UNIVERSITY OF TURKU

Improving internal vulnerability scanning and optimal positioning of the vulnerability scanner in the internal network

Cyber Security

Master's Degree Programme in Information and Communication Technology

Department of Computing, Faculty of Technology

Master of Science in Technology Thesis

Author:

Ali Aqeel Zafar

Supervisors:

Maciej Wlodarczyk (Sanoma, Finland)

Dr. Seppo Virtanen (University of Turku, Finland)

Petri Sainio (University of Turku, Finland)

Dr. Peter Ligeti (Eotvos Lorand University, Hungary)

May 2023

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
University of Turku

Subject: Cyber Security
Programme: Master's Degree Programme in Information and Communication Technology
Author: Ali Aqeel Zafar
Title: Improving internal vulnerability scanning and optimal positioning of the vulnerability scanner in the internal network
Number of pages: 53 pages,
Date: May 2023

The art of vulnerability scanning is an integral part of any organization's internal network security, and it cannot be underestimated. It is vital to use a dependable vulnerability scanner and carefully select the most appropriate one for the task. This thesis seeks to gain a profound understanding of Sanoma Media's internal network and subsequently enhance its vulnerability scanning capabilities by first comprehending the different Tenable products. After acquiring a firm understanding of the various products, the Nessus Scanner was chosen based on Sanoma's business requirements. With the scanner in hand, the optimal location for it had to be carefully determined. To achieve this, several scenarios were developed, and a combination of factors from the business, technical, and financial perspectives were used to select the most effective scenario for implementation within the internal network.

The implementation of the selected scenario involved meticulous setup of the scanner, from both a hardware and software perspective. This thesis also presents an analysis of the Host Discovery Scan and Basic Network Scan results, alongside a security analysis of the Basic Network Scan. Furthermore, it offers a detailed explanation of the selected scenario, including the parameters that were carefully determined before the implementation process commenced.

Finally, the thesis outlines future work that needs to be undertaken, including the challenges that were encountered during the practical portion of the study.

Keywords: Network Scanning, Internal Network, Security Scanning Tool-Tenable Nessus, Vulnerability Scanning, Other Tenable Products (Tenable.io, Nessus Agent, Nessus Network Monitor)

Table of contents

1 Introduction

Digitalization has become a widespread phenomenon across all sectors, leading to numerous benefits for individuals and organizations alike. Notably, this shift towards digitalization has resulted in a constant stream of innovative ideas and solutions. Through the adoption of digital technologies, companies have been able to achieve significant growth and success both locally and globally. However, digitalization has also brought with it several significant challenges, such as the increasing prevalence of cyber-attacks. These attacks can severely impact a company's normal operations and compromise the digital services they provide or rely on. Consequently, companies must continually monitor and implement security measures to safeguard against these threats. In cases where organizations lack the expertise or resources to fight cyber-attacks on their own, they may choose to outsource their security services to third-party providers.

Alternatively, some companies may implement in-house security solutions to keep their infrastructure secure and protect sensitive information from third-party entities. One effective method for achieving this level of protection is through vulnerability scanning. By regularly conducting vulnerability scans, companies can proactively identify and address potential security threats before they can cause significant damage.

1.1 General introduction to the topic

A vulnerability can be defined as a weakness or flaw within a system, software, or network that can be taken advantage of by attackers. The exploitation of vulnerabilities can lead to a variety of security breaches, including data theft, network intrusion, and other cyber-attacks. To combat these threats, the process of vulnerability scanning is utilized, which involves the identification of potential weaknesses within a network, system, or software.

Vulnerability scanning is an essential component of the overall Vulnerability Management Process, which is designed to identify, evaluate, and treat security vulnerabilities in systems or software. By conducting regular scans, organizations can proactively identify and address potential security threats, ensuring that their systems and software are adequately protected against attacks.

The focus of this thesis is to improve the internal vulnerability scanning process within the Sanoma organization, with the ultimate goal of optimizing the position of the vulnerability

scanner within the internal network. This involves identifying and evaluating current vulnerabilities within the network, as well as determining the most effective placement of the vulnerability scanner to ensure comprehensive coverage and efficient scanning. By implementing an effective internal vulnerability scanning process, Sanoma can improve its overall security posture and reduce the risk of cyber-attacks and data breaches.

1.2 About Sanoma

Sanoma Media Finland is a prominent media company that leads the way in digital media transformation. The company's primary focus is on delivering high-quality media content to its audience. Among the many publications produced by Sanoma Media are Helsingin Sanomat, a daily newspaper, Ilta Sanomat, a digital and mobile news outlet, and a range of local and regional newspapers and magazines including Aamulehti and Satakunnan Kansa. Founded in 1998, Sanoma Media has become a major player in the Finnish media landscape, generating large net sales.

1.3 Motivation behind the topic

In a big organization that has various internal services and IT assets, such as servers, databases, and network devices, it's possible for these assets to have vulnerabilities for a variety of reasons, such as misconfigurations, unpatched software, and compromised credentials. As a result, it's possible for these vulnerabilities to be exploited, leading to a disruption in the normal operation of these assets. To prevent this, it's crucial to identify and address these vulnerabilities before they can be discovered and taken advantage of by cyber attackers. These vulnerabilities can take many forms and may exist in multiple locations, so it's important to properly address and mitigate them.

The initial action to address this problem is to conduct a vulnerability scan of all crucial components. This scan enables the organization to detect vulnerabilities and implement the necessary steps to address them. By identifying these vulnerabilities, the organization can efficiently target such susceptible components, adjust its security protocols, and implement the appropriate security measures to safeguard itself from potential cyber-attacks.

The main objective of this thesis is to assist Sanoma Media in conducting vulnerability scanning of their internal network by utilizing various Tenable products, such as Nessus Scanner. These products will be deployed within the organization's internal network, allowing

for vulnerability scanning and analysis to be conducted. By doing so, Sanoma Media can take appropriate measures to address the vulnerabilities identified within their network. The positioning of Tenable products, such as Nessus Scanner, within the internal network is crucial to minimize any negative impact on network performance. This thesis includes a security analysis comparing practical and theoretical vulnerabilities, as well as suggestions for expediting vulnerability scanning with Tenable products like Nessus Scanner.

1.4 The overview of the upcoming chapters

The structure of the thesis is as follows:

Chapter 2 comprehensive overview of vulnerability, vulnerability scanning, types of vulnerability scanning and how attackers can exploit vulnerabilities. The chapter emphasizes the significance of vulnerability scanning for companies. Moreover, the chapter discusses potential vulnerabilities that could exist in office devices and network devices such as switches based on theoretical concepts.

Chapter 3 introduces Tenable, a company specializing in vulnerability scanning, and its products which will be utilized in this thesis. Chapter 4 of this thesis presents a review of literature on previous research in the area of vulnerability scanning and the tools used for this purpose.

Chapter 5 gives an overview of the internal network of Sanoma Media is presented. It also explains the reasoning behind the experiment and why Nessus Scanner was selected over Nessus Agent. It goes on to describe the various deployment scenarios for Nessus Scanner in the internal network, as well as the necessary requirements for deploying it, such as general requirements, port configurations, antivirus software, and firewall configurations. Furthermore, the chapter provides reasoning behind the selection of a specific scenario for deploying Nessus Scanner within the internal network of Sanoma Media, taking into consideration various factors such as the scope of the project and the network architecture. Lastly, the chapter concludes with a list of parameters that will be analysed in the subsequent chapters of the thesis, providing a roadmap for the rest of the research. Overall, the chapter provides a comprehensive understanding of the decision-making process and the requirements for deploying Nessus Scanner within Sanoma Media's internal network.

Chapter 6 is dedicated to showcasing the practical implementation of the selected scenario. It describes the deployment of Nessus Scanner in a server and outlines the technical

specifications of the Linux based server. The chapter also covers the installation process of Nessus Scanner in a Linux-based operating system. Furthermore, it provides a subsection on how to link Nessus Scanner with Tenable.io. The chapter concludes by providing a step-by-step guide with accompanying images on how to conduct a Host Discovery Scan and Basic Network Scan.

Chapter 7 provides an in-depth explanation of the outcomes produced by Host Discovery Scan and Basic Network Scan. Additionally, it includes a security assessment process of the outcomes generated Basic Network Scan. The security analysis involves contrasting the vulnerabilities discovered in theory with examples of those identified in real-world. At last the chapter also discusses about the ways to increase the scan speed.

