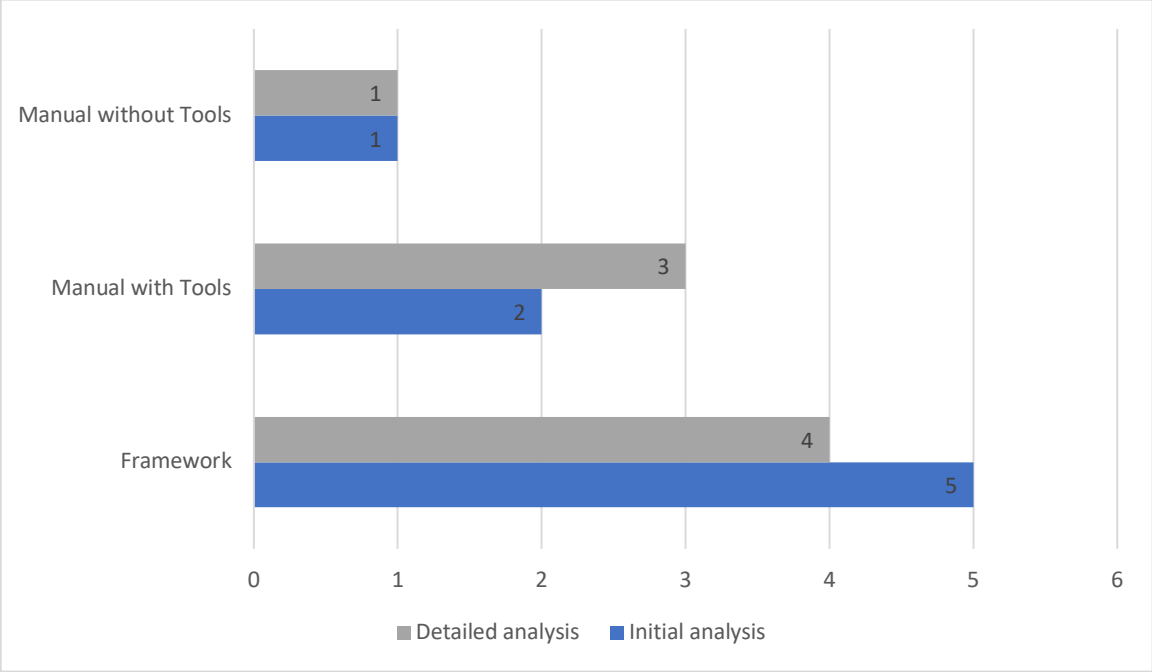


Table 4.8 shows the usefulness of the analysis gathering portion of the case study. The gray bar depicts the number assigned to the detailed analysis based on the usefulness of the information, while the blue bar is the number for the initial analysis. The first conclusion that can be drawn from this diagram is that manual collection without tools is not helpful at all, either in initial analysis or with detailed analysis. This diagram also shows that manually searching with tools is only helpful after some time is spent with the data collected, but not at initial analysis. Using the framework and reporting method discussed in this paper gave the best analysis results, especially in the initial analysis category. The quality of the initial analysis reduced the need for a great length of time during the detailed analysis, so the framework improved the detailed analysis score.

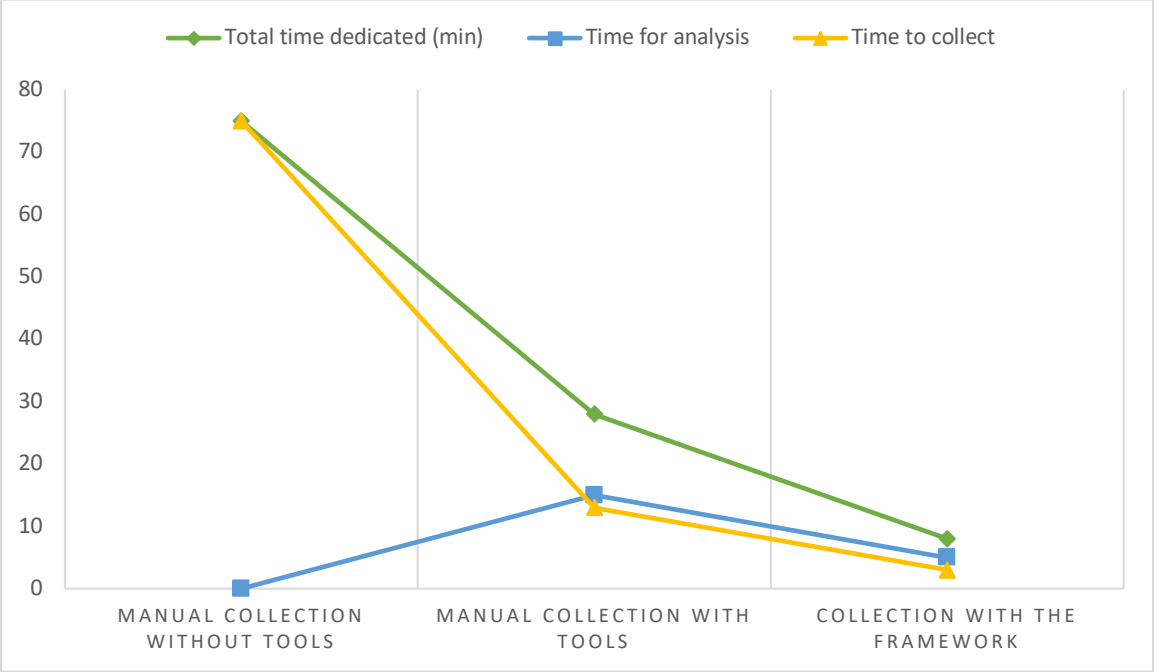
Table 4.8: Analysis of information gathered (1-5 scale)



This case study shows that using tools in an OSINT investigation is invaluable to investigators. Searching for information without any tools resulted in little information after over an hour of searching. An analysis of the collection time and resulting information analysis shows that this method provides little value to the investigation.

The framework and manual tools collected the same information, but at different speeds and with different data representation. This difference caused an increase in the collection speed and a decrease in the time for detailed analysis. These changes improved the efficiency of the information collection by 28.6% decrease in required time for collection and analysis between the manual tools collection without the framework and the collection using the framework. Table 4.9 shows a depiction of this decrease in time required for collection and analysis.

Table 4.9: Trend in time for collection and analysis



There are some challenges presented in any of these investigative methods that come from the type of investigation being performed in this case study. When preforming an OSINT investigation, investigators are partially guessing at potential usernames if that information is unknown at the start of the case. Individuals often choose usernames that are not their real name or have no other way of being associated with them. Another challenge with this type of investigation is language differences. The internet and social media are global networks that encompasses many different languages. The individual who is the subject of the investigation may not use the same language, which can create challenges with finding content or accounts from this person.

These challenges are something investigators must face regardless of the type of OSINT investigation they are preforming. There are some tools that can help with overcoming them, such as facial recognition web scrapers, but they are also limited in the potential help they can provide. Furthermore, in some cases, the individual might not use social media at all, giving them a small digital footprint.

Once one or more social media accounts are identified for the individual in question, the information from these sites can be added to the graph. This information can provide valuable insights into the individual, their habits, places they frequent, latest post location information, or people they know. This information can be useful in determining the possible last actions of the person in question, which can be used to retrace their last steps and there are many different tools that can be used to automate the collection of this information. Unfortunately,

there were no social media accounts determined to belong to this individual. However, this idea is demonstrated in the next section of this chapter.

4.4. COLLECTION ON A KNOWN INDIVIDUAL

Now that we have established the collection time and sources from using this method, we need to test the usefulness of the information gathered. This requires information that can be separately verifiable for the purpose of testing. To do this, we must use a public figure who has at least one verified social media account.

To compare the data and metrics collected from the missing person case study, we collected information on a public figure for comparison. The individual we chose for this portion was Elon Musk, the CEO of Tesla and SpaceX. He has public social media accounts that we can verify as belonging to him in the information collection stage. By preforming this on a known individual, testing allowed us to show the report generated by AWS Neptune. The queries used to make this report are gremlin queries, showing the importance of formatting the data according to gremlin requirements.

The data was collected using the same tools and resulted in 169 possible accounts and 15 possible account usernames. These results are similar to the information gathered in the missing person case earlier in this case study. This shows that regardless of the individual in question, the tools collect a large amount of information that investigators can analyze and verify. Verifying this information involved reviewing this data to see if any of the accounts were verified by the social media organization to belonging to Musk. There was only one account, a Twitter account, that was verified. The username for this account was also one of the usernames generated using the tools. Table 4.10 shows this information and the percentages of useful information from what was gathered.

