****

**FACULTY OF SCIENCE, ENGINEERING AND COMPUTING**

**School of Computer Science and Mathematics**

**BSc (Hons)**

**Cyber Security and Digital Forensics**

Bilal Amir

K2010293

Detection of DOS attacks on IOT devices using Log analysis

28/04/2023

Hu Yuan

KU London Logo

**Declaration**

I have read and understood the University regulations on plagiarism, and I understand the meaning of the word *plagiarism*. I declare that this report is entirely my own work. Any other sources are duly acknowledged and referenced according to the requirements of the School of Computer Science and Mathematics. All verbatim citations are indicated by double quotation marks (“…”).  Neither in part nor in its entirety have I made use of another student’s work and pretended that it is my own.  I have not asked anybody to contribute to this project in the form of code, text, or drawings.  I did not allow and will not allow anyone to copy my work with the intention of presenting it as their own work.

Date: 20/11/2022

Name: Bilal Amir

Signature: BAmir

|  |
| --- |
|  |
| Table Of Contents  [List of Figures 4](#_Toc133793593)  [**1.Introduction and Background** 6](#_Toc133793594)  [1.1. Background 8](#_Toc133793595)  [1.2 Aims and Objectives 9](#_Toc133793596)  [1.3 legal and ethical concerns 10](#_Toc133793597)  [**2. Project Management Strategy** 11](#_Toc133793598)  [**3. Literature Review:** 14](#_Toc133793599)  [**4. Methodology:** 20](#_Toc133793600)  [4.1 Hardware requirements 21](#_Toc133793601)  [4.2 Software requirements 21](#_Toc133793602)  [4.3 Technology comparison section 22](#_Toc133793603)  [**5. Implementation of Artefact:** 25](#_Toc133793604)  [5.1 Launching the DoS attack using Metasploit 25](#_Toc133793605)  [5.2 Using Hping3 to launch DOS attack 30](#_Toc133793606)  [5.3 Monitoring Normal traffic 34](#_Toc133793607)  [5.4 Splunk enterprise and splunk forwarder 36](#_Toc133793608)  [5.5 Network Traffic analysis using Splunk 43](#_Toc133793609)  [5.6 Analysis of the Network traffic from the synflood attack 44](#_Toc133793610)  [**6. Comparison of Results** 50](#_Toc133793611)  [**7. Critical Review and Conclusion** 52](#_Toc133793612)  [7.1 Study limitation 52](#_Toc133793613)  [7.2 Future work 53](#_Toc133793614)  [**8. Bibliography** 54](#_Toc133793615) List of Figures [Figure 1: what is Internet of things from (Avsystem,2019) 6](#_Toc133319930)  [Figure 2: Gantt chart “planned schedule for the project” 10](#_Toc133319931)  [Figure 3: Distinctive communication patterns between an infected IoT device and Mirai’s loader component. SYN (synchronize), FIN (finish), PSH (push), and ACK (acknowledge) are standard TCP packet types, from (Kolias ,Kambourakis and Voas, 2017) 13](#_Toc133319932)  [Figure 4: Attacker virtual machine 19](#_Toc133319933)  [Figure 5: Victim virtual machine 19](#_Toc133319934)  [Figure 6: pinging virtual machines to ensure connection 20](#_Toc133319935)  [Figure 7: Starting Metasploit on the attacker machine 20](#_Toc133319936)  [Figure 8: using synflood on attacker virtual machine 21](#_Toc133319937)  [Figure 9: Setting the target info and preparing the exploit command to launch the attack. 21](#_Toc133319938)  [Figure 10: using tcpdump to monitor the network traffic on the victim ip and storing it in a file. 22](#_Toc133319939)  [Figure 11: Listening on the ip address of the victim 22](#_Toc133319940)  [Figure 12: launching the attack from the attacker’s virtual machine 22](#_Toc133319941)  [Figure 13: Result of the attack 23](#_Toc133319942)  [Figure 14: launching various attacks 23](#_Toc133319943)  [Figure 15: setting permissions for the file 24](#_Toc133319944)  [Figure 16: Attacker virtual machine 24](#_Toc133319945)  [Figure 17: Victim virtual machine 25](#_Toc133319946)  [Figure 18: Attacker using icmp request to flood. 26](#_Toc133319947)  [Figure 19: using Tcpdump to record the network traffic 26](#_Toc133319948)  [Figure 20: Attacker flooding the victim with random source 27](#_Toc133319949)  [Figure 21: Using Tcpdump to record save it to a file 27](#_Toc133319950)  [Figure 22: Attacker flooding with udp packets 27](#_Toc133319951)  [Figure 23: Attacker flooding with sctp packets 27](#_Toc133319952)  [Figure 24: Setting file permissions 28](#_Toc133319953)  [Figure 25: Checking the IP address on the victim machine with Nat network 28](#_Toc133319954)  [Figure 26: using tcpdump to record the network traffic 28](#_Toc133319955)  [Figure 27: Loading a browser on the victim machine 29](#_Toc133319956)  [Figure 28: setting file permissions 29](#_Toc133319957)  [Figure 29: configuring the network settings for Splunk server 30](#_Toc133319958)  [Figure 30: Downloading Splunk from the browser 30](#_Toc133319959)  [Figure 31: installing Splunk with the wget command 31](#_Toc133319960)  [Figure 32: Starting Splunk server 31](#_Toc133319961)  [Figure 33: Enabling port to receive data. 32](#_Toc133319962)  [Figure 34: Checking port status 32](#_Toc133319963)  [Figure 35: installing Splunk forwarder on victim virtual machine 33](#_Toc133319964)  [Figure 36: Starting Splunk forwarder 33](#_Toc133319965)  [Figure 37: Adding forwarding server (Splunk server) 34](#_Toc133319966)  [Figure 38: Adding monitor of Synflood attack traffic 34](#_Toc133319967)  [Figure 39: Adding monitor of Hping3 attack traffic 34](#_Toc133319968)  [Figure 40: Adding monitor of Normal traffic 34](#_Toc133319969)  [Figure 41: Updated data summary 35](#_Toc133319970)  [Figure 42: Updated sources 35](#_Toc133319971)  [Figure 43: visualisation using Splunk’s search and report feature 35](#_Toc133319972)  [Figure 44: creating a dashboard using Splunk 36](#_Toc133319973)  [Figure 45: Synflood Dashboard 38](#_Toc133319974)  [Figure 46: Hping3 Dashboard 40](#_Toc133319975)  [Figure 47: Normal traffic Dashboard 42](#_Toc133319976)  [Figure 48: Hping3 network traffic visualisation 43](#_Toc133319977)  [Figure 49: normal network traffic visualisation 43](#_Toc133319978)  [Figure 50: Synflood vulnerable port visualisation 44](#_Toc133319979)  [Figure 51: ports used during normal network traffic. 44](#_Toc133319980)  **Glossary of Terms**   1. Distributed denial of service (DDoS): A common type of cyber-attack which involves utilising numerous compromised systems, usually infected with malware, to flood a particular website or server with a substantial flow of traffic. The objective of this attack is to overpower the target's resources, making it inaccessible to users. 2. Internet of Things (IoT): it is a system consisting of interconnected devices, including household appliances, transportation vehicles, and other devices. These devices are embedded with sensors and software enabling them to communicate and exchange information. 3. Intrusion Detection System (IDS): It is a component of hardware or software used to monitor malicious activity or potential policy breaches on a network or system. It examines a range of data sources, including system logs and network traffic and finds any possible security risks. The IDS can also notify system administrators or security teams in the case of a security breach, enabling them to take preventive steps to limit damage. 4. Denial of service (DoS): This type of cyberattack aims to prevent the users from accessing a website or network. In a DoS attack, one attacker machine overwhelms the targeted system with a massive amount of network traffic. This can then exhaust the system's resources of the target to the point where it crashes or becomes inaccessible. |
| **1.Introduction and Background** Companies have long relied on client input to enhance their products and comprehend their clients' needs. Since the benefits of IoT devices are more widely recognised, businesses are increasingly deploying them to accomplish a broader spectrum of tasks. These goals include increasing productivity, automating repetitive processes to save time, and even analysing data from IoT devices to enhance their products. Despite their benefits, IoT devices' reliance on internet access makes them vulnerable to cyberattacks and data breaches. A company's reputation might suffer significantly if sensitive client data is lost. Security measures must therefore be given top priority by companies developing IoT devices (Gills, 2022).  Internet of Things (IoT) security is crucial for several reasons, and companies must consider various security concerns while constructing these devices. IoT devices' frequent lack of adequate security measures are one of the fundamental causes for concern. Many devices are pre-configured with weak default passwords that might be cracked or quickly guessed, and they might need to get software upgrades to patch security flaws. This makes them a simple target for attackers trying to access a network without authorisation or stealing personal data. Another security issue regarding IoT devices is the possibility of widespread Distributed Denial of Service attacks. These attacks involve the deployment of several infected devices to overload a network or website with traffic, blocking it from being accessed by authorised users. IoT devices are more susceptible to this type of attack due to their limited processing and memory capacities, making them easy targets for attackers seeking to obtain control over them. In addition to these concerns, IoT devices have the potential to be utilised to collect private information like login credentials and personal data. Once a hacker obtains access to this data, they can use it to perform financial fraud or identity theft (strike, 2022).  