In Chapter 8, various potential avenues for future work are explored to suggest ways to further improve the process. The chapter also delves into the difficulties and issues that were encountered during the practical implementation.

2 Vulnerabilities in computing system

Broadly speaking, the field of cybersecurity involves individuals and organizations working to mitigate the likelihood of a cyber-attack. To accomplish this, individuals and organizations employ cybersecurity measures to safeguard their digital information, devices, and online services. Vulnerabilities in software or devices are often the cause of the risks that cybersecurity seeks to minimize.

2.1 Introduction to Vulnerability

A vulnerability refers to a weakness or defect that exists in an information system, internal controls, security procedures, or implementation, which can be manipulated by an adversary. Various types of vulnerabilities can be taken advantage of using different methods, such as buffer overflows, remote code execution, SQL injection, and so on. The initial step to execute a cyber-attack involves recognizing the vulnerability of a network, software, or system in order to disrupt an organization's assets. Vulnerabilities are defined due two reasons. The first reason is that, software development is a complex process that involves creating software according to client needs while adhering to strict deadlines. In order to meet these deadlines, software developers may have to make compromises, such as skipping certain testing or quality assurance steps. This can result in software bugs that go undetected, ultimately leading to the emergence of vulnerabilities [1].

The second reason is that, complex systems like networks and systems designed by organizations require a high level of technical expertise to set up and maintain. Misconfigurations in these systems can occur due to a variety of reasons such as human error, insufficient training, or lack of understanding of the system's security requirements. Misconfigurations may result in vulnerabilities that can be exploited by attackers, potentially leading to unauthorized access or other security incidents [1].

To identify vulnerabilities, two perspectives are considered: hardware and software. When it comes to hardware, vulnerabilities can be identified through either proprietary vulnerability scanning tools or in-built vulnerability scanning tools that are designed by individuals or organizations. Regarding software, there are several techniques available to identify vulnerabilities. These techniques include Dynamic Analysis, Static Analysis, Code Review and Reverse Engineering, and Fuzzing. Dynamic Analysis is a technique used to identify vulnerabilities in software by executing the software or its components and monitoring the

execution simultaneously. During the analysis, various aspects of the software being executed are checked, such as whether it creates files with write permissions or if it is able to access previously freed memory [1]. The following technique is Static Analysis, which examines whether the software code contains poor design patterns or not. The third technique used to identify vulnerabilities is called Code Review and Reverse Engineering, which involves a thorough examination of all code and binary files. The final technique is called Fuzzing, where a target program uses a normal document or webpage, or any data that it can access. These files are randomly modified, and the program is then opened using them. If the program crashes, a security vulnerability may have been discovered. This process can be repeated until all possible vulnerabilities are found [1].

The process of publishing vulnerabilities is a thorough one. When a security researcher or an organization discovers a vulnerability, they usually notify the Common Vulnerabilities and Exposures (CVE) organization to obtain an ID for the vulnerability. Then, the product owner is given a timeframe of around two to three months to fix the vulnerability and inform other relevant vendors about the solution. Once the CVE organization receives the vulnerability information and possible exploits, it forwards it to the National Vulnerability Database (NVD) for analysis. Finally, the CVE or vulnerability is published on the NVD website. However, if someone can provide a detailed and valid reason for its removal, the vulnerability can be taken down from the website [2].

There are different types of vulnerabilities like:-

- Unpatched Software [3][4]: If a software has an unpatched security vulnerability, it creates an opportunity for an adversary to exploit a known security flaw and execute malicious code. The attacker will search the environment for unpatched security issues through enumeration and then launch a direct or indirect attack.

- Misconfiguration [3][5]: Misconfiguration is another type of security vulnerability that occurs when a system has been incorrectly configured. Misconfigurations can be exploited by adversaries, leading to a breach of an organization's network infrastructure. Misconfiguration occurs due to improper or incomplete implementation of security measures, or when system or security administrators neglect to change default configurations.

- Weak Credentials [3]: If a user sets weak passwords for their user account, an attacker can use brute force or dictionary attacks to obtain the desired credentials. Weak passwords are often a result of people choosing easily remembered credentials such as their birthdate, pet's name, or simple passwords like "abc123". Privilege escalation can occur through weak credentials, which enables the attacker to obtain root access and retrieve critical information from the system.

- Easy-to-phish-users [3]: Phishing targets non-technical security personnel. The attackers trick users into executing malicious code or giving away their credentials without their knowledge. These attacks can be carried out through emails or text messages, where the attacker impersonates a genuine company or person, such as a friend, relative, or a legitimate employee of the target's company.

- Trust Relationship [3]: An attacker can take advantage of trust configurations that are set up to enhance system accessibility, enabling them to infiltrate an organization's network. They can then move from one system to another, gradually compromising the security of the entire network.

- Compromised Credentials [3][6]: To gain unauthorized access to a system within an organization's network, an attacker can utilize compromised credentials. The attacker's plan would be to obtain passwords from channels that are either unencrypted or incorrectly encrypted between systems or from software or users that handle credentials in an unsecured manner.

- Malicious insider [3]: An insider refers to an individual such as an employee or a vendor who possesses access to crucial systems within an organization, and can potentially abuse their access by stealing important data or causing damage to the system.

- Missing or Poor Encryption schemes implemented [3][7]: When there is either no encryption or even poor encryption is being used an attacker would be able to intercept the communication between systems and will retrieve information. The attacker can intercept information, which is not encrypted or even poorly encrypted, and from it he or she can extract bits of critical and useful information. In addition, it impersonates itself to be a legitimate system and can disrupt the communication between the systems by injecting false and incorrect information.

- Zero-days and Unknown Methods [3][8]: This security vulnerability is unique to software and is currently only known to the attacker, and there is no available fix for it. The reason being that the flaw in the software has not been reported to the software vendor. The attacker takes advantage of this vulnerability by attempting to gain access to the organization's network environment through the software system, using zero-day attacks either directly or indirectly.

2.2 Common Vulnerabilities

There many vulnerabilities related to software and OS present in office devices, Network Devices and also with open ports which can be:

- Bug: A software bug is essentially an error in the code of a program, which can have severe consequences such as the theft of sensitive information or the failure of a system. When a software bug occurs, the program may behave unexpectedly, providing an opportunity for attackers to exploit the vulnerability and cause harm to both the software and the system on which it is running [9].

- Flaws in injection: The presence of flaws in an application's code injection process can create opportunities for attackers to insert malicious code into the system. This happens when the system receives a command or query with malicious data and executes it using the interpreter. As a result, attackers may gain unauthorized access to protected data and compromise the system's security [9].

- Buffer overflow: When the allocated memory space of a program is inundated with an excessive amount of data, it results in a buffer overflow vulnerability. This vulnerability can be exploited by attackers to overwrite the program's storage capacity and gain unauthorized access to the system [9].

- Security Misconfiguration: This issue can be caused by configurations that are either left incomplete or set to default values [9].

- Broken access control: When the restrictions of a user are broken, it can lead to numerous problems within the software. Adversaries can exploit this type of vulnerability and gain access to sensitive data and the system, as there are no restrictions in place to prevent unauthorized access [9].

- Insecure Deserialization: Attackers can use this vulnerability to carry out DDoS and injection attacks. Untrusted data is typically used as the main vector for these attacks to be successful [9].

- Remote Code Execution: The attacker can modify and execute the code on the device from a remote location, which may lead to an escalation of privileges [10].

- Privilege Escalation: The attacker's ultimate objective is to obtain complete user access to the system, with the end goal being gaining control over the administrator privileges [10].

- Denial of Service: The attacker overwhelms the system, causing exhaustion of its resources and bandwidth, thereby preventing legitimate requests from being processed [10].

- Memory Corruption: The adversary can collect sensitive data as a result of the negative impact on the device's memory [10].

- Overflow: The operating system cannot control the amount of code generated, which can lead to a system crash or data loss as a consequence [10].

- Cisco SSH Vulnerability Allows Device Reload: A vulnerability has been discovered in the Cisco IOS and Cisco IOS XE software, which serves as the standard operating system for switches and routers. The vulnerability is related to the implementation of SSH, which enables an attacker to reload a device that is affected due to mishandling of resources [11].

- Juniper Devices Vulnerable to OSPF DoS Attack: This vulnerability is present in Junos OS, the operating system used by Juniper network devices. It can be exploited by attackers to carry out a Denial-of-Service attack. The vulnerability affects the OSPF routing protocol, causing the peer interface to frequently fluctuate, leading to an increase in the memory consumption of the Routing Protocol Daemon, ultimately resulting in a Denial of Service attack [11].