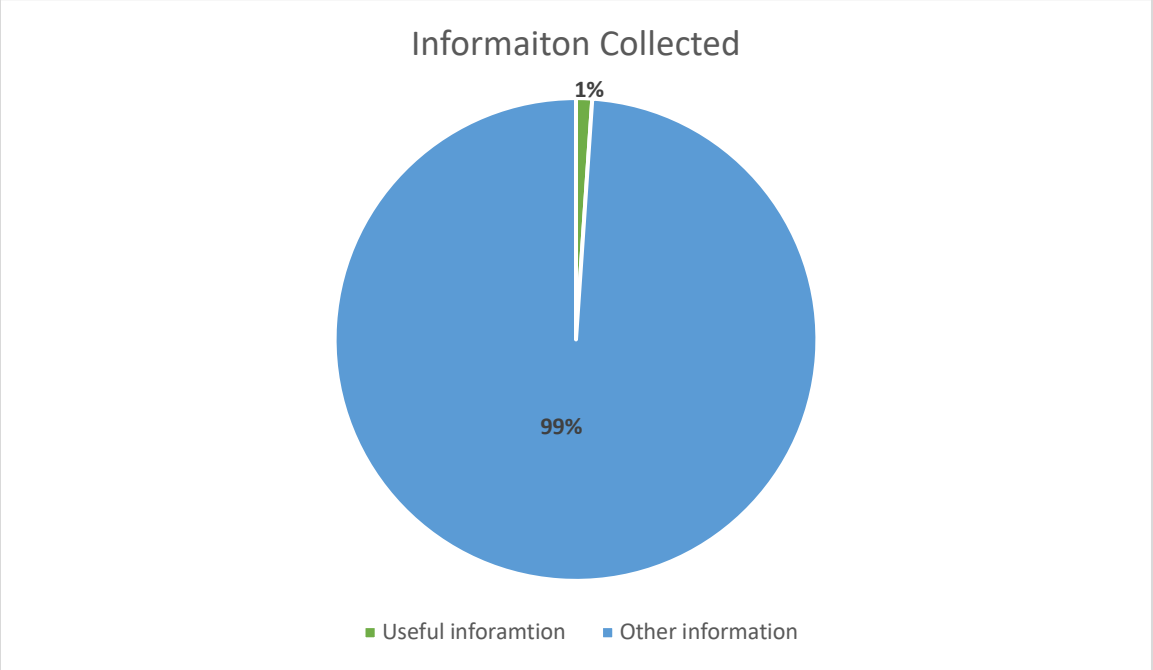
Table 4.10: Verified information from known individual collection metrics

Information collected with tools	Verifiable Information	Percent of useful valid information	Information collected after verified account identified
Account information	Verified Twitter	0.6% (1/169)	Company information
Potential nicknames	Verified username (elonmusk)	6.7% (1/15)	Verified Twitter for companies

Table 4.11 shows a depiction of the percent of useful information from what we gathered using the tools. There were many items that were not associated with the individual,

but 1% was verified. This 1% is enough in a case to pivot from and find more relevant information for the case. For example, there are two verified Twitter and YouTube accounts for the two companies where Musk is the CEO. These accounts may provide useful information on recent events or locations related to Musk.

Table 4.11: Usefulness of the information collected from general collection



From the verified information gathered earlier, we could then pivot and discover other accounts associated with this individual and relevant information from these accounts. This information is then gathered and added to the relational graph. This process can continue until all the relevant information is found and a cluster diagram can be achieved with this information and its relationships. In this specific case, Musk was used as the starting point and the nickname connections were added. The accounts using the nickname "elonmusk" were then added with a connection to that nickname.

Figure 4.3 shows the console and graph of the nicknames generated and their relationship with Musk within the database. The query at the top of the image is used to show all connections associated with the name "Elon Musk." The blue bubble represents the Musk and the yellow are the potential nicknames. The red bubble represents the accounts associated with Musk, but these will be explored further in this section. The graph tab shows the graphical representation of the information found by the query, while the console tab shows the raw data.

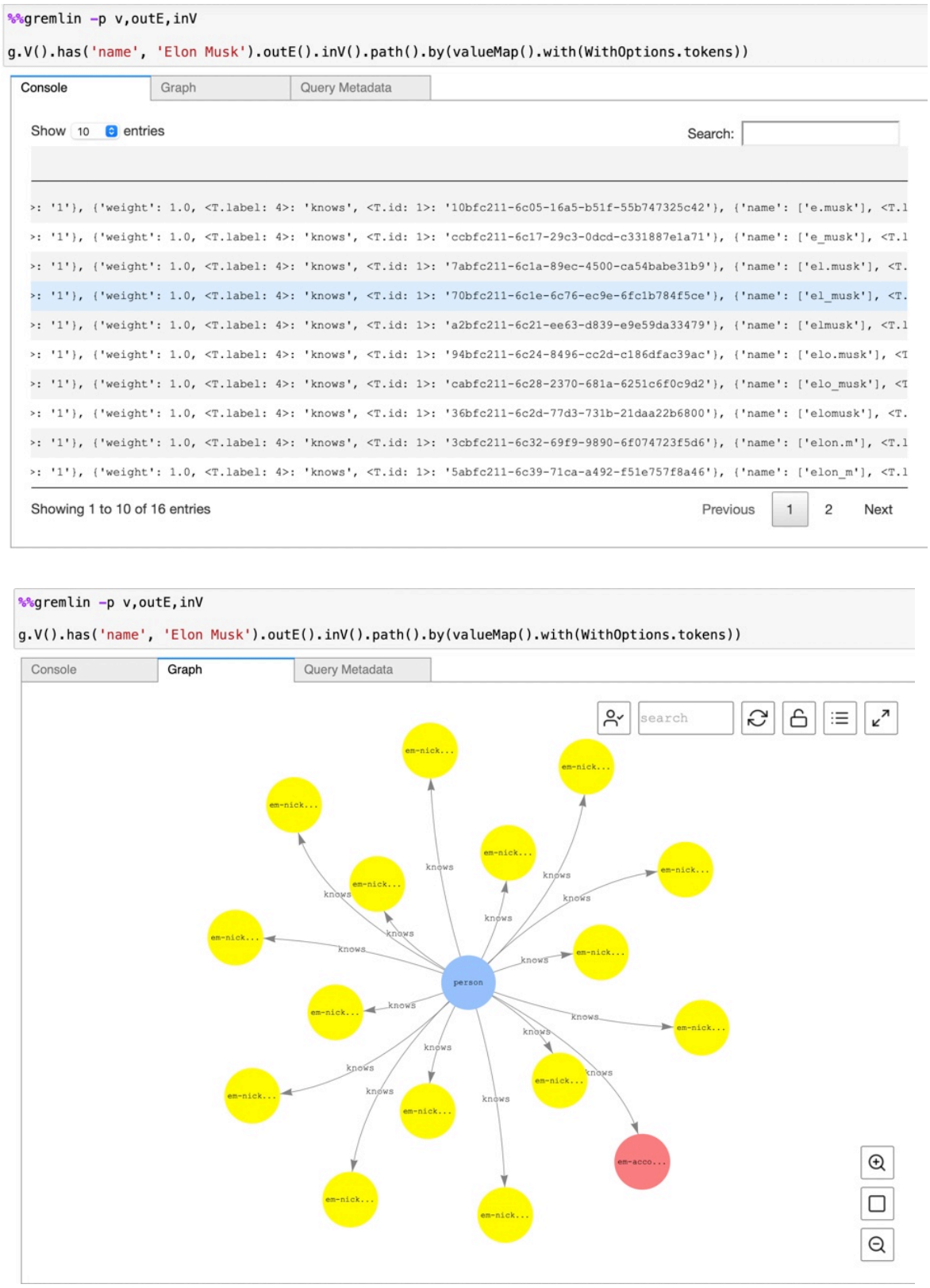


Figure 4.3: Graphical representation of the nicknames

Figure 4.4 shows the console representation and graphical representation of all the accounts associated with the nickname “elonmusk.” The query at the top of the image is what is used to show all associated connections with the specified username. In this diagram, the blue bubble represents the nickname chosen to search for accounts and the yellow are the possible accounts. One of these accounts is the verified Twitter account owned by Musk. From this verified account, information about Musk can be scraped, including information about companies he runs, locations he has been, or relationships with other accounts on Twitter. All this information can be added to the graph and connected to the verified account.