The growing popularity of the Internet of Things industry has raised significant concerns regarding the potential for Denial-of-Service attacks targeting IoT devices. IoT devices are vulnerable due to their limited processing and memory capabilities, which cyber attackers can leverage these devices to initiate botnets that have the ability to carry out distributed denial-of-service attacks, flooding an entire network or website with a continuous stream of traffic and making it inaccessible. Since IoT devices are deployed in significant quantities, cyber attackers can exploit them to initiate extensive attacks that may potentially disrupt even the most durable networks. Therefore, Companies must implement strong security measures, such as secure authentication mechanisms and frequent software upgrades, to guarantee the safety and reliability of IoT devices. (Gupta, 2021).  The prevention of Denial of Service (DoS) attacks on IoT devices is critical, and effective log management is a key component in this effort. Log management systems can analyse traffic patterns from diverse IoT devices and servers in order to detect any indication of a potential DoS attack. This analysis provides vital insights to security teams, allowing them to act swiftly to mitigate the attack's effects.  However, the benefits of log management go beyond identifying DoS attacks. These systems can also identify compromised devices within an IoT network by monitoring for unusual activity that could indicate a breach. With this information, security teams can identify potential vulnerabilities in IoT devices and networks and take appropriate steps to prevent an attack. Moreover, log management insights are vital to proactive security measures, allowing security teams to isolate compromised devices and take necessary steps to prevent attacks from happening and mitigate any damage caused. Therefore, log management systems are essential for securing IoT devices, and their valuable insights can help prevent DoS attacks and other security breaches (Tools, 2020).  This report aims to examine the threats of cyber-attacks that can arise from not properly securing IoT devices. The report also aims to discuss the effectiveness of log management and how it can be used to detect cyber-attacks such as DoS. There is a list of objectives that would contribute towards achieving this goal. Through each of these increments, the report will provide a more comprehensive understanding of IoT devices and the security vulnerabilities surrounding them. 1.1. Background What is IoT? A Simple Explanation of the Internet of Things  Figure 1: what is Internet of things from (Avsystem,2019)  This section of the report aims to provide background knowledge regarding the evolution of IoT and its potential to grow over the years. The significant events that portray the evolution of IoT can be seen in *Figure 1*.  The development of the IoT is due to several decades of advancements in the field of computing. The early computer networks formed the basis for the Internet and communication technologies, which eventually enabled the creation of IoT devices. In 1968, Theodore Paraskevakos pioneered machine-to-machine (M2M) communications, leading to the development of electronic meters that could communicate with power plants. Bluetooth and Wi-Fi emerged in the 1990s, enabling wireless IoT devices to share data over the internet without physical connections. These advances have transformed device interaction and control, making IoT a rapidly growing field with vast potential (Gupta, 2021).  IoT development can be traced back to the early 2000s and the rise of Radio-Frequency Identification technology. Initially, IoT devices were used in industrial settings to monitor machinery via closed networks. In 1999, Kevin Ashton created the term "Internet of Things" during an Radio-frequency identification(RFID) presentation, sparking interest in the possibilities of interconnectivity. Wireless sensor networks (WSNs) emerged as another significant development, allowing tiny, low-power sensors to gather data in physical environments. Smart homes also emerged in the 2000s as a significant area for application for internet-connected IoT devices, showcasing the potential for IoT to transform daily life (Futures, 2016).  The COVID-19 pandemic of 2020 has had a significant impact on the use of IoT devices in healthcare and remote work. These devices were used to track hospital supplies, monitor patients remotely, and provide real-time data to medical practitioners. They were also employed to track the availability of medical equipment and monitor the virus's spread. In remote work, IoT devices allowed employees to operate from home while staying connected to colleagues and clients. The pandemic underscored the value of IoT devices in facilitating remote work and healthcare and is expected to lead to increased usage in the coming years. IoT devices have the potential to become increasingly vital to companies' efforts to provide better healthcare services and adapt to changing work environments (Javaid and Khan, 2021). 1.2 Aims and Objectives The overall purpose of this project has been broken down into the following more straightforward goals:   * Providing a detailed study regarding security vulnerabilities within IoT devices * Research regarding IoT devices’ infrastructure and vulnerabilities. * Review IoT software and analysis of different prior research. * Research regarding DoS and DDoS attacks. * Research into the implementation of different types of DoS attacks. * Research into log management and analysis of logs. * A detailed Log Analysis of a cyber-attack on a virtual machine with IoT software. * Discussion of Log analysis results with comparison.   The project aims to educate IoT users about the safety of their devices. The stated objectives include examining a cyber-attack scenario, which would demonstrate IoT device vulnerabilities and data loss risk. The cyber-attack would be carried out in a virtual environment to fully comprehend the victim's perspective. This research will also examine cyberattack-detection strategies. IoT developers can discover how to recognise attacks through this investigation. The report will also provide a network traffic analysis of the virtual machine. This will reveal major trends and security issues associated with IoT devices.  The Internet of Things could impact almost every sector and element of people's life. Therefore, this research could have a significant impact. Furthermore, cyberattacks such as DDoS, Botnets, malware and others could additionally impact IoT devices. Since IoT has many vulnerabilities and a significant impact, this study will narrow its scope to focus on DoS attacks and how log analysis can be used to detect them, offering a more in-depth examination of a particular vulnerability in IoT devices.  1.3 legal and ethical concerns **IoT devices facilitate many people's lives, but they also present severe issues with privacy since they capture a lot of personal data that developers must consider. IoT developers must establish a privacy policy since they collect and utilise personal data. Therefore, the regular privacy policy will apply to any data collected by the IoT operating system for this project.**  **When performing a DoS experiment in a virtual environment, ethical considerations and procedures must be followed. The target system was simulated in a controlled, not a live system. The authorisation should be obtained to avoid violating the rights of others or engaging in illegal activity. The attack's duration and intensity were limited to avoid damaging the target system. Detailed documentation is essential to keep track of the investigation. The report aims to consider ethical concerns and utilise appropriate steps while conducting the experiment.**  When designing IoT devices, the GDPR must be taken into account. This regulation addresses data protection and communications between machines. IoT developers must follow a strict development cycle when creating IoT devices, which includes selecting a Data Protection Officer. The General Data Protection Regulation promotes data privacy and security for manufacturers and end-users, acquiring trust and demonstrating a commitment to protecting personal data. This project aims to comply with GDPR policies while utilising IoT devices or software (Stelmashchuk, 2020). **2. Project Management Strategy** This project was managed using the waterfall methodology, which is a project management technique. The Waterfall methodology follows a linear approach, where a sequential process of gathering requirements, designing, implementing, testing, and deploying takes place. It is ideal for individual projects with a well-defined scope, requirements, and timelines, as it ensures clarity in understanding project goals and objectives, simplifies project management, and maintains control over progress. However, the methodology can be rigid and less adaptable to complex projects requiring frequent feedback and collaboration (Hoory and Bottorff, 2021). Since the project objective is well-defined and unchanging, it is beneficial to employ this methodology. This methodology also facilitates the project to be completed in increments, meaning that one task must be completed prior to commencing the next. This is also time efficient because it enables all aspects of the project to be completed by their specified deadlines.  