Performancey Analysis Of Information Gathering Tools

Submitted in partial fulfilment for the requirements of the degree of

BSc (Hons) in **Computer Science & Artificial Intelligence**

At

University of Sussex
2023

Syed F. Shah - 231012

Supervisor: Dr. Imran Khan

Word Count: 11879

Statement

This report is submitted as part requirement for the degree of Computer Science & Artificial Intelligence at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged. I hereby give / withhold permission for a copy of this report to be loaned out to students in future years.

- Syed Fahd Shah

Acknowledgements

I would like to express my sincere appreciation to my supervisor, Professor Imran Khan, for his unwavering guidance, support, and mentorship throughout this research. His indispensable expertise, constructive criticism, and unrelenting encouragement were instrumental in shaping my comprehension of the topic and ensuring the successful completion of this thesis paper. I am extremely appreciative for the assistance he has provided, as well as the hours he has spent discussing my work and challenging my ideas. In addition, I would like to express my deepest gratitude to my parents, whose love, support, and sacrifice have made it possible for me to pursue my academic goals at the University of Sussex. Their faith in my abilities and constant encouragement have been the propelling force behind my academic journey. I will be eternally appreciative for the opportunities they have granted me and their unwavering belief in my potential.

Finally, I would like to thank the faculty and staff, and at the University of Sussex for establishing a stimulating and supportive environment that has facilitated my research growth and development. Their collective knowledge, insights, and camaraderie have significantly enhanced my educational experience and contributed to my success in this endeavour.

Contents

Abstract **6**
1 - Introduction **7**
1.1 - Objective Statement & Research Aims 8
1.2 - Relevancy 9
1.3 - Extensions 10
1.4 - Structure of Thesis 10

2 - Background **11**
2.1 - Origin 11
2.1.1 - White Box 12
2.1.2 - Black Box 13
2.1.3 - Grey Box 13
2.2 - Stages of an Attack 14
2.2.1 - Planning 15
2.2.2 - Information Gathering 15
Footprint (Sector 1) 16
Scan (Sector 2) 17
Enumeration (Sector 3) 17
2.2.2.1 - Passive Reconnaissance 19
2.2.2.3 - Active Reconnaissance 20
2.2.2 - Penetration 22
2.2.3 - Post-Penetration (Analysis) 23
2.2.4 - Cleaning House 23
2.3 - Tools 24
2.3.1 - Kali Linux 24
2.3.2 - VMware 25
2.3.3 - Nmap 26
2.3.4 - Unicornscan 27
2.3.5 - DMitry 27
2.3.5.1 - Google (?) 28
2.3.6 - Recon-NG 31

2.3.7- Maltego 33
2.3.8 - Wireshark 35
2.3.9- SQLmap 36
2.3.10 - Social Engineering 37

3 - Requirement Analysis 40

4 - Implementation 42
4.1 - Installation 42
4.2 - Maltego 45
4.2.1 - Maltego with Shodan 51
4.2.2 - Maltego with theHarvester 56
4.3 - DMitry 61
4.4 - Recon-NG 64
4.5 - NMap 68
4.5.1- Aggressive Scanning 71

5 - Conclusion 72
5.1 - Results 72

6 - Future Work 76
6.1 - Wireshark & SQLMap 76
6.2 - Social Engineering 76
6.3 - Firewalls & Preventive Measures 77

References 78
Appendices 83
6.1 Commands/Code 83
6.2 Original Proposal 83
6.3– Meeting Log 86

Abstract

Information gathering is a fundamental aspect of computer security and intelligence gathering, consisting of the systematic acquisition and analysis of data to identify potential vulnerabilities, system configurations, and other vital intelligence about a target. This procedure enables security professionals and ethical hackers to assess risks, develop appropriate mitigation strategies, and safeguard digital assets from unauthorised access or malicious activities. This thesis seeks to conduct a performance analysis of various information gathering tools within computer security, concentrating on both passive and active techniques in order to evaluate their efficiency in accomplishing their intended tasks. This analysis will be informed by a comprehensive background review as well as demonstrating the practical implications and outcomes of these methodologies. The investigation will also address future work considerations regarding the issues with particular methods of information gathering, and suitable applicable methods of countermeasures.

Keywords: Gathering, AWS, Passive, Active, Ethical, Malicious, Hackers, Efficiency, analysis

Chapter 1

1 - Introduction

Information gathering is the use of integrating the required tools to gain otherwise illegal and sensitive information[1]. The significance of effective information gathering instruments in computer security cannot be overstated in the swiftly changing digital landscape. As the number and complexity of cyber threats continue to rise, it becomes increasingly essential to comprehend the strengths and weaknesses of various information gathering techniques. This paper's primary objective is to investigate both passive and active forms of information gathering methods, comparing their efficacy and intended tasks in order to provide valuable insights for the design and implementation of effective security measures.

Reconnaissance is a crucial component of a penetration test because it sets the groundwork for the subsequent phases of evaluation. During this phase, the infiltrator attempts to gather as much information as possible about the target system, network, or organisation in order to identify potential vulnerabilities, entry points, and other valuable intelligence that can be utilised during the test. This method typically combines the usage of techniques such as open-source intelligence collection, network surveillance, and social engineering, to develop a comprehensive understanding of the target environment.

The direct impact of some of the effects of cyber attacks, can be seen to have a direct significant impact on the financial losses of individuals, businesses, and even governments. Cybercriminals have been able to exploit vulnerabilities in computer systems and networks by employing sophisticated techniques and cutting-edge tools, resulting in substantial monetary losses. These losses may result from the seizure of confidential financial data, the disruption of

[1] Laxmi Kowta, A.S. *et al.* (2021) ‘Analysis and Overview of Information Gathering & Tools for Pentesting’, in *2021 International Conference on Computer Communication and Informatics (ICCCI)*. *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–13. Available at:

essential services, or even the exfiltration of funds directly. In addition, the costs associated with detecting, mitigating, and recovering from such intrusions contribute to the victims' financial burden. In order to develop effective countermeasures and protect digital assets from malicious actors, it is essential to have a comprehensive comprehension of information gathering methods in computer security, as the far-reaching effects of hacking-induced financial loss highlight this necessity.

Since these techniques can be perceived as exceedingly malevolent, it should be stated that they should only be replicated in an educational setting and never for malicious purposes. There has been an overlaying emphasis with being diligent in addressing professional concerns and ensuring that our research adheres to the BCS Code of Conduct throughout the duration of this thesis. In all aspects of work, there has been a strict adherence to the BCS guidelines, which emphasise ethical behaviour, professional competence, and due diligence. It is essential to note that no human participants were put at risk or in peril throughout the duration of the project, as our research focused on the theoretical foundations, practical applications within virtual machines, and ethical considerations of information gathering tools and techniques within computer security. In addition, the necessary measures to complete the necessary paperwork, including obtaining all necessary approvals and permissions, in order to ensure that the research is conducted in a responsible and compliant manner.

1.1 - Objective Statement & Research Aims

With the rapid increase of digitization and storing information on platforms like the cloud, there is an inherent risk of attracting unwanted attention in the form of direct cyber attacks. Therefore, becoming familiar with some of the most notorious gathering applications can allow one’s knowledge of potential risks they may encounter to vastly grow. Therefore the objective of this thesis is to investigate the value of various information gathering tools within the realm of computer security in order to provide a thorough comprehension of their strengths, weaknesses, and applicability in a variety of situations. By comparing and contrasting these

tools, there is a hope to provide valuable insights and guidance for the deployment of these techniques as well as the creation of potential countermeasures to protect digital assets & sensitive information from potential attacks. With this understood, the thesis aims to follow these particular goals:

- Examine both passive & active information gathering techniques through the Kali Linux OS
- Use Public Sites such as Spacex.com, as a meaning of collecting Information and demonstrating how information on a target can be collected
- Compare the effectiveness of different tools and methods, taking into account factors such as speed, precision, resource consumption, and usability.

1.2 - Relevancy

The undertaking will combine information learned in third-year courses with knowledge acquired over the past several years of study at Sussex. Specifically, Computer Security, Database Structures and Algorithms, Introduction to Programming (and modules focusing on Python programming), and web services and applications would be integrated. Computer security has proven to be an invaluable asset to this project, providing in-depth guidance on applications such as DVWA and how to apply it to the cloud for penetration testing. Furthermore, the understanding on how to use the Kali Linux platform has also demonstrated to be instrumental.There is also a noticeable close relationship with studies in Psychology and Cognitive Science's foundations, in which we could understand a victim's mentality when exposed to various types of phishing techniques and how they may be susceptible to being deceived has been facilitated by the knowledge acquired from these courses. Lastly, this assignment will evaluate the report-writing and professional writing skills I've acquired during my time at university.

1.3 - Extensions

The research will investigate the use of social engineering as active reconnaissance techniques. Social engineering techniques have emerged as a prevalent method for gaining unauthorised access to sensitive data, frequently by exploiting the human element of security systems. By analysing the tactics employed in social engineering attacks and their role in active reconnaissance, our study aims to cast light on the growing importance of this aspect of cybersecurity. Furthermore, a look into Firewalls, as a crucial layer of network security defence, and how they have the potential to significantly impact the efficacy of reconnaissance efforts. Improving overall security posture necessitates a more in-depth understanding of how firewalls can be configured and deployed to deter active reconnaissance attempts.

1.4 - Structure of Thesis

This research project aims to follow the following structure throughout its entire life cycle:

- **Chapter 2 (Background) -** Conduct a thorough literature review of extant information gathering tools and techniques, focusing on their underlying principles, methodologies, and potential limitations.

- **Chapter 3 (Requirement Analysis) -** Showing what is needed to accomplish the task at hand, if there are any external problems and how they might be addressed.

- **Chapter 4 (Implementation) -** Discusses the methods and tools that will be integrated and tested to determine their effectiveness and intended tasks. Further showing how they are used in a professional setting with appropriate evidence.

- **Chapter 5 (Conclusion) -** Concludes the thesis and presents a summary of the results obtained.
- **Chapter 6 (Future Work) -** Suggests potential recommendations for future work.

Chapter 2

2 - Background

To understand the significance and ramifications of information gathering instruments in computer security, it is necessary to comprehend the historical, technological, and theoretical foundations that have influenced their development and application. The purpose of this section is to provide the reader with context and background information on the evolution of information gathering techniques, their role in the broader landscape of cybersecurity, and the fundamental principles that govern their application. This section will conclude with a very brief introduction to some of the most widely used tools that are going to be featured throughout this thesis, and their specific applications.

2.1 - Origin

Penetration testing, also known as "Pen Testing," is a proactive method of evaluating the security posture of a computer system, network, or application by simulating real-world intrusions in order to identify vulnerabilities and potential hazards. In the early days of computing and networking, organisations and governments recognised the need to safeguard digital assets from unauthorised access and exploitation. As the complexity of computer systems and the number of cyber threats increased over time, penetration testing became a more structured and specialised discipline, with dedicated methodologies, tools, and frameworks designed to facilitate the systematic evaluation of security defences. As seen in figure 1, pen testing can be distinguished throughout its 3 different variants.

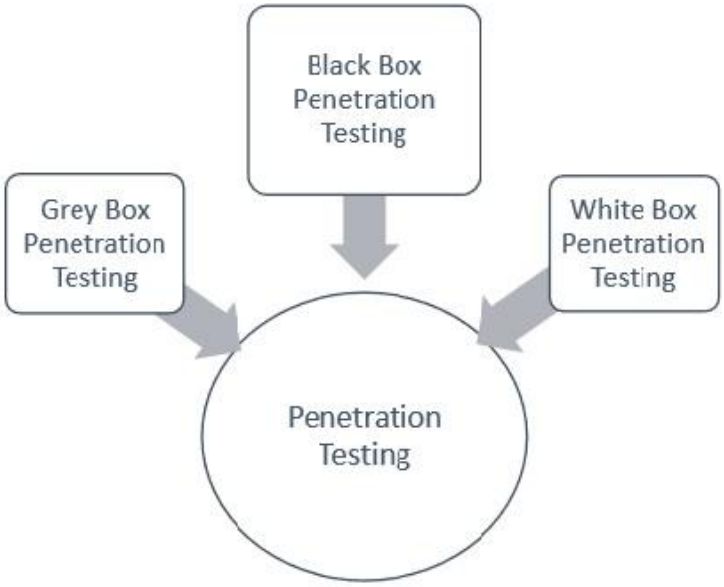

Figure 1: *Types of Pen Testing*[2]

2.1.1 - White Box

White box penetration testing, also known as clear box or transparent box testing, is a type of pen testing in which the infiltrator has in-depth knowledge of the target system, network, or application before conducting the test. This information may include IP address', documentation, networking layouts, source code and so on, giving the tester an "insider's perspective" of the target environment. White box hacking is also seen as the culmination of extensive information gathering/reconnaissance. Furthermore, White-box penetration testing provides a thorough evaluation of both internal and external vulnerabilities, making it the best option for calculation testing. The intimate relationship between white-box pentesters and

[2] https://www.tutorialspoint.com/penetration_testing/types_of_penetration_testing.htm

developers provides a high level of system knowledge but may influence testers' behaviour, as testers operate based on information not accessible to hackers[3].

2.1.2 - Black Box

In contrast to white box penetration testing, black box pen testing, also known as closed box or opaque box testing, is a technique where no prior knowledge of the target system, network, or application is given to the tester. The tester simulates the viewpoint of an external adversary who attempts to obtain unauthorised access or exploit vulnerabilities without intimate knowledge. This could be seen as the most practical form of attacking due to the nature of having to evaluate a target system for its security. Information gathering within this stage would be perceived as a necessity due to being subjected to infiltrating whilst blind. During the information gathering phase of a black box penetration test, the tester accumulates as much data as possible about the target through passive and active reconnaissance techniques. The tester then uses the collected information to identify potential entry points, vulnerabilities & probing the system for security vulnerabilities[4].

2.1.3 - Grey Box

As can be inferred by how it is referenced, Grey box penetration testing is a hybrid between white box and black box testing in which there is a limited knowledge of the target system. This limited information may include a target system’s architecture diagram, or access credentials, but not the entire source code or other details like that of white box testing. The purpose of this is to create a scenario wherein lies an attacker who has acquired some information about the target, conceivably by means of surveillance, or prior exploits

[3] https://resources.infosecinstitute.com/topic/what-are-black-box-grey-box-and-white-box-penetration-testing/
[4] Midian, P. (2002) ‘Perspectives on Penetration Testing — Black Box vs. White Box’, *Network Security*, 2002(11), pp. 10–12.

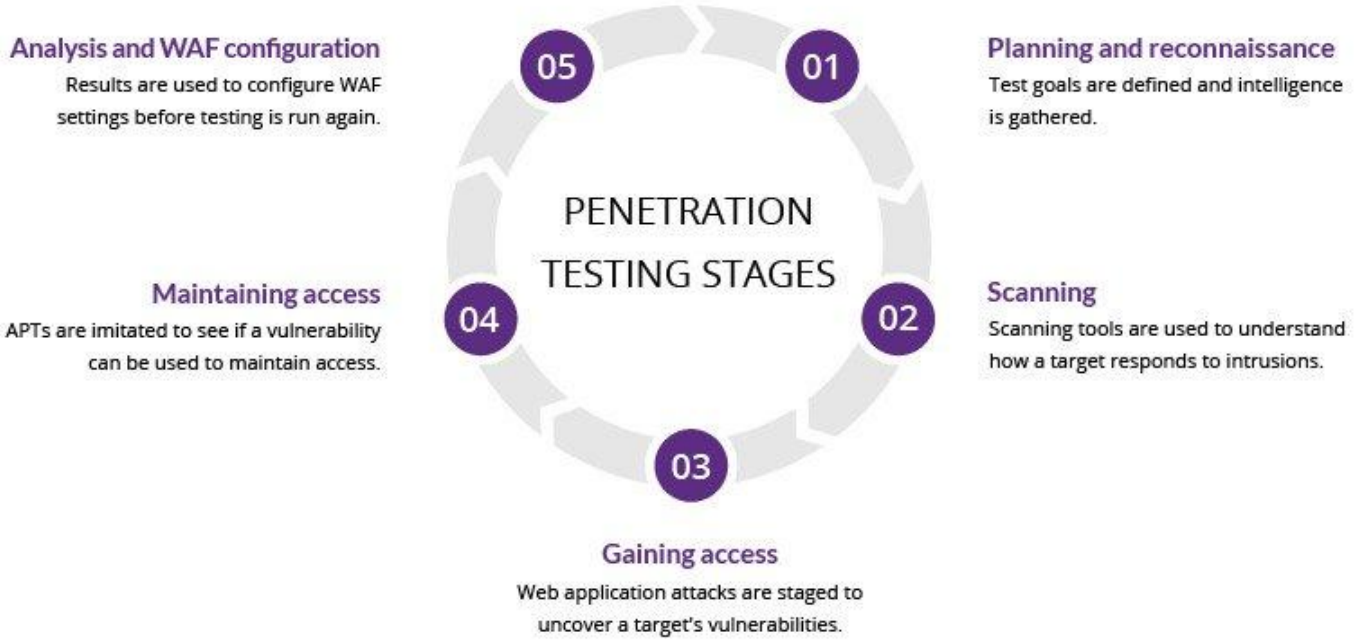

Figure 2: *The Process of a Penetration Test*[5]

2.2 - Stages of an Attack

Pen testing is a methodical procedure consisting of multiple phases, each of which is designed to ensure a thorough and comprehensive evaluation of the security posture of a target system. Engebretson (2013) states that penetration testing can come down to the following steps: Planning and reconnaissance, information gathering, vulnerability analysis/scanning, gaining access, attempting to maintain said access, post-exploitation analysis, and reporting[6]. By adhering to a systematic methodology, pen testers can identify vulnerabilities in a logical manner, maximising the test's efficacy, providing valuable insights to assist organisations in prioritising and addressing security risks. In addition to ensuring that every aspect of the target

[5] https://www.imperva.com/learn/application-security/penetration-testing/
[6] Patrick Engebretson. The Basics of Hacking and Penetration Testing, 2nd Edition. 2nd ed. 225 Wyman Street,

environment is thoroughly examined, the phased process of penetration testing contributes to a robust and resilient cybersecurity strategy that is better outfitted to defend against actual cyber threats.

2.2.1 - Planning

The preparation stage of a penetration test is essential for laying the groundwork for a successful evaluation of the defensive status of the intended target entity. During this phase, the pen tester, in combination with the clientele, collaborates to establish the test's scope, objectives, and boundaries[7]. Key considerations include identifying the target systems or applications, determining the scope of access to be granted, establishing a timeline, and delineating any compliance or regulatory requirements that must be met during the testing process. Therefore, in layman's terms, the planning stage acts as a phase in which the tester becomes wholly acquainted with a target's background, planning which vulnerabilities could be further examined for the bona fide penetration attempt. The contractor requiring the service of a Pen Test, may also be explicitly clear on the deadline of the attempt as well as setting the boundaries.

2.2.2 - Information Gathering

As the main section of the Pen Test for the purpose of this thesis, the Information Gathering stage, also commonly referred to as the "reconnaissance " phase, involves the entails collecting as much pertinent information as possible about the target system, network, or application, allowing the tester to develop a thorough comprehension of the target environment and identify potential access points, attack vectors, and vulnerabilities[8]. Active and passive reconnaissance are the two primary classifications of information acquisition. Observing network traffic or investigating publicly available information on the Internet are examples of passive

[7] Geer, D., and J. Harthorne. "Penetration Testing: A Duet." Proceedings Of The 18Th Annual Computer Security Applications Conference, 2002 (2002): 185.
[8] Saindane, Manish. "Penetration Testing - A Systematic Approach." (n.d.): n. pag. Www.infosecwriters.com. 2009.

reconnaissance also referred to as OSINT (open source intelligence)[9]. Active reconnaissance involves direct interaction with the target system, such as scouring for accessible ports, detecting operating services, and probing for vulnerabilities. Both approaches are essential for constructing a comprehensive profile of the target environment and informing the tester's strategy for the subsequent phases of the penetration test. The information gathering phase has become increasingly complex, with a vast array of tools and techniques to aid security professionals/perps in their reconnaissance efforts. This evolution is a result of the realisation that a thorough and well-informed information gathering phase is essential to the overall success of a penetration test, as it enables testers to prioritise their efforts, concentrate on high-risk components, and ultimately improve the efficacy of security assessments. Based on the differential between OSINT and Active intelligence, the variation of Intelligence gathering could be split into 3 different sectors:

Footprint (Sector 1)

Sector 1 of information collection involves extracting target information and how far the span of the target system is. It is a method for collecting restricted information about the target system passively. Due to the nature of this, extracting this level of information, primarily using automated tools to extract data, would be mostly beneficial. With that in mind, social engineering techniques could also be used to deceptively retrieve information without the target noticing anything suspicious . Whois & NsLookup, are both examples of footprinting techniques commonly used within the infosec environment.

[9] Weidman, G. 2014. Penetration testing: a hands-on introduction to hacking. San Francisco: No Starch Press, Inc.

Scan (Sector 2)

Sector 2 of information gathering integrates the usage of Scanning, a method of gathering classified information about the target system, such as exposed ports or running bespoke toolkit applications. With integrating knowledge acquired from sector 1, automation can be used to perform a network scan on the target system as well. This level necessitates extensive knowledge of the Infrastructure to be compromised, its physical location, organisational behaviours, and relationships. This will enable the penetrator to learn about their security strategy. Popular network & system scanning surveillance utilities include Recon-NG, NMap/ZenMAP, Wireshark etc.

Enumeration (Sector 3)

In the context of information gathering during a penetration test, Enumeration refers to the process of extracting detailed information about a target system/network in order to obtain a greater comprehension of its structure and potential vulnerabilities. Beyond the initial reconnaissance phase, enumeration focuses on amassing more specific information that can be used to refine the tester's attack strategy and identify potential entry points or vulnerabilities in the target environment[10]. Enumeration involves the previous sectors, in further understanding these vulnerabilities, and as such, can be split into different types of enumeration methods. Network-based enumeration involves identifying the services operating on the target system and locating any exploitable vulnerabilities associated with those services. Using instruments like Nmap, this can be accomplished. Through using the file dictionary system, identifying files, directories, and their respective permissions on the target system can provide valuable information about potential vulnerabilities or sensitive data that may be exploited during the penetration test, which can be accomplished with applications such as DirBuster or dirsearch.

[10] 'Scanning and Enumeration Phase' (2019) in *Constructing an Ethical Hacking Knowledge Base for Threat Awareness and Prevention*. IGI Global, pp. 149–177. Available at: https://doi.org/10.4018/978-1-5225-7628-0.ch006.

Whilst both of these concern system networking and the file directory on a target network, DNS enumeration typically Includes collecting information on the domain infrastructure of the target organisation, including domain names, subdomains, mail servers, and IP addresses. This can be accomplished using DNSenum, Fierce, DNS recon, Etc. Overall, the process of enumeration holds significant importance in the information gathering phase of a penetration test. It enables the tester to construct a comprehensive and precise profile of the target environment, thereby facilitating the development of an effective strategy for the subsequent stages of the assessment[11]. The successful execution of enumeration can be seen to increase the probability of a fruitful penetration test.