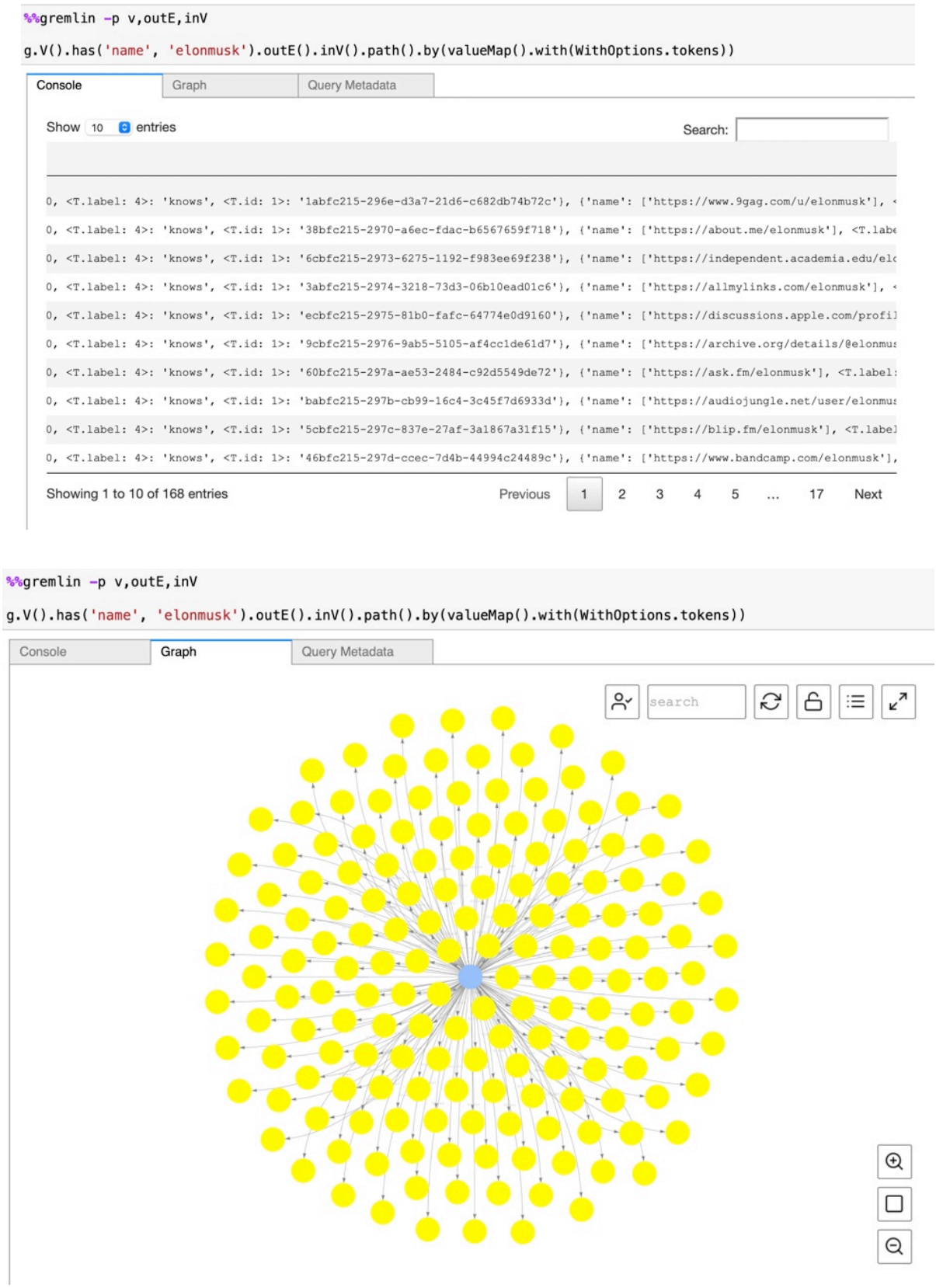


Figure 4.4: Graphical representation of accounts that belong to one username

This graph shows how just one nickname can have hundreds of accounts associated with it that require analysis. However, this type of report reduces the amount of time required to analyze the information from the initial gathering of information. Once a verified connection is found, this connection can be used to add information from that source to the database and discover other connections regarding the individual that could be useful in a missing person case. An example of this is depicted in Figure 4.5, which shows the connection between Tesla, SpaceX, and the verified Twitter account belonging to Musk.

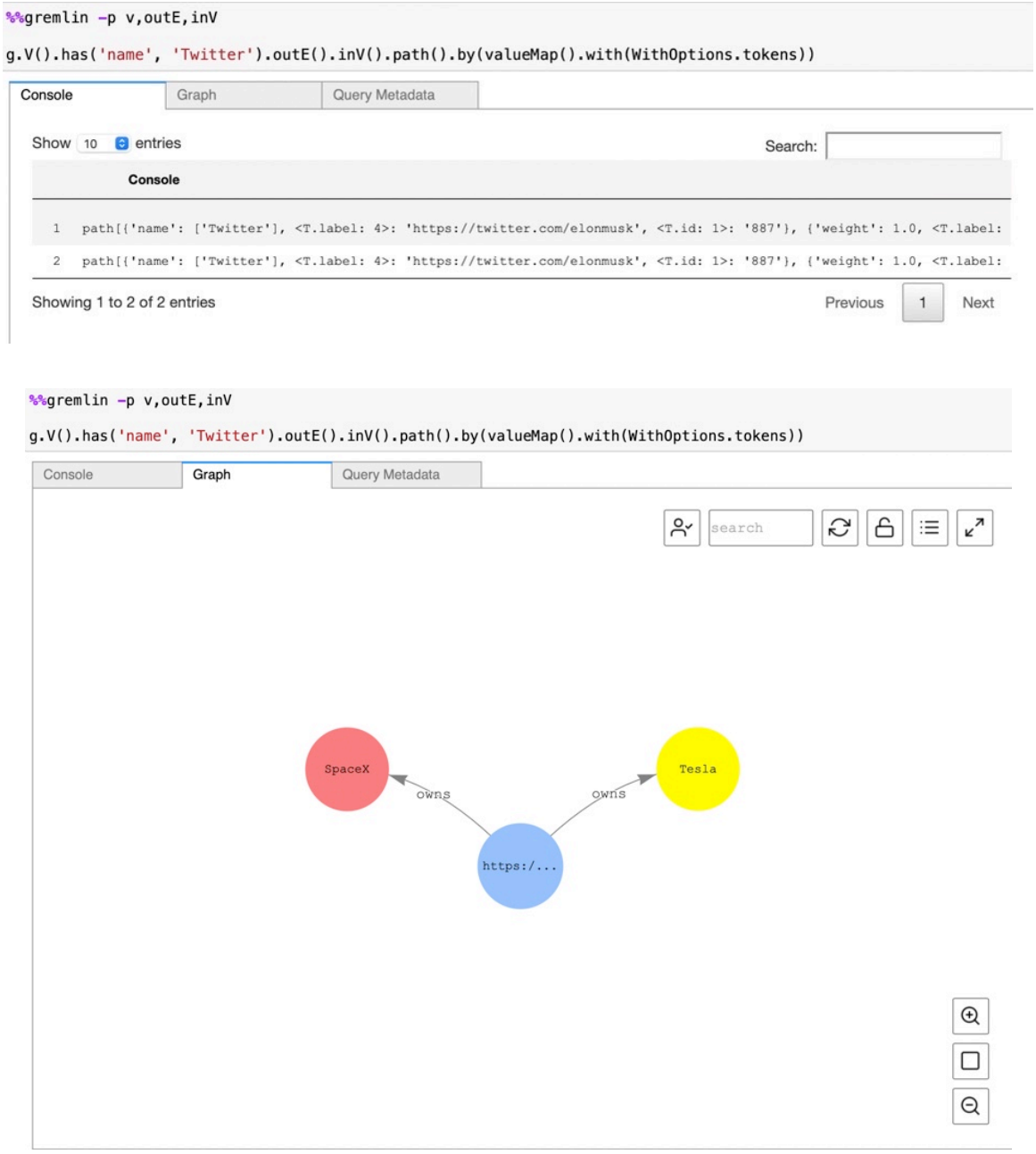


Figure 4.5: Connection between verified account and the related accounts of Tesla and SpaceX

There are also methods that investigators can use to find connections between entities within AWS Neptune. Figure 4.6 shows the search for a connection between Musk and Tesla. It shows that within the database, Musk is connected to possible nicknames, possible Twitter accounts, and Tesla connected to the verified Twitter account belonging to Musk. This method can be used to find connections within the database between two entities that seemingly do not have a connection. By being able to query the database for connections, investigations can search for connections within the useful information that can help advance the case. However, this can only be done after the information is verified to be connected to the individual from the missing person case, or a related individual.