- Port 20 and 21: These ports are used for File Transfer Protocol (FTP) and are vulnerable to various exploits. Passwords can be brute-forced by attackers to gain unauthorized access. In addition, the ports are susceptible to cross-site scripting and

directory traversal attacks. A directory traversal attack involves an attacker creating a malicious file with a crafted path. When the user accesses the malicious path, the malicious file executes a code which results in remote code execution, allowing the attacker to gain control of the system [12].

- Port 22: On this port SSH service is running and the attacker can exploit this by leaking the SSH keys or even by brute forcing the credentials[12].

- Port 23: The Telnet service typically runs on this port. An attacker can gain access to Telnet credentials by repeatedly guessing different combinations of usernames and passwords until they find a valid one. Once the attacker has obtained valid credentials, they can impersonate the authorized user [12].

- Port 25: Simple Message Transfer Protocol runs on port 25. The port can be vulnerable to spamming and spoofing [12].

- Port 137 and 139: These two ports are running the NETBIOS service over TCP protocol, which enables communication between different devices. However, they are vulnerable to brute force attacks targeting the credentials for the Server Message Block [12].

- Port 3389: This port is used for Remote Desktop service, which becomes vulnerable when an unauthorized user creates a malicious RDP server and connects to it via remote desktop. Once connected, the unauthorized user gains access to the file systems of all the victim users who are connected to the malicious RDP server [12].

2.3 Vulnerability Scanning

Vulnerability Scanning is a procedure used to discover security vulnerabilities in a network, network devices, and systems, as well as the software running on those systems. It is part of vulnerability management program. The primary objective of vulnerability scanning is to protect the organization by identifying potential security vulnerabilities in network devices, hosts, and software. The process aims to safeguard these assets and prevent situations where sensitive data can be exposed or breached. [13]. Vulnerability scanning is typically conducted using automated tools to identify potential security weaknesses in network devices, systems, and software, with the goal of safeguarding the organization and protecting sensitive data from exposure or breach. Many companies offer vulnerability scanning services, with some of

the most popular providers being Astra Pentest, Intruder, Symantec, Qualys, Rapid7, AlertLogic, and Tenable [14].

There are different types of vulnerability scanning. The types of vulnerability scanning are as follows:

- Host-based vulnerability scanning [15]: This process involves examining a network host to identify any weaknesses or vulnerabilities present. The primary objective of this type of scan is to detect potential vulnerabilities that may exist within the host. Through host-based vulnerability scanning, it is possible to gain insight into various aspects of the host, including the impact a vulnerability could have, the patch history of the host, any outdated patches that could lead to vulnerabilities, ways that an attacker might gain access to the host, and ways to address and minimize the risks associated with these vulnerabilities.

- Network-based scanning [15]: Network-based scanning is a critical type of vulnerability scanning that is commonly used to identify vulnerabilities within an organization's network. The process starts by identifying all the devices within the network infrastructure and examining their connections. The scanning tool then proceeds to scan the endpoint devices, analyzing their ports, services, software, applications, as well as any weak passwords or authentication errors that may exist. After the scan is complete, a report is generated which provides a list of the vulnerabilities that were detected and offers recommendations for addressing those vulnerabilities. This comprehensive approach helps to identify potential weaknesses in the network and minimize the risk of a successful cyber-attack.

- Database Vulnerability Scanning [15]: The main aim of this process is to identify vulnerabilities in a database application. This is achieved through the use of database scanners, which work in a manner similar to password cracking tools. The database scanner examines the internal configuration of a database to identify any potential vulnerabilities that could be exploited by attackers to compromise the database. The scanner considers various aspects of the database, such as unauthorized passwords, logon hours violations, account and role permissions, object ownership, remote server login, system table permissions, extended stored procedures, cross-database ownership chaining, authentication methods, login attacks, admin account security, password aging, trail auditing, configuration auditing, and the deliberate testing of

buffer overflows in usernames and database links. By analyzing these attributes, the database scanner can detect potential weaknesses in the database that could be exploited, and help users to take appropriate measures to address these vulnerabilities.

- Cloud Vulnerability Scanning [15]: Cloud vulnerability scanning is a technique employed as part of a cloud security strategy to detect vulnerabilities in a cloud deployment. The objective of this process is to enhance the overall security of the cloud infrastructure through monitoring and management. The scanning procedure is capable of detecting various types of vulnerabilities, including weak passwords, SQL injections, misconfigurations in servers, Cross-site scripting (XSS), CSRF bugs, and outdated or unpatched software. Any of these vulnerabilities can be exploited to compromise the cloud services, underscoring the significance of this scanning approach in cloud security management.

- Application Vulnerability Scanning [15]: Web and mobile application vulnerability scanning is a widely employed type of vulnerability scanning aimed at detecting vulnerabilities in web and mobile applications. Despite frequent updates from developers, some vulnerabilities may still exist due to human error or oversight. Moreover, web and mobile applications often incorporate third-party APIs, themes, and plugins that may contain exploitable vulnerabilities. To address these potential risks, web and mobile application vulnerability scanning is used to monitor the security status of these applications. The scan results are presented in a comprehensive report that includes information on the identified vulnerabilities, their impact score, risk analysis, and suggested mitigation solutions.

Prioritizing vulnerabilities is a crucial step in the vulnerability management process. Once vulnerabilities are detected through scanning, they need to be prioritized based on their severity, potential impact, and exploitability. This helps organizations focus their resources on addressing the most critical vulnerabilities first, reducing the risk of a successful attack.

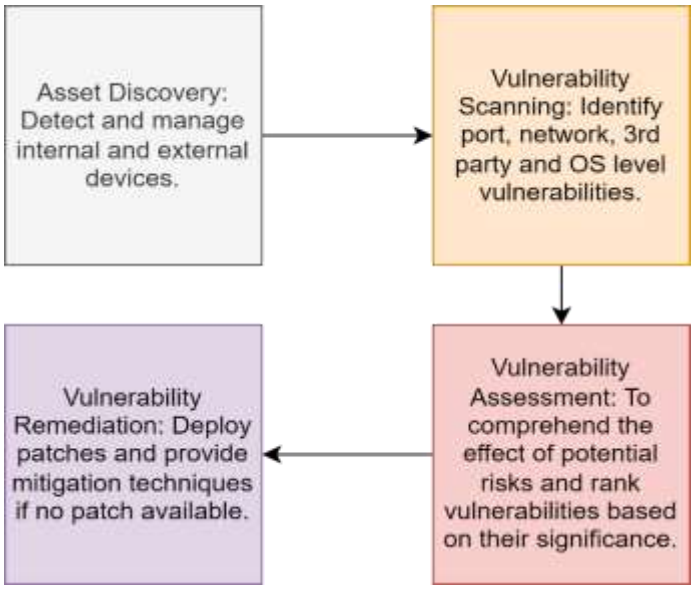

Figure 1. Diagram of the Workflow for the Vulnerability Management Process. This figure is self-drawn.

As it can be seen from Figure 1 that vulnerability management is a process used by organizations to identify, evaluate, treat, and report on vulnerabilities in their network, system, or software. The goal is to prioritize the threats and reduce the attack surface. Firstly, all the assets such as network devices, servers, endpoints are identified. Then, a vulnerability scan is performed on all these assets using a vulnerability-scanning tool. The tool then identifies all the vulnerabilities present in the assets and classifies them based on parameters such as CVSS score, impact score, age of the vulnerability, etc. Further, the vulnerability-scanning tool generates a vulnerability scan report that lists all the vulnerabilities found, which can be sorted according to the parameters from the previous step. Finally, this report is used to plan and mitigate the vulnerabilities appropriately. The vulnerability management process helps organizations to prioritize their threats and decrease the attack surface. Performing a vulnerability scan is necessary whenever vendors release patches, firmware, and updates. Subsequently, vulnerability scanning ensures that critical patches are not missed by the management process of a company if any changes are made.