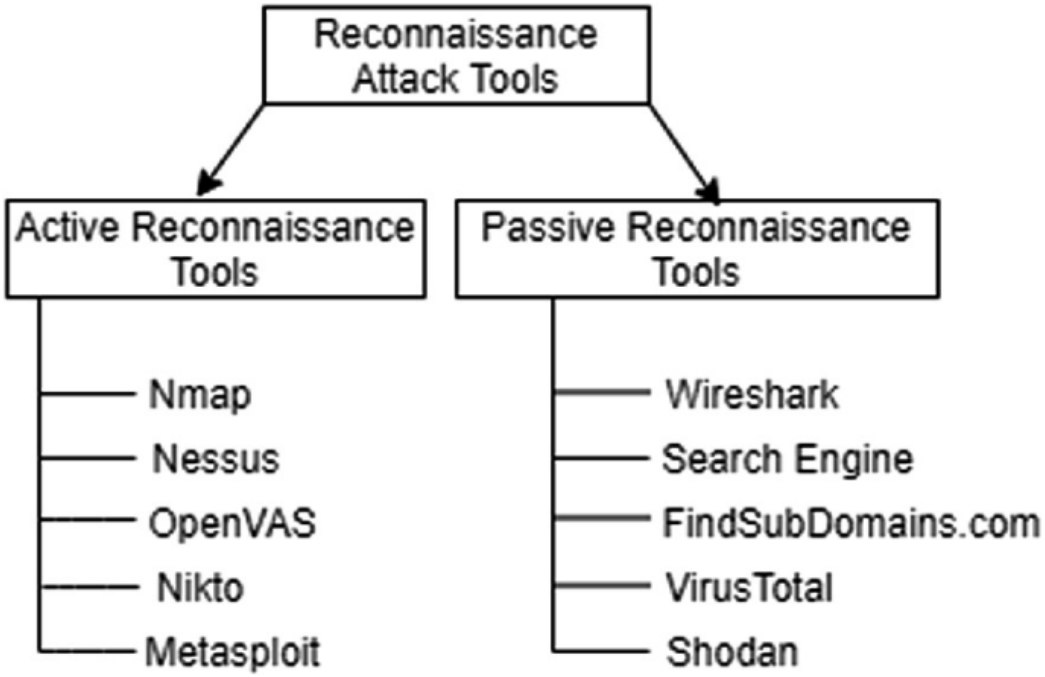

Figure 3: *Reconnaissance Techniques*[12]

[11] Sinha, S. and Arora, Dr.Y. (2020) 'ETHICAL HACKING: THE STORY OF A WHITE HAT HACKER', *International Journal of Innovative Research in Computer Science & Technology*, 8(3). Available at: https://doi.org/10.21276/ijircst.2020.8.3.17.

[12] Rana, S., Garg, U. and Gupta, N. (2022) 'Reconnaissance Attacks: A First Step to Hack IoT Devices and Cyber Crime', in A.K. Das et al. (eds) *Computational Intelligence in Pattern Recognition*. Singapore: Springer (Advances in Intelligent Systems and Computing), pp. 183–194.

2.2.2.1 - Passive Reconnaissance

Passive reconnaissance is the process of gathering intelligence on a target system without interacting with it directly. As mentioned previously, passive reconnaissance is typically the first stage in a penetration testing engagement, as it serves to create a profile of the target and identify potential vulnerabilities without alerting the target to the tester's presence[13]. Since passive reconnaissance does not involve direct interaction with the target's systems, it is less likely to be detected by the target or to set off security countermeasures. However, the information collected through passive reconnaissance may be obsolete, or insufficient, necessitating additional validation through active reconnaissance techniques. Methods of Passive Recon can include the following:

- **OSINT (Open Source Intelligence)** - Information gathered from publicly accessible sources, including websites, social media, public databases etc. This may involve browsing for employee information, corporate structure, or the technology platform that a company frequents[14].
- **Network Infrastructure** - Analysation of network-based traffic to and from the target system in order to identify patterns or protocols. This method necessitates access to network traffic data without engaging the target system directly.
- **Search Engine** - Using specialised search engines, such as Shodan, can be used to discover information about the target, exposed web applications, and connected devices are just a couple of instances. Advanced search operators can assist in refining results and locating additional details.
- **Certificates** - Examining SSL/TLS certificate transparency records in order to identify domain names and subdomains belonging to the target organisation. This can reveal the

[13] Hidayah Zulkiffli, S.N., Ahmad Zawawi, M.N. and Rahim, F.A. (2020) 'Passive and Active Reconnaissance: A Social Engineering Case Study', in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*. *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pp. 138–143..

[14] Schaurer, F. and Störger, J. (2013) 'The Evolution of Open Source Intelligence (OSINT)'.

web infrastructure and potential attack surfaces of the target. Tools like Netcraft, are used to discover information concerning a domain's network, SSL/TLS, hosting history, associated addresses and email, parent organisation, and so on.

2.2.2.3 - Active Reconnaissance

As discussed, active reconnaissance is the act of directly interacting with a target system or network in order to collect information regarding its configuration & vulnerabilities. Active reconnaissance is more intrusive than passive, and therefore carries a significant risk of detection via intrusion detection systems and firewalls. Conducting active reconnaissance in a careful and ethical manner is crucial, particularly when carried out within the framework of authorised security assessments, usually discussed during the planning stage. Tools used within active reconnaissance environments include:

- **Port Scan -** In order to identify the services operating and their corresponding versions, port scanning a machine entails sending packets to certain ports on a target machine. For port scanning, programmes like Nmap, Unicornscan, and Netcat are often used. Ports are the sites of communication that enable the exchange of data over a particular network. By identifying the exposed ports on a target machine, it is possible to determine which services are being used. This information is useful for attackers looking to determine what exploit vulnerabilities can be used[15].
- **Packet Sniffing** - Packet sniffer, also known as packet capturing or network traffic analysis, is the process of interpreting and analysing the data packets transmitted over a network between devices. Packets can be analysed by examining their Source and

[15] Smith, B., Yurcik, W. and Doss, D. (2002) 'Ethical hacking: the security justification redux', in *IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No.02CH37293). IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No.02CH37293)*, pp. 374–379. Available at: https://doi.org/10.1109/ISTAS.2002.1013840.

Destination IP addresses, Protocol types, Payload, and other performance metrics. An example of a tool would be wireshark or ettercap.

- **Network Analysis** - Network mapping is the process of constructing a visual representation or schematic of a target network, including its connections and interconnections. It provides a distinct comprehension of the structure of the target network, which can be used to identify potential weaknesses or misconfigurations. Maltego, an example of a networking tool, is.a potent open-source intelligence (OSINT) and graphical link analysis application that can be used to visualise and analyse network relationships and data collected from a variety of sources. While it incorporates elements of passive recon, it allows for a more aggressive approach to collecting and displaying that data.
- **Website Scanning**- Website scanning results in the manipulating and analysing a target web-based application to gather information about its structure, functionality, and potential security vulnerabilities. Burp Suite is an extremely comprehensive web application security testing platform comprising an automated vulnerability scanner, a proxy for intercepting web traffic and other tools needed for a web-based test. Furthermore, sqlmap is an open-source penetration testing tool that automates the process of detecting and exploiting SQL injection vulnerabilities in web applications. SQL injection is a prevalent web application security vulnerability that enables an adversary to inject malicious SQL queries into user input fields, thereby acquiring access to the application's database.
- **Social Engineering** - Whilst there is an understanding that computers will always be vulnerable, there are also arguments that concern human behaviour as a larger target for exploitation. This entails coercing members of the target to provide confidential information or do activities that might result in a security breach. Phishing, baiting, and quid pro quo are some of the methods most commonly found. The Social Engineering Toolkit (SET), an open-source framework made for security analysts and penetration testers. It already has Kali Linux installed and focuses on social engineering attack

vectors. The goal of SET is to automate and streamline the process of carrying out different social engineering assaults[16].

Using any of the aforementioned tools within reconnaissance, would also result in a stage of the “vulnerability assessment” being achieved. This utilises a portion of information gathering period to actively assess the vulnerabilities that have been discovered. By utilising vulnerability databases and a succession of active tests, vulnerability scanners attempt to identify client system vulnerabilities. If this step fails, the risk of injecting erroneous exploits increases, and errant exploits can cause services to malfunction and trigger intrusion-detection alerts. Therefore in essence, critical reasoning is another mandatory attribute a pen tester must have.

2.2.2 - Penetration

The Exploitation/Actual Penetration phase of a penetration test is the phase in which the perpetrator, having identified vulnerabilities and weaknesses in the target system during reconnaissance and vulnerability detection, actively attempts to exploit these security holes[17]. During this phase, the penetration tester uses a combination of tools, techniques, and knowledge to obtain unauthorised access to the target environment, extract sensitive data, or disrupt services. The goal is to simulate the actions of a real-world adversary in order to evaluate the system's resilience against cyber threats and identify potential security flaws. By effectively exploiting identified vulnerabilities, the tester can demonstrate the potential repercussions of a real attack, providing the benefactor with invaluable data. The information collected during this phase enables the organisation to prioritise and resolve vulnerabilities, thereby improving the security posture and decreasing the likelihood of a successful attack by malicious actors.

[16] https://www.trustedsec.com/tools/the-social-engineer-toolkit-set/

[17] . Maynor, David, K. K. Mookhey, Jacopo Cervini, Fairuzan Roslan, and Kevin Beaver. "Metasploit Toolkit for Penetration Testing Exploit Developement." (2007): n. pag. Www.syngress.com.

2.2.3 - Post-Penetration (Analysis)

In the instance of either a successful or incapable attempt on a target system, the appropriate documentation and recording of the event needs to take place. During this phase, the tester examines the compromised systems, accessed data, and level of control over the target environment and whether that could be maintained over a long period of time, sufficient enough to wreck significant havoc. Within this report, the tester investigates the root causes of exploited vulnerabilities, which may include software defects, misconfigurations, or in the instance of social engineering, human factors. Identifying root causes such as the aforementioned reasons, tends to be crucial in successfully fixing vulnerabilities and preventing future dilemmas.

2.2.4 - Cleaning House

After which the target system is compromised and dealt with, it is the responsibility of the tester to “clean house” or restore the system back to its original state. This means that eliminating any tools, exploits, or artefacts introduced during testing, and ensuring that no residual impact remains, is extremely important. This is also critical as it allows the organisation or affected target, to recommence normal operations/”business as usual” without any residual effects and demonstrates the tester's dedication to ethical practices.

2.3 - Tools

Tools are crucial to the efficacy and effectiveness of a penetration test, as they aid ethical hackers in automating processes and systematically identifying vulnerabilities. Testers have access to a variety of tools, including network mapping utilities, exploit frameworks, and decryption software. These instruments not only accelerate the testing process, but also ensure a comprehensive and consistent evaluation of the target setting. Most tools, besides being split between active and passive recon, tend to be open or closed source. The main inherent difference between open and closed source platforms/software, is that the open source tools have their source code publicly available, meaning that their firmware tends to be regularly updated, either by the author or community. On the opposite end of the spectrum, closed-source programmes are the author's property, in which they hold the legal right to modify or improve the programme at any moment. Since this dissertation will introduce practical elements and cross-compare the usage of active and passive recon tools, this portion will serve to introduce the tools that shall be used and compared. Due to the interlink into active & passive tools respectively, 5 tools will be examined and the results they produce will be cross–analysed. There will also be an opportunity to delve into social engineering which tends to be an entire sector itself.

2.3.1 - Kali Linux

Serving as the main operating system for the entire thesis, the Kali Linux system is a debian-based distribution of linux that is primarily used for the purpose of penetration testing[18]. Offensive Security, the organisation that maintains it, offers an extensive set of tools, utilities, and resources for educational or professional attacks. Kali Linux comes pre-installed with more than 600 security tools, including well-known applications that appeal to various aspects of security testing, vulnerability assessment, and exploitation. Due to the open source nature, the distribution is routinely moderized, ensuring that its users have access to the most up-to-date

[18] https://www.kali.org

security tools and features. Kali Linux is highly configurable, allowing users to configure the system according to their requirements, and it supports multiple desktop environments, including GNOME, KDE, and Xfce.

2.3.2 - VMware

VMware is a provider of cloud computing and virtualization software used to reconstitute the instance of specific software categories, which would otherwise necessitate the configuration of a physical machine. By decoupling operating systems and applications from the underlying hardware, virtualization enables more efficient utilisation of resources and enhanced flexibility.

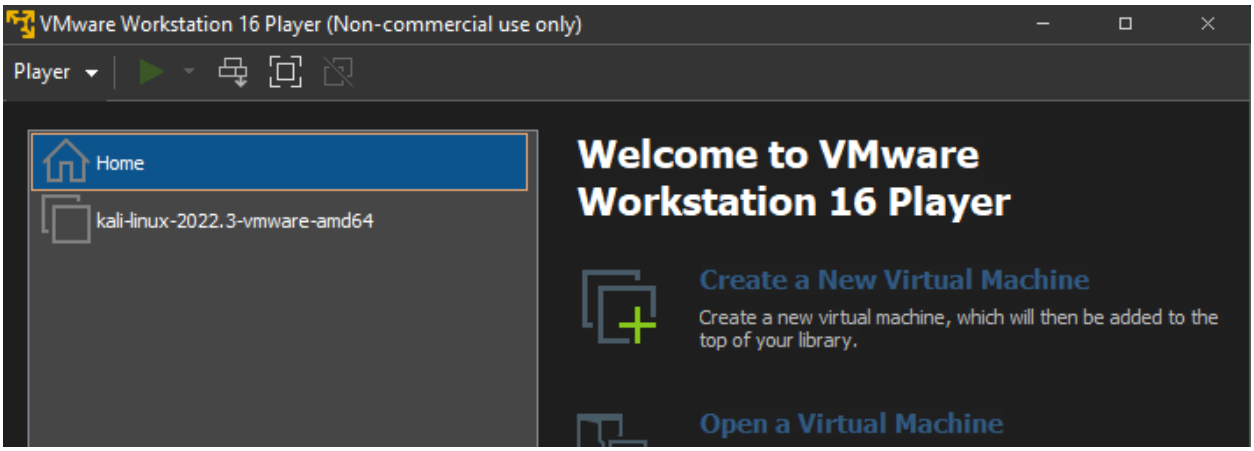

Figure 4: *Instance of VMware Running Kali Linux*

As seen in Figure 4, there is an active running workstation hosted in the cloud, which is currently coping with an instance of Kali Linux. This allows the user to access their saved instance of this particular Kali Linux, from anywhere, as long as they have a VMware account.

2.3.3 - Nmap

Nmap, which stands for "Network Mapper," is a popularly-used, open-source network scanning utility designed to identify hosts, services, and security flaws on network systems. Nmap can scan vast networks or a single host with ease. It is a highly effective tool that is backed by an active community of users and developers. The Nmap programme contains multiple utilities that can be used for scanning and querying.

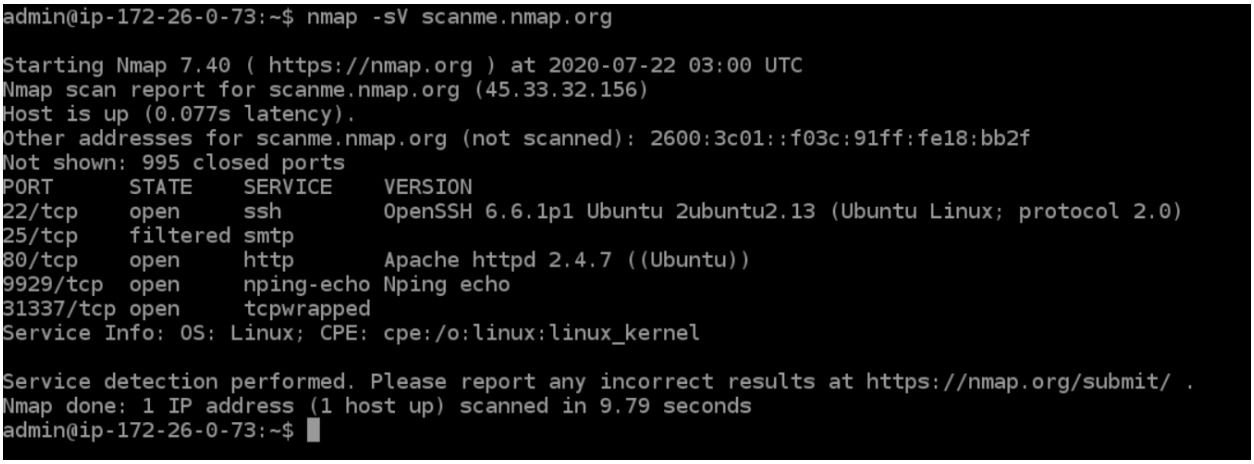

Figure 5: *Open Ports on an Nmap Scan*[19]

As seen in Figure 5, Nmap utilises raw IP packets in novel ways to identify what hosts are accessible on a network, which applications those hosts are running, what operating systems (and their respective versions) they have running, and dozens of additional information. It is intended to scan large networks quickly, but it also functions well against solitary hosts. Nmap is compatible with all main computer operating systems and is available in both console and graphical formats.

[19] https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/

2.3.4 - Unicornscan

Unicornscan is another port scanning method that could be pegged on the same level of NMap. That being said, there are arguments as to why an individual would use Unicornscan as opposed to NMap, as it is already an extremely well documented port scanning tool. The reason for this is due to how “noisy” (as previously mentioned) nmap truely is. When examining large IP ranges or several ports at once, or when scanning multiple ports simultaneously, Unicornscan is especially well suited.While Unicornscan has advantages in some situations, Nmap is a well-liked and versatile network scanning tool. Because Unicornscan uses an asynchronous scanning method, it can send out several connection requests simultaneously without having to wait for responses. Additionally, it can handle many concurrent connections, making it appropriate for scanning large networks or performing fast port scans.

2.3.5 - DMitry

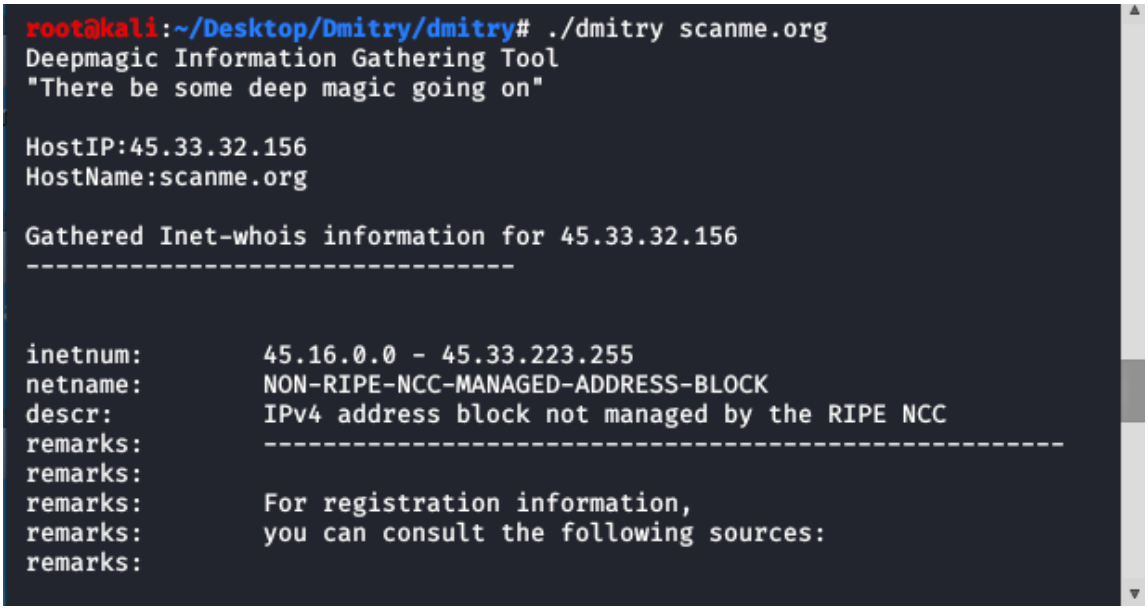

Figure 6: *A Typical Scan on DMitry*[20]

[20] https://www.geeksforgeeks.org/dmitry-passive-information-gathering-tool-in-kali-linux/

DMitry (Deepmagic Information Gathering Tool) is a C-programmed UNIX/(GNU)Linux command-line application. Dmitry can acquire as much knowledge as feasible regarding a host. Basic capabilities can collect potential subdomains, email addresses, uptime information, tcp port scans, whois lookups, Etc. Figure 6 shows the UI used within DMitry when using the whois lookup functions, furthermore the entire tool is technically run only within a command line and doesn't hold a GUI interface like other common tools. Besides using the whois lookup function, DMitry could also resolve IP addresses and enumerate subdomains, making it an essential tool for a pen tester.

2.3.5.1 - Google (?)

While potentially seen as an unconventional form of reconnaissance, Google is still an extremely effective form of researching and acquiring passive information about a target, other than being an effective search engine. The use of Google and other search engines (such as shodan), allow for pen testers to identify information through public means that should not have otherwise been disclosed[21]. Google can be helpful when searching for statistics as well as additional valuable information, such as disc space usage or system records generated by network tracking software. Google is also capable of searching for HTTP messages of error, which can provide highly useful data about the system, database structure, and configuration, as well as searching for usernames and passwords, personal info, and confidential documents. In addition to Google, accessible resources such as centralops or digitalpoint can be used in tandem to acquire Internet-based information about a target. Google Dorks is a method of searching that employs sophisticated search operators in Google and other search engines to discover particular data or security flaws. These search operators assist in refining queries by adding additional parameters, allowing users to obtain more precise and pertinent outcomes. An example of using a

[21] Piotrowski, M. (2005) 'Dangerous Google – Searching for Secrets'. Retrieved from https://doc.lagout.org/security/DangerousGoogle-SearchingForSecrets.pdf

dork method is by searching for file types by using particular queries. By searching for files using keywords like *'filetype:pdf "secret",* then any pdf file that contains the word secret will be returned to the user. In the same instance, using the term *'inurl:login intitle: "login"*, will return any login URL's that contain the term "login".

In addition to Google, other search engines like Shodan (Sentient Hyper-Optimised Data Access Network), have been used for infosec purposes. Commonly referred to as the "Hacker's Version of Google", Shodan has the capability of finding open, obsolete, and unsecure devices such as phone systems, network drives, routers and switches, CCTV cameras, open network printing devices, and surveillance systems. These tend to always have a finite (yet increasing) number of platforms that are either open or configured with a default password that is straightforward to obtain , through cracking or basic decryption techniques.