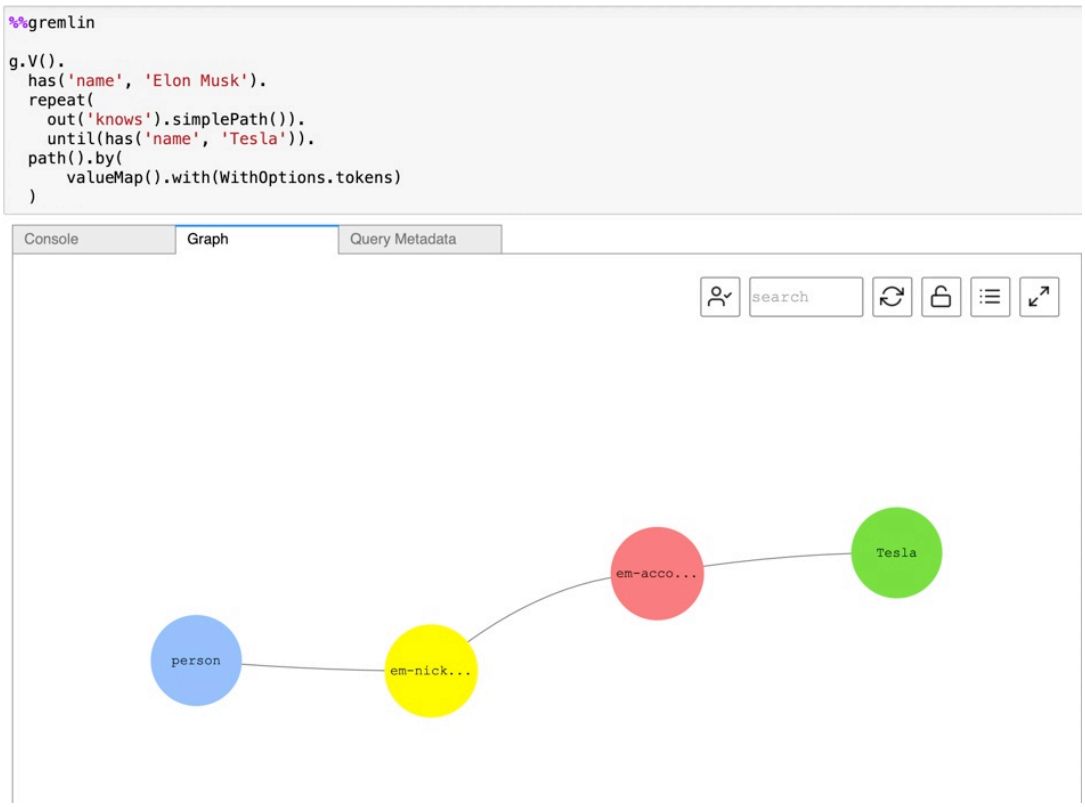


Figure 4.6: Connection between Musk and Tesla

The framework reduced the time of collection by 28% overall, with the collection of information with tools being a 23% reduction and the reporting and analysis method being a 33% reduction. The collection of tools reduced the time to collect and the reporting method reducing the required time for analysis. This increase in efficiency can help advance the tasks of an investigation, which allows investigators to use their resources in later stages of the investigation, allowing for the potential of faster progress in a case.

The effectiveness of the information collected for a cyber investigation using this framework and reporting method was increased, only after verified information was identified. The graphical reporting method increased the effectiveness of searching for connections and relevant information. However, with the amount of information collected and most of it being unhelpful, this could not be utilized until the information is verified to belonging to the individual in question.

In the future, when using this tool, we want to note several details to improve the graph output. One item to note is that, to ensure the proper name is displayed in the graph, the label should be defined as the name, nickname, account, or information item. This can be seen in Figure 4.5 where the label “person” is used for Elon Musk, but we used the label “Tesla” to represent Tesla. Also, connections can be bulk loaded into Neptune in the same way that the CSV information files were added earlier in this chapter.

Even if the framework does not provide useful information to investigators, it is still helpful to perform this data gathering in an investigation. Investigators and analysts must look at all the data available to them when working on a case, and this provides a way for them to collect that data quickly and presents it in a format that is easy to analyze. If they do not find any helpful information, at least they made an effort to check all the online sources available to them.

4.5. COST

The cost of this framework is the aggregated cost of deployment of the system and the cost of using the system per investigation. One of the goals of this framework was to reduce the overall cost of OSINT investigation tools in order to make the tools more accessible to individual analysts and investigators. The reduction in cost not only came from using open-source tools, but also using free-tier tools within AWS where possible. Within AWS, there were only two tools that needed monetary support: Neptune and SageMaker.

Lowering the cost of the system is an important aspect of the success of this framework. The cost of the system adds to the accessibility of the system. Having a cost-effective alternative to enterprise tools allows individuals to volunteer their skills and time to gathering and analyzing information for cyber investigations. With more analysts and investigators applying their time to an investigation, the chances of that investigation being successful are increased.

Deployment was the costliest part of using the AWS resources. Figure 4.7 shows the cost of one month during the deployment phase of this framework. The total cost of that month was 142.80 dollars. The deployment and testing of this system spanned two and a half months and this number is representative of the average over this time. However, after deployment, utilizing the framework cost only between a few cents and less than 5 dollars, depending on the time spent using the resources. This shows that the usage of Neptune and SageMaker for querying and reporting is not only useful for the investigation, but also cost effective.

▾ Neptune		$93.87
▾ US East (Ohio)		$93.87
Amazon Neptune		$93.87
$0.021 per GB / month (Amazon Neptune)	0.017 GB-Mo	$0.00
$0.098 per instance hour (or partial hour) running Amazon Neptune	54.569 Hrs	$5.35
$0.100 per GB / month (Amazon Neptune)	0.121 GB-Mo	$0.01
$0.20 per 1 million I/O requests (Amazon Neptune)	1,434,321.000 IOs	$0.29
$0.696 per instance hour (or partial hour) running Amazon Neptune	126.751 Hrs	$88.22
▾ SageMaker		$48.93
▾ US East (Ohio)		$48.93
Amazon SageMaker CreateVolume-Gp2		$1.13
$0.14 per GB-Mo of Notebook Instance ML storage	8.105 GB-Mo	$1.13
Amazon SageMaker RunInstance		$47.80
$0.00 for Notebk:ml.t3.medium per hour under monthly free tier	250.000 Hrs	$0.00
$0.05 per Notebook ml.t3.medium hour in US East (Ohio)	955.962 Hrs	$47.80

Figure 4.7: Cost during deployment of the architecture

The leading OSINT tool is Maltego, and the pro version of this tool is $999 per user. TraceLabs currently has over 2,400 users signed up on their Discord server. If every one of those users wanted to use a tool, the total cost of using Maltego would be 2.4 million dollars. The total cost of using this framework every day for a year would be 438,000 dollars. Even with roughly 150 dollars for deployment of the system, the overall cost of deployment and use for the entire TraceLabs community would be 798,000 dollars. Table 4.12 shows this cost comparison. The framework decreases the cost by 1.6 million dollars. A reduction in cost would allow this system, and thus faster methods of cyber investigations, to be more accessible to a wider range of analysts and investigators.

Table 4.12: Cost analysis of the framework

	Cost of single use	Cost of large-scale deployment and use
Our framework	~3 dollars	798,000 dollars (438,000 dollars post deployment)
Maltego	1,000 dollars	2.4 million dollars

4.6. PERFORMANCE

Through our deployment and testing of this framework, there were several metrics for performance and usage that we collected to show how the framework performance. These metrics showed resource usage of each of the elements within AWS, which shows the efficiency of the system itself. The performance and cost of the framework show how the framework compares to other enterprise competitors. Also, by breaking down the usage of the system based on the tasks done at the time, we are able to see which parts of the deployment and investigations are most intensive on the system.

During the deployment and testing of the framework, there were several usage and performance metrics captured. These metrics show how the framework performed while it was being constructed, and while information was being collected and analyzed. The details of this performance timeline are shown in Figures 4.9, 4.10, and 4.11. These metrics can be broken down into the different elements of the deployment and testing based on the timeline shown in the reports from AWS CloudWatch.

By breaking down the performance timeline, we can determine which elements are the most intensive. This can lead to future research into areas where improvement can be used to decrease the cost or increase the performance of the underlying infrastructure of the framework. The underlying elements add to the accessibility of the system. Having an efficient soliton to enterprise tools allows individuals to volunteer their skills and time to gathering and analyzing information for cyber investigations.