2.4 Need for Vulnerability Scanning

One of the most important things for organizations is to regularly scan their network devices, systems, and software to keep up with the ever-evolving strategies of hackers and to ensure that their hardware and software remain up to date. Performing a vulnerability scan is necessary when vendors release patches, firmware, and updates. Subsequently, vulnerability scanning

ensures that critical patches are not missed by the management process of a company if any changes are made. Large organizations use multiple types of hardware, increasing the chances of misconfiguration. Vulnerability scanning helps identify these misconfigurations[16]. Vulnerability scanning can also be used to verify whether the service levels and systems are being protected in accordance with the contract signed between the organization and a third-party IT service provider. This helps ensure that the provider is fulfilling its obligations as per the agreement and provides an opportunity to address any discrepancies or areas of concern [16]. As customer demand and concerns about data protection continue to increase, organizations need to reassure their customers that their data is being safeguarded. One way to build this trust is by using vulnerability scanning to identify vulnerabilities and then taking steps to mitigate these vulnerabilities in order to protect customer data in a more sophisticated manner. Performing regular vulnerability assessments demonstrates to an organization's employees that the management and the organization as a whole are committed to ensuring cybersecurity [17].

3 Introduction to Tenable Tools

Established in 2002, Tenable is a company specializing in Exposure Management. It assists numerous organizations, both on-premises and in the cloud, in comprehending and alleviating their cybersecurity risks, spanning from IT to OT. More than 40,000 organizations worldwide rely on Tenable for guidance and assistance in comprehending and mitigating their cybersecurity risks across their potential attack surface. Tenable's products offer comprehensive and detailed visibility of all the attack surfaces within an organization. All services provided by Tenable are delivered through its range of products [18]. The products provided by Tenable are described in the following sections.

3.1 Nessus Scanner

This vulnerability scanner boasts a high level of accuracy, with a low incidence of false positives. Tenable has tested and demonstrated that it only produces 32 false positives in a million scans [19]. Nessus offers a vast range of vulnerability coverage at an affordable price point. As an open-source platform, it utilizes the Common Vulnerabilities and Exposure Architecture. Nessus can be integrated with other Tenable products such as Tenable.io and Tenable.sc. It is available in various versions, including Nessus Expert, Nessus Professional, Nessus Manager, and Nessus Agent [20]. Nessus offers several benefits, including its high accuracy and low incidence of false positive ratings. In addition to its accuracy and low false positive rating, Nessus also features a well-designed and user-friendly GUI. It offers pre-built scanning templates that can be readily used by its customers. Once vulnerabilities have been detected within an organization, Nessus also recommends solutions on how to mitigate those vulnerabilities from the network and systems that are part of the organizational network [21].

3.2 Nessus Agent

Nessus Agent is offered in both Tenable.io and Nessus Manager. It is a software module that is installed on hosts within an organization's network. The primary objective of the agent is to report on vulnerabilities within the host's internal infrastructure. Using Nessus Agent can make scanning the hosts of an organization easier because it does not require host credentials like the Nessus Scanner. The Nessus Agent is capable of scanning hosts that are offline as well [20]. Nessus Agent provides a convenient solution for conducting comprehensive vulnerability scans within large organizations, as it enables concurrent scanning without negatively impacting network performance.

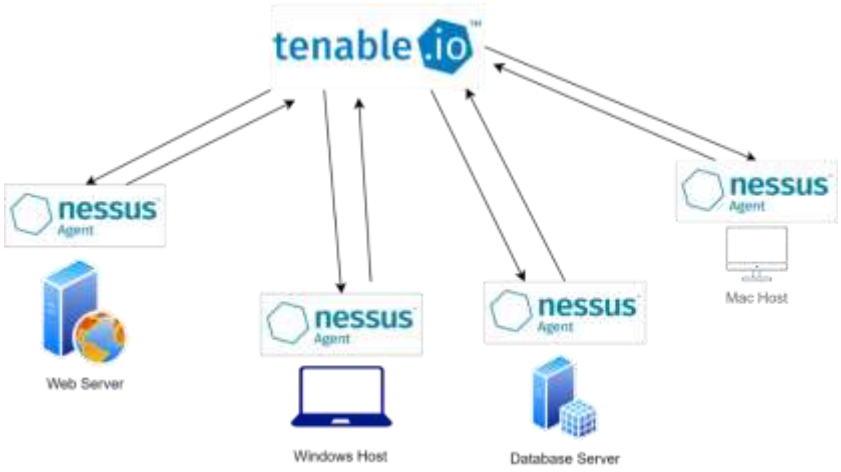

Figure 2. An instance of varied Nessus Agents deployed across distinct devices connected to Tenable.io. This figure is self-drawn.

As shown in Figure 2 agents are able to provide more coverage by scanning the host in which they are installed. The agents gather data about vulnerability compliance and system information from the host and then transmit it to either Tenable.io or Nessus Manager, depending on the organization's needs. Nessus Agent can be installed on various types of hosts, such as servers, laptops, and desktop computers, to collect information related to vulnerability compliance and system details [20].

3.3 Nessus Network Monitor

This tool is utilized to identify vulnerabilities and explore networks. The software offers real-time network profiling and keeps a watch on the security rules and policies of the organization. Nessus Network Monitor can be implemented on cloud platforms such as AWS, Google Cloud Platform, and Microsoft Azure. Afterwards, it can be installed on virtual machines like VMware ERSPAN, VMware ESXi, VMWare vSphere, and Microsoft Hyper V. Additionally, it can be installed on network visibility and traffic monitoring software such as Gigamon. Finally, it can also be installed on Docker [22].

3.4 Lumin

This tool is designed for viewing dashboards and is used to evaluate the risk of cyber exposure. It also allows for comparison of health and performance in terms of remediation with other organizations within the community. Lumin acquires vulnerability data from Tenable.io and Tenable.sc, and presents it in a visually appealing format. Lumin employs various scoring systems like Cyber Exposure Score, Vulnerability Priority Rating, Asset

Criticality Rating, Asset Exposure Score, Assessment Maturity Grade, and Remediation Maturity Grade to create its own measurement, known as the Lumin Metric. The Lumin Metric provides a consolidated view of scores from all the aforementioned metrics in a single table [23].

3.5 Tenable.sc

This is a tool used for managing vulnerabilities that is installed on a network within an organization. It helps to prioritize which vulnerability to mitigate first. There are many functionalities and features by this Vulnerability Management Software. Like it shows the result of the scan result in a detailed and elaborate way. Tenable.sc offers the feature of Vulnerability Analysis, which presents the vulnerabilities as a compilation of either Cumulative Vulnerabilities or Mitigated Vulnerabilities. The software tool is capable of analysing mobile devices, provided they are connected to a Mobile Device Management system that can transfer their information to Tenable.sc [24].

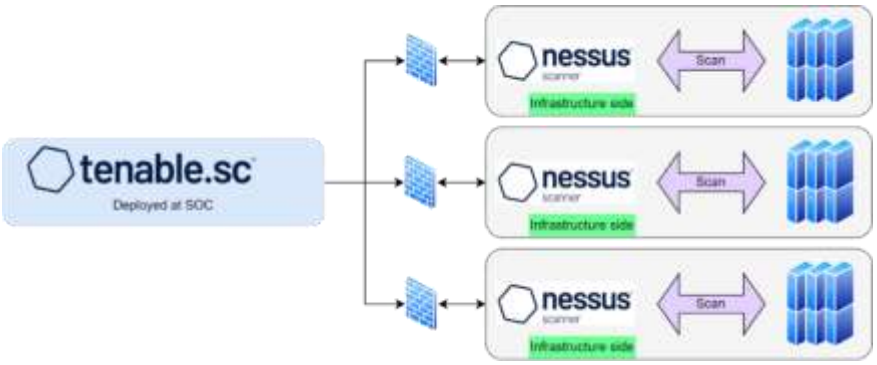

Figure 3. Diagram of the logical architecture of Tenable.sc with connection to Nessus Scanner. This figure is re-drawn [24]. (c) Tenable, used with permission.

As shown in Figure 3, Tenable.sc is installed in the Security Operation Centre (SOC), where Nessus Scanner or Nessus Agent conducts vulnerability scans on the target host and sends the collected data to Tenable.sc through the firewall. It is important to set up firewall rules that allow data traffic between Nessus Scanner and Tenable.sc.

3.6 Tenable.io Vulnerability Management

Tenable.io Vulnerability Management has an extensive database that contains details of over 73000 vulnerabilities. This database combines vulnerabilities gathered from Nessus Scanner, Nessus Agent, and other third-party scanners integrated with the software. The software provides a comprehensive overview of the vulnerabilities, based on their criticality, which is

assessed and measured through risk score metrics. This leads to the ability to rapidly evaluate risks and gain a clear understanding of which vulnerabilities should be addressed as a priority. Also, Tenable.io Vulnerability Management can be deployed on cloud [25].