There are also a variety of project management methodologies, and each has its own advantages. Agile methodologies, such as Scrum and Kanban, are more flexible and adaptive, prioritising collaboration, feedback, and continuous improvement. Scrum provides a framework for organising work in short sprints, which is focused on delivering value in the shortest time possible. Kanban visualises workflows, limits work in progress and manages flow to increase productivity and improve quality. Both Scrum and Kanban emphasises iterative and incremental development, allowing teams to respond to changes and adapt to new requirements as they arise.  In contrast to the linear process of Waterfall, Agile, Scrum, and Kanban offer a more iterative approach, allowing teams to continuously learn, adapt, and improve throughout the project. Through emphasising collaboration and feedback, Agile methodologies enable teams to work more closely together and respond quickly to changing circumstances. While Waterfall may be more structured, these alternatives prioritise flexibility, enabling teams to adapt to changing requirements and deliver high-quality results.  However, the Waterfall methodology provides a more structured and disciplined approach to project management. This is ideal for this project as it ensures clarity in understanding the project goals and objectives, which simplifies project management and maintains control over progress. Utilising the distinct stages and deliverables, Waterfall also makes it easier to track progress and measure success.  This section of the report aims to divide the project into smaller, more manageable tasks that, which will contribute to the completion of the entire project. The tasks were completed following the waterfall methodology, where the completion of one task started another. The table and the Gantt chart below display the split of the tasks.   |  |  |  |  | | --- | --- | --- | --- | | TASK ID | TASK | TASK DURATION | TASK START DATE | | 1 | Complete the project development form | 7 | 26/09/2022 | | 2 | Draft for project proposal | 7 | 03/10/2022 | | 3 | Research into IoT | 7 | 10/10/2022 | | 4 | Research into IoT security | 7 | 17/10/2022 | | 5 | Research into DOS attack | 14 | 24/10/2022 | | 6 | Research into security vulnerabilities | 7 | 07/11/2022 | | 7 | Research into defending against attacks | 7 | 14/11/2022 | | 8 | Research into IoT operating systems | 28 | 21/11/2022 | | 9 | Research into cyber attacks | 21 | 19/12/2022 | | 10 | Methods of implementing a cyber attack | 14 | 09/01/2023 | | 11 | Researching virtual environments | 14 | 23/01/2023 | | 12 | Research into log management and analysis | 14 | 06/02/2023 | | 13 | Demonstration of a cyber-attack log analysis | 14 | 20/02/2023 | | 14 | Discussion of results | 7 | 06/03/2023 | | 15 | Conclusion for report | 7 | 13/03/2023 |   *Table 1: planned schedule for project*    Figure 2: Gantt chart “planned schedule for the project” |
| **3. Literature Review:**  This section of the report contains literature on the application of cyber-attacks attacks, DoS and DDoS in particular, it also reflects prior studies conducted by researchers regarding the detection of DoS attacks using log analysis. This section serves as the foundation for this project since it discusses the usability and effectiveness of log management in detecting DoS attacks.  Cyber-attacks have increasingly become more prevalent in society, and it is essential for individuals to consider these threats when using online services. There are various types of attacks a cybercriminal can utilise to harm a target; for example, DoS, DNS tunnelling, and malware are all attacks that can be used to disrupt, infect, and gain unauthorised access to vulnerable systems. Malicious users can also access sensitive data to exploit companies and individuals. This could have devastating consequences for a company’s reputation and ruin the trust between a company and its clients. Therefore, it is critical for companies and individuals to consider the security vulnerabilities within their systems and attempt to mitigate these security flaws (IBM, n.d.). It is also important to understand the functionality of such attacks and how they can be detected through log analysis and network activities.  According to an article by Shishir Kumar Shandilya (Shandilya et al., 2023), demonstration of various cyber-attacks can be conducted in a controlled environment to understand the impact of such attacks and develop defence mechanism. The article consists of a comprehensive dataset of real-time network data, which includes a variety of cyber-attacks, such as distributed denial-of-service attacks. The dataset simulated a controlled network environment with ten computers connected to a router in a Docker Bridge network, each targeted with different attack scenarios such as SQL injection, account takeover, service exploitation, DNS, and ARP spoofing. Through the analysis of the network traffic logs of the dataset, the article highlighted how the traffic patterns generated by DDoS attacks can be identified, providing valuable insights into the development of effective detection mechanisms that can recognize similar activity. Furthermore, the article also emphasised the importance of regular updates and patching vulnerabilities to prevent exploitation by attackers. Through the comparison of the vulnerabilities of older versions of applications to newer and updated ones, the significance of staying up to date was further acknowledged. The information regarding trends and patterns of various cyber-attacks through this article is beneficial as it offers valuable knowledge for the detection and prevention of similar events.  **An article by** Elisa Bertino and Nayeem Islam **(Bertino and Islam, 2017) discusses how IoT systems due to their low computational power are at a higher risk to cyber-attacks. This also makes them an “attractive target” for attackers to compromise and utilise as a botnet.**  **A botnet is a collection of machines that have been hacked and are controlled by a botmaster for criminal purposes, attacks such as email spam, distributed denial of service attacks, password cracking, key logging, and cryptocurrency mining could be used by botnets. Bots can spread throughout a network by exploiting security weaknesses and vulnerable passwords. Once a machine is compromised, the botmaster can activate it to launch attacks, such as DDoS attacks, on a target website (Crowdstrike, 2020). The article discusses the "Mirai botnet", a notable example of how compromised IoT devices can be weaponized for cyber-attacks, resulting in a massive 600-Gbps assault on Brian Krebs's security blog. The article highlights the alarming lack of security in IoT devices, with up to 25 vulnerabilities per device, such as weak passwords, absence of encryption, and vulnerable firmware. This highlights the urgent need for incorporating extensive security features within IoT devices. While changing default passwords and updating IoT devices with security patches can address common vulnerabilities, a comprehensive security strategy is essential to ensure IoT systems' safety. The article also recommends conducting a thorough risk assessment and implementing security mechanisms such as encryption, authentication, access control and application security that can be scaled up to match the IoT ecosystem. This article is a valuable source of insights and recommendations for securing Internet of Things devices. However, the article also acknowledges the diversity of IoT applications, and the challenges involved in selecting the appropriate combination of techniques and tools to secure IoT systems.**  An article by Constantinos Kolias and Georgios Kambourakis (Kolias ,Kambourakis and Voas, 2017) further discusses the increasing threat of botnets, especially the Mirai botnet and its different variants. Mirai has become a significant threat to the Internet of Things (IoT) landscape, infiltrating devices such as webcams, DVRs, and routers. Its sophisticated attack techniques involve utilizing a limited dictionary of username-password combinations to exploit vulnerable IoT devices, using brute force tactics to gain administrative access. Once access is granted, the botnet launches a devastating distributed denial of service (DDoS) attack on a targeted group of servers by propagating through poorly configured IoT devices. The botnet is composed of four distinct components, including the bot, the command and control (C&C) server, the loader, and the report server, which work together in a coordinated effort to launch attacks on the Internet (Usenix Association, 2005). The article also highlights the digital footprint left by mirai while communicating with compromised IoT devices.  