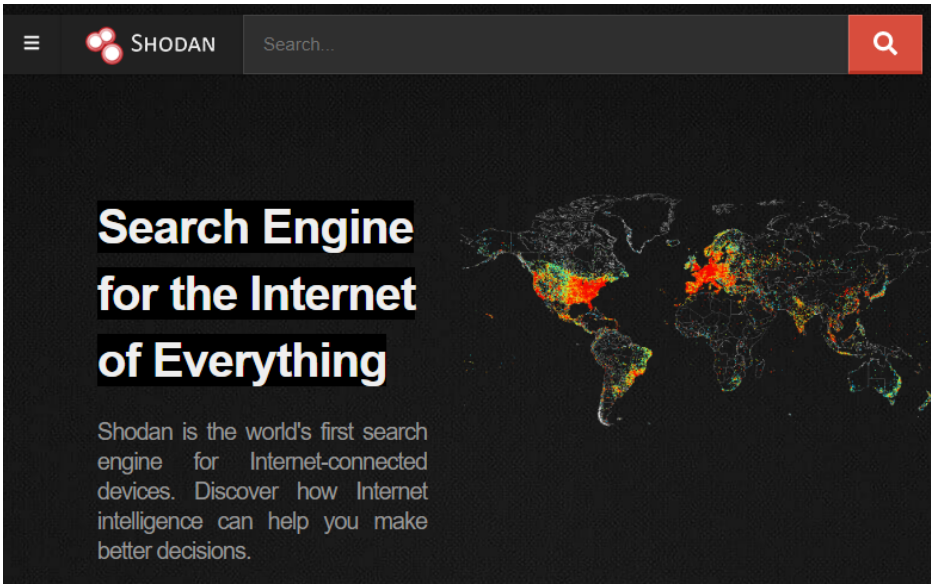

Figure 7: *Front Page of Shodan*

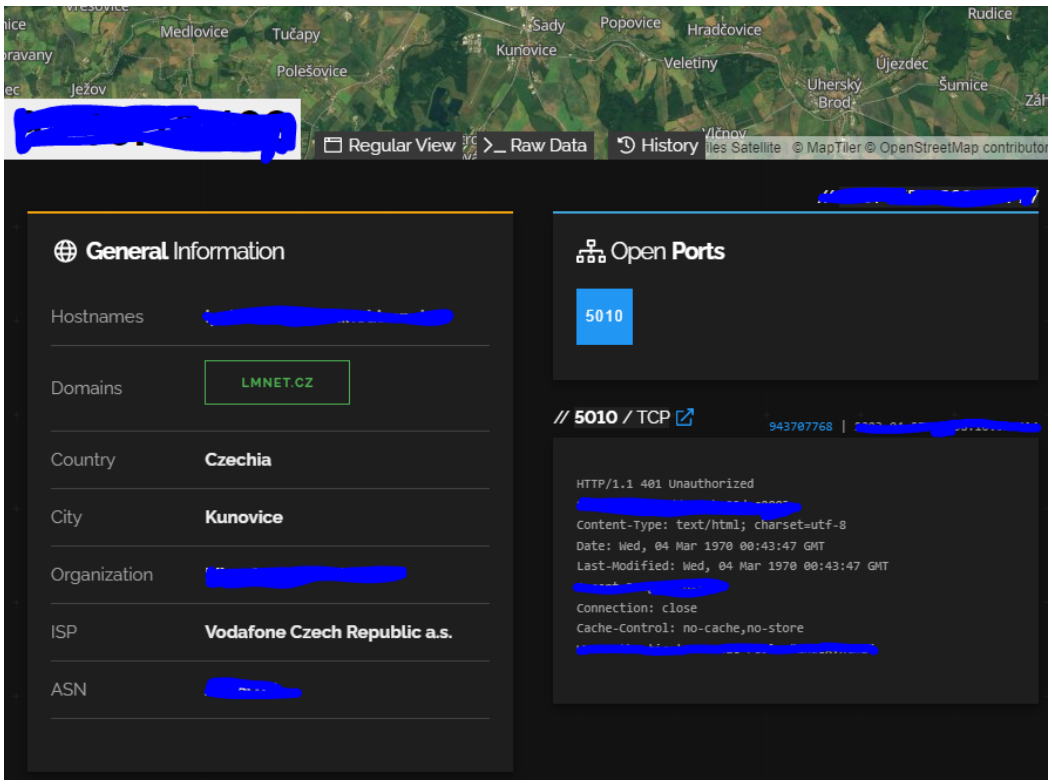

Figure 8: *An Example of a Unsecure System (Shodan)*

Figure 7 shows the front page of Shodan, presenting an extremely simple UI to get started, making it almost entirely virtually impossible for anybody to have any serious difficulties with. To get started, Shodan offers an explore tab whereby users are able to select a variety of different obsolete/unsecure devices they could manipulate. In Figure 8 (hiding sensitive data to protect integrity), a live example of what it's like to take over a unsecure device is seen. Here, a camera in Czechia was insecure and hasn't been updated since 1970, meaning the firmware could be a potential breaching point for a pen tester, proving how platforms like Shodan could be extremely beneficial for looking into security flaws.

2.3.6 - Recon-NG

Figure 8: *Intial Instance of Recon - NG*

As seen in the figure 8, Recon-NG is another command line, open-source reconnaissance framework created to gather information about those being targeted during the early phases of a security evaluation. It employs a similar modular architecture to the Metasploit Framework, allowing users to readily extend its capabilities with bespoke extensions or modules. The tool comes built-in with numerous modules that simplify the procedure of accumulating data from diverse sources, including social media, internet search engines, and publicly accessible databases. The aforementioned modules enable users to effectively gather data pertaining to target organisations, networks, domains, and people in general, thereby contributing to the development of a comprehensive profile of the intended setting[22]. Although it is a lengthy

[22] Chauhan, S. and Panda, N.K. (2015) *Hacking Web Intelligence: Open Source Intelligence and Web Reconnaissance Concepts and Techniques*. Syngress.

procedure, automatically executing data with Recon-NG tends to save overall time and hassle, with the obtained results being able to be saved for later review. This autonomous passive collection of information on the target websites, can be done in the following example:

> Google can be firstly and primarily used to enumerate sub-domains. Then to display other subdomains, using switches such as "site:" and "inurl:" will remove all other sub-domains (-inurl) found in the search field.

Finally, Recon-NG's connection with external application programming interfaces and services strengthens its capabilities, allowing users to take advantage of additional data sources and tools for a more comprehensive reconnaissance procedure. Recon-ng is a valuable tool for professionals in cybersecurity because it provides an extensible and efficient platform for amassing comprehensive information during the reconnaissance phase of a security evaluation.

2.3.7- Maltego

One of the most powerful & versatile programs (that even features its own GUI with visualisation) is Maltego. Serving as a data mining open source piece of kit that utilises primarily OSINT, Paterva (the company that maintains the software) has enabled Maltego to assist clients in gathering, analysing, and visualising intricate sets of data from different sources, permitting them to uncover concealed trends, relationships, and links between those entities such as individuals, organisations, domains, and Internet Protocol (IP) addresses.

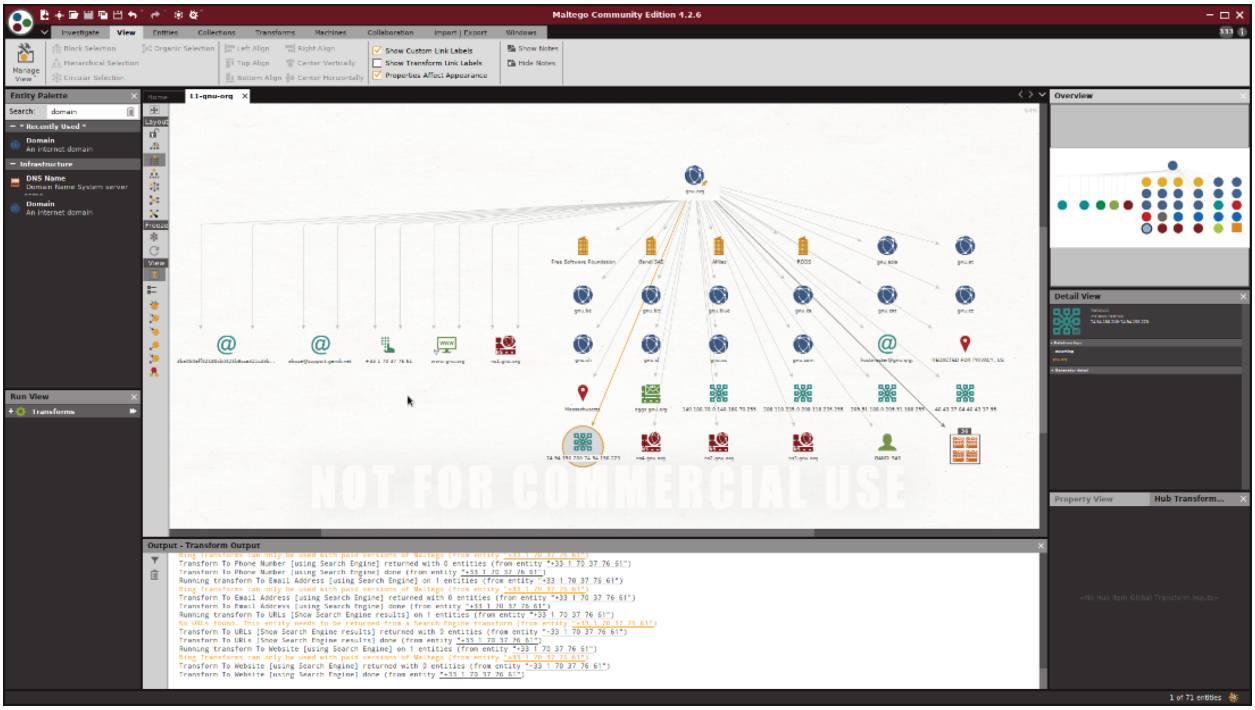

Figure 9: *A Example of a Transformation Tree in Maltego*[23]

The way in which Maltego works is in allowing users to begin by selecting one or more entities to act as their investigation's starting point. Entities may encompass a variety of data categories, including individuals, email addresses, IP addresses, web addresses, and social networking accounts. The next phase is by applying Transformations. Transforms are algorithms

[23] https://www.maltego.com/blog/beginners-guide-to-maltego-setting-up-maltego-community-edition-ce/

or processes that are applied to entities in order to collect additional data or identify relationships. Maltego examines various sources of information as well as APIs depending on the specified entity when executing a transform. For instance,a transform may discover all the email addresses associated with a specific domain name or the social media profiles associated with an email address.

Maltego creates a dynamic graph that visually depicts the connection between entities as a user executes transformations. Individual groups are represented by nodes in the graph, while edges denote their connections. To better comprehend the links and relationships, users are able to interact with the visualisation by scaling in/out, rearranging nodes, and filtering data. Maltego further provides a variety of statistical instruments that enable users to recognise data patterns, trends, and anomalies. These instruments, which include algorithms for clustering, link evaluation, and statistical measures, can be used to obtain insights and draw conclusions from collected data.

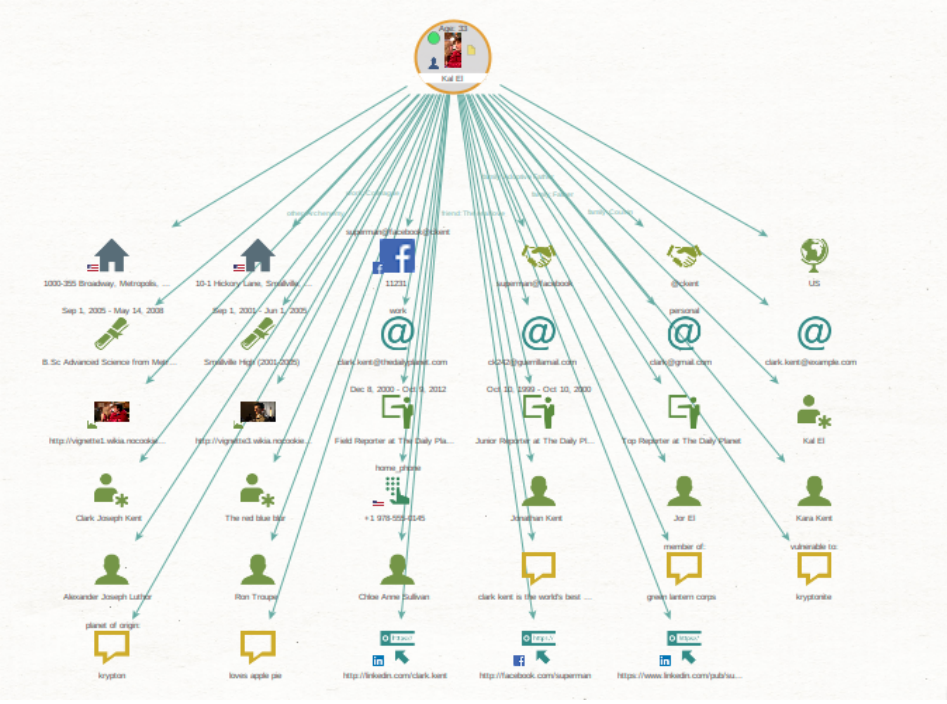

Figure 10: *An Example of a Detailed Graph in Maltego*[24]

[24]https://pipl.com/resources/blog/optimizing-person-of-interest-investigations

Figure 10 represents what it is visually like to denote an organisation or person of interest. In this graph, the investigation began with a single organisation entity that expands out into tracking OSINT (and other publicly available information) to create connections within the organisation. In this example, the user of Maltego obtains information about who in particular works at this organisation, what their title at their job may be, what their social media platforms are, what their emails are and other pertinent data required for an reconnaissance operation. Therefore it can be observed how useful Maltego can be to take the physical take of searching through data, by automating the process.

2.3.8 - Wireshark

Networking traffic is one of the most commonly used methods of active information gathering. Wireshark itself is an open-source network protocol analyzer that allows users to capture, analyse, and debug real-time network traffic. It is accessible on several different platforms. Network engineers frequently use Wireshark to monitor, diagnose, and comprehend network behaviour, as well as detect dangers and threats. Wireshark is valuable for monitoring traffic, but it also has additional applications. It may be employed to conduct a man-in-the-middle attack[25] by performing an Address Resolution Protocol (ARP) poisoning. ARP is a protocol used for routing traffic that converts IP addresses to MAC addresses. The attacker uses Wireshark to deceive the target machine into believing that the traffic originates from their machine. Thus, every connection destined for the target is redirected via Wireshark, allowing the attacker to capture activity meant for the original target.

[25] Saputra, D. and Riadi, I. (2019) 'Network Forensics Analysis of Man in the Middle Attack Using Live Forensics Method', *International Journal of Cyber-Security and Digital Forensics*, 8, pp. 66–73. Available at: https://doi.org/10.17781/P002558.

2.3.9- SQLmap

Just like many of the tools featured, SQLmap is another open-source tool for vulnerability analysis that automates the detection and exploitation of SQL injection vulnerabilities in web-based applications. An SQL injection attack is a common security flaw that takes place when an attacker inserts malevolent SQL code into a user input box or URL parameter, enabling themselves to alter the database that underlies the application and access/delete private information.

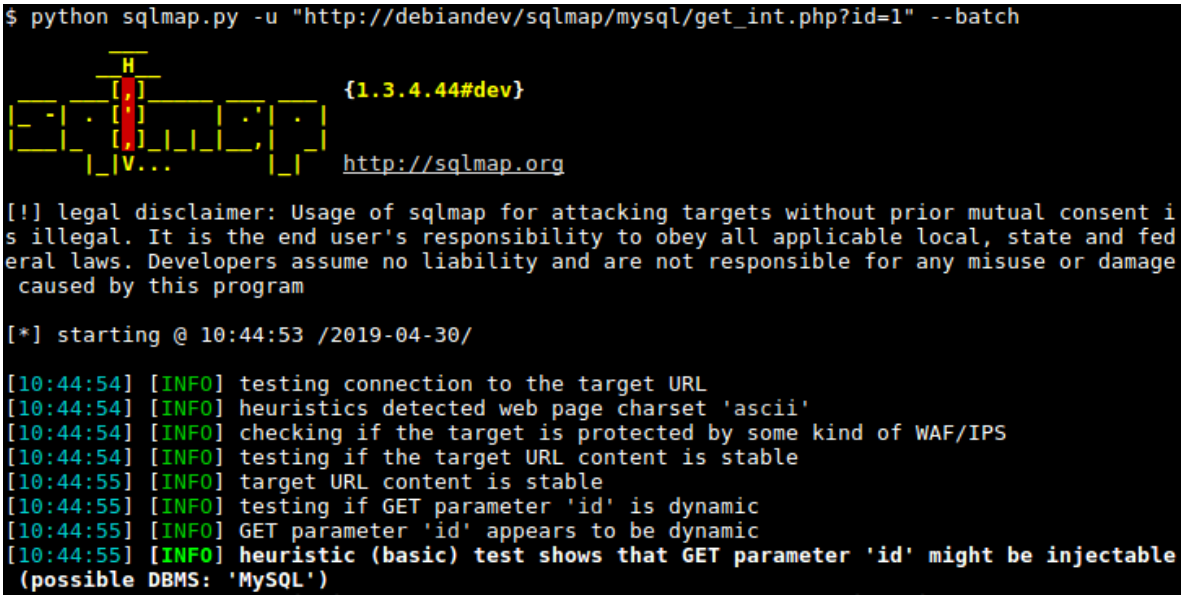

***Figure 11**: SQLmap on Command Line*[26]

By analysing the target application's responses to crafted payloads, SQLMap can automatically identify several varieties of SQL injection vulnerabilities, including error-based, blind, and duration-based. Once an SQL injection flaw has been identified, SQLMap can exploit it to retrieve data from the database's structure, including table and column identifiers. Furthermore, it can create fresh records, alter existing ones, and delete records. SQLMap can also enumerate numerous database characteristics, including users, privileges, and roles allowing for a better comprehension of the targeted database's structure and level of security.

[26] https://sqlmap.org

2.3.10 - Social Engineering

Deception is the belief or act to deliberately convince someone of a falsehood, typically for personal gain, strategic advantage, or to manipulate the individual's perception or behaviour[27] . Keeping this in mind, it is typically the belief that computer systems and devices will always be the most vulnerable systems to compromise, how the field of social engineering techniques provides a new insight into how information gathering techniques have altered. Social Engineering is a psychological deception technique used by attackers to deceive and manipulate individuals into disclosing confidential data, granting them unauthorised entry, or taking actions which compromise the integrity of an organisation or system. Instead of focusing on technology, social engineering exploits the psychology of humans and our inherent capacity to have confidence in others or react to certain situations. In the words of the United States Department of Justice, attacks using social engineering are among the world's most hazardous threats. The US was the nation that was targeted by the most social engineering attacks and had the greatest cost per attack in 2016[28], with the estimated cost pegged at approximately $121 billion. Based on research conducted by the Federal Bureau of Investigation (FBI), the US faced the largest portion of social engineering attacks in the form of fraud & email scams where the perpetrator attempts to pretend to be a co-worker or CEO, which in turn lead to losses of up to $2 Billion were expected alone[29]. Examples of the most common engineering attacks are as seen:

- **Quid Pro Quo -** The offender provides a service or benefit in return for sensitive information or entry to a target system. They may pose as IT support solving a false virus on someone's machine & offering to resolve this in exchange for extremely large sums of money or login credentials.
- **Baiting -** A victim is typically enticed to interact with malicious software-infected material objects (e.g., USB drives) or sites by offering free applications.

[27] https://www.dictionary.com/browse/deception#:~:text=Deception%20is%20the%20act%20or,or%20more%20complicated%20cover-ups.
[28] Arana, M., 2017. How much does a cyberattack cost companies. *Open Data Security*, pp.1-4.
[29] Costantino, G., La Marra, A., Martinelli, F. and Matteucci, I., 2018, June. CANDY: A social engineering attack to leak information from infotainment system. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)* (pp. 1-5). IEEE.

- **Phishing -** Considered the most prevalent type of psychological manipulation. In an attempt to fool receivers into divulging sensitive information such as credit card numbers, passwords, attackers send emails or messages posing as legitimate correspondence from renowned organisations, institutions, or providers of services.

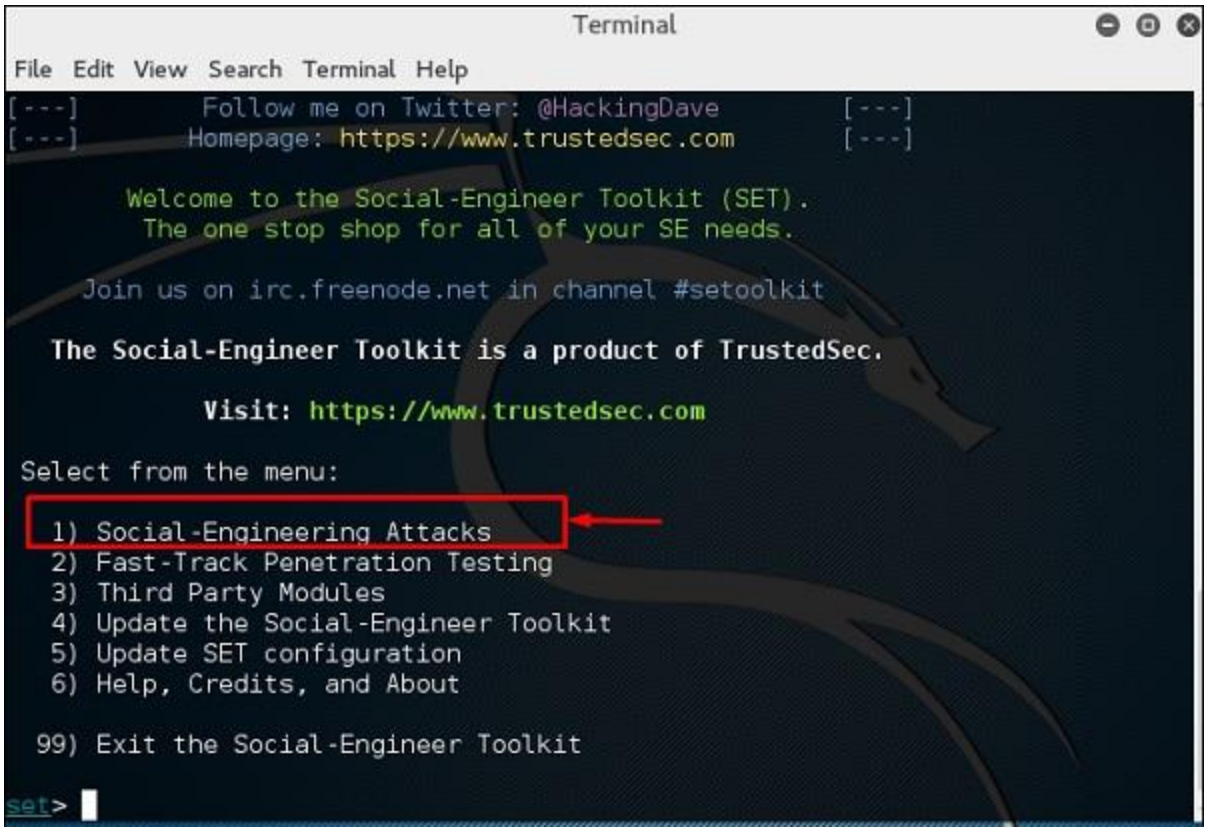

Figure 12: *The Social Engineering Toolkit (SET) on Kali Linux*[30]

The Social Engineering Toolkit (SET), is a Python-based open-source framework intended to help facilitate and automate a myriad of social engineering attacks. Developed by TrustedSec, SET offers experts a potent instrument for designing and executing social engineering initiatives in order to assess a business's susceptibility to this type of attack.