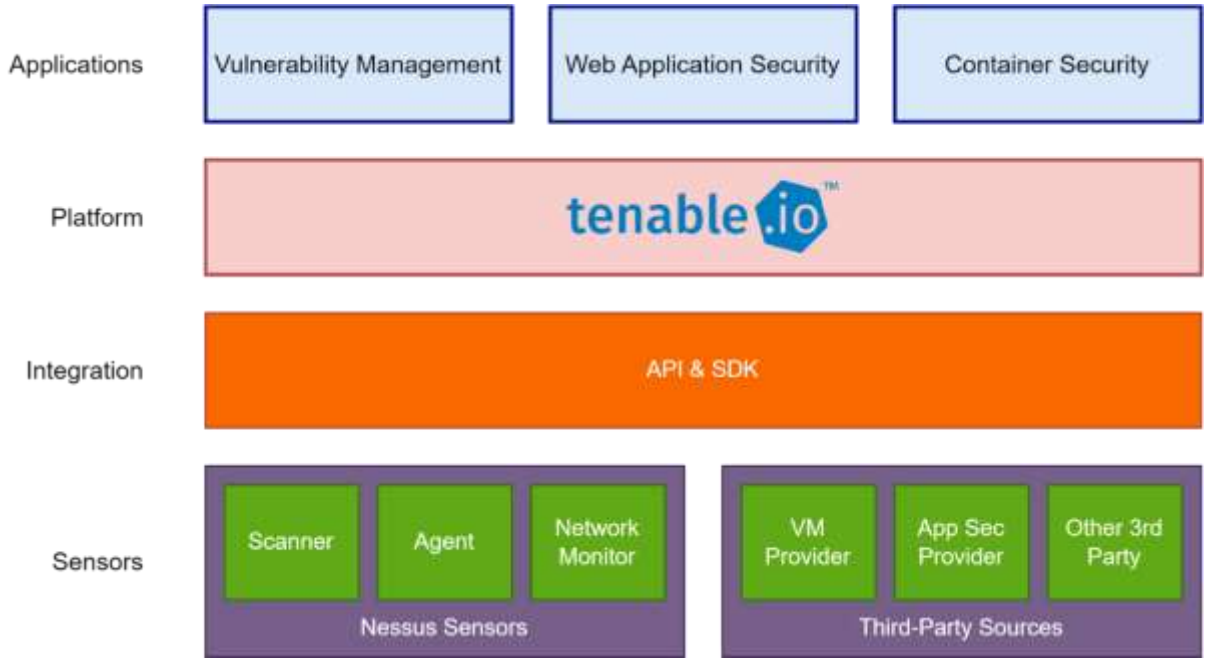

Figure 4. Diagram of the logical architecture of Tenable.io [25]. This figure is re-drawn. (c) Tenable, used with permission.

The above Figure 4 illustrates that Tenable.io Vulnerability Management is part of the Tenable.io platform, which also includes Web Application Security and Cloud Security tools for scanning organizational servers and cloud vulnerabilities. The platform is connected to various sensors such as Nessus Scanner, Nessus Agent, Nessus Network Monitor, and third-party sources through APIs and SDKs. The sensors conduct vulnerability scans on target devices, and the resulting information is transmitted to the Tenable.io platform through APIs. This information is then displayed in applications such as Tenable.io Vulnerability Management.

Tenable.io Vulnerability Management offers a variety of features, including the ability to share vulnerability findings with other users and Tenable-related applications. It also allows for the creation of rules, such as recast and accept rules. Tenable.io Vulnerability Management contains a record of all assets discovered by the Discovery Scan from sensors linked to it. It enables generating, editing, and designing custom reports, and can also transfer asset information to other users and Tenable-related applications.

4 Literature Review

Authors Bairwa and Mewara [26] in 2014 discussed a method of using vulnerability scanners for securing web applications. They selected popular scanners and proposed a detailed approach including manual testing, security audits and vulnerability scanning. In 2017, Al-Alami and Hadi [27] concentrated on scanning the vulnerabilities of IOT devices utilizing Shodan, a search engine, and scrutinized the effectiveness and efficiency of various vulnerability scanning tools. They restricted their focus to IOT devices because of the increasing usage of such devices in Jordan where they were located. In the subsequent year, Wang and Yang [28] evaluated open source vulnerability tools in their own virtual laboratory. They examined the effectiveness of the OpenVAS tool and the results it produced. Their study did not involve any Tenable products or actual organizational networks. Due to the small scale of their experiment, they did not analyze the optimal location for the OpenVAS vulnerability tool.

In 2017, author Kushe [29] conducted a comparative analysis of two software-based vulnerability scanning tools, Nessus and Retina. The analysis focused on three parameters: the tools' ability to identify vulnerabilities, their search capacity, and the time required to scan a device. The analysis was conducted on a single device, and did not consider vulnerability scanning of an organizational network. Additionally, since only one device was used, the optimal placement of the tools was not evaluated.

In 2018, Harell and Patton [30] conducted a study of nearly 300 higher education institutions to assess their vulnerability. They used Nessus and Burp for their analysis and found that many vulnerabilities lacked proper mitigation techniques. The authors generated vulnerability reports and devised mitigation techniques for each one. Although they focused on the institutions' networks for vulnerability scanning, they did not address ways to increase the efficiency of their tools or the optimal positioning of Nessus.

In 2018, Qasaimah and Shamlawi [31] conducted an evaluation of various web application scanners using black-box evaluation. They selected popular scanners such as Acunetix, WVS, Burp, NetSpark, and Nessus and compared their outputs with established security standards. In 2019, Aksu and Altuncu [32] conducted an analysis of the usability of OpenVAS. They detected the challenges that come with using the vulnerability scanning tool and proposed recommendations to enhance its functionality. Authors Amankash and Chen [33] conducted a

comparative analysis in 2020, which evaluated the efficiency of eight open source web vulnerability scanners used in large organizations. The experiment was conducted on web applications, and the authors analysed the performance of the scanners for detecting vulnerabilities. Their main focus was on web application vulnerability scanning rather than network vulnerability scanning. In the subsequent year, Wang and Yongbin [34] developed a network vulnerability scanning system for conducting Network Vulnerability Scan Tests. The system was tested in a small network with only a few devices. In 2022, Subhangani and Chaudhary [35] performed an analysis on open-source vulnerability scanning tools, evaluating their ability to assess vulnerabilities, identify security risks, and generate vulnerability reports. The tools they used was Nessus and Nextpose. Instead of analysing a large organizational network, they carried out their analysis on a single device.

The current body of literature highlights that a significant number of research studies have concentrated on the security of Web Applications and IOT devices. Then the studies also concentrated solely on evaluating the efficacy of vulnerability scanning tools through conducting experiments on a single device. A research article also explained its own method of evaluating the efficiency of vulnerability scanning tools. A research paper also provided recommendations on the usability of vulnerability scanning tools, identifying potential challenges users may face when utilizing the tool and proposing suggestions for enhancements. One study also created their own system for vulnerability scanning. Their testing was conducted on a small virtual network rather than a larger organizational network. Some studies utilized Nessus, a product from Tenable, for their research and evaluation but did not deploy it as the sole tool in an internal network of an organization. The authors of the research did not emphasize on the optimal placement of the vulnerability scanning tools in a large organizational network, nor did they provide any suggestions to enhance the scanning speed of the tools they employed. The motivation behind this thesis was to bridge the research gap by conducting vulnerability scanning within the internal network of Sanoma, a selected organization, using the Tenable product while considering its features and capabilities. Additionally, this thesis aimed to determine the best placement of the chosen Tenable product within Sanoma's internal network, as well as provide recommendations for improving scan speed which part of selected Tenable product.

5 Target network, Specification and Design of the experiment

5.1 The target network environment

When conducting internal vulnerability scanning for an organization, it is essential to comprehend its internal network. This entails gaining an understanding of the host, network devices, and the connections between them.