The article delves into the communication patterns and signatures of Mirai, this type of malware can be detected through basic network analysis. By examining Mirai's behaviour on a network, the article identifies specific patterns and signatures that can be used to identify its presence. These include sequentially testing credentials, sending reports, downloading binary code, receiving attack commands, and generating attack traffic with a few random elements. Additionally, the article includes a valuable diagram (Figure 3) illustrating the communication patterns between an infected IoT device and Mirai's loader component. This diagram can be used to identify network traffic associated with Mirai infections. A deeper understanding of the communication patterns and behaviour of malware like Mirai, valuable insights can be gained regarding developing methods that detect and prevent similar types of cyber-attacks.    Figure 3: Distinctive communication patterns between an infected IoT device and Mirai’s loader component. SYN (synchronize), FIN (finish), PSH (push), and ACK (acknowledge) are standard TCP packet types, from (Kolias ,Kambourakis and Voas, 2017)  Log management could also play a critical role in identifying denial-of-service attacks; it allows organizations to monitor and evaluate events across their IT infrastructure. Through the collection and real-time analysis of log data from diverse sources, security teams can detect abnormal trends or behaviours that may indicate an ongoing attack. The Log data also provides valuable historical context, different attack vectors, and facilitates incident response. Through analysing log data, security teams can determine the extent of the attack, the affected systems, and the most suitable measures to counteract the attack (Security Centre, 2021). Log management could be conducted on all sizes of networks and could be performed manually in a cost-effective manner. However, as the size of the network grows larger and data collected becomes more complex, it becomes impractical and time-consuming to manage data manually. The utilisation of log-management tools become effective in such cases as they efficiently analyse and collect data from various sources.  **According to an article by Vasileios Anastopoulos and Sokratis Katsikas (Anastopoulos and Katsikas, 2017), a structured methodology for deploying a log management could be used on even wider-area-networks. The article discusses a structured methodology for implementing a vendor-independent log management infrastructure in wide area networks (WANs) and could offer valuable insight into research regarding log management for IoT security. While the article focuses on WANs, it also discusses concepts and best practices that could also be applied to IoT environments. The methodology proposed in the article covers various aspects of log management, such as requirements gathering, network topology, log generation, collection, and storage. Additionally, the article highlights the significance of log management for network security, real-time monitoring, and contingency planning, which are all crucial aspects for securing IoT devices and networks. The case study presented within the article also provides further insights and inspiration for practical log management implementations, which could be beneficial for future experiments. The proposed methodology in the article could also serve as a framework for designing a log management infrastructure for IoT devices which would contribute towards the development of detecting cyber-attacks.**  However, Nadia Chaabouni, Mohamed Mosbah offer an alternative solution for detecting cyber-attacks within their research (Chaabouni, Mosbah and Faruki, 2019), suggesting the use of Intrusion Detection Systems (IDSs) for the Internet of Things devices using machine learning techniques. The article discusses the challenges of securing IoT systems with traditional defence mechanisms, such as encryption and access control, which could be insufficient due to the complexity and dynamic nature of these systems. In order to overcome these challenges, the article suggests that IDSs are a more effective security solution for IoT systems and highlights the potential of Machine learning algorithms, specifically deep learning, to enhance the effectiveness of IDSs. Additionally, the article emphasizes the importance of Intrusion detection systems and ML techniques for protecting IoT systems against cyber-attacks. The study also evaluates existing defence techniques and explores the different types of IDSs that can be deployed within IoT systems. This article provides valuable insights into the use of IDSs and ML techniques for detecting cyber-attacks in IoT systems and can serve as a source of inspiration for the development of detection systems.  Furthermore, there are various challenges while implementing log management on a large scale. As stated in an article by Andriy Miranskyy (Miranskyy, Hamou-Lhadj and Larsson, 2016); “big data systems are complex and have many dynamic components”. The article presents the challenges associated with managing large logs in big data systems and their impact on various tasks. The challenges identified within the article were limited storage issues, scalability issues, and multiple log formats. These issues are also prevalent in the management of logs from IoT systems. furthermore, the article urges the importance of logs in detecting security breaches, which reinforces the effectiveness of log management in detecting cyber-attacks.  The article also suggests potential solutions to address these challenges, such as building converters for each log format encountered in IoT environments, distributing large logs across various storage devices to cope with limited storage, and using homomorphic encryption on the log data for privacy, which is relevant in the context of IoT systems that collect sensitive data. The article provides valuable insights into potential solutions for managing large log data in IoT systems, which can contribute to the development of effective IoT security strategies.  **The research examined in this section of the report provided significant insight into the objective of the project and efficient methods for accomplishing it. The research and article outcomes highlighted potential limitations in the suggested solution and offered alternatives. This information was crucial since it contributed to the project's success and helped provide a solid foundation for any further investigation.** |
| **4. Methodology:** This experiment intends to conduct a DOS attack within a virtual environment and investigate both the attacker’s and the victim's perspectives. During this experiment, the victim, which was configured to be an IoT device, was flooded with packets over the network, replicating a DOS attack. These types of cyber-attacks present a significant threat towards devices connected to the internet, especially IoT devices as they contain various security vulnerabilities. The inability to secure devices against these assaults can have significant repercussions, including financial loss, service interruptions, and a detrimental effect on a company's reputation. Therefore, companies must investigate these concerns and develop preventative measures.  To comprehend the functionality of DOS attacks, it is beneficial to examine their influence in a scenario. During this experiment, two types of DOS attack was launched against a virtual machine representing an IoT device. The attack was initiated by flooding the network with a massive volume of data, which was then analysed for important information regarding the attack, such as its origin, packet types, and impact on the system. It was also important to record normal network traffic as it demonstrated the difference in network activity during a cyber-attack. The log analysis of the attacks and normal network traffic provided additional insight into the characteristics of DOS attacks, including attack patterns and duration in comparison to normal network behaviour. The results of the experiment could potentially assist in the creation of defences against these attacks.  The purpose of this experiment was to identify security flaws within IoT devices and demonstrate how DOS assaults can be identified using log data. The results obtained from this experiment would be beneficial for researchers, developers, and users who value the security of Internet of Things devices.  The literature review produced significant insights regarding the experiment's methodology and anticipated outcomes. The research highlighted the severe consequences of cyber-attacks, particularly DDoS attacks, highlighting the urgent need to create defensive strategies to combat them. The review has also emphasised the absence of security measures incorporated in IoT devices, revealing the potential threat these devices are exposed to. Furthermore, the effectiveness of log management was reviewed, highlighting log analysis as an effective method for detecting cyber-attacks. As a result of these discoveries, an investigation was developed to evaluate the effectiveness of log analysis in detecting cyber-attacks, specifically denial of service. 4.1 Hardware requirements There were various hardware requirements that were considered while conducting the experiment. This experiment required a computer with at least a 5-core processor and 8GB of RAM. The experiment required these system specifications due to the various applications that were downloaded and utilised during the process. Alternatively, the experiment could have been conducted within a cyber lab, which will be able to host several virtual machines with IoT device operating systems. A strong Wi-Fi connection was also recommended as the virtual machines needed to be connected to the network to download applications and send data collected during the experiment. 