[30] https://www.tutorialspoint.com/kali_linux/kali_linux_social_engineering.htm

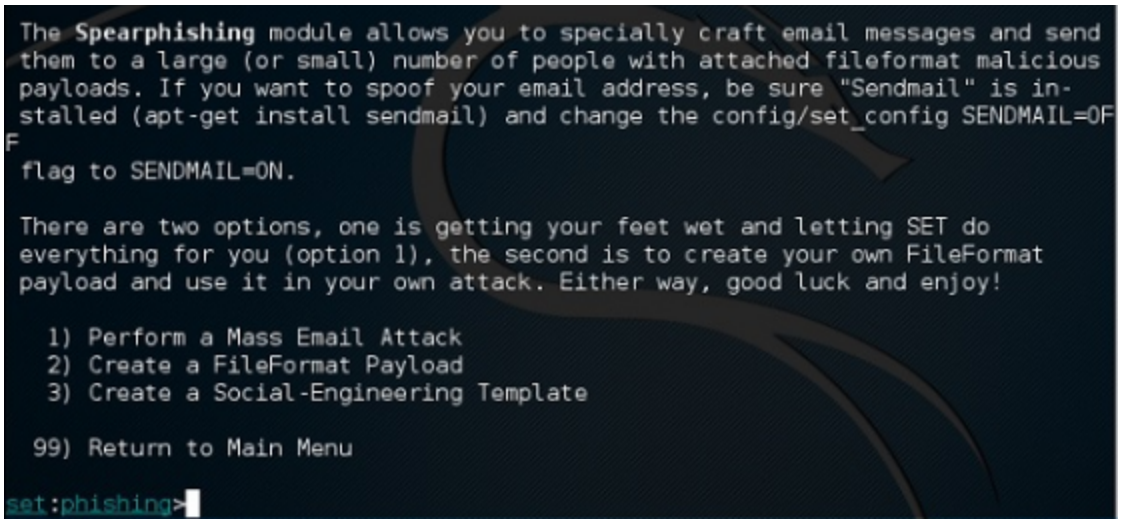

Figure 13: *An Example of what SET can perform*

SET is capable of performing a large variety of different attacks, with the sole purpose of manipulating someone or an organisation for a particular reason. These attacks could be:

- **SpearPhishing (Figure 12) -** Is the development and distribution of focused phishing messages, which are frequently intended to fool the recipient into selecting a link that is malicious, launching a malicious program/link, or divulging private data. This can also take the form of fraudulent emails sent in bulk to multiple targets.
- **Malicious Payload -** Permits the construction of malicious USB drives and other removable media that, when inserted, can compromise a target's computer or network. An example of this is configuring custom payloads, reverse shells or keyloggers and imbedding them in usb drives or outright purchasing premade variants like the Bash Bunny or Rubberducky[31].
- **Spoofing -** SET has the functionality to send forged/spoofed messages that look to seem to come from an established source. This is done in order to deceive receivers into disclosing sensitive information or deceiving them into conducting a particular action.

[31] https://www.theverge.com/23308394/usb-rubber-ducky-review-hack5-defcon-duckyscript

Chapter 3

3 - Requirement Analysis

In this section, there will be a description of the research project plan for the thesis, including the objectives, methodology, and anticipated results. The project plan acts as a road map for the research process, ensuring that each phase is well-structured and centred on investigating the expected results obtained from the information gathering tools (At the end of Chapter 2). Furthermore, there will be an intent to investigate the complexities of active & passive reconnaissance and uncover valuable insights to further our collective understanding of security itself. Whilst, there cannot be a large select amount of tools covered due to the time constraints, by focusing on the most highly regarded, it will still aim to provide beneficial insight that could serve to educate those curious. The particular objectives and benchmarks hoped to accomplish during the study, as well as the approaches and assets that will be employed to achieve these goals. By delineating a comprehensive plan for the project, there should be sufficient groundwork for a rigorous, well-organised, and successful study effort that contributes significantly to the existing body of knowledge. As a minimum, I look to complete the following:

1. Set up a running version of Kali Linux on VMware. Then create another virtual instance of windows 10/11 (Depending on cost and availability of the OS).
    a. Use the windows box to host a DVWA and log in to my personal website for testing
2. Test 5 different methods of capturing information gathering techniques and record the data/information obtained from using them
    a. Maltego (With Shodan & theHarvester), DMitry, Recon-NG, Social Engineering

3. Develop a conclusion of all the tools used, and rank them upon factors:
    a. Ease of use
    b. Amount of data that could be harvested the most efficiently (examination of time complexity against space complexity).
    c. Open vs Closed Source

In terms of an extension, I would hope to look to be able to complete the following as well:

1. Collect data produced by Wireshark & SQLMap
2. Demonstrate the effects of a Social Engineering strike (within in a contained environment)
    a. Harder to obtain numeral evidence due to the nature of these attacks being inherently psychological
3. Document the effects of a firewall mitigating a malicious attack
    a. Demonstrate the creation of a firewall/detection software

Chapter 4

4 - Implementation

Within this portion, appropriate documentation is provided of the comprehensive overview of how the research was executed, commencing in the initial installation of an isolated workstation within VMware, to establish a secure and reliable research environment. Establishing an adequate workstation was essential to guaranteeing that investigation and experimentation with various tools could be conducted smoothly and safely, without any forms of external interference (including but not limited to harming other outside computer networks/systems/devices etc.). Within this portion, appropriate documentation is provided of the comprehensive overview of how the research was executed, commencing in the initial installation of an isolated workstation within VMware, to establish a secure and reliable research environment. Establishing an adequate workstation was essential to guaranteeing that investigation and experimentation with various tools could be conducted smoothly and safely. After the establishment of the workstations, An in depth look into information gathering tools took place. This exhaustive analysis intended to evaluate the capabilities, strengths, and limitations of each tool in relation to the initiative's objectives, as well as to the tools to each other.

4.1 - Installation

Figure 4 shows an installation of Kali Linux running on VMware (a virtualization software previously mentioned, used to recreate the environment of a user’s select choice in operating system (OS)).

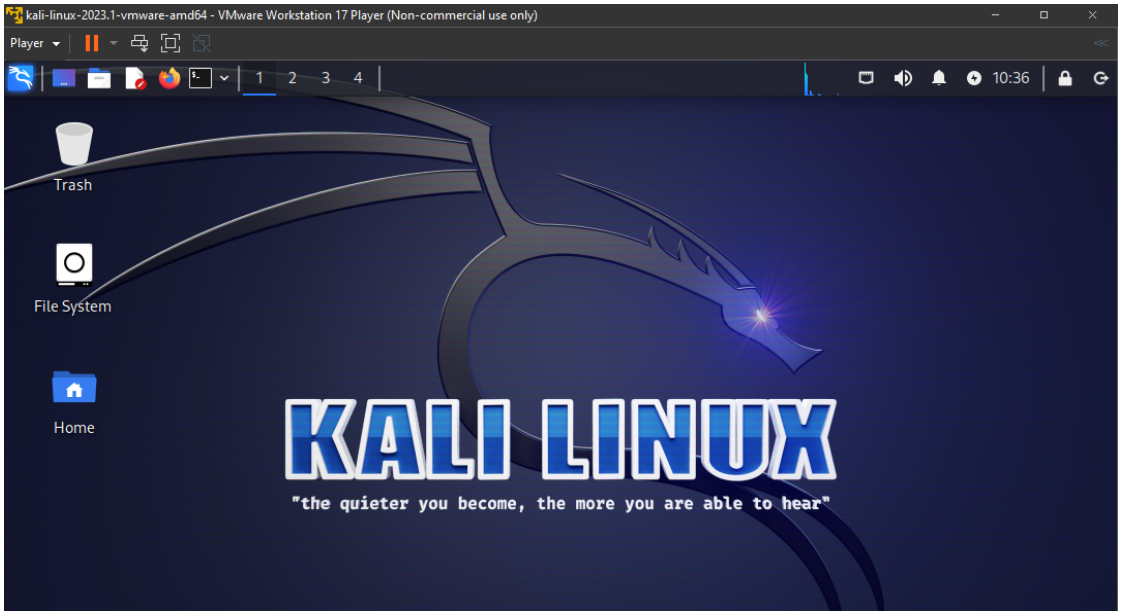

Figure 14: *Kali Linux Desktop*

Upon the successful installation of Kali Linux, users will be met with the infamous OS background. As previously mentioned, Kali comes with over 600 tools pre-installed, however, some of the tools covered throughout this research are not standard installation options. By using the commands:

sudo apt intall git

sudo apt install kali-linux-large

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install kali-linux-large
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware firmware-ath9k-htc firmware-atheros firmware-brcm80211
  firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek
  firmware-sof-signed firmware-ti-connectivity firmware-zd1211 kali-linux-firmware
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  0trace aapt afflib-tools amap android-framework-res android-libaapt
  android-libandroidfw android-libbacktrace android-libbase android-libcutils
  android-liblog android-libutils android-libziparchive apache-users apktool armitage
  arpwatch asleap avahi-daemon avahi-utils backdoor-factory bed beef-xss bluelog
  blueranger bluesnarfer braa btscanner bytecode-viewer cabextract caldera certipy-ad
```

Figure 15: *Mass Installing Tools on Kali*

Using both these commands will allow Kali to check for both dependencies and install the entire suite of tools available for usage, this is normally a highly recommended step for new users as a general “setting up” requirement. To finish this particular initialization for Kali Linux, Maltego has to be installed a touch differently. This is done by simply using the command:

sudo apt intall maltego

Which will in turn, prompt figure 16 and complete the basis of the installation needed for this research:

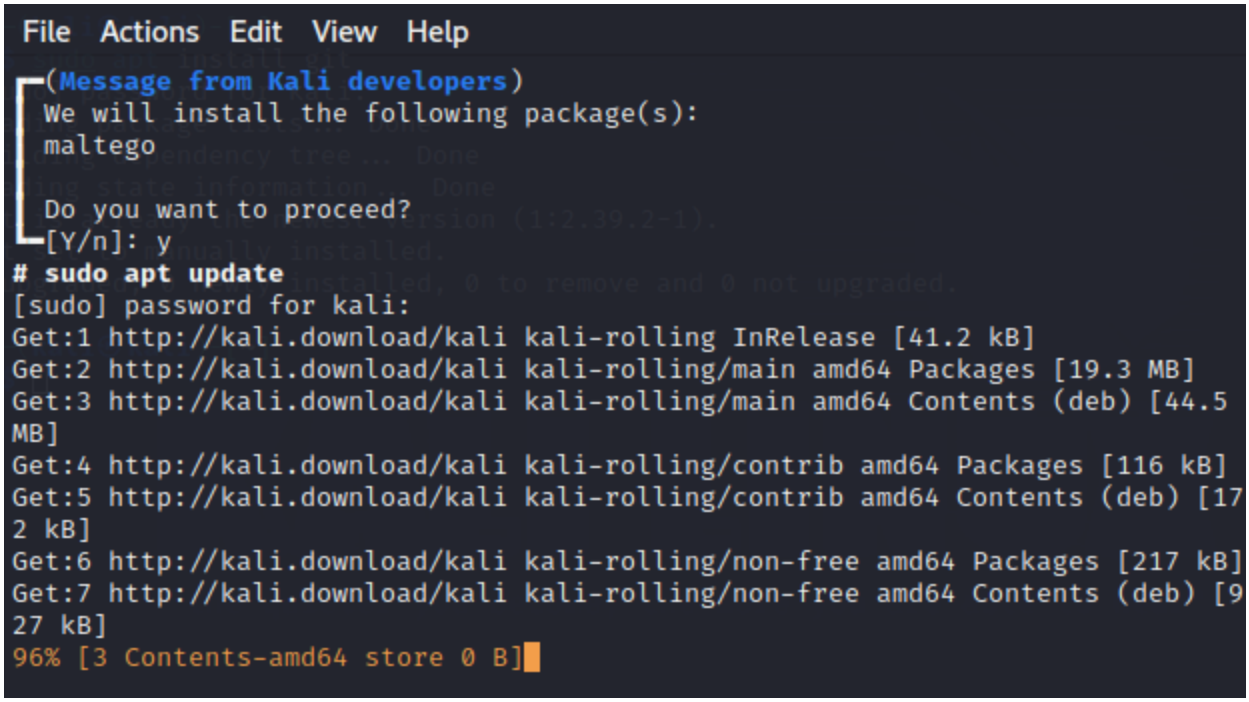

Figure 16: *Maltego installation*

4.2 - Maltego

Upon running and using maltego for the first time, it will present you with an example graph which can serve as a key for aiding the user into using and understanding the information that it retrieves.

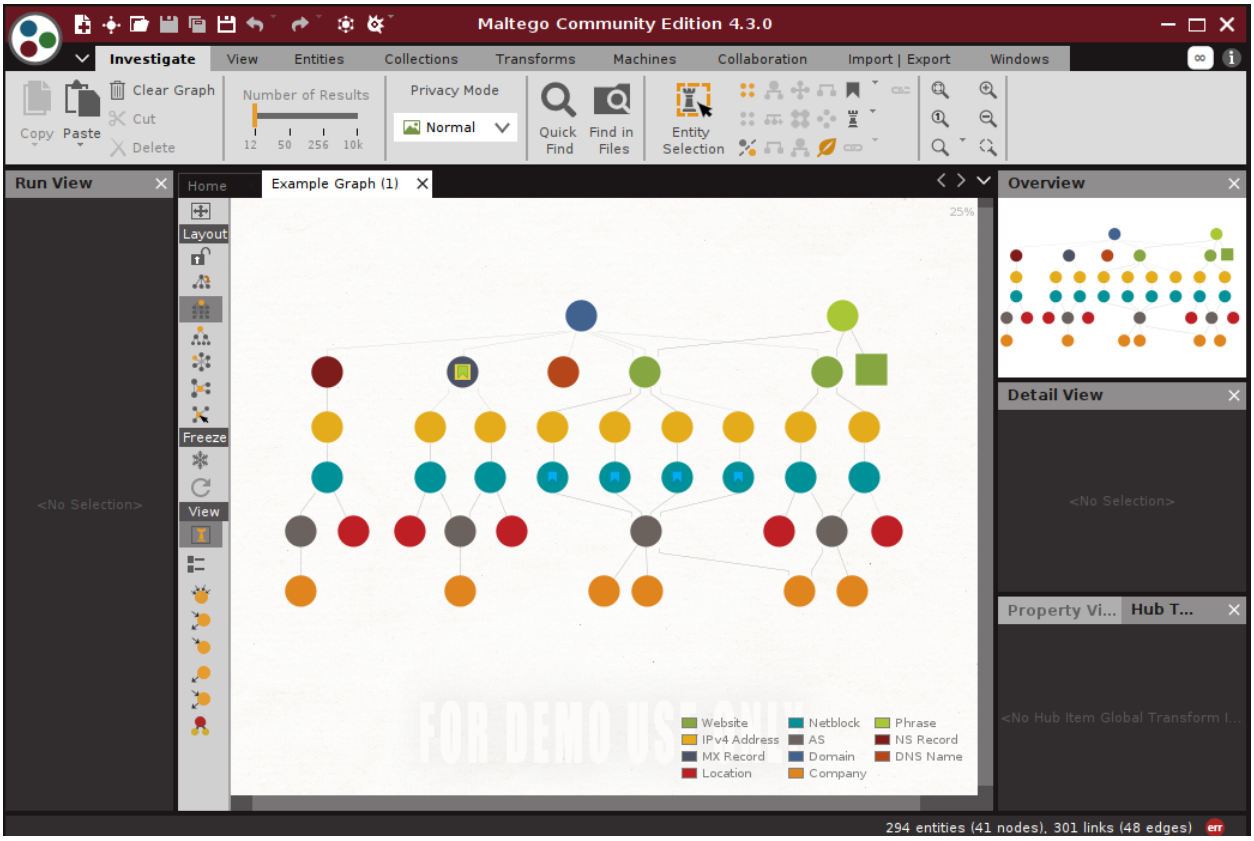

Figure 17: *Maltego example graph*

Figure 17 shows a blank/empty graph of connections, from an original domain, connecting potential locations, IPv4 Addresses of the company in question. It is worth denoting the structure of the diagram and how similar it will appear when creating a new tree on a potential target.

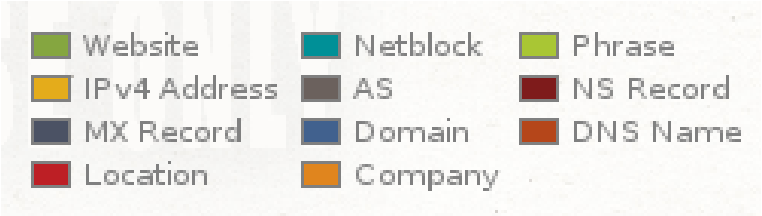

Figure 18: *Maltego Key*

Due to the nature of maltego and having blocks of information coded by key, it is worth memorising or referring to figure 18, to ensure that it is understood what each part of a graph correlates to in regards to a subject target. Now, the first stage of utilising Maltego is to construct a new graph. A graph can be stored locally and subsequently imported for modification. In Maltego, a fresh graph is generated by clicking on the new tree icon.

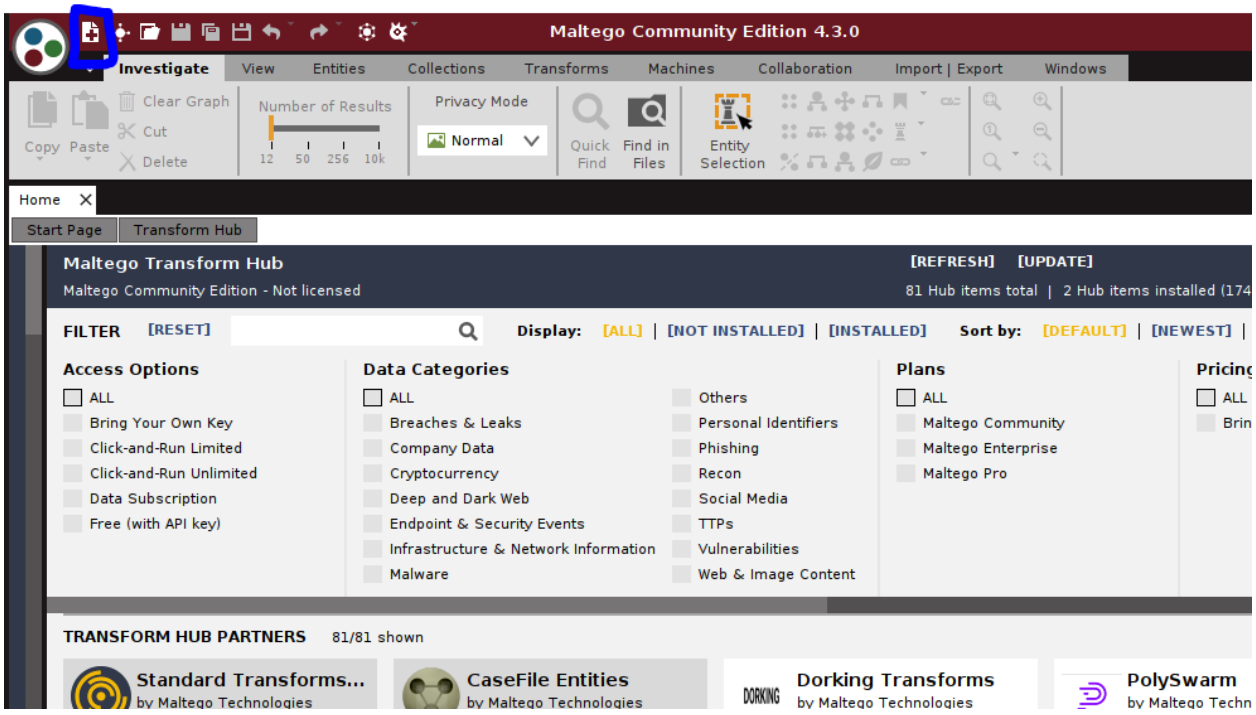

Figure 19: *New Maltego Graph*

When presented with a new blank page, the user is going to want to understand entirely what their target objective is at this point. For the purpose of this research project, we are first going to correlate the information surrounding a subject, which is going to be none other than Mark Zuckerberg. It should be noted that, the documentation of information gathering techniques on publicly available information of an influential person(s), is legal (do not use this for malicious intent, as it could result in serious jail time or prosecution).

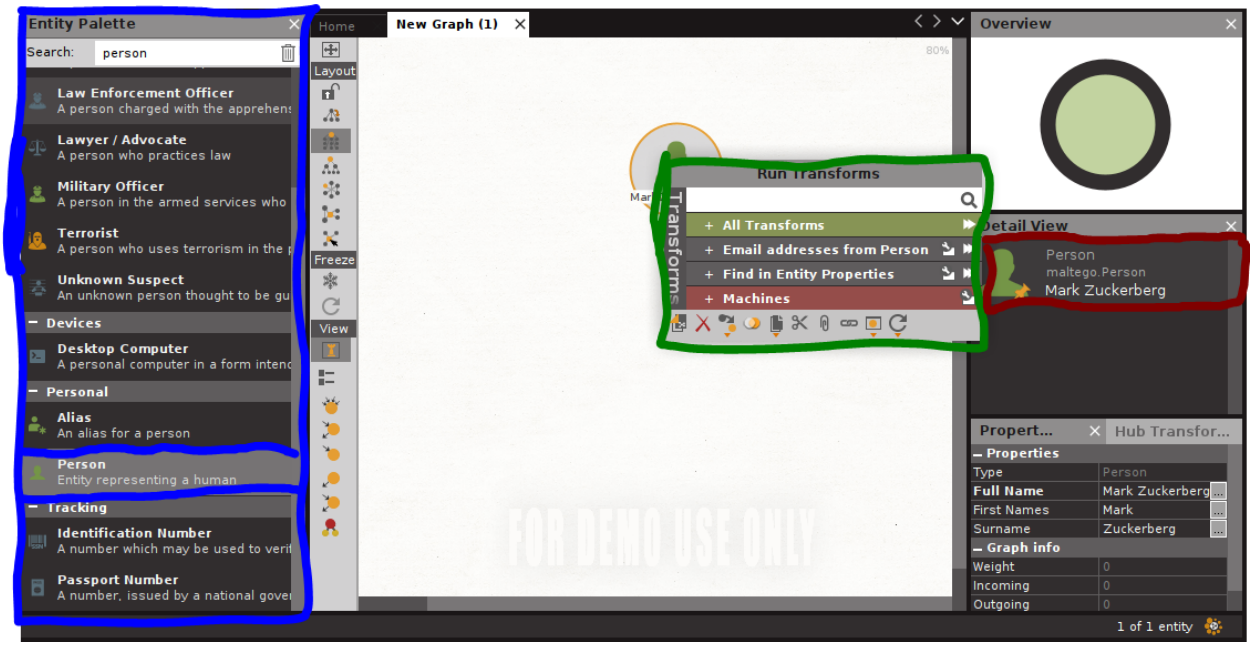

Figure 20: *Preparing a transformation*

To prepare a search in maltego, we are defining Mark Zuckerberg as an entity first. This tells maltego that the subject is a person, of which we will be using OSINT to complete a picture about who exactly this person is. The green box in figure 20 is the transformations section. Transformation in Maltego describes the act of accessing and modifying data inside the tool in order to determine relationships between various entities and reveal concealed connections. In most use case scenarios, the beginning of a graph will start by trying to retrieve as much information about a target at once, before being more specific about what the attacker needs to know about the target.

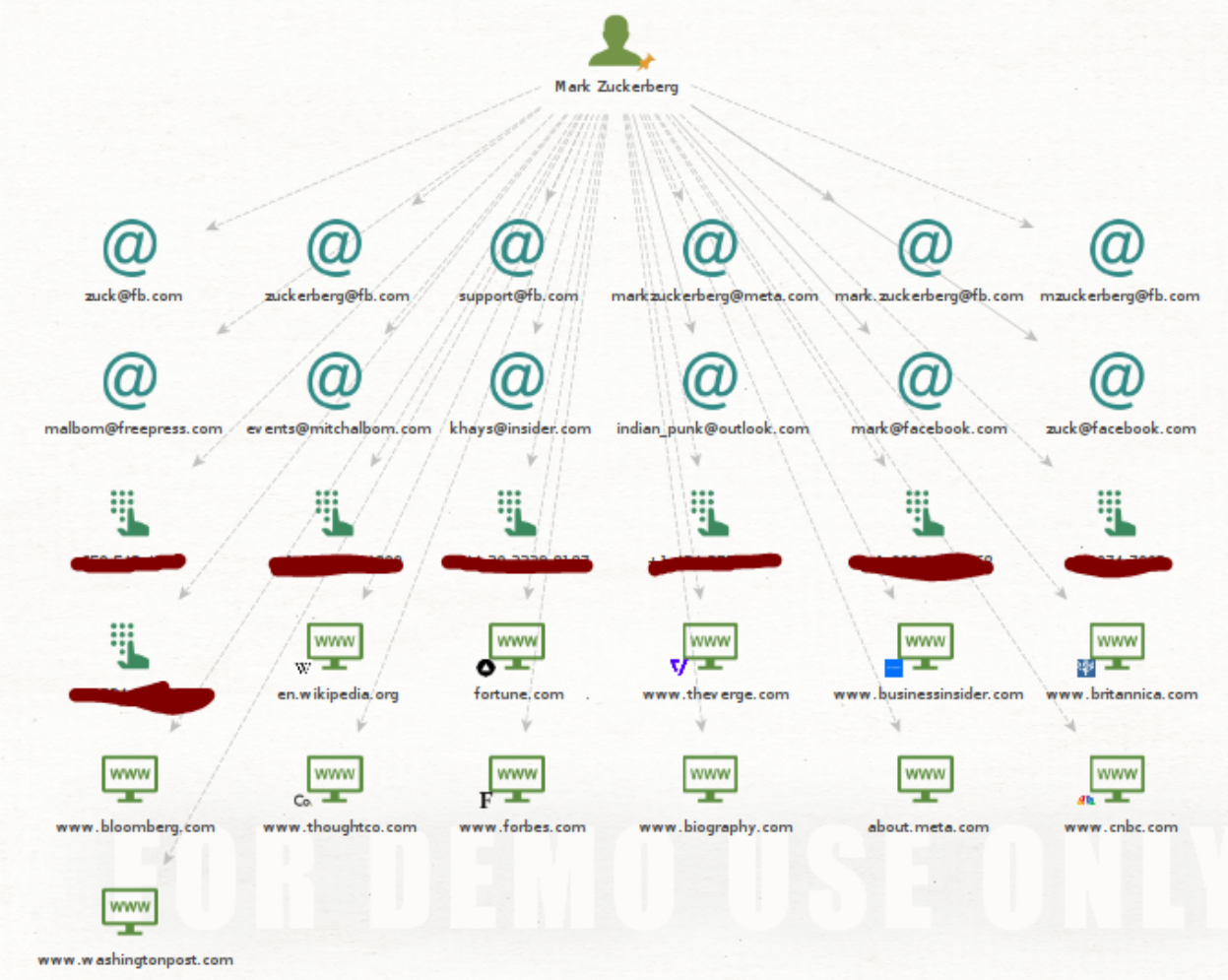

Figure 21: *Profile of Target in Maltego*

Within **4.18 seconds** of confirming the selection to pursue information on Zuckerberg, maltego returns with all forms of public information possible on the subject. This ranges from the various email addresses, phone numbers (blurred for safety/privacy) and social media accounts across several platforms. This allows us to also run individual transformations on any pieces of data that we obtained from a previous transformation. For example, running another transformation on the first email address, ***zuck@fb.com***, would result in a new set of data being attached to the graph; this can be seen in figure 22.

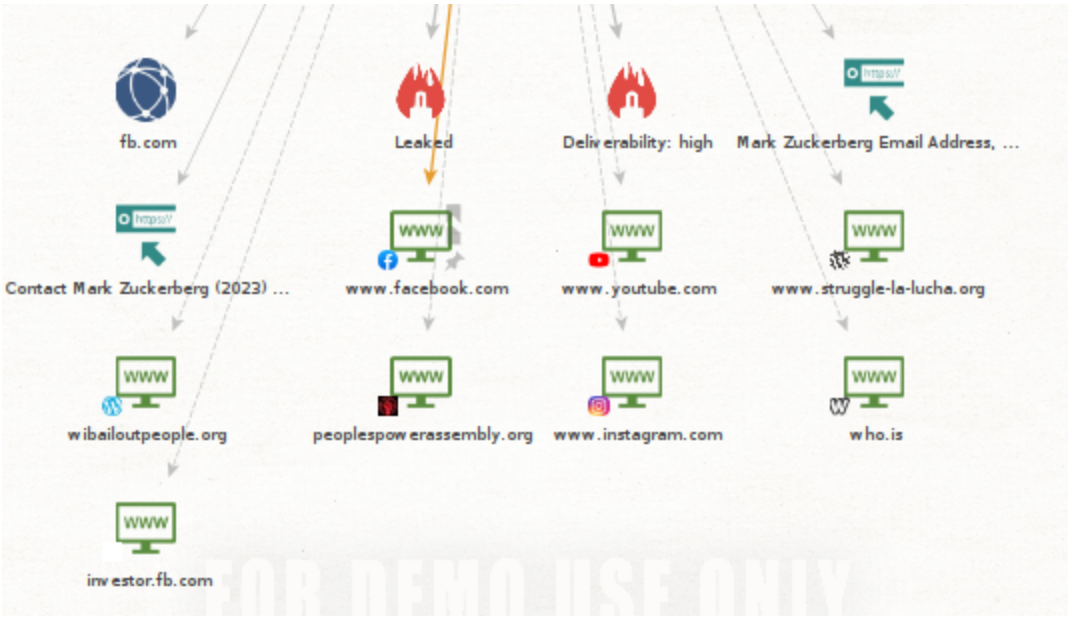

Figure 22: *New Transform*

With this in mind, we are now going to turn our attention to the target of this investigation, domain/website-based targets. By starting a new tree/graph, the domain of *lovejantiques.000webhostapp.com,* is going to be used. For contextual purposes, this site is a personal site of mine that I created for the purpose of coursework during my time at Sussex, which is now going to be the target of an OSINT operation to access how much information can be derived from it.

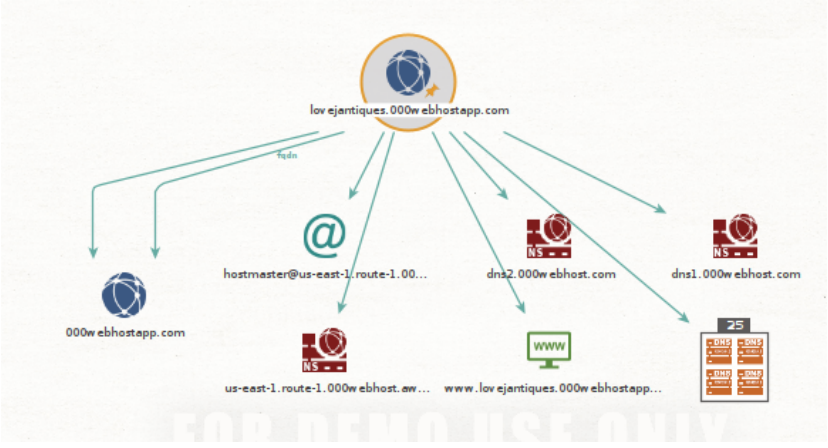

Figure 23: *Domain Search*

Figure 23 shows the result upon searching my personal domain site using maltego. The returned information varies from the type of hosting as well as the location of where the site is hosted, DNS name & Email addresses. Upon further inspection and running another transformation on the web address of the site (figure 24), maltego was able to resolve the actual hosting site I paid, hostinger, as well as the email addresses that were registered on the website. Maltego even went as far as retrieving IP addresses that accessed and used the site at some point (due to the lack of IP Encryption my site had to offer). The figure also displays that maltego retrieved a netblock. A netblock refers to the specifics of a range of IP addresses administered by a singular entity or organisation. To facilitate the effective distribution, management, and routing of internet resources, internet protocol (IP) addresses are separated into chunks. Therefore, to reiterate, all IP's that accessed the site, were accounted for, even those that had accessed the site on a virtual private network (VPN).

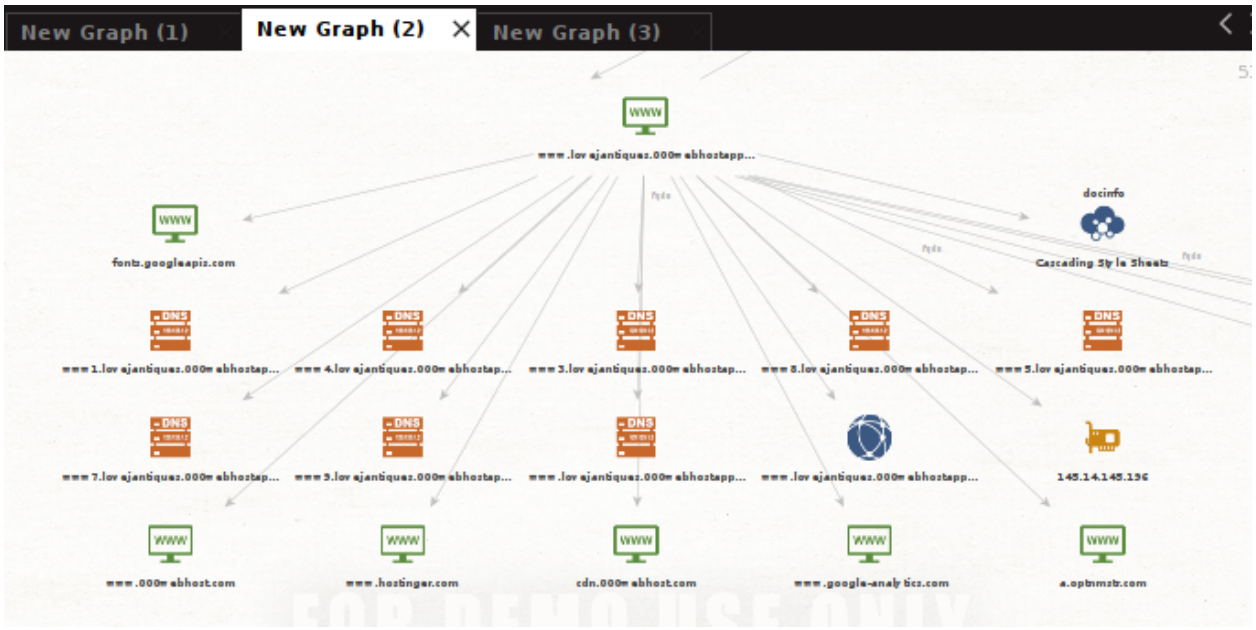

Figure 24: *Further Look into the Domain*

4.2.1 - Maltego with Shodan

The intelligence that Maltego is able to provide is without a doubt extremely reliable and accurate; however, to fully optimise the amount of information that can be garnered, the assailant would need to use it in conjunction with other tools, such as Shodan. As previously mentioned, Shodan is a search engine used to access information that shouldn't always be public (and functions as a 'darker version of Google'). Shodan is available as a plugin for Maltego due to its ability to search for insecure devices, and provide IP addresses, ports being used by exploitable device; however, it requires a full working copy of Maltego (and not the community edition being used for this research topic), though this does not prevent the separate use of both platforms and the sharing of information as needed. For the purpose of this experiment, the free variant of Shodan will be used, however the more expensive payment plans allow for more in depth analytical procedures (like vulnerability filters, batch IP lookups & extensive increase in search parameters).

The target for this assessment will be an undisclosed telecommunications company based in Spain. The main reason this target in particular was selected, was to remain unbiased and impartial when selecting a subject of interest, so by using Shodan’s Explore function, the first available target was selected. Not only this, but my personal site, despite being quite barren with content, has a few protective measures to prevent a lingering eye from breaching it, therefore it would serve more useful to demonstrate how outdated technology can prove to be a potential target’s achilles heel.

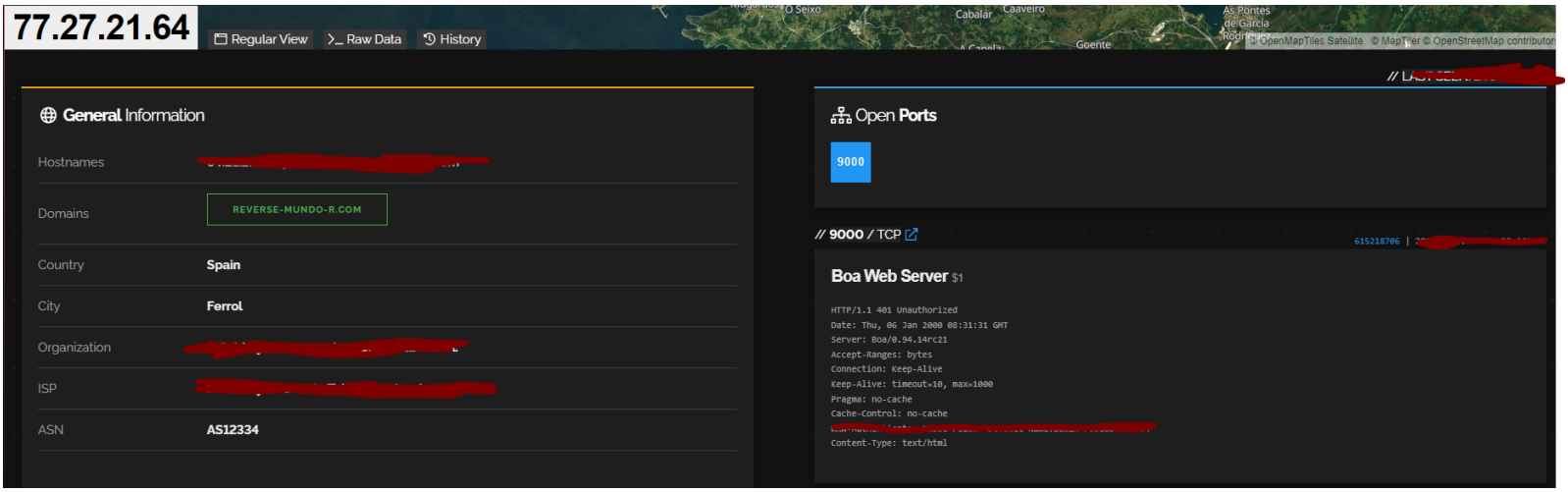

Figure 25: *Vulnerable telecom provider*

As seen with figure 25, our subject based in Ferrol (Spain), has a list of all public information regarding itself as an organisational entity, as well as the unsecure devices/platforms they could be using. In this case, it can be observed that Shodan identified an open port, which typically means a network port on a computer or another device that is always watching for new connections from clients or other systems. An open port indicates that a certain service or application is accessible and prepared to receive interactions from distant systems. Communication between different hardware or services throughout a network must be enabled through open ports. An open port could pose an invasion of privacy since it exposes the service or application to possible threats or intrusion. In this case, the open port of *:9000* can be attributed to a Boa Web Server. Boa is a now discontinued web server platform, meaning any platform running through it, could be extremely likely to be infiltrated.

By taking the IP address provided by Shodan (77.27.21.64), and plugging it into a new graph in Maltego, we can notice (figure 26), that the entire network for the telecomunications company appears. Confirming that a name and location appears to correlate with the public information on Shodan, it can be confirmed that the graph being run is accurate for the information being obtained on our target.

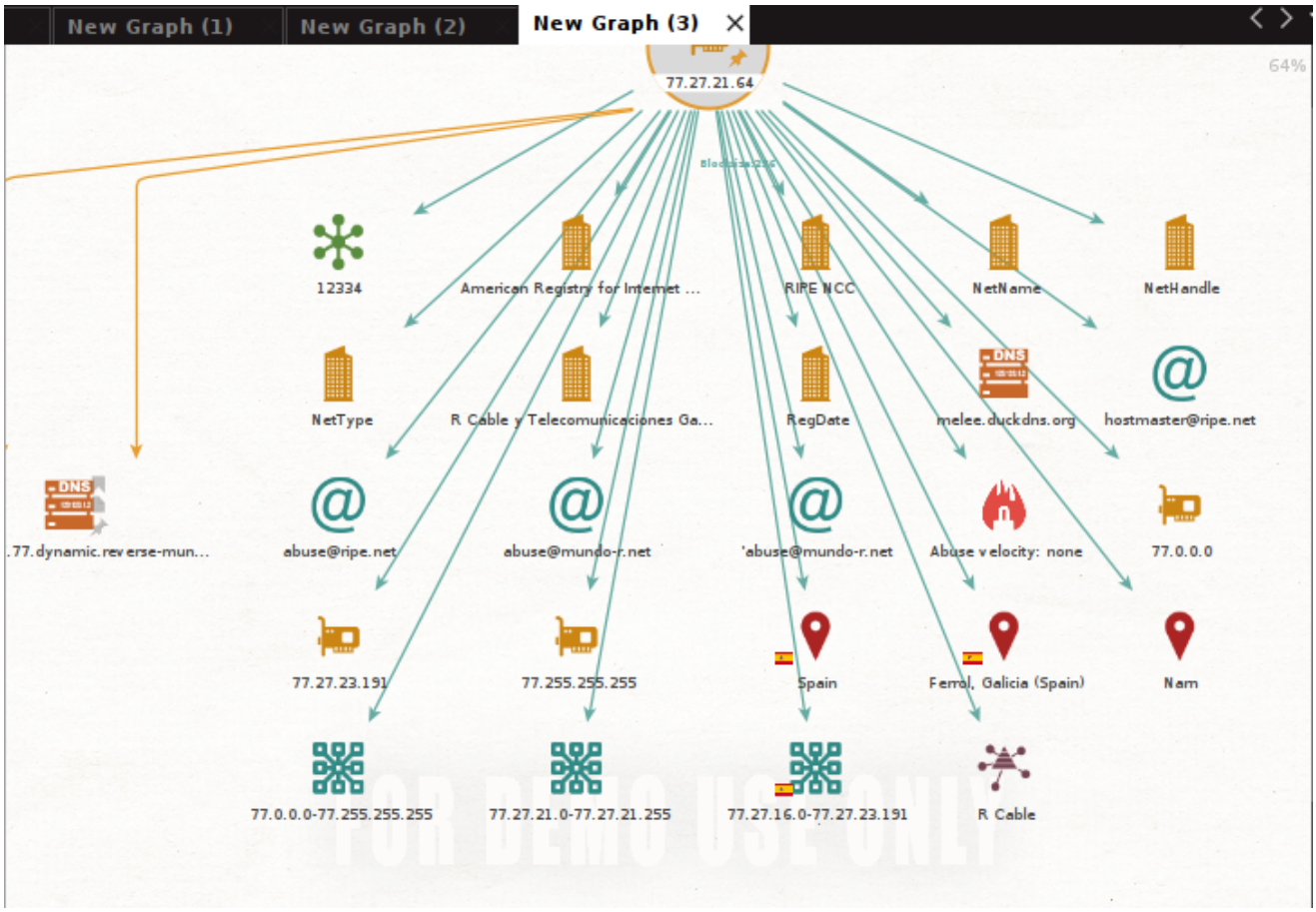

Figure 26: *Graphing the Telecommunication Company*

A utilising of a further transformation plugin known famously as “Have I Been Pwned?” in Maltego may be used to determine if a certain username or email address has been subject to a data breach. When a breach involving an email address or username is detected, the transform will provide details about it, including the name of the compromised site, the date of the breach, and the kind of data that was exposed. In this instance, Maltego has returned the possible name and password (hidden for privacy), for access to our open port web server.

Figure 27: *Opening the TCP Open Port*

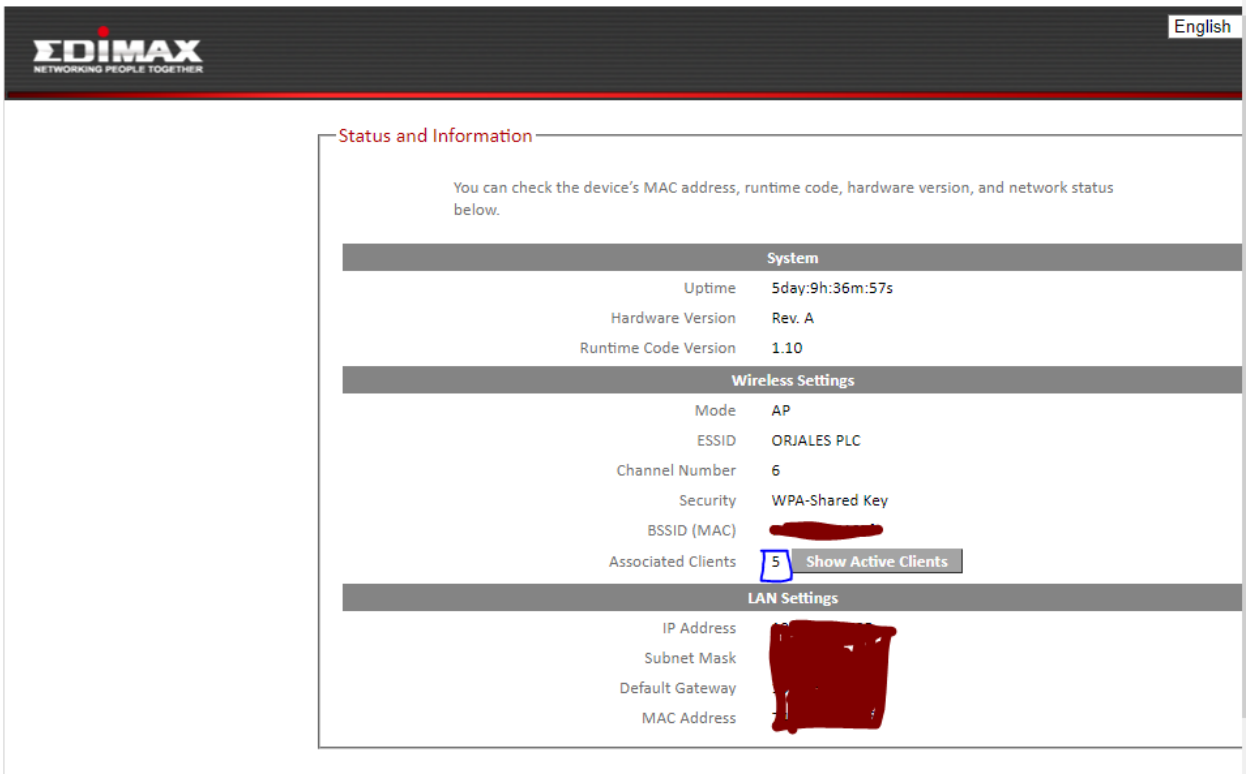

Figure 28: *Getting into to the Web Server*

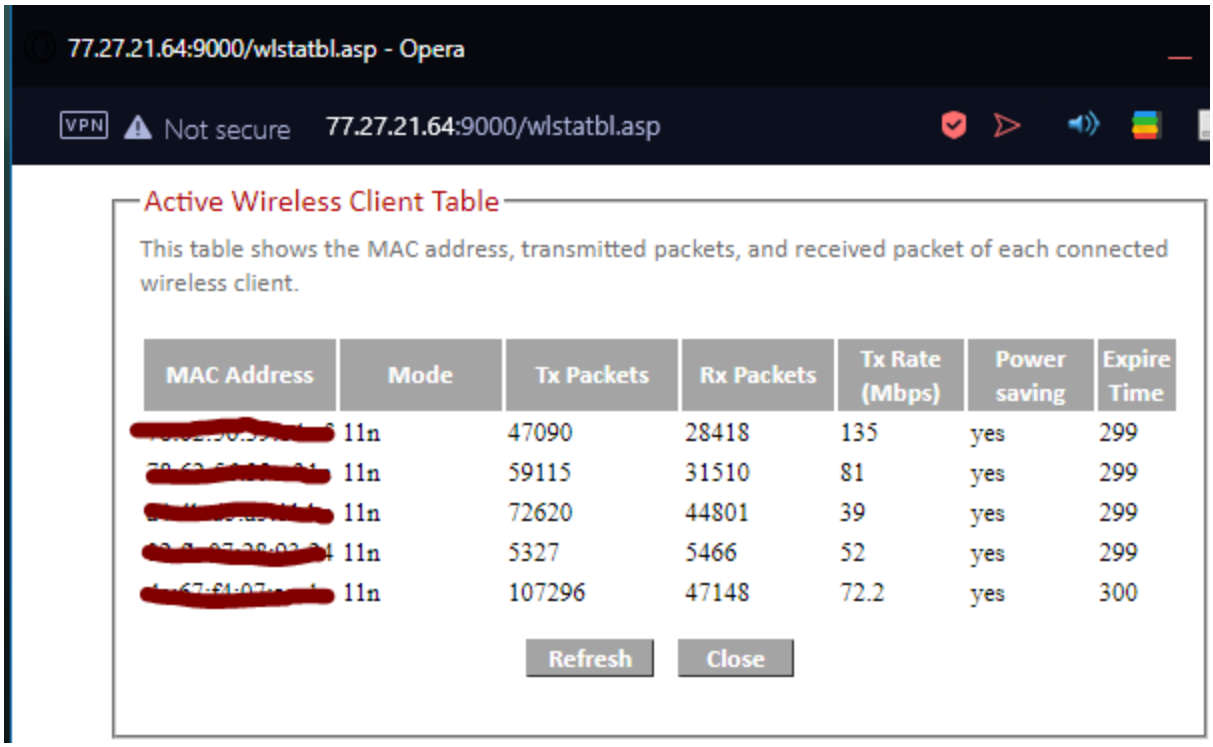

Figure 29: *Active Clients*

Figures 27 & 28 shows the steps taken to effectively gather information from the target. Using the account details provided by Maltego, they were inputted into sign in when attempting to access the open port. The details, proving to be correct/accurate, presented details about the web–server, IP address, Mac addresses, how the security works and more. It is also worth noting, I was told the number of people actively connected (blue box indicator) at the time of the intrusion. Figure 29 shows a closer look at these clients, indicating their personal MAC Addresses as well as their Rx (receiver) & Tx (transmitter) packets. A network interface controller (NIC) is given a specific identification called a MAC (Media Access Control) address, which is used as an address for the network in communication inside the network segment in question. It is a hardware address given to a network equipment, such as a computer, smartphone, or router, by the manufacturer. Therefore these MAC addresses could range from anything to be another computer, a smartphone or router. Although an attacker couldn't penetrate or compromise a system on a MAC address, the use of it allows them to know of multiple