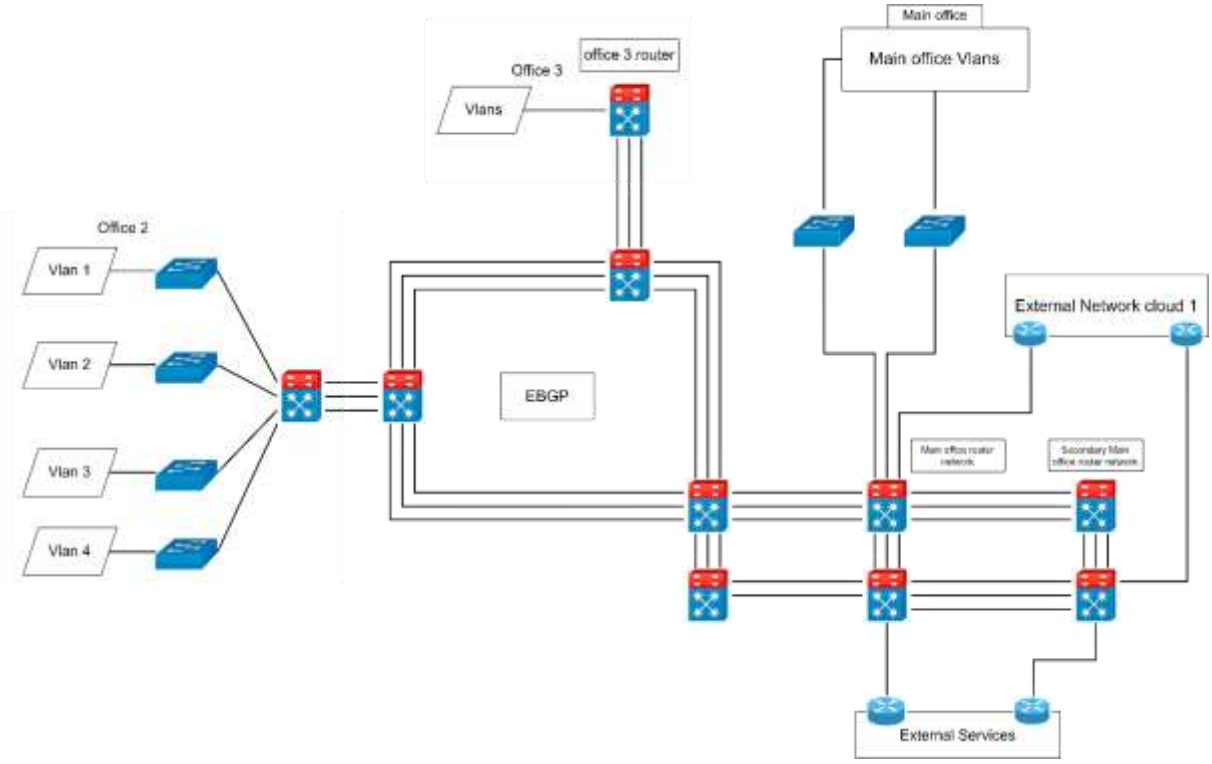

Figure 5. Sanoma Internal Network. This figure is self-drawn.

The above Figure 5 has been altered because of Sanoma's need for confidentiality. The diagram above illustrates the Sanoma Office Network in Finland. To effectively position Tenable.io and Nessus during experimentation, it was crucial to first comprehend the internal network topology. The diagram reveals that Sanoma's network is partitioned into Vlans, a common practice among organizations for network segmentation. Implementing Vlans has enhanced the network's reliability, efficiency, and security. The three border routers, labeled with EBGP, are responsible for routing network packets between Vlans, despite their physical locations within the organization.

5.2 Motivation behind Experiment

There were multiple justifications for designing and carrying out experiments. The primary rationale was to gain a better understanding of how to enhance the Internal Vulnerability Scan utilizing Tenable products, specifically Tenable.io and Nessus Scanner. Another objective was to determine the most optimal and efficient location for the Nessus Scanner within the internal network infrastructure of Sanoma Media. Once the Nessus Scanner was positioned, the experiment aimed to identify strategies for expediting vulnerability scans with Tenable products like Nessus Scanner.

5.3 Nessus Scanner Vs Nessus Agent

When deciding between Nessus Agent and Nessus Scanner for experimentation, several reasons led to selecting Nessus Scanner. The reasons are listed in the following table below.

Table 1. Difference between Nessus Scanner and Agent

Nessus Agent	Nessus Scanner
• Does not scan the entire network but only a particular asset. • Allows Basic Network Scan but cannot perform Host Discovery Scan. • By configuring credentials of a particular asset it then provides the inside view of that asset only. • Does not allow to perform network checks.	• Can scan entire network. • Allows to carry out external and remote security checks. • Provides an external view with the help of port scanning which is part of Host Discovery Scan. • By configuring credentials, it can provide an inside view of the network and its assets. • Allows Basic Network Scan and Host Discovery Scan.

First, Nessus Scanner can scan the entire network of Vlans, and it enables external and remote security checks, allowing scanning of target systems deployed on remote servers. Host Discovery Scan's port scanning provides an external view of assets, and users can perform Basic Network Scan and Host Discovery Scan on the network. When provided with a specific asset's credentials, the scanner can deeply scan the network and its assets. In contrast, the Nessus Agent only scans the asset in which it is installed and does not allow Basic Network

Scan or Host Discovery Scan. It can only provide deep scanning results for the particular asset in which it is installed and has been given credentials. Lastly, it cannot perform network checks.

5.4 Possible Scenarios for positioning the Tenable Nessus Scanner

Once the internal network was understood, various scenarios were created to determine the best placement for the Tenable Nessus Scanner. The focus was on office devices and network devices, specifically switches. Typically, Nessus Scanner(s) will be connected to Tenable.io to provide a centralized view. The scenarios are presented below.

5.4.1 Scenario 1: One Nessus Scanner without Vlan grouping

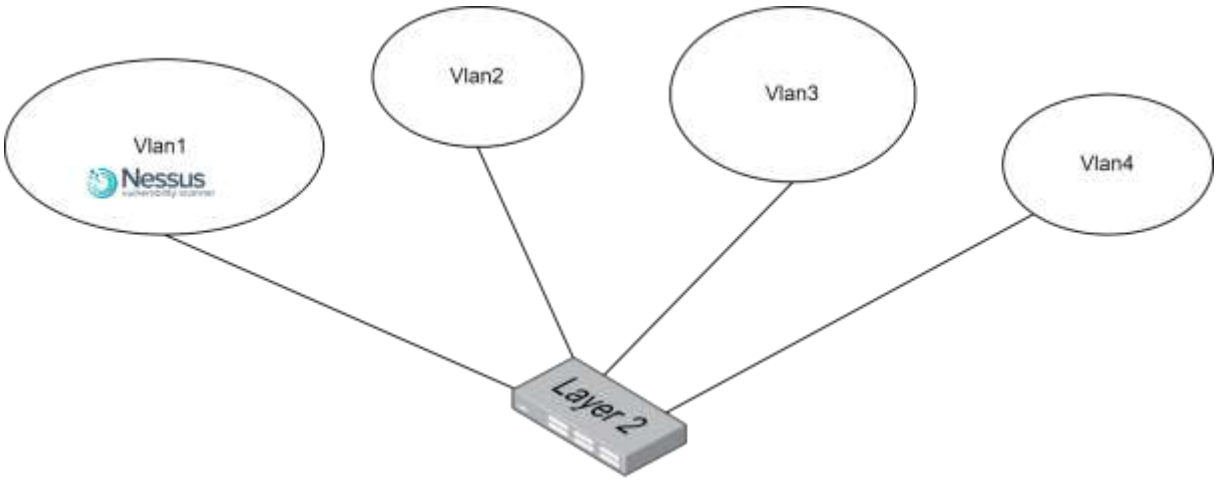

Figure 6. A diagram depicting a single Nessus Scanner without VLAN grouping for vulnerability scanning. This figure is self-drawn.

Figure 6 depicts the deployment of the Nessus Scanner in Vlan1, enabling it to perform vulnerability scanning of its own vlan as well as other vlans. In this scenario, the Nessus Scanner is installed on a server. The benefits of this approach include the ability to easily scan all vlans with a single scanner by inputting the IP addresses that require scanning. Additionally, fewer hardware resources are necessary as only one scanner is needed and deployed on the server. Lastly, vulnerability scanning is more manageable because only one scanner and its results need to be monitored.

On the flip side, there are drawbacks to this approach. Firstly, if the Nessus Scanner performs vulnerability scans on all vlans simultaneously, the scan time will increase due to the usage of hardware and software resources, such as the scanner's scan libraries, across all vlans during vulnerability scanning. Additionally, there is a risk of switches dropping the scanning packets

of the Nessus Scanner during scans of all vlans, resulting in poor network performance. Lastly, when scanning the switch, there will be no redundancy in the scan results since only one Nessus Scanner will be scanning the switch.