4.2 Software requirements There were also various software requirements that were considered while conducting the experiment. The experiment was conducted within a virtual environment using the virtual box application. The virtual box was chosen for its speed, dependability, and its ability to host a diverse range of virtual machines. The Kali virtual machine was also utilised during this experiment. Kali is a popular Linux operating system utilised by many cybersecurity professionals as it contains several pre-installed penetration testing tools which can be used to access the security of a system.  In the context of this experiment, the attacker was on a Kali Linux machine and using an open-source penetration testing tool known as Metasploit. Metasploit is used to exploit security vulnerabilities within a target system for testing purposes. However, in this experiment, Metasploit was utilised to demonstrate a DoS attack. Metasploit is pre-installed in Kali Linux and therefore doesn’t require installation. In order to gain a more thorough understanding of the functionality of a DoS attack, it was advantageous to employ a different tool. Therefore, another example of a DoS attack demonstration was carried out by using the Hping3 tool. This tool is also preinstalled inside Kali Linux and can be utilised through the terminal.  it was also essential to monitor the network traffic from the victim’s perspective to record the attack and the packets sent during the attack. The victim, in this case, was also a Kali Linux virtual machine, this machine was chosen as the victim because most IoT devices utilise the Linux operating system. Therefore, this virtual machine was a great representation of an IoT device. The Tcpdump network packet analyser tool was used to record the network traffic during the attack. This tool is pre-installed within kali Linux but can be downloaded through the browser if not present. 4.3 Technology comparison section The objective of this section is to critically evaluate the advantages and drawbacks of the software applications and tools employed in the investigation. This section will also provide alternative tools that could be utilised for future experiments.   |  |  |  | | --- | --- | --- | | Tool Utilised | Advantages and Disadvantages | Alternatives | | Hping3 | Advantages:  - Versatility: The Hping3 tool is a versatile network tool that supports various protocols and can be utilised for multiple purposes, including firewall testing, remote operating system identification, and port scanning. (PhD, 2009).  -Capacity: This tool also offers a diagnostic capacity that allows for efficient network problem diagnosis. Additionally, its advanced features, such as the ability to add data payloads to packets and fragment attacks, make it highly skilled at evading detection by IDS and firewalls. | Nping is a tool that can be used on the command line to create network packets, analyse responses, and test networks.  Nping is part of the Nmap security scanner software; it is user-friendly and suitable for simple tasks like ping sweeps and port scanning. Nping is more efficient than hping3 as it can handle more complex protocols like TCP/IP, UDP, and ICMP. However, it doesn't have the advanced packet crafting abilities of hping3 and isn't as versatile for testing firewalls or identifying remote platforms (documentation, n.d.). | | Tcpdump | Advantages:  - the ability to record and analyse network traffic live.  - Allows the filtering of captured packets using various criteria.  - supports a command line interface which allows for efficient use of scripts and automation.  Disadvantages:  - The traffic recorded by Tcpdump could be hard to interpret for a beginner.  - It has no graphical user interface, which makes it less user-friendly (Buch, 2018). | Wireshark is another network packet analyser tool that can be utilised to review network traffic.  The graphical user interface offered by Wireshark could provide a more user-friendly experience in comparison to tcpdump's command-line interface, which might present an extra difficulty for beginners.  In addition, Wireshark provides advanced functionalities, such as the ability to save captured packets for further examination and to display packets based on a specific criteria. Overall, Wireshark provides numerous features for monitoring and analysing network traffic (Educba, 2021). | | Splunk | Advantages:  - Splunk is highly scalable and capable of processing large volumes of data from diverse sources.  - Splunk facilitates real-time data monitoring, allowing users to quickly identify issues as they arise and take corrective measures.  Disadvantages:  - The cost could be a concern with Splunk, especially for large enterprises that have a substantial amount of data to analyse.  - Learning skills in Splunk can be difficult for beginners, which could require a considerable amount of time and effort. | Elastic stack is another log management tool and could be used as an alternative to Splunk.  Splunk provides a licencing model based on stream processing quantity, which could result in significant costs for larger organisations. In contrast, the Elastic Stack is an open-source solution that is freely available for use.  Splunk also presents a comparatively inflexible architecture and limited adjustability. However, Elastic Stack demonstrates a high degree of flexibility and can be customised to meet the specific requirements of an organisation. | |
| ***Table 2: Comparison of tools utilised within the experiment*** **5. Implementation of Artefact:**5.1 Launching the DoS attack using Metasploit 1) Assuming that the attacker is on the same network as the victim, network settings for both parties were configured to use an internal network. Additionally, IP addresses were modified to ensure internal network usage. (figures 4 and 5).  A screenshot of a computer  Description automatically generated  A screenshot of a computer  Description automatically generatedFigure 4: Attacker virtual machine    Figure 5: Victim virtual machine  2) The ping command was then used to ensure that the connection between the two virtual machines has been established (figure 6)  Text  Description automatically generated  Figure 6: pinging virtual machines to ensure connection  3) To access root privileges, the sudo su command was used (figure 7). This was necessary since some Metasploit features require admin privileges. The msfconsole command was then used to launch Metasploit.  Diagram  Description automatically generated with medium confidence  Figure 7: Starting Metasploit on the attacker machine  4) The search command (figure 8) was used to find the directory of the synflood module in Metasploit, which is used to overload the target machine with TCP packets and cause a denial of service. The use command was then executed to access the synflood module and prepare for the DoS attack.  *Graphical user interface, text, website  Description automatically generated*  Figure 8: using synflood on attacker virtual machine  5) The set command (figure 9) configured variables for the attack, including RHOST (target machine's IP address) and NUM (number of packets to flood the target machine). The RPORT variable can specify a port to flood with packets. RHOSTS was set to the victim's IP address, and NUM was set to 1000 packets. Entering the exploit command would initiate the attack.  Text  Description automatically generated  Figure 9: Setting the target info and preparing the exploit command to launch the attack.  The Victim’s perspective  6) Using the sudo su command (figure 10), tcpdump was run with admin privileges to avoid errors while recording network traffic. The tcpdump –help command was used to see available options.  Text  Description automatically generated  Figure 10: using tcpdump to monitor the network traffic on the victim ip and storing it in a file.  7) Using the tcpdump command (figure 11) with various variables, network traffic was recorded on the victim's network. The -w variable specified the file path for recording, and the -i variable specified the interface (eth0 in this case). The dst variable specified the packet destination (victim's IP address) and to record traffic only for this IP. Once executed, tcpdump began recording network traffic.  Text  Description automatically generated  Figure 11: Listening on the ip address of the victim  8) On the Attacker’s side, the exploit command was now entered (figure 12), and the DoS attack was launched, and packets were sent towards the victim’s ip address.  Text  Description automatically generated  Figure 12: launching the attack from the attacker’s virtual machine  9) On the victim’s side, the network packets sent by the attacker were received and recorded (figure 13). These packets were then saved in the networktraffic.pcap file specified earlier.  Text  Description automatically generated  Figure 13: Result of the attack  10) The attacker launched various attacks (figure 14) targeting different ports of the victim's network using different IP addresses to demonstrate how attacks could occur from various machines. Since there were no resources available to launch another attack from a different machine, this method showed how multiple machines could attack the victim simultaneously, leading to a more dangerous attack like a DDOS attack.  