potential entry points into a network by investigating the other devices. If an attacker were to discover their point of entry, they would likely need to take advantage of a flaw in the device's software or firmware or use social engineering methods to get the user to divulge private data.

4.2.2 - Maltego with theHarvester

TheHarvester is another OSINT application aimed at gathering information about target organisations, including email addresses, subdivisions, IP addresses, and employee identities, from various public sources such as web search engines, domain name system (DNS) records, and social media platforms. It allows a user in essence, to search across a variation of platforms like search engines, to retrieve information about a target which can be inputted straight into maltego.

For this portion, another domain/website will be used, in particular it will be *spacex.com*.

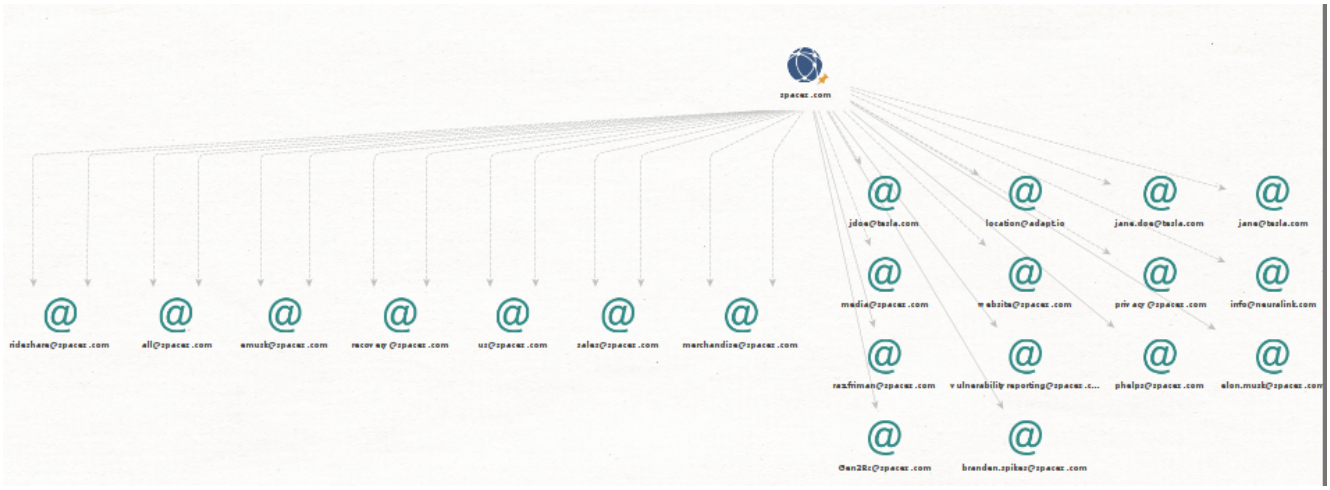

Figure 30: *Tree of the spacex domain*

Figure 30 shows a select number of emails picked up by maltego. The reason the number of emails could be seen as so few, is due to the fact that I ran a transformation that looked for email addresses in the PGP keyserver & could be found via a search engine. Users may store and

access public PGP keys in a central database known as a Pretty Good Privacy (PGP) keyserver. PGP is a popular encryption tool that enables users to encrypt their email messages, files, and other sorts of data using a public key in order to safeguard them. Once there is adequate starting ground for email addresses collected, the search could be expanded using theHarvester.

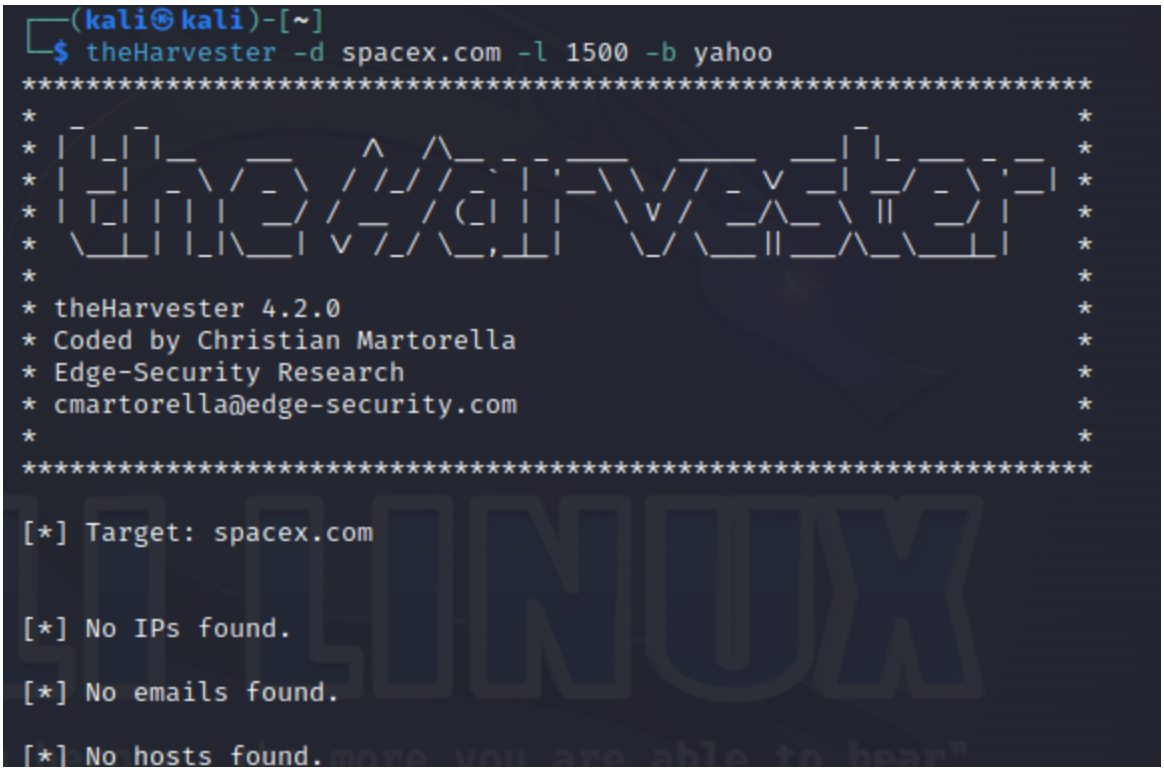

Figure 31: *Running a search query with theHarvester*

Figure 31 demonstrates how a typical search is queried in theHarvester. By using

theHarvester -d spacex.com -l 1500 -b yahoo

A query is set to ask theHarvester to search the domain (-d) *spacex.com* for any IPs, emails or hosts, with a limit(-L) of 1500 searches using the source(-b) yahoo. Since the search parameters are relatively miniscule by using only one domain, there is also an information gap

applied to ourselves on how much could be collected. To resolve this, a simple change to the domain source can done, shown with the following example:

*theHarvester -d spacex.com -l 1500 -b **all***

By changing the parameter from "*yahoo*" to "*all*", we allow theHarvester to exhaust all available resources in collecting information from OSINT sources. Whilst some methods of collection require an API Key, an overwhelming majority do not, resulting in the result shown in figure 32.

Figure 32: *The Returned Results from a larger Query*

The attacker is able to quite simply copy and paste the result, whether it is an IP Address found or email, they can be dragged into an instance of Maltego for transformations to take place.

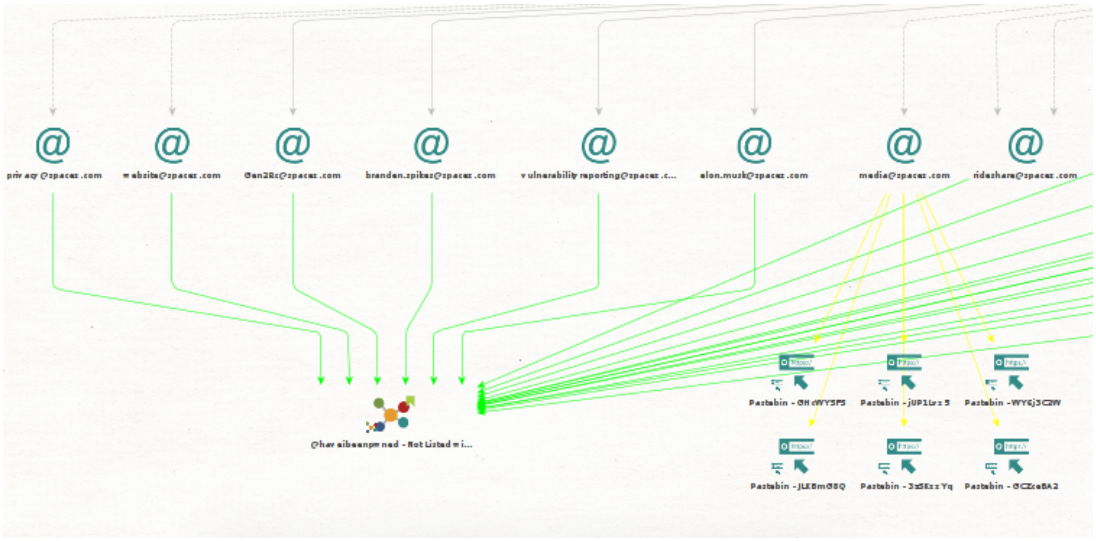

Figure 33: *Paste bin search*

Figure 33 shows the result of computing a transformation on the emails obtained from theHarvester, and checking to see if any of them have been used within a pastebin. In most cases during a data breach, information concerning account details of distinguished corporations, find themselves being leaked to pastebin or sources like wikileaks. In this case, 6 pastebin posts were found in connection to our harvested email addresses. While these pastebins are now thankfully taken down (protecting their breached email accounts once more), performing a further "Have I Been Pwned?" transformation can help to identify what were the causes of the breach.

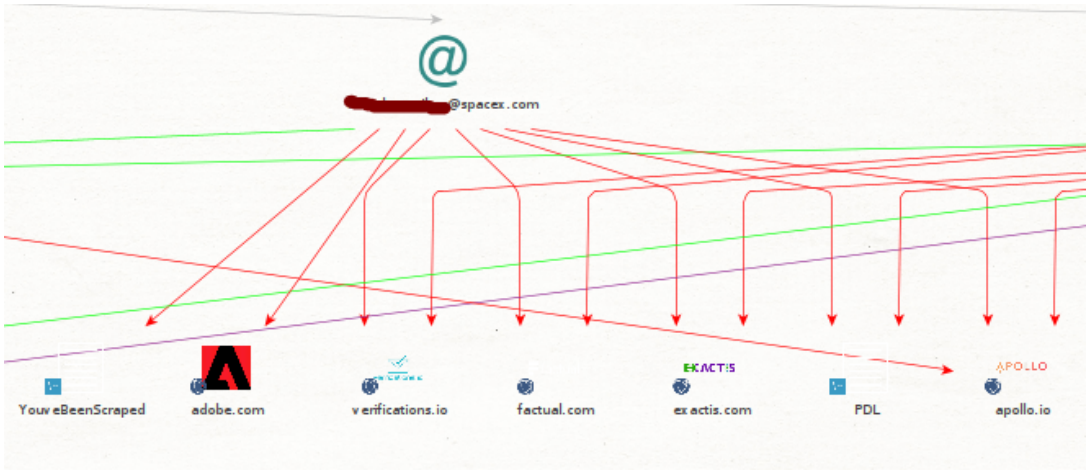

Figure 34: *Pwnd Transformation on spacex.com*

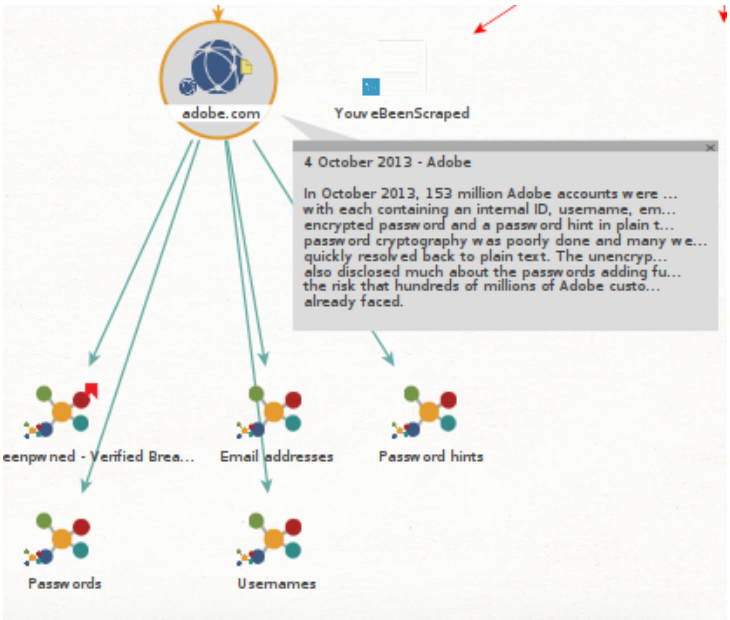

Figure 35: *Enriching the Transformation*

After taking only **6.213 seconds,** Maltego returned with all the forms of ways a particular email had been breached from Spacex (figure 34). By using a method of “enriching”, it allows us to get information on particular breaches of security. For example, figure 35 shows how the adobe leak of this particular Spacex email led to the breach of passwords, usernames & email addresses of this individual.

To summarise, Maltego offers an incredibly user-friendly platform that can be customised to the needs of the attacker in real-time even if it doesn't produce numerical data for examination. Maltego is an extremely flexible tool that enables the effective visualisation and analysis of links between numerous data sources, making it an important tool for security experts and investigators. The quantity of data that can be gathered at any given time is unlimited and solely dependent on the user's ultimate objectives, which must be understood. Maltego's usefulness depends on the strategic application of its features and transformations to collect and analyse the most pertinent data, eventually enabling users to make defensible decisions based on the connections and correlations discovered. Should the attacker, upgrade their version of Maltego, there are endless possibilities on what could be accomplished, as a consequence of its built in plugins/transformations platform.

4.3 - DMitry

To reiterate, DMitry is used to acquire information about a target, including sub-domains, email addresses, and open ports. It enables a number of data sources, including as search engines and other internet services, as well as direct system-to-system communications for services like WHOIS and network port scanning. DMitry may be used actively to do TCP scans, but by default it will be a passive information collecting tool and TCP scanning is a more specialised form of scanning that can be given to NMap (as it servers to only perform open port scans). It could be quite similarly linked to theHarvester, which shows the idea of a domain to scrub for emails, IP's, and hosts. For the purpose of testing DMitry, we are going to be using the previously used spanish telecoms as a method of not only cross-referencing the information retrieved to see which tool provided the most amount of information in a least amount of times and permutations/actions, but it servers to standardise the experimentation.

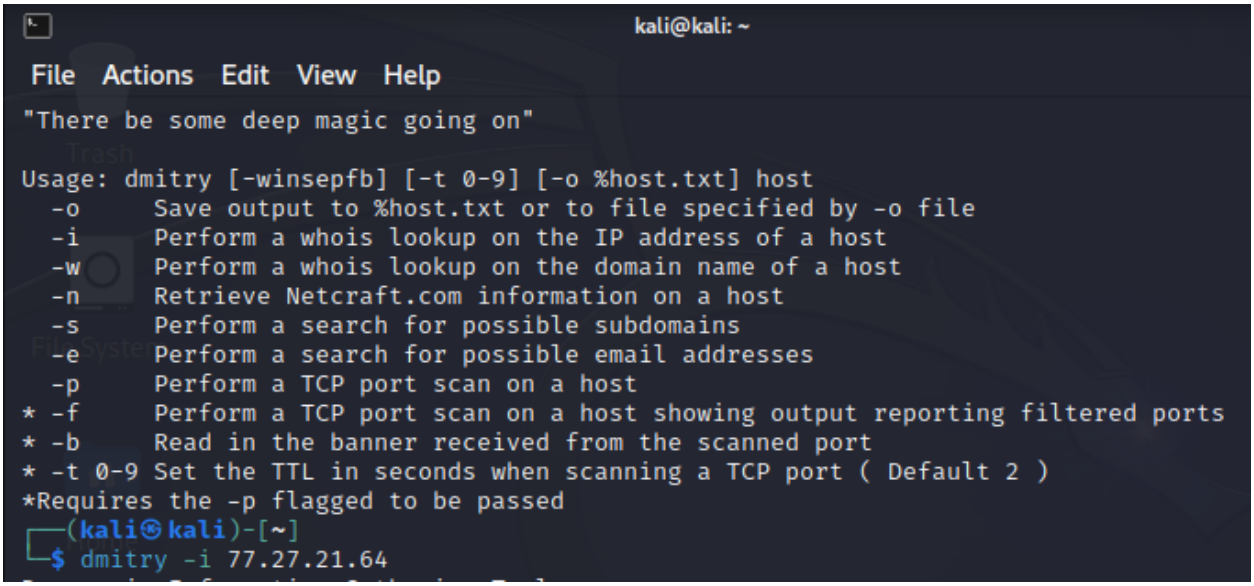

Figure 36: *DMitry Help Guide*

Figure 36 shows that a command is already inputted into the bottom line which is:

-i *77.27.21.64*

This simple line of code, asks to run a whois lookup on an IP address and attempts to tell us everything DMitry can using it, as can be seen by the resultant:

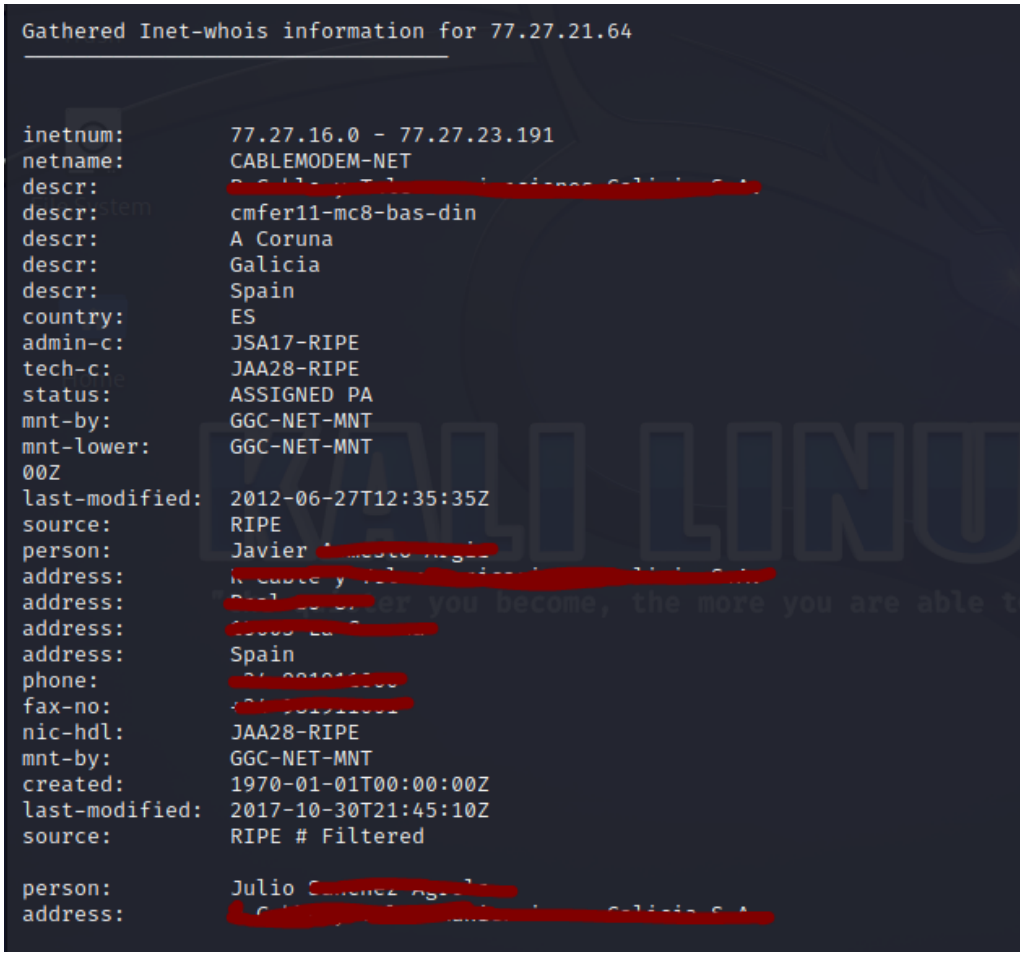

Figure 37: *whois on 77.27.21.64*

The amount of information obtained from a simple IP lookup can be extremely extensive. We retrieve the names of potential workers/employees of the organisation, their physical address, the range of IP's that accessed their web server and more. We also now know their web server was last maintained up until 2017, meaning it is extremely likely it is defunct or out of maintenance, posing a security risk for the company.

The other feature that could be utilised here, is the use of the email scrubbing service, similarly to theHarvester. Within **0.512 seconds**, DMitry returns with 2 emails that it found in connection to the site (figure 38). When acting upon this information in a setting like that of Maltego, it is able to use the email addresses found to create relationships between, the site, the organisation, and the breached email addresses. Therefore, providing the efficiency of these passive information gathering techniques

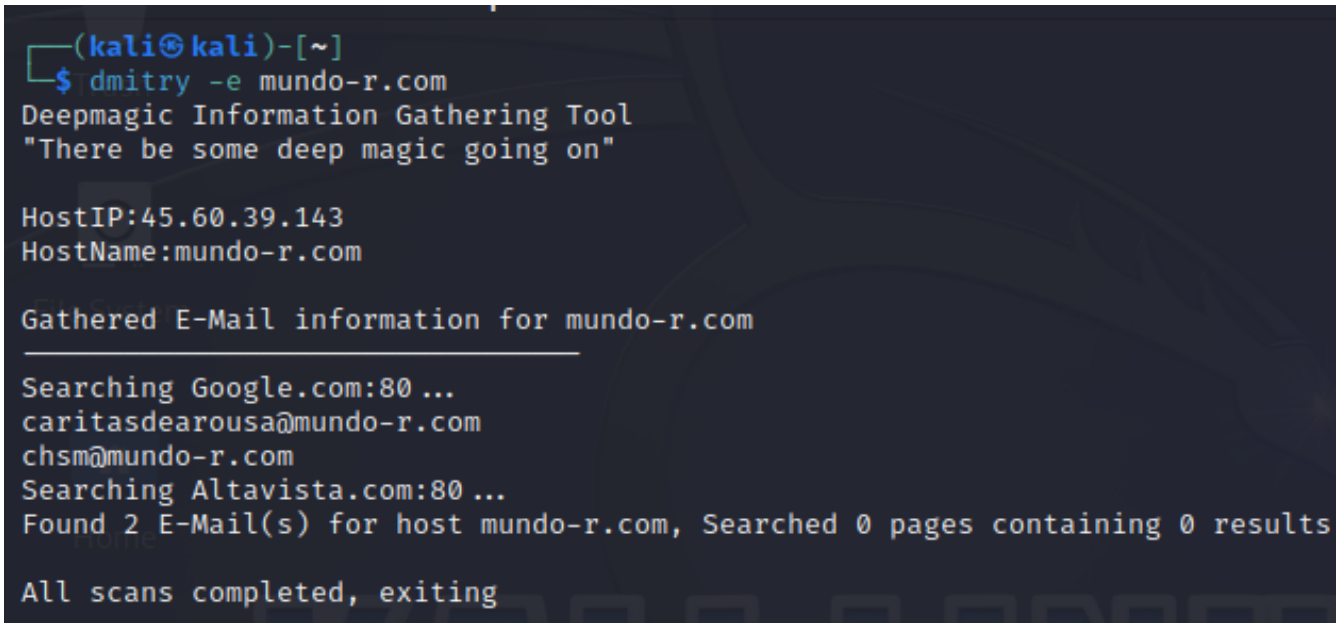

Figure 38: *email scrub on 77.27.21.64*

DMitry is an open-source command-line programme that is extremely user-friendly and simple to use. By reducing the procedure for gathering crucial data about a target, DMitry streamlines the reconnaissance portion of a security assessment. DMitry provides a simple interface that enables users to rapidly comprehend a target's digital footprint by supporting a variety of data sources. DMitry is an essential tool in the toolbox of security experts due to its simplicity and efficiency.

4.4 - Recon-NG

Recon-ng is a potent open-source framework for reconnaissance that was created to help with information collecting, as was already indicated. Recon-ng streamlines the process of gathering information on targets from numerous open sources, including search engines, social networks, and online databases, thanks to its modular architecture and user-friendly interface. For the purpose of testing Recon-NG to the fullest (with free modules/plugins), we are going to be implementing the SpaceX example (allow for a comparison with Maltego to access results), however in the case that there are limitations when showing SpaceX, the BBC will be used instead.

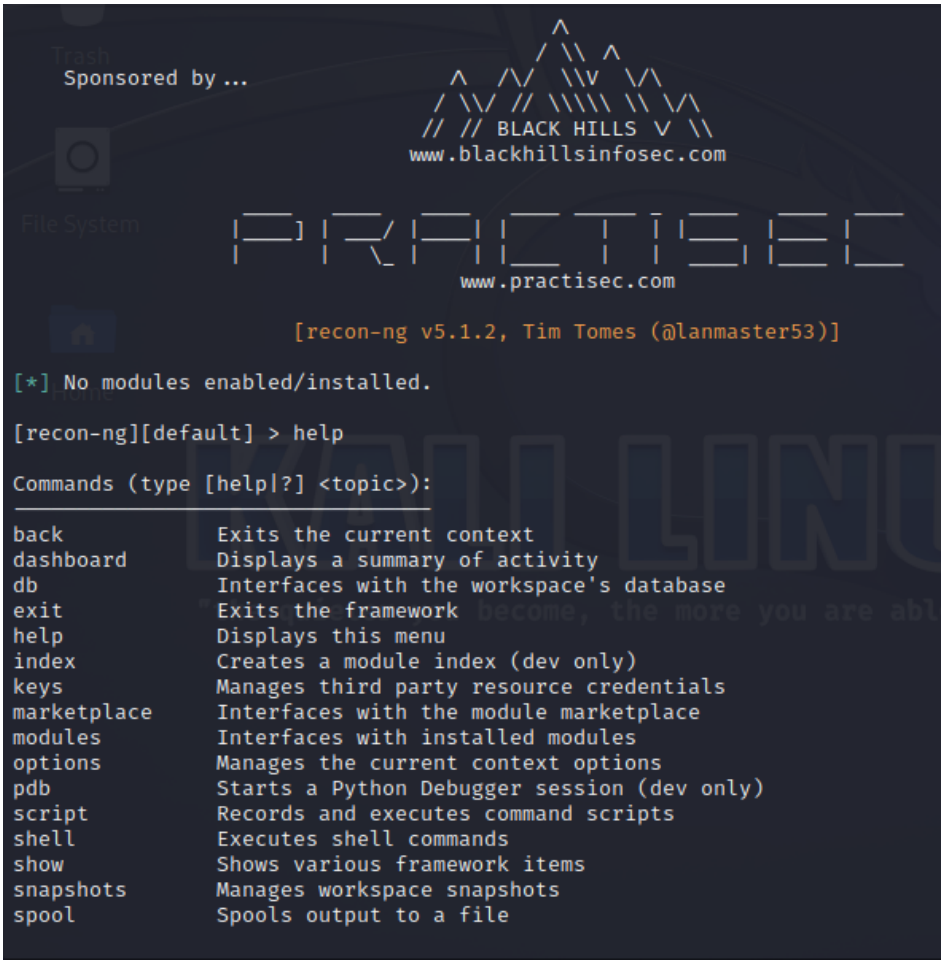

Figure 39: *Running Recon-NG -help*

Due to the complexity of Recon-ng, getting accustomed to the functions that it can perform is usually the first step to intel gathering. By default, the User will have to create a new workstation to define their topic of interest.

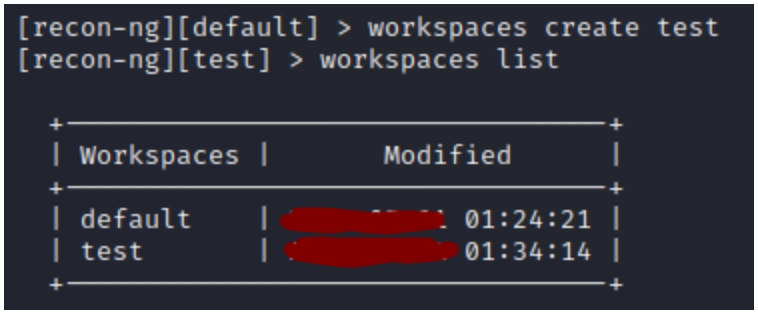

Figure 40: *Recon-NG Workstations*

After defining and creating a new workstation, a large percentage of the actions that Recon-NG can perform has to be installed. Using the marketplace functionality, it allows the attacker to decide which methods they would like to choose when beginning their reconnaissance attempts.

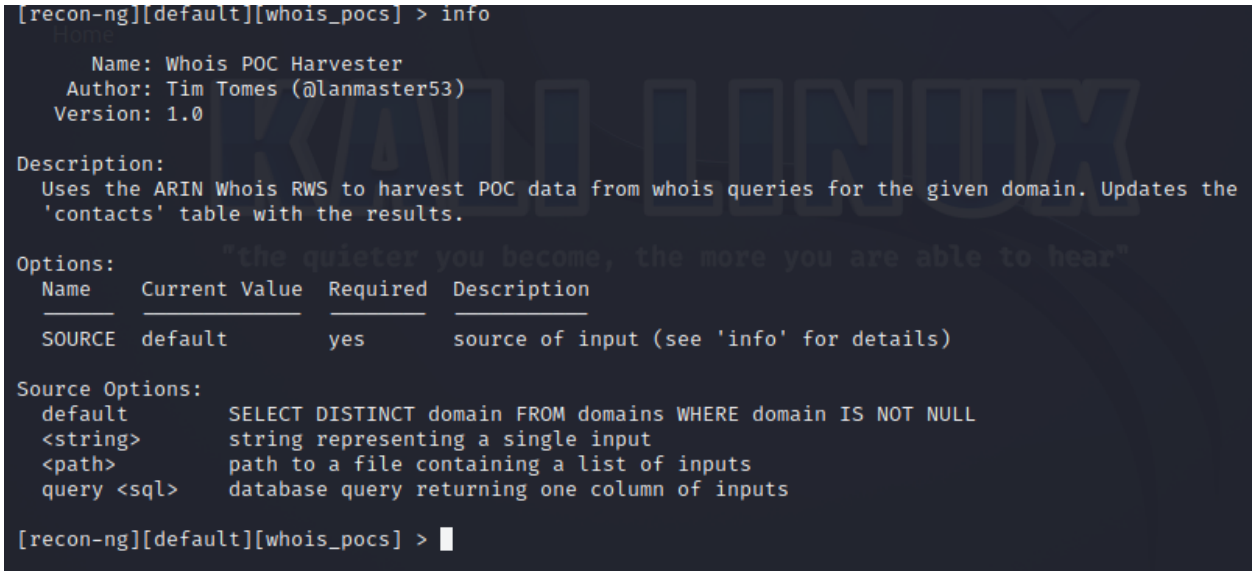

Figure 41: *Recon-NG whois*

After defining and creating a new workstation, a large percentage of the actions that Recon-NG can perform has to be installed. Using the marketplace functionality, it allows the attacker to decide which methods they would like to choose when beginning their reconnaissance attempts. Figure 41 shows that the module being used is a Whois lookup, with the results listed in Figure 42, showing that emails were obtained from the lookup request (12 email addresses). Cross-referencing this with the maltego results showed 2 identical addresses, meaning that the information gathered was accurate, the remaining addresses could simply be new domains not previously identified.

Recon-NG will automatically save all this information in tables within it's database, allowing users to generate reports as well as statistics on the number of hosts, IP addresses, Emails and more, that had been identified in recon attempts.

Figure 42: *Recon-NG whois running*

Recon-NG also has the ability to brute force certain aspects of information gathering, this can be applied to attempting to find and resolve what domains potentially belong to our target.