5.4.2 Scenario 2: Multiple Nessus Scanners with Vlan grouping

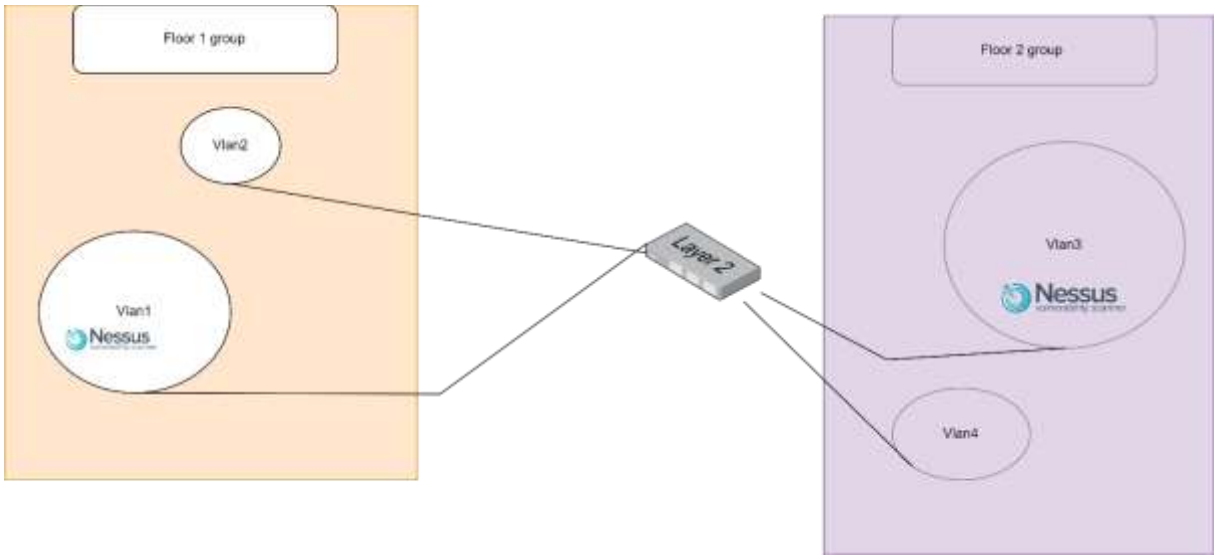

Figure 7. A diagram of multiple Nessus scanners grouped by VLANs. This figure is self-drawn.

Figure 7 demonstrates that VLANs can be classified based on their proximity, such as by floors or buildings, for instance. Within each group, one Nessus Scanner will be used to perform vulnerability scanning. The hardware and installation expenses of the Nessus Scanner will depend on the number of floors or buildings in the network. For instance, if an office has two floors and two Vlans on each floor, the Vlans will be grouped based on the floor. Thus, two Vlans on each floor will be grouped together and will share one scanner. The Nessus Scanner will be installed on a server with its own dedicated IP ports and configurations based on the operating system being used.

This case scenario offers several advantages. The first one is that multiple scans can be conducted on different groups simultaneously since each group has its own hardware, time and locality and is independent of other groups. Secondly, there will be an enhanced network coverage as the scanner will concentrate on the group of vlans it is deployed on. Lastly, scan scheduling will be more efficient as the scanner will be focused on the specific groups it is assigned to.

The scenario also has several drawbacks. Firstly, if all scanners are running simultaneously based on locality, the network load will increase. Secondly, any misconfiguration in the

Nessus Scanner can lead to network degradation. Additionally, the scan time will be directly proportional to the number of hosts in each vlan group. Moreover, it will be a laborious task to configure scanners for each group as each locality may have a different network address, port, and access rules for firewalls. Lastly, to prevent redundancy in vulnerability results and negative impact on the network, only one scanner from any group should scan the switch.

5.4.3 Scenario 3: Nessus Scanner in every Vlan without Vlan Grouping

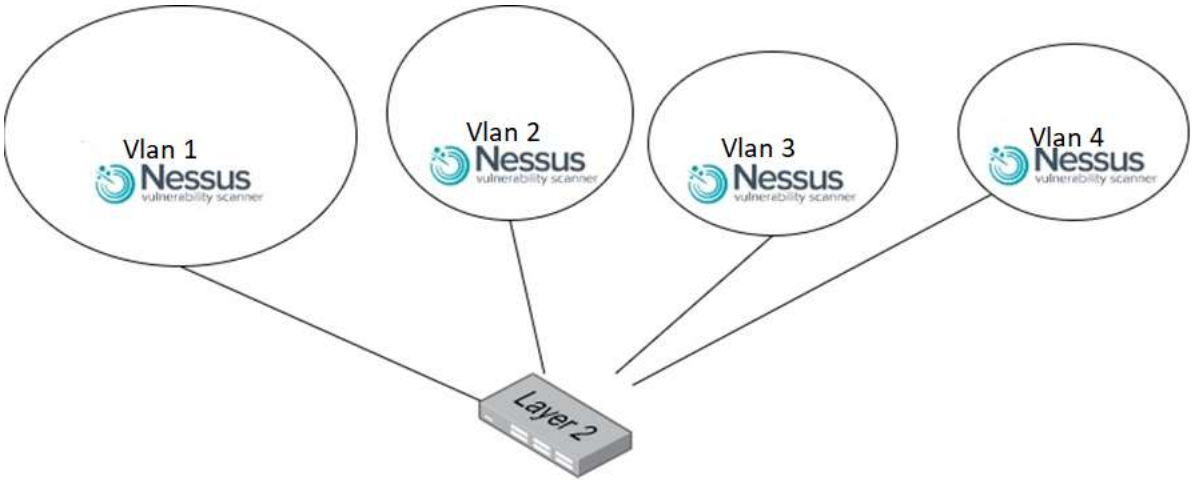

Figure 8. Diagram of Nessus Scanner deployment in multiple VLANs without VLAN grouping. This figure is self-drawn.

Figure 8 depicts a scenario where a Nessus Scanner is installed in each VLAN, presenting various benefits. Firstly, it provides better vulnerability scanning management as each scanner is deployed in its respective VLAN, making it easy to monitor all the scanners. Secondly, each VLAN has its own scanner, resulting in greater scan coverage for every VLAN. Additionally, the scan scheduling can be better since the VLAN activity will be dynamic with respect to time and day. Different scans on different VLANs can be conducted at different times.

To summarize, deploying a Nessus Scanner in every VLAN has some advantages and disadvantages. One advantage is that it offers better vulnerability scanning management as each VLAN has its own scanner, making it easy to monitor all of them. Additionally, there is greater scan coverage for every VLAN since each one has its own scanner, and better scan scheduling can be achieved since different scans can be done on different VLANs at different times.

However, this approach also has its downsides. If all the scanners run scans simultaneously, it can increase network traffic, and more hardware is required since every VLAN needs its own scanner. The load on the switch will also increase if all scanners are running in different VLANs at the same time, which can be expensive in terms of hardware costs and scalability. Additionally, configuring the scanner for each particular VLAN can be time-consuming. Finally, to avoid redundancy in the vulnerability results of the switch and prevent negative network impact, it is necessary for only one scanner from any group to scan the switch.

5.4.4 Scenario 4: One Nessus Scanner with VLAN Grouping based on No. of Hosts

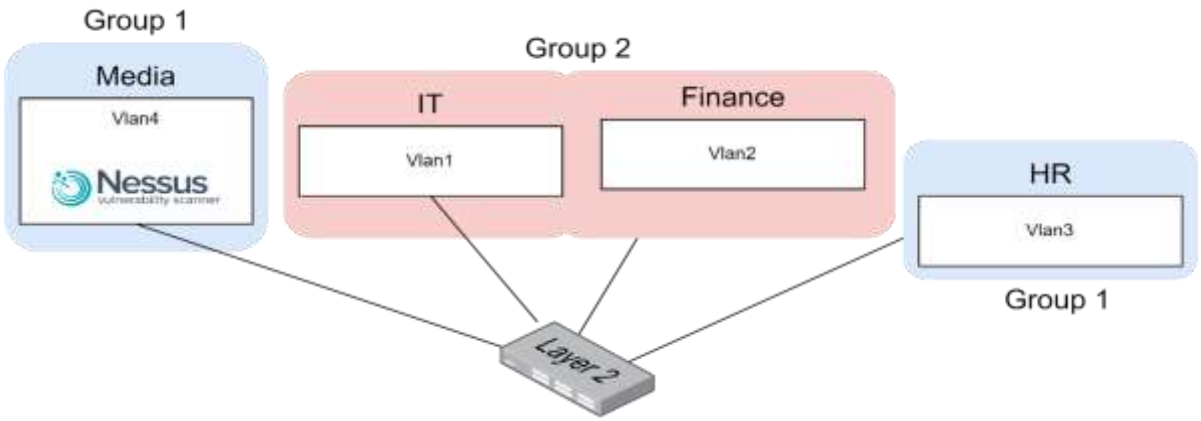

Figure 9. Diagram of Nessus Scanner deployment with VLAN grouping based on number of hosts. This figure is self-drawn.