Text  Description automatically generated  Figure 14: launching various attacks  File permissions set.  11) The victim's recorded network traffic was stored in the networktraffic.csv file. To send this file to the Splunk server, file permissions had to be changed to allow anyone to read, write, and execute the file. Using the chmod command (figure 15), ugo+rwx allowed anyone to access the networktraffic.csv file.    Figure 15: setting permissions for the file 5.2 Using Hping3 to launch DOS attack This attack demonstration's findings were also saved inside a file on the compromised machine. The following is an attempt to illustrate the methodology behind such an attack.  1) The virtual machines from earlier were reused to demonstrate Hping3's performance compared to the synflood module. The attacker and victim (figure 16 and 17) were on the same internal network with the same gateway. Hping3 is preinstalled in Kali Linux and requires admin access for some features. Using the sudo su command accessed the root directory. Tcpdump was re-enabled on the victim side (figure 17) to record network traffic for analysis using Splunk.  Graphical user interface, text  Description automatically generated  Figure 16: Attacker virtual machine  Text  Description automatically generated  Figure 17: Victim virtual machine  2) The hping3 tool was launched on the attacker's interface (figure 18) with specific variables. The -1 indicated the protocol for packet transmission during the attack (ICMP protocol in this case). The –flood variable instructed the tool to flood the target IP with packets as fast as possible, followed by the victim's IP address to specify the attack target.  Text  Description automatically generated  Figure 18: Attacker using icmp request to flood.  3) On the victim's side, the tcpdump command (figure 19) recorded network traffic caused by the attack. The -w attribute specified the file path to store network traffic, the -I attribute specified the monitored interface. Finally, the dst attribute recorded all packets with the victim's IP address as the destination.  A screenshot of a computer screen  Description automatically generated with medium confidence  Figure 19: using Tcpdump to record the network traffic  3) Another Hping3's feature is to mask the attacker's identity. The –rand-source variable (figure 20) was used in another attack, where multiple IP addresses made the attacker undetectable. This also represented a DDOS attack, where multiple machines flooded the victim with packets.  A screenshot of a computer  Description automatically generated with medium confidence  Figure 20: Attacker flooding the victim with random source  4) On the victim’s side, the results of the previous attack were also stored in a file using the tcpdump command (figure 21).  Text  Description automatically generated  Figure 21: Using Tcpdump to record save it to a file  5) Hping3 allowed the attacker to send various types of packets. Two more attacks (figures 22 and 23) used UDP and SCTP packets to flood the victim's network and demonstrate this feature.  Text  Description automatically generated  Figure 22: Attacker flooding with udp packets  Text  Description automatically generated  Figure 23: Attacker flooding with sctp packets  6) Results from all attacks were recorded in one file, which was converted to a CSV file for log analysis on the Splunk server. Since tcpdump was launched with root privileges, only the root user could access the CSV file, which was changed using the chmod command (figure 24). ugo+rwx allowed anyone to read, write, and execute the hping3traffic.csv file.    Figure 24: Setting file permissions 5.3 Monitoring Normal traffic 1) Monitoring and analysing normal traffic on the victim's machine would provide insight into network traffic changes during cyber-attacks like DDoS or DoS. The victim machine's network settings were on NAT, with DHCP allocating the IP address. The ifconfig command (figure 25) was used to check the victim machine's IP address.  A screenshot of a computer  Description automatically generated with medium confidence  Figure 25: Checking the IP address on the victim machine with Nat network  2) Tcpdump recorded and stored the victim machine's network traffic during normal functions, such as browsing or downloading. The tcpdump command (figure 26) saved network traffic in a file named normalTraffic.pcap, with the dst variable specifying the receiving machine's IP.  Text  Description automatically generated  Figure 26: using tcpdump to record the network traffic  3) The browser was loaded (figure 27) on the victim machine and one drive was used to download a random file for demonstration purposes. The network traffic caused by these actions was recorded and stored within the normalTraffic.pcap file.  Graphical user interface, application  Description automatically generated  Figure 27: Loading a browser on the victim machine  4) The normalTraffic.pcap file was converted to a CSV file for easy indexing by Splunk. The chmod command (figure 28) allowed anyone to read, write, and execute the normalTraffic.csv file.    Figure 28: setting file permissions  The files were ready to be sent to the Splunk server for log analysis. The Splunk universal forwarder was used to monitor the files directly from the victim machine. This tool enabled live monitoring of logs, which was beneficial for scenarios where the network traffic was updated. 5.4 Splunk enterprise and splunk forwarder 1) Another virtual machine was set up as a Splunk server and configured to be on the same network as the victim machine, to allow data transfer through the universal forwarder.  A screenshot of a computer  Description automatically generated with medium confidence  Figure 29: configuring the network settings for Splunk server  2) Splunk Enterprise can then be downloaded on the Splunk server virtual machine via the browser or using the wget command (figure 30 and 31).  A screenshot of a computer  Description automatically generated  Figure 30: Downloading Splunk from the browser  Text  Description automatically generated  Figure 31: installing Splunk with the wget command  3) Once splunk has been downloaded and extracted within the /opt directory. The cd command can be used to navigate to the /opt/splunk/bin directory, once inside the directory the ./splunk start command was used (figure 32) to start splunk enterprise.  Text  Description automatically generated  Figure 32: Starting Splunk server  4) The Splunk server was configured to listen from a specific port to enable the transmission of the network traffic data collected by the victim. This was done by navigating to the forwarding and receiving section and configuring the receiving data section inside the Splunk interface (figure 33).  Graphical user interface, text, application, email  Description automatically generated  Figure 33: Enabling port to receive data.  Additionally, the netstat -plnt command (figure 34) could also be used to ensure that the port is enabled and ready to receive data.    Figure 34: Checking port status  Starting Splunk forwarder  Text  Description automatically generated5) Splunk forwarder was installed on the victim machine to forward network traffic data collected during the cyber-attacks. It can be downloaded from the browser or by using the wget command. The sudo apt install command (figure 35) was used to install the forwarder.  Figure 35: installing Splunk forwarder on victim virtual machine  6) The splunk forwarder was started in the /opt/splunkforwarder/bin directory using the sudo ./splunk command with the –accept-license attribute (figure 36). The account for the splunk forwarder was created by specifying a username and password.  Text  Description automatically generated  Figure 36: Starting Splunk forwarder  Text  Description automatically generated7) The ./splunk add forward-server command (figure 37) added the splunk server to the forwarding server list by specifying its IP address and port, allowing the network traffic files to be sent.  Figure 37: Adding forwarding server (Splunk server)  Graphical user interface, text, chat or text message  Description automatically generated8) The ./splunk add monitor command (figure 38) was then used alongside the login credentials for splunk and the target file that requires monitoring. The file was now sent to the splunk server for log analysis.  Figure 38: Adding monitor of Synflood attack traffic  Add monitor command (figure 39) was also used to send the Hping3traffic.csv file and the normalTraffic.csv file (figure 40) to the Splunk server for log analysis along with the network traffic files collected during cyber-attacks.  Text  Description automatically generated  Figure 39: Adding monitor of Hping3 attack traffic  Text  Description automatically generated  Figure 40: Adding monitor of Normal traffic  9) Data summary (figure 41 and 42) on the Splunk server was updated with the Kali host and the source menu was also updated with the indexed network traffic files ready for analysis.  