The EC2 instance used in this framework was a t2.micro. The details of this are shown in Figure 4.8. This instance has only one CPU and it has access to one gibibit of memory with low or moderate network performance. Overall, this is a bottom end instance type relative to the other options available in AWS. It has minimal resources and is free to use. However, with these few resources, the framework performed well, and the resource usage was good. The tools used in this framework do not have a GUI, so they do not require a lot of processing power or memory. Also, they were able to perform well with low to moderate network performance as demonstrated in the tests described above.

Instance name ▲	On-Demand hourly rate ▽	vCPU ▽	Memory ▽	Storage ▽	Network performance ▽
t3a.nano	$0.0047	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t3a.micro	$0.0094	2	1 GiB	EBS Only	Up to 5 Gigabit
t3a.small	$0.0188	2	2 GiB	EBS Only	Up to 5 Gigabit
t3a.medium	$0.0376	2	4 GiB	EBS Only	Up to 5 Gigabit
t3a.large	$0.0752	2	8 GiB	EBS Only	Up to 5 Gigabit
t3a.xlarge	$0.1504	4	16 GiB	EBS Only	Up to 5 Gigabit
t3a.2xlarge	$0.3008	8	32 GiB	EBS Only	Up to 5 Gigabit
t2.nano	$0.0058	1	0.5 GiB	EBS Only	Low
t2.micro	$0.0116	1	1 GiB	EBS Only	Low to Moderate

Figure 4.8: EC2 instance specs

The overall performance of this framework was good. Figures 4.9, 4.10, and 4.11 depict the usage of CPU and network resources during the deployment and testing of the framework over two months. The average CPU usage was 2.59% while the highest spike was 5.17%. The

average network in was 6.67M while the highest was 13.3M. Finally, the highest network out metric was 219k while the average was 109k. These are also depicted in Table 4.12. The performance of the S3 was also well under the upper requirements. As Pelle et al. (2019) discussed, the S3 bucket has an upper limit for a single PUT command of 5GB per command. The total file size of all the information collected with this framework was 10.1KB. This leaves plenty of room to expand the functionalities of the framework without exceeding the limit set by AWS.

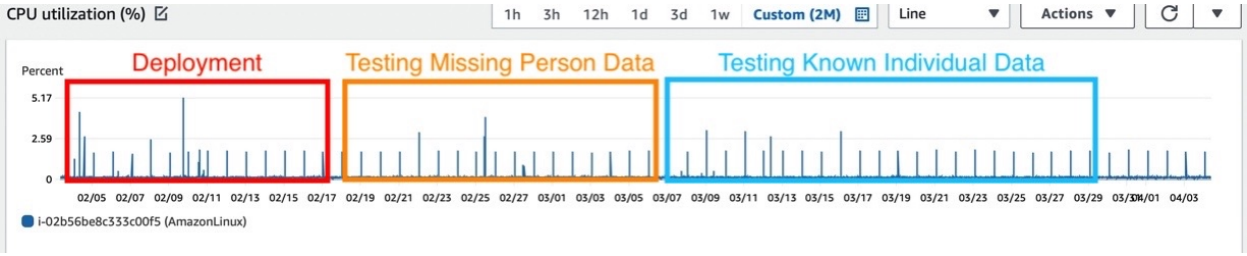


Figure 4.9: Overall CPU utilization during deployment and testing

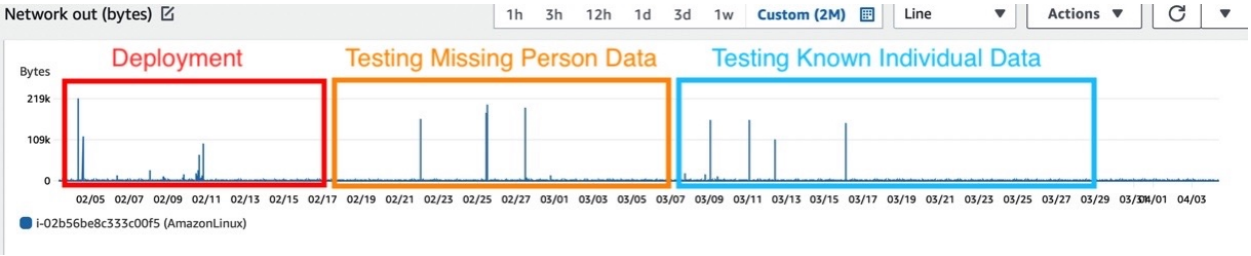


Figure 4.10: Overall network out usage

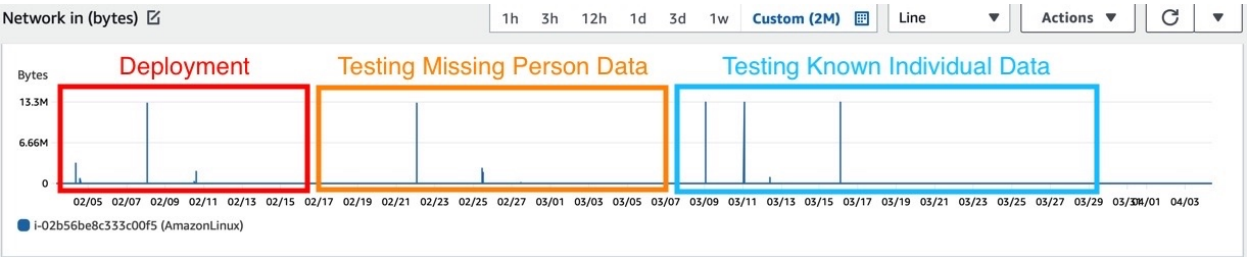


Figure 4.11: Overall network in usage

Figures 4.9, 4.10, and 4.11 also show the usage of resources during the different phases of deployment, testing with the missing person data, and testing with data on a known individual. Resources were used mostly during the deployment phase as the resources were being created and configured and as the scripts were being written. This correlates with the cost of deployment versus the cost of performing an investigation. For the testing of the two

data sets, the resource utilization is similar because the processes used were the same for both data sets.

Table 4.13: EC2 instance resource usage over two months

	CPU Usage	Network In (bytes)	Network Out (bytes)
Highest	5.17%	13.3M	219k
Average	2.59%	6.67M	109k

Overall, the performance of the system was more than sufficient for the automation, collection, and reporting tasks. It performed well under capacity and did not come close to reaching the storage or push limit. The cost of using the framework was well under the cost of competitor products. This decrease in cost over a large group of individual contributors creates a distinct improvement in the accessibility of this framework. The framework is designed to be accessible to individual analysts and investigators. By creating a framework that, not only decreases the time for collection and analysis, performs well and reduces the cost of using the platform, more investigators and analysts can contribute to cases. There is a large community, as can be seen with TraceLabs, that want to contribute their time and skills to these cases and making the tools to help them more accessible ultimately assists in the investigator's success with the case.

4.7. RELEVANT PRATICAL PROBLEM

We found that OSINT is not a well-researched topic, especially in the field of cybercrime investigations. Cybercrime is a topic that affects many organizations and individuals, and research in the area of receding and combating this crime is critical to protecting these organizations and individuals. It was also determined in the literature review that automation is a key element of the future of technology. This motivated us to integrate automation into the framework, especially in areas that we identified as being redundant. These redundant areas include information collection and storage as well as multiple reports for one case. By consolidating these tasks with automation, we found that we could make the investigative process more efficient.

In our literature review, we did not find any research on the topic of improving efficiency in OSINT cyber investigations or building a framework for this in the cloud. The framework, including the cloud architecture and reporting method, are novel. With the number of services available through AWS, there are thousands of ways to create a system with these pieces. It was critical for us to develop the simplest framework by using the least number of pieces possible. This would ensure that not only is the framework simple to use, but it would also cut down on the potential cost and digital footprint.

The automation used in this framework is focused on the most time-consuming and difficult tasks in OSINT cyber investigations: information collection, storage, and reporting. These processes are critical to the investigation, but they often require the investigator to perform the same tasks repeatedly. By automating these repeated tasks, we were able to reduce the amount of effort required to perform these tasks, which reduced the time required. The improvement in time for collection and analysis as reported above show how this automation helps the investigation.

Through research into the topic of cybercrime investigations, especially with increasing accessibility to OSINT tools and methods, researchers can add to the growing effort to reduce cybercrime and its impact. As shown in this research, investigating cybercrime is one effective way to combat cybercrime and reduce its effects on organizations and individuals. With tools and methods more accessible to analysts and investigators, that allows them to contribute their time and skills more easily to these investigations.