```
Description:
  Brute forces host names using DNS. Updates the 'hosts' table with the results.

Options:
  Name      Current Value                          Required  Description
  --------  -------------                          --------  -----------
  SOURCE    spacex.com                             yes       source of input (see 'info'
  WORDLIST  /home/kali/.recon-ng/data/hostnames.txt  yes       path to hostname wordlist
```

Figure 43: *Recon-NG using Brutal_Hosts*

As seen above, brutal hosts is a module that allows for the usage of a wordlist and source domain, to attempt to resolve any other domains that could be connected to the target.

```
[*] ------------------------------------------------------
[*] operations.spacex.com ⇒ (A) 20.140.147.200
[*] Country: None
[*] Host: operations.spacex.com
[*] Ip_Address: 20.140.147.200
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] ------------------------------------------------------
[*] radius.spacex.com ⇒ No record found.
[*] ras.spacex.com ⇒ No record found.
[*] raptor.spacex.com ⇒ No record found.
[*] r01.spacex.com ⇒ No record found.
[*] rapidsite.spacex.com ⇒ No record found.
[*] rc.spacex.com ⇒ No record found.
[*] rd.spacex.com ⇒ No record found.
[*] rcs.spacex.com ⇒ No record found.
[*] read.spacex.com ⇒ No record found.
[*] realserver.spacex.com ⇒ No record found.
[*] re.spacex.com ⇒ No record found.
[*] recruiting.spacex.com ⇒ No record found.
[*] ref.spacex.com ⇒ No record found.
[*] red.spacex.com ⇒ No record found.
[*] reference.spacex.com ⇒ No record found.
[*] reg.spacex.com ⇒ No record found.
[*] register.spacex.com ⇒ No record found.
[*] registro.spacex.com ⇒ No record found.
[*] regs.spacex.com ⇒ No record found.
[*] remote.spacex.com ⇒ No record found.
[*] relay.spacex.com ⇒ No record found.
[*] rem.spacex.com ⇒ No record found.
[*] remstats.spacex.com ⇒ No record found.
[*] reports.spacex.com ⇒ No record found.
```

Figure 44: *Recon-NG using Brutal_Hosts*

Figure 44, shows how during the **19.23 second** search, an overwhelming majority of domains are not found during the brute force, however the word “operations”, was able to be identified. Upon further expectation of the returned IP address, it places it in San Antonio, Texas, a known location for where Space X is based, meaning the information returned is valid on our target. Any domains that are actually valid will be automatically stored in a DB that allows for review post-gathering stage.

4.5 - NMap

As previously mentioned, the use of NMap as a footprinting tool, is almost regarded as a must have. It serves to provide invaluable data about anything that is possibly conceivable about a target system. For the purpose of this research, NMap will be on free resources as network scanning a device without correct permission can lead to legal consequences. Furthermore, the usage of Nmap on a server could result in the unintentional creation of DOS attack, whereby the excess packets sent to scan a network, actually crash it. Despite how rare this instance is, it is not a risk worth taking in the pursuit of exploring the educational uses. NMap in general, could be perceived as also as being more invasive/active with the target. Not only this, but NMap is understood to be relatively “noisy” compared to other alternatives, this means that NMap can be detected by firewalls or web servers, however, there are methods of using it to prevent detection. The first thing to do is to ensure that the ip address you want to investigate is online and can be reached. This is done by:

ping 77.27.21.64 **or** *ping mundo-r.com*

The command "ping" is a useful resource for network investigating, connection testing, and latency measurement. It should be noted that certain hosts could block ICMP packets, which could result in ping attempts failing although the device is actually running. In our particular

case, a free networking scanning site called *scanme.nmap.org* has returned packets, meaning the device is transponding correctly (figure 45).

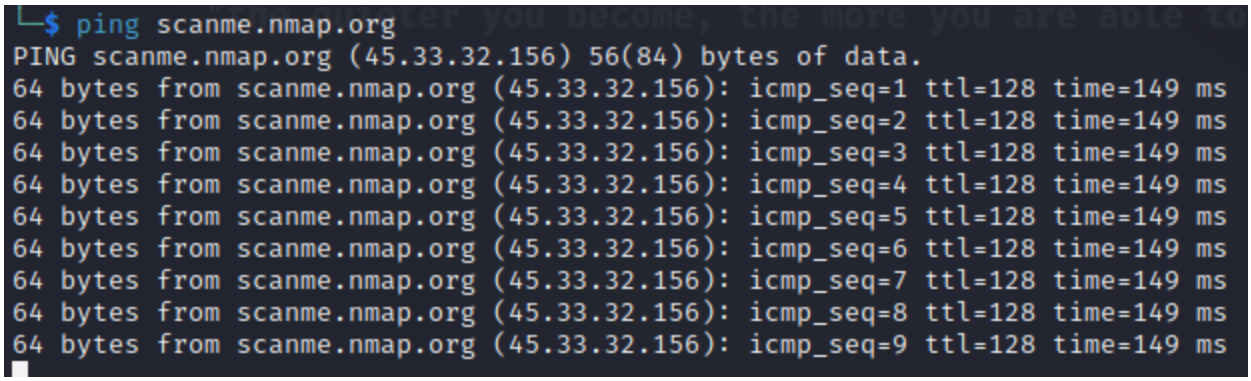

Figure 45: *Pinging a Host*

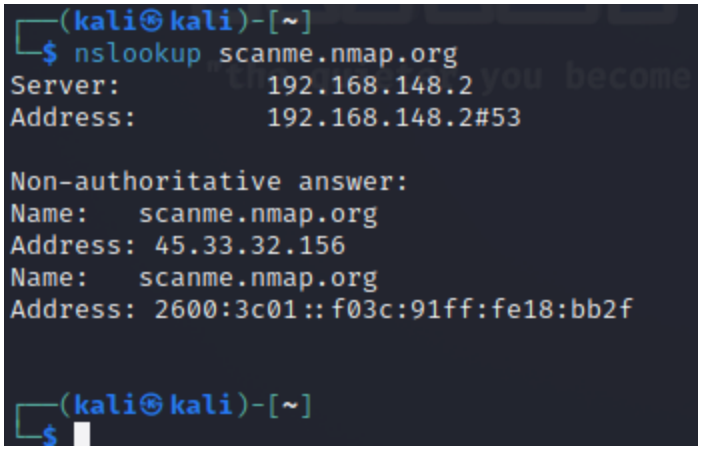

Figure 46: *nslookup*

The nslookup tool has allowed us to see the name of a domain, it's IP address, as well as its server address.

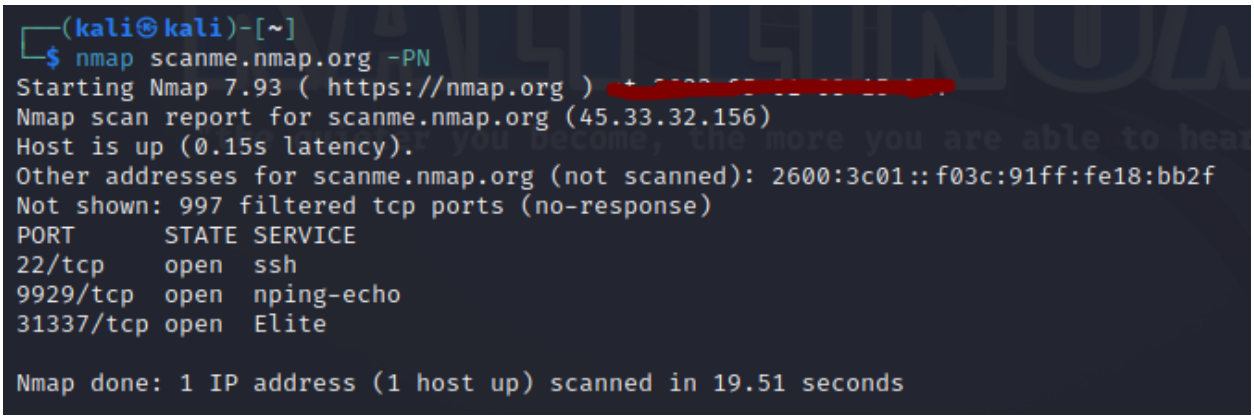

Figure 47: *nmap scan*

Figure 47 describes the view of when a full nmap scan is completed on the target. Using the simple code written in the terminal, Nmap returns with 3 open ports in **19.51 seconds**.

4.5.1- Aggressive Scanning

Scanning ports can become a lot more aggressive by using the *-A,* which according to documentation, allows for knowledge on factors like the OS being used.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A scanme.nmap.org
Starting Nmap 7.93 ( https://nmap.org ) a[redacted]
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.15s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac00a01a82ffcc5599dc672b34976b75 (DSA)
|   2048 203d2d44622ab05a9db5b30514c2a6b2 (RSA)
|   256 9602bb5e57541c4e452f564c4a24b257 (ECDSA)
|_  256 33fa910fe0e17b1f6d05a2b0f1544156 (ED25519)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-favicon: Nmap Project
|_http-title: Go ahead and ScanMe!
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.55 seconds
```

Figure 48: *OS detection*

In the **29.55 seconds** it takes, based on the returned results, it shows that the domain running *scanme.nmap.org*, is currently built on a ubuntu linux platform. Furthermore it shows the open 22/tcp SSH server as well. Running this command on the Spanish telecommunications network *77.27.21.64,* shows that there is an open :9000 TCP port also running on Ubuntu. This is information that Shodan & Maltego were unable to provide, truly demonstrating that NMap is able to syphon a lot more information about a target than meets the eye.

Chapter 5

5 - Conclusion

5.1 - Results

Due to the fact that there were two different methodologies present during the integration of the tools, the retrieved results will be split into two different manners. Firstly, there will be an observation into port-based scanning tools that were used and they will be accessed based on their ability to:

- Detect OS types
- Time Taken to complete the Scan
- Number of Ports they scanned
    - Number of ports they found were open
- Ease of Use

This should allow for an identification into their strengths and weaknesses, as well as determine the most efficient scanner available. The following are the results collected based on usage of IP addresses of:

- *77.27.21.64*
- *83.56.11.7*
- *185.237.180.175*
- *195.55.120.106*

The reason for this is, testing proved that the original IP for the telecommunications company showed a singular open port. Therefore by expanding the implementation to multiple IP addresses, it can show the range of ability for these port scanners.