Graphical user interface, application  Description automatically generated  Figure 41: Updated data summary  Graphical user interface, application, Teams  Description automatically generated  Figure 42: Updated sources  10) Splunk was used to analyse the indexed data and identify trends. The analysis revealed security vulnerabilities with the victim’s system and characteristics of the attacker. Spl statements (figure 43) were utilized for this purpose.  Graphical user interface  Description automatically generated  Figure 43: visualisation using Splunk’s search and report feature  11) Visualizations created using spl statements can be grouped and displayed in a user-friendly dashboard (figure 44) to highlight important trends. This is especially useful for victims who may not have experience interpreting log data.  Graphical user interface, text, application, email  Description automatically generated  Figure 44: creating a dashboard using Splunk 5.5 Network Traffic analysis using Splunk This section explains how the network traffic collected during the experiment was analysed using Splunk enterprise. It covers the functionality of the Spl commands used for data analysis and the benefits of Splunk's visualisation features in identifying trends and security vulnerabilities within a system.  Some of the key commands used within the analysis:   * **Rex command**: the rex command is used to extract fields from data by searching for specific patterns provided within the command such as digits, alphabets and even special characters. * **Iplocation**: this command was used to pinpoint the location of specified ip addresses. * **Count by**: this command was used to count events by a specified fields and visualise the results. * **Rename**: this command was used to change the labels of specific graphs to provide a more user-friendly visualisation. * **Top**: this command was used to display the top results within a specified field. * **Timechart**: this command was used to display events of a specified field over time. * **Sort**: this command was used to sort the result from biggest to smallest  5.6 Analysis of the Network traffic from the synflood attack Most vulnerable port  source="networktraffic.csv" sourcetype="csv" | rex ">\s\s(?<port>\w+)\s" | chart count by port | rename count as "Packets sent"  The source and sourcetype attributes used at the start of this command specify the filename and its type, in this case, the networktraffic with the source type csv was being analysed. The | pipe command then allowed for further modification this command, which was then followed by a rex command. The rex command was used to extract port numbers and store them in a field named port. The chart count by command was then used to display the result and the rename command was used to make the visualisation more user-friendly.  Ip used to attack.  source="networktraffic.csv"sourcetype="csv"|rex"Tell\s(?<attacker\_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})" | stats count by attacker\_ip | rename count as "Attack attempts"  This spl command shows all the ip addresses used during the attack. The rex command extracts the ip addresses from the data and stores them in a field. The stats count by command displays the results and the rename command makes the visualisation user-friendly.  ip location of each attacker\_ip  index=\_\* OR index=\* sourcetype=csv | iplocation source\_ip | geostats count by Country  This SPL command displays the location of the attackers' IP addresses by using the iplocation command to determine the location and the geostats count by command to display the results by country.  Packets sent by each source ip  index=\_\* OR index=\* sourcetype=csv Protocol=TCP OR Protocol="ARP" | chart count by source\_ip | rename count as "Packets sent"  This spl command shows TCP and ARP events by each source IP. The chart count by command is used to display the events for each source IP, and the rename command is used to improve the visualization.  Synflood Dashboard  Graphical user interface, chart  Description automatically generated  Figure 45: Synflood Dashboard  Analysis for the Network traffic from hping3 attack  Types of packets received  index=\_\* OR index=\* sourcetype=csv | stats count by Protocol | rename count as "Total packets"  This Spl command was used to display all packet types in the network traffic. Stats count by command was used to count all events by the protocol field and rename command used to make the visualisation more user-friendly.  Ip addresses used during attacks  index=\_\* OR index=\* sourcetype=csv| top limit=20 source\_ip  This spl command was used to display the ip addresses used the most during the DoS attack. The top command was then used to display the top results by the source\_ip field.  Potential attacks over time  index=\_\* OR index=\* sourcetype=csv | timechart count | rename count as "Potential attack"  This spl command shows network traffic over time using the timechart command to display events occurring over time, and then the rename command was used to improve the visualisation.  Most vulnerable port  index=\_\* OR index=\* sourcetype=csv | rex ">\s\s(?<port>\d+)\s"| chart count by port | sort -count | rename count as "Packet received"  This Spl command showed the most attacked port. The rex command extracts the port number, which is then displayed using the chart count by command. The results are sorted and renamed for user-friendliness.  Hping3 Dashboard  Graphical user interface, chart  Description automatically generated  Figure 46: Hping3 Dashboard  Analysis for normal traffic  Types of packets being received.  source="normalTraffic.csv" sourcetype="csv" | chart count by Protocol | rename count as "Total packets"  This Splunk command was used to display different packet types in network traffic. Chart count by command used to display results with a user-friendly visualization using rename command.  Ip addresses sending packets.  source="normalTraffic.csv" sourcetype="csv" | stats count by source\_ip  This spl command was used to count all the event by each ip address. The stats count by command was used to count all the events by the source\_ip field.  Network traffic over time  source="normalTraffic.csv" sourcetype="csv"| top limit=20 Time showperc=f | rename count as "Network traffic"  This spl command showed top events by time using the top command and removing percentages using showperc=f. The results are made more user-friendly using the rename command.  Packets received by each port  source="normalTraffic.csv" sourcetype="csv" | rex ">\s\s(?<port>\d+)"| chart count by port | rename count as "Packet received"  This spl command displays packets received by each port of the victim. The rex command extracts the port number and stores it in the port field. The chart count by command counts events by the port field and the rename command makes the visualization user-friendly.  Normal Traffic Dashboard  Graphical user interface, application, Teams  Description automatically generated  Figure 47: Normal traffic Dashboard |
| **6. Comparison of Results**  **The results analysed through log analysis revealed interesting trends within the network traffic. The visualisation also highlighted the characteristics of DoS attacks and how they can be detected through monitoring the network traffic.**    Figure 48: Hping3 network traffic visualisation  **One of the visualisations displayed through the log analysis of the hping3 network traffic file (figure 48) demonstrates potential attacks over time. The graph highlights the network packets received by the victim over time, the spike of data received around 8 am indicates a potential denial of service as nearly 200 thousand packets were sent to the victim in an attempt to overload the machine. In contrast, a visualisation from the normal network traffic analysis (figure 49) demonstrates the normal network packets received while using the victim machine. The number of packets received is also significantly lower, with no significant spike in packets received, indicating normal network behaviour. This comparison offers valuable information as it shows the difference between normal network traffic and network traffic during an attack. The information can also be used to grasp a better understanding of a DoS attack.**    Figure 49: normal network traffic visualisation    Figure 50: Synflood vulnerable port visualisation  **Another visualisation created during the analysis of the Synflood attack network traffic revealed the packets received by each port. The results (figure 50) revealed that port 80 was flooded with packets during the attack. In contrast, the normal network traffic behaviour (Figure 51) highlighted the use of various ports, which means several applications were being used, and network traffic was being divided among ports. This comparison highlights the presence of an attack attempt where one port is being flooded with traffic to cause a denial of service. This information can be used by a security team to implement security strategies such as authentication to protect a vulnerable port from future attacks.**    Figure 51: ports used during normal network traffic.  **The comparison of different types of network traffic behaviour revealed significant patterns and characteristics of DoS attacks. The section also offered a more comprehensive understanding of how normal network traffic is disrupted during a denial-of-service attack. The knowledge gained through the log analysis could be used by security teams to understand various attack vectors and security vulnerabilities within the system, resulting in the creation of defence mechanisms to combat these vulnerabilities.