4.8. VALIDATION PROCESS

The process of validating this framework and the underlying research included testing the framework with a real missing person case as well as with an individual with verifiable information. This allowed us to measure and analyze the results and determine how the framework would perform in an actual missing person case. Using the missing person case allowed us to show the amount of information that can be gathered on an individual and the time differences of collection and analysis using the framework versus using other methods. Also, testing with verifiable information showed how much information that is gathered using the framework is accurate and useful in an investigation.

This research used information associated with real individuals and real situations. It showed how this framework works in as close to a real-life case as possible. Using this information allowed us to simulate as best we could how the framework would function when used as part of a case. Doing this is an important part of ensuring the framework is useful in real investigations and identifying where it improves investigations the most. Identifying where improvements can be made helps direct where this research and framework can benefit from future research.

We tested this research using information that is verifiable in order to determine how much information collected is relevant to the investigation. This is a critical process in any OSINT investigation. This is due to the amount of data collected and how important it is that information is proven to be related to the individual or the case. OSINT investigations gather a large amount of information that may or may not be related to the case, which is why information validation must be done. Without the ability to verify information for a case, investigators will not know where to continue with the investigation.

By using real and verifiable information, we were able to test the framework using two sets of data that had different reporting outcomes, but similar efficiency outcomes. The collection time was decreased in both tests and the reporting was also shown to be more efficient in both. The amount of effort for analysis using the framework was decreased with both sets of data, and the verifiable information showed that connections can be found using the relational nature of the database.

4.9. BROADER IMPACT

Researching the efficiency of OSINT and the accuracy of the information gathered is a critical area that affects many other fields. OSINT is used, not only in cyber investigations, but also in many other areas of cybersecurity. For example, penetration testers use OSINT to perform reconnaissance on the organization they are testing. The same areas we identified as being redundant are still present in this case and can benefit from automation and centralized reporting.

This research focuses on the efficiency of cybercrime investigations and accuracy of the information gathered. Both of these elements are critical to an investigation's success. OSINT is an incredibly useful tool in an investigation. Therefore, it is important to ensure the tools and methods used are efficient and any redundant areas are reduced with technology such as automation. Also, investigators cannot use information that they are unable to verify as being relevant to the investigation.

The framework deployed in this research is designed to make cyber investigation tools more accessible to individuals who wish to contribute. In cases such as missing person cases, many individual researchers, analysts, and investigators are motivated to contribute their time and skills to the case. By making tools that decrease the time of collection and analysis while costing less than the commercial option, this contribution becomes more accessible to these individuals who want to contribute.

The researched performed and the framework we developed also decrease the amount of time required when performing OSINT investigations regarding missing person cases. This has an impact with other cyber investigations that use OSINT as a tool. The principles of this framework research can be applied to other investigations, such as those discussed in Chapter 2. This will reduce the required time for those other investigation types.

CHAPTER 5: CONCLUSIONS AND RESULTS OF OSINT TOOL RESEARCH INTO INVESTIGATION EFFICIENCY AND EFFECTIVENESS

5.1. CONCLUSIONS AND RESULTS

Cybercrime affects many areas of technology and other industries. Many industries, such as the financial and health fields, rely on technology and require security and privacy to function properly. One way to reduce the rate of cybercrime, which reduces the effect on individuals and these industries, is to investigate these crimes. Reducing the effects of cybercrime on people or businesses or reducing the rate of cybercrime is important for individuals and industries to function with a sense of safety. One method to reduce cybercrime is through cyber investigations. An effective method used in cyber investigations is open-source intelligence (OSINT).

This thesis proposed and tested a method to improve the efficiency and effectiveness of a framework of open-source OSINT tools. It demonstrated the importance of efficiency and effectiveness in OSINT investigations. By improving the efficiency and effectiveness of OSINT investigations and using open-source tools, these investigations can become faster, produce better results, and be more accessible to other people outside of professionals working in cybersecurity.

5.1.1 Reduction in Time for Collection

One of the results we accomplished from this research was a reduction in the time for collecting the information. The framework we built streamlined the tasks required to collect the information, save it into the correct format, and move it to a centralized repository. By doing this, the number of commands an investigator must enter are reduced and the amount of time to collect information on accounts is also reduced. As shown in Chapter 4, there was a 23% increase in efficiency for collection time with the framework.

Using the framework of open-source tools has no extra cost for investigators and analysts. The tools are completely free to download onto any Linux instance. It gives them the functionality of collecting the information without being time consuming. It also gives them the ability to modify the functionality of the modules if that becomes necessary. The framework automates and streamlines the tasks of collecting the information, storing the artifacts, and properly formatting the information for analysis.

5.1.2 Reduction in Time for Analysis

Another reduction that we saw was a reduction in the time to analyze the reported information. Without the reporting mechanism we used in this research, investigators would have to look at raw data output from the tools. The reporting mechanism we employed showed a graphical representation of the data and connections. This assists investigators in understanding the data and finding important pieces and connections faster than with just the raw data. As described in Chapter 4, there was a 33% increase in efficiency for the time to analyze the data using the reporting method of this framework.

After the database instance is created, the cost of using AWS Neptune and SageMaker is minimal. The database can be started whenever investigators need to use it and stopped whenever their analysis is complete. It only costs a few cents to start the instance, run some commands in the Jupyter notebook, retrieve the results, and stop the instance again. This allows investigators and analysts to get the functionality of a graphical database report without costing a large amount of money.

5.1.3 Effectiveness Improved Only After Information Verification

We tested two elements of improvement in this research: efficiency and effectiveness. The efficiency is the time for collection and analysis, while effectiveness is the amount of useful information collected and identified. As mentioned above, the efficiency was improved, but the effectiveness of cyber investigations was not changed if a verified account was not identified. Without an account that is verified to belonging to the individual in question, it is impossible to find useful information to add to the graphical relational database. Our research could not verify accounts.

The report generated cannot draw any conclusions automatically based on the information given to it unless a known and verified account is identified. An investigator still needs to manually analyze the information in the report and assess what information is useful and what is not. This is an area we hoped to improve, but we found this did not happen. Once a verified account is identified, then the reporting mechanism becomes useful. However, if a verified account is not identified, the report is not helpful.

Before beginning this research, we believed the reporting mechanism would help in the effectiveness of investigations and make it easier for an investigator to find verified information and connections. We believed that if we could gather verified information into a graphical database, we could discover connections between information elements faster or other connections that otherwise would not be found, therefore increasing the effectiveness of the reporting. The reporting method increased the time to manually analyze the information; however, we did not find it faster in terms of validating the information. We had no way of identifying information that was verified as being connected with the individual from the missing person case.

Improving effectiveness is an important aspect of improving the capability of cyber investigative methods. Efficiency is only useful if the information being collected and presented to investigators is valuable and relevant to the case. If researchers can improve the effectiveness of information collection, the efficiency improvements can be applied to create an entire OSINT program that can quickly provide valuable results.

5.2. FUTURE RESEARCH DIRECTIONS

This thesis leaves some open issues that can lead to further research, such as information verification and determining cause. As described above, information verification comes from the effectiveness of cyber investigation methods. Determining cause is where investigators analyze data on the individual and the circumstances to try to determine why the person went missing. From these open issues, there are several areas where this research can be expanded and further explored, such as expanding the information collection capabilities and implementing information validation. These improvements can further improve the efficiency and effectiveness of cybercrime investigations.

5.2.1 Open Issues

Information Verification. As demonstrated in this research, verifying information is something that currently must be done by an analyst when using open-source tools. Verifying information is where an analyst or investigator determine if an account or other piece of information is associated with the missing individual from the case. Our research showed that after information verification, investigators can use our framework to perform more information collection and analysis. If verification can be performed with automation, then the investigative process can occur faster, allowing the investigator to allocate their resources in other areas and more cases.

Determining Cause. One of the first thing investigators must do in a missing person case is determine the cause of the person being missing, such as kidnapping or running away. This helps determine where they need to look next and what their next steps should be in the investigation. Before this can occur, however, the information and account have to be verified as being associated with the missing individual.

5.2.2 Future Research

Expanded Capabilities. This framework of tools can be expanded on in the future by adding more tools and functionalities. With more tools and methods within the framework, investigators and analysts can gather more information and different kinds of information from a variety of sources. More information in an OSINT investigation gives investigators a greater chance of success.

Some of the functionalities to target with these tools were described in Chapter 2 of this research. This research focused on social network analysis and information extraction, but computer vision or natural language processing can be added for another layer of analysis. For example, AWS Comprehend is a tool that utilizes natural language processing, and because it is an AWS tool, it can easily be integrated into the existing architecture.