	Detected OS?	Time Taken (s)	Open Ports	Total Ports
Unicornscan	Yes	19.55	1	0 - 65535
Nmap	Yes	79.61	1	0 - 65535
DMitry	Yes	6.79	1	0-500

Table 1: *Results of Scanning 77.27.21.64 (1 Port only)*

	Detected OS?	Time Taken (s)	Open Ports	Total Ports Scanned
Unicornscan	Yes	41.59	6	0 - 65535
Nmap	Yes	87.12	10	0 - 65535
DMitry	Yes	11.21	3	0 - 1500

Table 2: *Results of Scanning 83.56.11.7 (10 Total Ports)*

	Detected OS?	Time Taken (s)	Open Ports	Total Ports Scanned
Unicornscan	Yes	22.07	6	0 - 65535
Nmap	Yes	77.41	7	0 - 65535
DMitry	Yes	9.11	4	0 - 1500

Table 3: *Results of Scanning 185.237.180.175 (7 Total Ports)*

	Detected OS?	Time Taken (s)	Open Ports	Total Ports Scanned
Unicornscan	Yes	24.6	7	0 - 65535
Nmap	Yes	101.12	9	0 - 65535
DMitry	Yes	11.21	4	0 - 1500

Table 4: *Results of Scanning 195.55.120.106 (9 Total Ports)*

As the beginning standard for testing a lot of the port scanning tools within the implementation section, table 1 shows the result from IP *77.27.21.64,* illustrating, that despite the 1 port that was open (and checked via shodan), it took NMap an incredible 79.61 seconds to check through all 65535 ports. This is an obvious reason as to why it took longer than say DMitry, that had an active limit of only 1500 ports, but fails in comparison to Unicornscan which performed the 65536 scans exactly **55.34 seconds** faster than NMap. Table 2, however, paints an entirely different image. In this instance, despite taking nearly over double the amount of time to scan as Unicornscan, Nmap has proven to be superior, able to resolve all open ports as opposed to both the others falling short. Unicornscan could have possibly missed these ports as they were UDP ports. While retrieving these particular ports are a rare occurrence, they can still transpire. Furthermore, the use of Unicornscan's unique asynchronous scanning methods to sweep through a large number of ports simultaneously may be its downfall if a particular port is unstable and doesn't respond. Due to DMitry's default settings of being configured to only 1500 ports, it was somehow still able to retrieve 3 open ports. That being said, the limitation of having to restrict the port quantity per scan by default, is an immediate/noticeable setback when comparing the scanning methods. Not only this, but DMitry doesn't have a UDP scan in its system, that means despite how rare finding open UDP ports may be, DMitry doesn't offer the functionality of finding it regardless.

The Trend continues with both tables 3 & 4, showing the dominance of just how powerful Nmap is, with being able to detect all the open ports without failure. The only noticeable disadvantage on paper with using Nmap appears to be the time taken to complete a singular scan, however, it could be seen as a tradeoff as it tends to have a solid amount of accuracy on returning open ports and what type of port they are. Therefore, based on this, it could be concluded that NMap is the most beneficial form of information gathering (within the perspective of port scanning). DMitry is unfortunately extremely limited, as it doesn't incorporate a TCP SYN scan, and cannot identify an OS type, making it the least effective of the trio. Finally, Unicornscan could be a potentially useful tool with how fast its scans take place, but it is overshadowed by the pure accuracy that NMap can put out.

In terms of information gathering outside of port scanning, it could be said this thesis has effectively shown how useful passive information collecting technologies like Maltego, Shodan, Recon-NG, and theHarvester are. Every instrument has unique qualities and characteristics that improve the efficacy and efficiency of the reconnaissance process. The adaptability of Recon-NG, which offers a variety of plugins and APIs that may be combined for greater usefulness, cannot be matched by theHarvester, despite theHarvester having some use cases. Outside of maltego and shodan being specific use cases in a league of their own, Recon-NG stands out among the research's instruments because it enables a more thorough information gathering approach. Recon-NG's wide variety of modules allow users to apply strategies like brute-forcing, which theHarvester does not support. This versatility allows users to customise their information-gathering strategy to meet particular goals and target systems. In the end, whether one tool is superior than another, their combination may greatly enhance the information collecting process. It should be noted that the user has the freedom to choose whatever tools they prefer, since the integrated harmony of using all of these tools together would allow for the collection of more information than using any one option alone.

Chapter 6

6 - Future Work

Significant advancement was made in examining and addressing the main objectives during the course of this research, but certain topics could still be broadened in subsequent work. Like in any comprehensive investigation, certain aspects must be given priority while others must be left open for further investigation due to the breadth of the inquiry and the constraints of both resources and time.

6.1 - Wireshark & SQLMap

A potent network protocol analyzer, Wireshark, makes it possible to collect and examine network traffic, giving important details about the underlying communication between objects and services. On the other hand, the process of identifying and taking advantage of SQL injection vulnerabilities in online applications is automated by the well-known SQL injection tool SQLmap. Future studies can go deeper into the technical elements of discovering security holes and broaden the scope of research by including these tools into the information gathering process. A more thorough and rigorous approach to security assessments and penetration testing may be achieved by combining the advantages of these technologies with other information collecting strategies.

6.2 - Social Engineering

Social Engineering is an extremely fascinating subtopic within the world of information gathering and penetration testing. Future studies might investigate the idea of using social engineering approaches in addition to more conventional technological means to acquire information. An example would be to examine the efficiency of various social engineering

techniques in acquiring relevant data from targeted people or organisations by fusing technological proficiency with a thorough grasp of human behaviour. In the end, such study may result in a more thorough method of data collection, bridging the gap between technological and human aspects, and improving the general efficacy of security assessments and penetration testing.

6.3 - Firewalls & Preventive Measures

As a key component of protecting organisational assets, the idea of assessing firewalls and other preventative measures to limit information collecting attempts might be investigated. Simulating various information collecting methods and gauging their performance at evading or breaching the deployed defences would be necessary to evaluate the efficacy of these security features. It is possible to identify any potential flaws or incorrect setups by methodically testing and analysing the performance of firewalls, intrusion detection and prevention systems, and other security measures. Improvement suggestions may then be made. Furthermore, such studies can aid in the creation of best practices and guidelines for businesses, enabling them to better safeguard their valuable data and assets from unauthorised access or online dangers. In the end, examining the effectiveness of preventative measures against information collecting will improve our comprehension of cyber defence tactics and help create more durable and secure systems.

References

[1] Ahmed, S., Khan, H. and Saeed, K. (2019) 'Penetration Testing Active Reconnaissance Phase -Optimized Port Scanning With Nmap Tool', in. Available at: https://doi.org/10.1109/ICOMET.2019.8673520. (Accessed: 16 October 2022).

[2] Ashraf, M. *et al.* (2021) 'Ethical Hacking Methodologies: A Comparative Analysis', in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*. *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pp. 1–5. Available at: https://doi.org/10.1109/MAJICC53071.2021.9526243.(Accessed: 16 October 2022).

[3] Bhusal, C.S. (2021) 'Systematic Review on Social Engineering: Hacking by Manipulating Humans'. Rochester, NY. Available at: https://papers.ssrn.com/abstract=3821594 (Accessed: 16 October 2022).

[4] Billig, J., Danilchenko, Y. and Frank, C.E. (2008) 'Evaluation of Google hacking', in *Proceedings of the 5th annual conference on Information security curriculum development*. New York, NY, USA: Association for Computing Machinery (InfoSecCD '08), pp. 27–32. Available at: https://doi.org/10.1145/1456625.1456634. (Accessed: 17 October 2022).

[5] Bishop, M. (2007) 'About Penetration Testing', *IEEE Security & Privacy*, 5(6), pp. 84–87. Available at: https://doi.org/10.1109/MSP.2007.159. (Accessed: 18 October 2022).

[6] Chauhan, S. and Panda, N.K. (2015) *Hacking Web Intelligence: Open Source Intelligence and Web Reconnaissance Concepts and Techniques*. Syngress. (Accessed: 19 October 2022).

[7] Choo, C.S., Chua, C.L. and Tay, S.-H.V. (2007) 'Automated red teaming: a proposed framework for military application', in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: Association for Computing

Machinery (GECCO '07), pp. 1936–1942. Available at: https://doi.org/10.1145/1276958.1277345. (Accessed: 19 October 2022).

[8] Chowdappa, K.B. and Lakshmi, S.S. (2014) 'Ethical Hacking Techniques with Penetration Testing', 5. (Accessed: 22 October 2022).

[9] Cisar, P. and Pinter, R. (2019) 'Some ethical hacking possibilities in Kali Linux environment', *Journal of Applied Technical and Educational Sciences*, vol. vol. 9. issue 4. ISSN 25605429. Available at: https://doi.org/10.24368/JATES.V9I4.139. (Accessed: 22 October 2022).

[10] Dalseth, M. (no date) 'The Development of a Reconnaissance Tool Aiming to Achieve a More Efficient Information Gathering Phase of a Penetration Test'. (Accessed: 16 October 2022).

[11] EC-Council (2022) 'What Are Footprinting and Reconnaissance?', *Cybersecurity Exchange*, 13 June. Available at: https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/basics-footprinting-reconnaissance/ (Accessed: 16 November 2022).

[12] Emoghene, O. and Nonyelum, O.F. (2017) 'Information Gathering Methods and Tools: A Comparative Study', (4), p. 14. (Accessed: 25 October 2022).

[13] Filiol, E., Mercaldo, F. and Santone, A. (2021) 'A Method for Automatic Penetration Testing and Mitigation: A Red Hat Approach', *Procedia Computer Science*, 192, pp. 2039–2046. Available at: https://doi.org/10.1016/j.procs.2021.08.210. (Accessed: 25 October 2022).

[14] Ghaznavi-Zadeh, R. (no date) *Kali Linux: Hacking Tools Introduction*. Primedia E-launch LLC. (Accessed: 22 October 2022).

[15] Hai-Jew, S. (2014) 'Using Maltego Tungsten to explore the cyber-physical confluence in geolocation', *SIDLIT Conference* [Preprint]. Available at: https://scholarspace.jccc.edu/c2c_sidlit/2014/Thursday/3. (Accessed: 08 November 2022).

[16] Hai-Jew, S. (2017) *Real-Time Sentiment Analysis of Microblog Messages with the Maltego "Tweet Analyzer" Machine*, *Social Media Listening and Monitoring for Business Applications*. IGI Global. Available at: https://doi.org/10.4018/978-1-5225-0846-5.ch012. (Accessed: 08 November 2022).

[17] Harper, D.A. *et al.* (2022) *Gray Hat Hacking: The Ethical Hacker's Handbook*. McGraw-Hill Education. Available at: https://www.accessengineeringlibrary.com/content/book/9781264268948 (Accessed: 08 November 2022).

[18] Heartfield, R. and Loukas, G. (2015) 'A Taxonomy of Attacks and a Survey of Defence Mechanisms for Semantic Social Engineering Attacks', *ACM Computing Surveys*, 48(3), p. 37:1-37:39. Available at: https://doi.org/10.1145/2835375. (Accessed: 02 November 2022).

[19] Hunt, E. (2012) 'US Government Computer Penetration Programs and the Implications for Cyberwar', *IEEE Annals of the History of Computing*, 34(3), pp. 4–21. Available at: https://doi.org/10.1109/MAHC.2011.82. (Accessed: 02 November 2022).

[20] Iommi, E. and Marcantoni, F. (no date) 'OSINT: Information gathering using Maltego'. (Accessed: 03 November 2022).

[21] Landau, S. (2013) 'Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations', *IEEE Security & Privacy*, 11(4), pp. 54–63. Available at: https://doi.org/10.1109/MSP.2013.90. (Accessed: 11 November 2022).

[22] Laxmi Kowta, A.S. *et al.* (2021) 'Analysis and Overview of Information Gathering & Tools for Pentesting', in *2021 International Conference on Computer Communication and Informatics (ICCCI)*. *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–13. Available at: https://doi.org/10.1109/ICCCI50826.2021.9457015. (Accessed: 11 November 2022).

[23] Mamilla, S.R. (no date) 'A Study of Penetration Testing Processes and Tools', p. 59. (Accessed: 18 November 2022).

[24] Mansfield-Devine, S. (2018) 'The best form of defence – the benefits of red teaming', *Computer Fraud & Security*, 2018(10), pp. 8–12. Available at: https://doi.org/10.1016/S1361-3723(18)30097-6. (Accessed: 18 November 2022).

[25] Midian, P. (2002) 'Perspectives on Penetration Testing — Black Box vs. White Box', *Network Security*, 2002(11), pp. 10–12. Available at: https://doi.org/10.1016/S1353-4858(02)11009-9. (Accessed: 18 November 2022).

[26] Mirjalili, M., Nowroozi, A. and Alidoosti, M. (2014) 'A survey on web penetration test', *ACSIJ Advances in Computer Science: an International Journal*, 3. (Accessed: 21 November 2022).

[27] Rana, S., Garg, U. and Gupta, N. (2022) 'Reconnaissance Attacks: A First Step to Hack IoT Devices and Cyber Crime', in A.K. Das et al. (eds) *Computational Intelligence in Pattern Recognition*. Singapore: Springer (Advances in Intelligent Systems and Computing), pp. 183–194. Available at: https://doi.org/10.1007/978-981-16-2543-5_16. (Accessed: 01 December 2022).

[28] S, A. and Dr. Bijimol T K (2021) 'A Research Work on Information Gathering Tools'. Available at: https://doi.org/10.5281/ZENODO.5101265. (Accessed: 01 December 2022).

[29] Saha, Sanchita *et al.* (2020) 'Ethical Hacking: Redefining Security in Information System', in M. Chakraborty, S. Chakrabarti, and V.E. Balas (eds) *Proceedings of International Ethical Hacking Conference 2019*. Singapore: Springer (Advances in Intelligent Systems and Computing), pp. 203–218. Available at: https://doi.org/10.1007/978-981-15-0361-0_16. (Accessed: 09 December 2022).

[30] Salahdine, F. and Kaabouch, N. (2019) 'Social Engineering Attacks: A Survey', *Future Internet*, 11(4), p. 89. Available at: https://doi.org/10.3390/fi11040089. (Accessed: 05 December 2022).

[31] Schwarz, K. and Creutzburg, R. (2021) 'Design of Professional Laboratory Exercises for Effective State-of-the-Art OSINT Investigation Tools - Part 3: Maltego', *Electronic Imaging*, 33, pp. 1–23. Available at: https://doi.org/10.2352/ISSN.2470-1173.2021.3.MOBMU-045. (Accessed: 10 December 2022).

[32] Shebli, H.M.Z.A. and Beheshti, B.D. (2018) 'A study on penetration testing process and tools', in *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–7. Available at: https://doi.org/10.1109/LISAT.2018.8378035. (Accessed: 11 December 2022).

[33] Toxen, B. (2014) 'The NSA and Snowden: securing the all-seeing eye', *Communications of the ACM*, 57(5), pp. 44–51. Available at: https://doi.org/10.1145/2594502. (Accessed: 11 December 2022).

[34] *Types of Penetration Testing* (no date). Available at: https://www.tutorialspoint.com/penetration_testing/types_of_penetration_testing.htm (Accessed: 11 December 2022).

[35] Udayana University, Bali, Indonesia, Eka Pratama, I.P.A. and Wiradarma, A.A.B.A. (2019) 'Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company)', *International Journal of Computer Network and Information Security*, 11(7), pp. 8–12. Available at: https://doi.org/10.5815/ijcnis.2019.07.02. (Accessed: 05 Feburary 2023).

[36] Upton, S.C., Johnson, S.K. and McDonald, M.L. (no date) 'Breaking Blue:Automated Red Teaming Using Evolvable Simulations', p. 3. (Accessed: 04 Feburary 2023).

[37] Vats, P., Mandot, M. and Gosain, A. (2020) 'A Comprehensive Literature Review of Penetration Testing & Its Applications', in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). 2020 8th International Conference on Reliability, Infocom Technologies and*

Optimization (Trends and Future Directions) (ICRITO), pp. 674–680. Available at: https://doi.org/10.1109/ICRITO48877.2020.9197961. (Accessed: 05 Feburary 2023).

[38] Verble, J. (2014) 'The NSA and Edward Snowden: surveillance in the 21st century', *ACM SIGCAS Computers and Society*, 44(3), pp. 14–20. Available at: https://doi.org/10.1145/2684097.2684101. (Accessed: 06 Feburary 2023).

[39] de Vivo, M. *et al.* (1999) 'A review of port scanning techniques', *ACM SIGCOMM Computer Communication Review*, 29(2), pp. 41–48. Available at: https://doi.org/10.1145/505733.505737. (Accessed: 14 Feburary 2023).

[40] Weidman, G. (2014) *Penetration Testing: A Hands-On Introduction to Hacking*. No Starch Press. (Accessed: 03 April 2023).

[41] Piotrowski, M. (2005) 'Dangerous Google – Searching for Secrets'. Retrieved from https://doc.lagout.org/security/DangerousGoogle-SearchingForSecrets.pdf (Accessed: 03 April 2023).

[42] 'Scanning and Enumeration Phase' (2019) in *Constructing an Ethical Hacking Knowledge Base for Threat Awareness and Prevention*. IGI Global, pp. 149–177. Available at: https://doi.org/10.4018/978-1-5225-7628-0.ch006. (Accessed: 14 March 2023).

[43] Sinha, S. and Arora, Dr.Y. (2020) 'ETHICAL HACKING: THE STORY OF A WHITE HAT HACKER', *International Journal of Innovative Research in Computer Science & Technology*, 8(3). Available at: https://doi.org/10.21276/ijircst.2020.8.3.17. (Accessed: 14 March 2023).

[44] Hidayah Zulkiffli, S.N., Ahmad Zawawi, M.N. and Rahim, F.A. (2020) 'Passive and Active Reconnaissance: A Social Engineering Case Study', in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*. *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pp. 138–143. Available at: https://doi.org/10.1109/ICIMU49871.2020.9243402. (Accessed: 15 March 2023).

[45] Smith, B., Yurcik, W. and Doss, D. (2002) 'Ethical hacking: the security justification redux', in *IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No.02CH37293)*. *IEEE 2002 International Symposium on Technology and Society (ISTAS'02). Social Implications of Information and Communication Technology. Proceedings (Cat. No.02CH37293)*, pp. 374–379. Available at: https://doi.org/10.1109/ISTAS.2002.1013840. (Accessed: 03 April 2023).

Appendices

6.1 Commands/Code

Due to the Usage of Kali linux for the purpose of this thesis, No actual code implementations were used. This is due to the fact that looking into the analysis of information gathering tools uses scripts and commands on command line programs/tools, as opposed to writing large software and programs.

Included in the code submission, is a zipped file of every single command that was run, allowing anybody with access to it, the ability to recreate everything that has occurred during the process of this thesis research.

6.2 Original Proposal

Candidate Num: 231012

Working Title: Performancy Analysis Of Information Gathering Tools

Supervisor: Khan, Imran

Aims

Information gathering is a technique used in the process of ethical hacking, to determine the quantity of knowledge a hacker can potentially learn about a target before inducing an attack. Gathering knowledge is split between two primary functions, passive & active gathering. During the passive gathering, the attacker looks to stay as subtle as possible and makes no direct contact with the subject/victim. Contrary to this, active attacks require the perpetrator to directly involve themselves with the target (through means like that of social engineering attacks). The project set forth looks to investigate the usage of the aforementioned information-gathering techniques against vulnerable systems, specifically using AWS cloud services to host the entirety of the lab. In addition, this particular subject matter allows for an in-depth analysis of professional-level

techniques used by penetration testers to examine how effective a variety of methods can be in breaching potential targets. After discussing and collating the required data, potential solutions will be suggested as a means of demonstrating “blue-teaming” techniques, a term used to suggest the role of a penetration tester in how to harden the defense of a system.

- Create a virtual testing environment that allows for safe penetration testing
- Demonstrate how to use a variety of tools, and their effectiveness in different scenarios
- Look at the difference between active and passive gathering
- Examine the cyber security functions built into cloud tools like AWS, that prevent and protect a user’s data from being accessed.

Objectives

Primary

- Research and learn the tools that are used professionally in the industry to penetration test as well as defend against.
- Create a virtual testing lab within VMware, but uploaded to AWS servers, such that all data used is encrypted. The testing lab will consist of tools like DVWA as well as a fresh install of Windows 10/11, with fake data made by myself in an attempt to see what penetration hacking methods can be used in an effort to obtain it.
- Annotate and describe the output retrieved by all the information gathered and how it is different from one another.
- Look at examples and information from other studies to see how they correlate and back up the work produced by myself during this project

Extensions

- Design an outline for a prototype firewall that could be used to prevent any hacking attempts. Furthermore, adapting the prototype to understand different types of malicious software, and how they can be identified
- Look at further methods of pen testing outside of active and passive gathering, and more direct attacks such as brute-forcing, dictionary-based attacks to do with personal information (accounts for websites, etc.)

- Look at how a virus works to gather information about a target (keyloggers, backdoors, RATS, Worms, etc.)

Relevance

The work required during this project will be closely linked to the previously completed courses taken during my time at Sussex. I will look to integrate the knowledge of SQL, Python, and Linux/bash scripting techniques learned throughout Y1-3 of my course. Furthermore, a large majority of this project will focus on skills and knowledge enhanced within Introduction to Computer Security, a year 3 elective module. This project as a whole will allow me to get a better understanding of computer hacking & vulnerability exploitation, hence strengthening my career in computer security/ penetration testing.

Resources Required

Due to the nature and potential risk of the project at hand, there is no need for the involvement of any participants. The need for a "target" host system will be replaced with tools and applications like **DVWA** (Damn Vulnerable Web Application), a free-to-integrate penetration testing solution used by professionals within the industry. Furthermore, due to the concept of having it cloud-hosted, AWS services will be utilized on the free tier, allowing for ease of access for users and allowing this project to be run without the need for a cost value. Furthermore, the final tool will be a host-installed copy of Kali Linux, a Linux distribution of a penetration testing operating system that is maintained by the Offensive Security company.

Weekly Timetable

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00					
08:30					Free Study
09:00	Human-Computer	Project	Knowledge &	Project	
09:30	Interation		Reasoning		
10:00					
10:30		Work	Free Study	Work	
11:00					Knowledge &
11:30					Reasoning
12:00				Intro to	
12:30	Project			Comp-Security	
13:00		Intro to	Project		
13:30		Comp-Security		Free Study	
14:00	Work				
14:30					
15:00			Work		Intro to
15:30					Comp-Security
16:00	Free Study			Knowledge &	
16:30		Project		Reasoning	Lab
17:00					
17:30					
18:00		Work	Free Study		
18:30					
19:00				Human-Computer	
19:30				Interation	
20:00					

6.3– Meeting Log

Zoom Meeting - 24/10/23 -

- Initial questions and the working title were discussed upon
    - Determined whether to stick with the working title or to integrate a portion of A.I. into the mix.
- Decided upon using AWS (Amazon Web Services), for the cloud hosting platform, as it is free for the first year and allows for custom installs of all the tools required for this project.

- Discussed how I might go about collecting results for this project and how the project could be outlined
- Discussed what ethics approval may be required and how to complete the forms needed

Zoom Meeting - 24/10/22 -

- Initial questions and the working title were discussed upon
    - Determined whether to stick with the working title or to integrate a portion of A.I. into the mix.
- Decided upon using AWS (Amazon Web Services), for the cloud hosting platform, as it is free for the first year and allows for custom installs of all the tools required for this project.
- Discussed how I might go about collecting results for this project and how the project could be outlined
- Discussed what ethics approval may be required and how to complete the forms needed

Zoom Meeting - 3/11/22 -

- Discussed where I currently was in my work and research. Looked at some of the literature I had examined for my project and how they were suitable.
- Discussed the progress I had made with my Interim report and how much I had completed.
    - Further looking at the ethical section and making it clear what forms had to be filled out for the specific task at hand.