** **7. Critical Review and Conclusion**  Centralising Log data for log analysis is stated to be an effective method for detecting early symptoms of DDOS attacks (ChaosSearch, 2021). The Introduction and background section of the study included a detailed explanation regarding the concepts covered during the investigation and the evolution of IoT devices. The Literature review includes thorough research regarding all aspects of the project, such as log management, IoT security etc. The review also investigated alternative methods and opposing perspectives to demonstrate a comprehensive understanding of the subject. The project covered all the methods and techniques used during the implementation of the artefact and provided a list of alternative tools that could be utilised to replicate the artefact.  The project was able to meet the set requirements and provide valuable results which could be used for future investigation. The investigation involved a demonstration of a DoS attack within a virtual environment which was successfully performed while considering all the ethical concerns discussed. The investigation was also able to provide valuable visualisations of the data and a comparison section which offered a detailed understanding of the impact of a DoS attack on a machine and how it differentiates from normal network traffic. The findings of this investigation can be used by researchers and security teams to understand DoS and DDoS attacks and potentially develop defence mechanisms to combat them. 7.1 Study limitation This was a small-scale investigation which was conducted within a virtual environment. Therefore, the investigation was unable to demonstrate a cyber-attack within a real-life scenario. The lack of computational power also restricted the use of more than two virtual machines, this limitation restricted the demonstration of Distributed denial of service attack where multiple machines would flood the victim with network traffic. The study was also limited to one network packet analyser tool called Tcpdump, which provided difficulty as Tcpdump’s output is harder to interpret. If this experiment was repeated, then these aspects would need to be considered, and any adjustments made to address these would lead to an improved outcome. 7.2 Future work The project could also be further expanded through the investigation of different tools and techniques, such as IDS. The importance of an intrusion detection system in securing network traffic cannot be disregarded. Through the utilisation of pattern correlation, an IDS is capable of detecting potential threats by analysing traffic patterns and cross-referencing them with known cyber-attacks. Through the use of this methodology, an intrusion prevention system can identify and avert cyberattacks by detecting any malicious behaviour that could indicate a security breach (Griffin, 2021). An investigation comparing the effectiveness of log analysis and IDS could offer valuable insight into developing the most effective detection system to prevent cyber-attacks. |
| **8. Bibliography** Anastopoulos, V. & Katsikas, S. (2017) A structured methodology for deploying log management in WANs. Journal of information security and applications. 34, 120–132. doi:10.1016/j.jisa.2017.02.004.  Avsystem (2019). What is Internet of Things (IoT)? Everything you Need to Know. [online] www.avsystem.com. Available at: https://www.avsystem.com/blog/what-is-internet-of-things-explanation/ [Accessed 14 Feb. 2023].  Bouazzaoui, S. (2020). *IoT-Related DDoS Ethical Issues: A System of - ProQuest*. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2018926334?pq-origsite=gscholar&fromopenview=true [Accessed 11 Apr. 2023].  Buch, R.R. (2018). *Network Sniffers and Tools in Cyber Security*. [online] www.jetir.org. Available at: https://www.jetir.org/view?paper=JETIR1811392 [Accessed 18 Apr. 2023].  ChaosSearch, T.H., Founder, CTO & Chief Scientist (2021). *How to Mitigate DDOS Attacks with Log Analytics | ChaosSearch*. [online] www.chaossearch.io. Available at: https://www.chaossearch.io/blog/how-to-mitigate-dDoS-attacks [Accessed 19 Apr. 2023].  Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C. & Faruki, P. (2019) Network Intrusion Detection for IoT Security Based on Learning Techniques. IEEE Communications surveys and tutorials. 21 (3), 2671–2701. doi:10.1109/COMST.2019.2896380.  Crowdstrike (2020). *Botnets: Definition, Types, How They Work | CrowdStrike*. [online] crowdstrike.com. Available at: https://www.crowdstrike.com/cybersecurity-101/botnets/ [Accessed 13 Apr. 2023].  documentation, nmap (n.d.). *Chapter 18. Nping Reference Guide | Nmap Network Scanning*. [online] nmap.org. Available at: https://nmap.org/book/nping-man.html [Accessed 18 Apr. 2023].  Educba (2021). *Tcpdump vs Wireshark | Key Differences Between Tcpdump vs Wireshark*. [online] EDUCBA. Available at: https://www.educba.com/tcpdump-vs-wireshark/ [Accessed 18 Apr. 2023].  E. Bertino and N. Islam, "Botnets and Internet of Things Security," in Computer, vol. 50, no. 2, pp. 76-79, Feb. 2017, doi: 10.1109/MC.2017.62.  Futures, C. (2016). *IoT Past and Present: The History of IoT, and Where It’s Headed Today*. [online] Channel Futures. Available at: https://www.channelfutures.com/msp-501/iot-past-and-present-the-history-of-iot-and-where-its-headed-today [Accessed 14 Feb. 2023].  Ghoshal, A. (2021). *Kali Linux Metasploit | How Kali Linux Metasploit Works?* [online] EDUCBA. Available at: https://www.educba.com/kali-linux-metasploit/ [Accessed 18 Apr. 2023].  Griffin, J. (2021). *What Is an Intrusion Detection System (IDS)? Definition and Software*. [online] SolarWinds. Available at: https://logicalread.com/intrusion-detection-system/ [Accessed 19 Apr. 2023].  Gills, A. (2022). *What is IoT (Internet of Things) and How Does it Work? - Definition from TechTarget.com*. [online] IoT Agenda. Available at: https://www.techtarget.com/whatis/definition/IoT-OS-Internet-of-Things-operating-system [Accessed 29 Jan. 2023].  Gupta, b. (2021). *How IoT is Making DDoS Attacks More Dangerous?* [online] Insights2Techinfo. Available at: https://insights2techinfo.com/how-iot-is-making-dDoS-attacks-more-dangerous/ [Accessed 10 Apr. 2023].  Hoory, L. and Bottorff, C. (2021). *What Is Waterfall Methodology And How Do I Use It?* [online] Forbes Advisor. Available at: https://www.forbes.com/advisor/business/what-is-waterfall-methodology/ [Accessed 11 Apr. 2023].  IBM (n.d.). *What is a cyber attack? | IBM*. [online] www.ibm.com. Available at: https://www.ibm.com/uk-en/topics/cyber-attack [Accessed 13 Apr. 2023].  Javaid, M. and Khan, I.H. (2021). Internet of Things (IoT) enabled healthcare helps to take the challenges of COVID-19 Pandemic. *Journal of Oral Biology and Craniofacial Research*, [online] 11(2), pp.209–214. doi:https://doi.org/10.1016/j.jobcr.2021.01.015.  PhD, J.S. (2009). *At the Hop» Linux Magazine*. [online] Linux Magazine. Available at: https://www.linux-magazine.com/Issues/2009/99/Hping [Accessed 18 Apr. 2023].  Kolias, C., Kambourakis, G., Stavrou, A. & Voas, J. (2017) DDoS in the IoT: Mirai and Other Botnets. Computer (Long Beach, Calif.). 50 (7), 80–84. doi:10.1109/MC.2017.201.  Miranskyy, A., Hamou-Lhadj, A., Cialini, E. & Larsson, A. (2016) Operational-Log Analysis for Big Data Systems: Challenges and Solutions. IEEE software. 33 (2), 52–59. doi:10.1109/MS.2016.33.  Security Centre, C. (2021). *Logging and protective monitoring*. [online] www.ncsc.gov.uk. Available at: https://www.ncsc.gov.uk/collection/device-security-guidance/managing-deployed-devices/logging-and-protective-monitoring [Accessed 12 Apr. 2023].  Security, L. (2023). *Metasploit Framework alternatives*. [online] Linux Security Expert. Available at: https://linuxsecurity.expert/tools/metasploit/alternatives/ [Accessed 18 Apr. 2023].  Shandilya, S.K., Ganguli, C., Izonin, I. and Nagar, Prof.A.K. (2023). Cyber attack evaluation dataset for deep packet inspection and analysis. *Data in Brief*, 46, p.108771. doi:https://doi.org/10.1016/j.dib.2022.108771.  SimonIoT (2020). *Rise of IoT: The History of the Internet of Things*. [online] SIMON IoT. Available at: https://www.simoniot.com/history-of-iot/ [Accessed 14 Feb. 2023].  Stelmashchuk, V. (2020). *Privacy and data protection measures for IoT; GDPR for IoT; Internet of Things and General Data Protection Regulation.* [online] Legal IT group. Available at: https://legalitgroup.com/en/gdpr-and-internet-of-things-iot/ [Accessed 15 Feb. 2023].  strike (2022). *What is IoT Security? Tips To Secure IoT Devices | CrowdStrike*. [online] crowdstrike.com. Available at: https://www.crowdstrike.com/cybersecurity-101/internet-of-things-iot-security/ [Accessed 29 Jan. 2023].  Tools, T. (2020). How Can You Prevent DDOS Attacks With Log Analysis. [online] TekTools. Available at: https://www.tek-tools.com/apm/detect-dDoS-attack-with-log-analysis [Accessed 11 Apr. 2023].  Usenix Association (2005). *Proceedings of the Second Workshop on Real Large Distributed Systems : December 13, 2005, San Francisco, CA, USA.* [online] Berkeley, Ca: Usenix Association. Available at: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/xu-lei. |