As discussed in Chapter 2, computer vision is using images or sound from a video to determine certain things, such as the identity of the person in the picture or video. This is a helpful tool when trying to identify an account belonging to an individual where the username does not contain their real name, but they have posts containing their face. The ability to identify an account that does not contain the name of the owner helps resolve the issue we faced in this research when we could not find a verified account belonging to the individual who was missing.

Natural language processing, as discussed in Chapter 2, uses text analyzing technologies to make determinations about the author, sentiment, or the text itself. This can be applied to social media posts or chat messages. Natural language processing can be used to determine sentiment, which can help investigators determine why the person went missing. This can help expand the research on the open issue described above and assist investigators determine the cause of a missing person case faster.

Furthermore, other open-source tools can be added to the information collection stage to increase the amount of information collected by the framework. OSINT relies on the amount of information collected for analysis. This is why it is critical for the reporting mechanism to be efficient for investigators to analyze. A consolidated and graphical reporting mechanism provides this. The more information that can be gathered on an individual, the more likely investigators are to find an important connection in the case.

Another way more tools can be added to this framework is to improve or repair the existing tools that we could not use because of compatibility or depreciation issues. There were several tools that were considered for the architecture that could not be used because they were depreciated or incompatible with the other elements of the architecture. These tools provided useful functionality that could provide other pieces of useful information to investigators.

Verifying Information. Another area where this research can be expanded and greatly improve cyber investigations is with verifying information. After investigators gather potential accounts on an individual, they must be able to determine which of the accounts belong to that individual. After information is verified, then more information collection and analysis can occur. The current framework does not have a method of automating this task.

As discussed in Chapter 2, the literature review showed that machine learning and artificial intelligence are two areas of technology that are growing and affecting all other areas

of technology. These two fields can be utilized to help validate data. For example, AWS Comprehend mentioned above, utilizes machine learning during natural language processing to make determinations about the text. This can be applied to the accounts to check if the account contains any factors that would determine if it belongs to the individual.

Another way this functionality can be added to the framework is to implement other tools and methods to expand the capability in this direction. For example, as mentioned above, the other online investigative methods from Chapter 2 can assist in validating information. For example, computer vision, paired with facial recognition, can be used to identify images of the individual from online sources. A tool such as Eagle Eye, described in Chapter 3, can be used for this. For it to be integrated into this framework, it must be modified to be compatible with the architecture of the framework.

If investigators can easily validate that information or accounts are associated with an individual, this information can be added to the database to increase the likelihood that important information will be discovered. The more verifiable information that can be gathered by investigators, the OSINT investigation will be more efficient and effective. This will increase the chances of having a successful case, which means that the missing person will be discovered.

REFERENCES

2020 internet crime report. Internet Crime Complaint Center. (2021, March). Retrieved November 10, 2021, from https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf.

Adhikari, M., Amgoth, T., & Srirama, S. N. (2019). A survey on scheduling strategies for workflows in cloud environment and emerging trends. *ACM Computing Surveys*, *52*(4), 1–36. https://doi.org/10.1145/3325097

Ak, S. (2021, September 8). *Copy files from EC2 to S3 Bucket in 4 steps: Devops junction*. Middleware Inventory. Retrieved November 22, 2021, from https://www.middlewareinventory.com/blog/ec2-s3-copy/.

Barika, M., Garg, S., Zomaya, A. Y., Wang, L., Moorsel, A. V., & Ranjan, R. (2019). Orchestrating big data analysis workflows in the cloud. *ACM Computing Surveys*, *52*(5), 1–41. https://doi.org/10.1145/3332301

Barmpatsalou, K., Cruz, T., Monteiro, E., & Simoes, P. (2018). Current and Future Trends in Mobile Device Forensics. *ACM Computing Surveys*, *51*(3), 1–31. https://doi.org/10.1145/3177847

Barua, H. B., & Mondal, K. C. (2019). A comprehensive survey on Cloud Data Mining (CDM) frameworks and algorithms. *ACM Computing Surveys*, *52*(5), 1–62. https://doi.org/10.1145/3349265

Bermudez, I., Traverso, S., Munafo, M., & Mellia, M. (2014). A distributed architecture for the monitoring of clouds and cdns: Applications to Amazon AWS. *IEEE Transactions on Network and Service Management*, *11*(4), 516–529. https://doi.org/10.1109/tnsm.2014.2362357

Bertolino, A., Angelis, G. D., Gallego, M., García, B., Gortázar, F., Lonetti, F., & Marchetti, E. (2019). A systematic review on cloud testing. *ACM Computing Surveys, 52*(5), 1–42. https://doi.org/10.1145/3331447

Bhattacharjya, S., & Saiedian, H. (2021). Establishing and validating secured keys for IOT devices: Using P3 connection model on a cloud-based architecture. *International Journal of Information Security*. https://doi.org/10.1007/s10207-021-00562-7

Billard, D. (2018). Weighted Forensics Evidence Using Blockchain. *Proceedings of the 2018 International Conference on Computing and Data Engineering*, 57-61. https://doi.org/10.1145/3219788.3219792

Brezo, F., and Rubio, Y. (n.d.). *I3visio/osrframework: OSRFramework, the open sources research framework is a agplv3+ project by i3visio focused on providing API and tools to perform more accurate online researches.* GitHub. Retrieved November 22, 2021, from https://github.com/i3visio/osrframework.

Bryant, B. D., & Saiedian, H. (2020). Improving siem alert metadata aggregation with a novel kill-chain based classification model. *Computers & Security*, *94*, 101817. https://doi.org/10.1016/j.cose.2020.101817

Bryant, B., & Saiedian, H. (2021). An evaluation of Videogame Network Architecture Performance and Security. *Computer Networks*, *192*, 108128. https://doi.org/10.1016/j.comnet.2021.108128

Caviglione, L., Wendzel, S., & Mazurczyk, W. (2017). The future of digital forensics: Challenges and the road ahead. *IEEE Security & Privacy*, *15*(6), 12–17. https://doi.org/10.1109/msp.2017.4251117

Chernyshev, M., Zeadally, S., Baig, Z., & Woodward, A. (2017). Mobile Forensics: Advances, Challenges, and Research Opportunities. *IEEE Security & Privacy*, *15*(6), 42–51. https://doi.org/10.1109/msp.2017.4251107

Creepy - OSINT - geolocation OSINT tool. Cyber Centere. (2021, March). Retrieved November 22, 2021, from https://centre.caribbeancspa.org/hc/en-gb/articles/360018796059-creepy-OSINT-Geolocation-OSINT-tool-.

Cybersecurity supply and demand heat map. Cyber Seek. (2021, March). Retrieved November 11, 2021, from https://www.cyberseek.org/heatmap.html.

DataSploit tutorial. Gather Intel & Find vulnerabilities. HackerTarget.com. (2019, November 19). Retrieved November 22, 2021, from https://hackertarget.com/datasploit-tutorial/.

Du, X., Hargreaves, C., Sheppard, J., Anda, F., Sayakkara, A., Le-Khac, N.-A., & Scanlon, M. (2020). SoK: exploring the state of the art and the future potential of artificial intelligence in digital forensic investigation. *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 1-10. https://doi.org/10.1145/3407023.3407068

Dushantha, S. (n.d.). *Sherlock-project/sherlock: Hunt down social media accounts by username across social networks*. GitHub. Retrieved November 22, 2021, from https://github.com/sherlock-project/sherlock.

Edwards, M., Rashid, A., & Rayson, P. (2015). A Systematic Survey of Online Data Mining Technology Intended for Law Enforcement. *ACM Computing Surveys*, *48*(1), 54 pages. https://doi.org/10.1145/2811403

Focus on criminals, not the crime, say researchers. (2012). *Computer Fraud & Security*, *2012*(7), 3. https://doi.org/10.1016/s1361-3723(12)70069-6

Furnell, S., Heyburn, H., Whitehead, A., & Shah, J. N. (2020). Understanding the full cost of cyber security breaches. *Computer Fraud & Security*, *2020*(12), 6–12. https://doi.org/10.1016/s1361-3723(20)30127-5

Garlan, D., Allen, R., & Ockerbloom, J. (1995). Architectural mismatch or why it's hard to build systems out of existing parts. *Proceedings of the 17th International Conference on Software Engineering - ICSE '95*. https://doi.org/10.1145/225014.225031

Gu, Y., & Lin, Z. (2016). Derandomizing Kernel Address Space Layout for Memory Introspection and Forensics. *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*: ACM, 62-72. https://doi.org/10.1145/2857705.2857707

Horan, C., & Saiedian, H. (2021). Cyber Crime Investigation: Landscape, challenges, and future research directions. *Journal of Cybersecurity and Privacy*, *1*(4), 580–596. https://doi.org/10.3390/jcp1040029

Hunton, P. (2011). The stages of cybercrime investigations: Bridging the gap between Technology Examination and law enforcement investigation. *Computer Law & Security Review*, *27*(1), 61–67. https://doi.org/10.1016/j.clsr.2010.11.001

Ignaczak, L., Goldschmidt, G., Costa, C. A., & Righi, R. D. (2022). Text mining in Cybersecurity. *ACM Computing Surveys*, *54*(7), 1–36. https://doi.org/10.1145/3462477

Infamous Emotet botnet taken down by law enforcement. (2021a). *Computer Fraud & Security*, *2021*(2), 1. https://doi.org/10.1016/s1361-3723(21)00012-9

International Law Enforcement Operation Targeting opioid traffickers on the darknet results in 150 arrests worldwide and the seizure of weapons, drugs, and over $31 million. The United States Department of Justice. (2021). Retrieved November 10, 2021, from https://www.justice.gov/opa/pr/international-law-enforcement-operation-targeting-opioid-traffickers-darknet-results-150.

Jeong, D. (2020). Artificial intelligence security threat, crime, and forensics: Taxonomy and open issues. *IEEE Access, 8*, 184560-184574. https://doi.org/10.1109/access.2020.3029280

Kamiyamane, Y. (n.d.). *JKAKAVAS/Creepy: A geolocation OSINT tool. offers geolocation information gathering through social networking platforms.* GitHub. Retrieved November 22, 2021, from https://github.com/jkakavas/creepy.

Khan, S., Gani, A., Wahab, A. W., Bagiwa, M. A., Shiraz, M., Khan, S. U., … Zomaya, A. Y. (2016). Cloud Log Forensics: Foundations, State of the Art, and Future Directions. *ACM Computing Surveys*, *49*(1), 1–42. https://doi.org/10.1145/2906149

Kumbhare, A. G., Simmhan, Y., Frincu, M., & Prasanna, V. K. (2015). Reactive resource provisioning heuristics for dynamic Dataflows on cloud infrastructure. *IEEE Transactions on Cloud Computing*, *3*(2), 105–118. https://doi.org/10.1109/tcc.2015.2394316

Li, X., Ji, S., Han, M., Ji, J., Ren, Z., Liu, Y., & Wu, C. (2019). Adversarial examples versus cloud-based detectors: A black-box empirical study. *IEEE Transactions on Dependable and Secure Computing*, *18*(4), 1933–1949. https://doi.org/10.1109/tdsc.2019.2943467

Liao, X., Yuan, K., Wang, X. F., Li, Z., Xing, L., & Beyah, R. (2016). Acing the IoC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*: ACM, 755-766. https://doi.org/10.1145/2976749.2978315

Manral, B., Somani, G., Choo, K.-K. R., Conti, M., & Gaur, M. S. (2020). A Systematic Survey on Cloud Forensics Challenges, Solutions, and Future Directions. *ACM Computing Surveys*, *52*(6), 1–38. https://doi.org/10.1145/3361216

Mittal, S., Chauhan, S., & Aggarwal, A. (n.d.). *DataSploit/datasploit: An #OSINT framework to perform various recon techniques on companies, people, phone number, bitcoin addresses, etc., aggregate all the raw data, and give data in multiple formats.* GitHub. Retrieved November 22, 2021, from https://github.com/datasploit/datasploit/.

National Crime Prevention Framework. (n.d.). Australian Institute of Criminology. Retrieved November 10, 2021, from https://www.police.qld.gov.au/sites/default/files/2018-10/NCP%20Framework.pdf.

Nazah, S., Huda, S., Abawajy, J., & Hassan, M. M. (2020). Evolution of dark web threat analysis and detection: A systematic approach. *IEEE Access, 8*, 171796-171819 https://doi.org/10.1109/access.2020.3024198

Neptune User Guide. Amazon. (n.d.). Retrieved November 22, 2021, from https://docs.aws.amazon.com/neptune/latest/userguide/load-api-reference-load.html.

Neptune. Amazon. (n.d.). Retrieved November 22, 2021, from https://aws.amazon.com/neptune/.

Nykodym, N., Taylor, R., & Vilela, J. (2005). Criminal profiling and insider cyber crime. *Computer Law & Security Review*, *21*(5), 408–414. https://doi.org/10.1016/j.clsr.2005.07.001

Pelle, I., Czentye, J., Doka, J., & Sonkoly, B. (2019). Towards latency sensitive cloud native applications: A performance study on AWS. *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. https://doi.org/10.1109/cloud.2019.00054

Police struggle to investigate cyber cases. (2018). *Computer Fraud & Security*, *2018*(12), 19. https://doi.org/10.1016/s1361-3723(18)30122-2

Qu, C., Calheiros, R. N., & Buyya, R. (2018). Auto-scaling web applications in clouds. *ACM Computing Surveys*, *51*(4), 1–33. https://doi.org/10.1145/3148149

Raaijmakers, S. (2019). Artificial Intelligence for Law Enforcement: Challenges and Opportunities. *IEEE Security & Privacy*, *17*(5), 74–77. https://doi.org/10.1109/msec.2019.2925649

Ransomware operators arrested in Ukraine amid wave of new campaigns and malware. (2021b). *Computer Fraud & Security*, *2021*(10), 1–3. https://doi.org/10.1016/s1361-3723(21)00101-9

Satyanarayana, G., Bhuvana, J., & Balamurugan, M. (2020). Sentimental Analysis on voice using AWS comprehend. *2020 International Conference on Computer Communication and Informatics (ICCCI)*. https://doi.org/10.1109/iccci48352.2020.9104105

Stoyanova, M., Nikoloudakis, Y., Panagiotakis, S., Pallis, E., & Markakis, E. K. (2020). A survey on the internet of things (IoT) forensics: Challenges, approaches, and open issues. *IEEE Communications Surveys & Tutorials, 22*(2), 1191-1221. https://doi.org/10.1109/comst.2019.2962586

Sun, D., Fu, M., Zhu, L., Li, G., & Lu, Q. (2016). Non-intrusive anomaly detection with streaming performance metrics and logs for DevOps in public clouds: A case study in AWS. *IEEE Transactions on Emerging Topics in Computing*, *4*(2), 278–289. https://doi.org/10.1109/tetc.2016.2520883

Tavabi, N., Bartley, N., Abeliuk, A., Soni, S., Ferrara, E., & Lerman, K. (2019). Characterizing Activity on the Deep and Dark Web. *Companion Proceedings of The 2019 World Wide Web Conference*: SIGWEB, 206-213. https://doi.org/10.1145/3308560.3316502

ThoughtfulDev. (n.d.). *Thoughtfuldev/EagleEye: Stalk your friends. find their Instagram, FB and Twitter profiles using image recognition and reverse image search.* GitHub. Retrieved November 22, 2021, from https://github.com/ThoughtfulDev/EagleEye.

Turan, F., Roy, S. S., & Verbauwhede, I. (2020). HEAWS: An accelerator for homomorphic encryption on the Amazon AWS FPGA. *IEEE Transactions on Computers*, *69*(8), 1185-1196. https://doi.org/10.1109/tc.2020.2988765

Wallace, B. (n.d.). *Bwall/ircsnapshot: Tool to gather information from IRC servers*. GitHub. Retrieved November 22, 2021, from https://github.com/bwall/ircsnapshot.

WebBreacher. (n.d.). *Webbreacher/WhatsMyName: This repository has the unified data required to perform user enumeration on various websites. content is in a JSON file and can easily be used in other projects.* GitHub. Retrieved November 22, 2021, from https://github.com/webbreacher/whatsmyname.

Xillwillx. (n.d.). *Xillwillx/Skiptracer: OSINT Python webscaping framework*. GitHub. Retrieved November 22, 2021, from https://github.com/xillwillx/skiptracer.

Zhang, F., Li, W., Zhang Y., & Feng Z. (2018) Data Driven Feature Selection for Machine Learning Algorithms in Computer Vision. *IEEE Internet of Things Journal*, *5*(6), 4262-4272. https://doi.org/10.1109/JIOT.2018.2845412

Zhang, L., Li, F., Wang, P., Su R., & Chi, Z. (2021) A Blockchain-Assisted Massive IoT Data Collection Intelligent Framework. *IEEE Internet of Things*, 15 pages. https://doi.org/10.1109/JIOT.2021.3049674

Zhang, X., Li, W., Ying, H., Li, F., Tang S., & Lu, S. (2020) Emotion Detection in Online Social Networks: A Multilabel Learning Approach. *IEEE Internet of Things Journal*, *7*(9), 8133-8143, https://doi.org/10.1109/JIOT.2020.3004376