Regarding the scenario depicted in Figure 9, the vlans are being grouped based on the quantity of hosts within them. As an illustration, in the Figure provided above, IT and Finance are combined because they each contain 256 hosts, whereas HR and Media have 128 hosts per vlan.

One of the advantages of this scenario is that scan scheduling would be effective as groups with more hosts would be given higher priority. Other advantages are similar to the first scenario where vlans are grouped based on department importance and only one scanner is deployed to scan all the groups.

Scan time may increase in groups with a larger number of hosts. Other drawbacks are comparable to the first scenario, where VLAN grouping is based on department importance and a single scanner is responsible for scanning all groups.

5.5 Nessus Scanner Requirements

A dedicated server is required to install the Nessus Scanner, and the server can run either Windows or Linux depending on the organization's needs. The rationale behind using a dedicated server is that servers have more processing power than desktop computers and can be solely dedicated to running the Nessus Scanner and the scans it performs.

- General Requirements

  If an organization needs to scan up to 50,000 assets with a Nessus Scanner, it requires a minimum hardware configuration of four 2GHz cores with 8GB RAM and 30 GB disk space. However, if the number of assets to be scanned is more than 50,000, then a minimum hardware configuration of eight 2GHz cores with 8GB RAM and 30 GB disk space is necessary.

- Port Openings

  To access the Nessus Interface and enable inbound traffic, it is necessary to open TCP port 8834. This port allows incoming traffic to connect to the Nessus interface and interact with it. However, to enable outbound traffic and allow the Nessus scanner to communicate with Tenable.io and the plugins.nessus.org server, it is necessary to open TCP port 443. This port is used for secure communication with external servers and is required for the scanner to access plugin updates and other important resources necessary for the scanning process. Therefore, both TCP ports 8834 and 443 should be open to allow both inbound and outbound traffic and ensure the proper functioning of the Nessus scanner.

- Antivirus

  To enable the Nessus Scanner to scan the target workstation, it is necessary to ensure that the anti-virus software being used on the workstation allows the scanner to perform the scan. This is important because anti-virus software can sometimes perceive scanning activities as malicious and block them, which can impede the scanning process. Therefore, it is important to make sure that the anti-virus software is configured to allow the Nessus Scanner to scan the device.

  Furthermore, the network administrator should also allow the Nessus Scanner traffic into the target workstation. This involves configuring the network to permit communication

between the scanner and the target workstation. Failure to do so will prevent the scanner from accessing the device and conducting the necessary vulnerability scan. Therefore, it is important to ensure that the necessary network configuration is in place to allow the scanner to communicate with the target workstation.

- Host Based Firewall

To perform a successful vulnerability scan on a targeted workstation, it is important that the host-based firewall within the workstation allows the Nessus Scanner's traffic to enter. This can be achieved by configuring the host-based firewall settings to permit the specific port(s) required by the Nessus Scanner for communication. The network administrator should also ensure that the firewall settings of the network do not block the Nessus Scanner's traffic to the targeted workstation, as it can result in a failed scan. It is essential to ensure that both the host-based firewall and network firewall settings are configured appropriately to allow the Nessus Scanner to conduct a vulnerability scan successfully.

- General Firewall

The process of designing the access rules is necessary to ensure that the ports required for the inbound and outbound traffic from and into the Nessus Scanner are open. These rules should be written in the organizational firewall to facilitate the easy transfer of inbound and outbound network traffic from the Nessus Scanner to Tenable.io. It is important to carefully determine the specific ports that need to be open for inbound and outbound traffic, and to ensure that the rules are accurately implemented in the firewall. By doing so, the Nessus Scanner will be able to effectively communicate with Tenable.io, allowing for the seamless transfer of information between the two systems.

- Licensing

Tenable offers licenses for scanning an organization's assets. It is important to note that each license is assigned to a single asset, and therefore, if there are a large number of assets, it is advisable to purchase a limited number of licenses and utilize them for different assets over time. These licenses do not have an expiration date. Once a license is assigned to an asset, it remains with that asset for a period of 90 days, after which it becomes available to be used for another asset. However, it is not possible to release a license before the 90-day period has elapsed.

5.6 Hosting Tenable.io

According to Tenable organization, Tenable.io is exclusively deployed on the cloud. Organizations have the freedom to choose the cloud services where they want to deploy it. Tenable.io is compatible with various cloud services such as Amazon Web Services, Microsoft Azure, Google Cloud, IBM cloud, Oracle Cloud, Red Hat, and Heroku. During the experimentation phase for viewing the vulnerability results from the Nessus Scanner, Tenable.io was also utilized. There are multiple reasons why Tenable.io is considered to be advantageous. Firstly, Tenable.io provides up to 70,000 vulnerability classifications. Secondly, since Tenable.io can be deployed on any cloud service, it is cost-effective compared to deploying other vulnerability management software within an organization. In order to transfer data from the Nessus Scanner to Tenable.io, it is mandatory to establish a connection between the Nessus Scanner and Tenable.io.

5.7 Types of Scans used in Nessus Scanner

The following two scans will be conducted sequentially: firstly, the Host-Discovery Scan, followed by the Basic Network Scan.

- Host Discovery Scan: The Host-Discovery Scan is designed to identify the active IP address, operating system and open ports of the target device and switch. It is a lightweight scan that does not disrupt the normal functioning of the device and switch. Furthermore, it does not require any licenses as it is being executed, which implies that it will not verify if the asset being scanned is licensed or not. The scan can also be used to scan the switches. The Host Discovery Scan utilizes various plugins such as OS Identification, Nessus SNMP Scanner, Nessus SYN Scanner, and Netstat Ports Scanner (SSH).

- Basic Network Scan: This type of scan is designed to perform a comprehensive scan on the target asset using all available built-in plugins. These plugins, which are categorized into various groups, are used by the Basic Network Scan to identify vulnerabilities in the target asset. There are approximately 165,000 plugins that the scan uses. However, it is not suitable for scanning switches because it can disrupt the packet-forwarding capabilities of a switch. Additionally, the scan generates a significant amount of TCP SYN packets, which may impact network performance by increasing network traffic. To minimize its impact, the Basic Network Scan offers a

Scan Window option that automatically stops the scan after a certain period of time. In all the situations mentioned above, scanning routers is not recommended because they are the backbone of a network and scanning them can have negative consequences. The reason is that routers are responsible for forwarding packets of all the VLAN networks of an organization, and when they are being scanned by Nessus Scanner, their hardware resources will also be utilized. Therefore, it is better to limit the scan to the VLANs and switches to which they are connected. This way, the router's responsibility will only be to transfer the network traffic, including scanning traffic and data traffic.

5.8 Selected scenario

In Section 3.3, various scenarios were discussed along with their pros and cons. Out of those scenarios, one scenario was chosen for the experiment. The scenario chosen, depicted in Figure 6, is "One Nessus Scanner without Vlan grouping". The reason for choosing this scenario was that from a business point of view, it is cost-effective as only one hardware is needed for installing the Nessus scanner. Additionally, only one Nessus scanner is required to be linked with the Tenable.io, which makes synchronization effective.

Multiple scenarios were considered, but they were not chosen due to several reasons. One of the primary reasons was that those scenarios required the use of multiple scanners, which could lead to increased costs from a business perspective since separate hardware would be necessary for each scanner. Furthermore, maintaining multiple scanners could be a challenging task from both hardware and software perspectives, which added to their drawbacks. Therefore, those scenarios were not selected for experimentation. The network is designed to work best with a single scanner that can scan all the VLANs being used. This implies that any attempt to use multiple scanners at the same time may lead to substantial network congestion. Therefore, to prevent network disruptions, it is suggested that the current practice of using a single scanner for all VLANs be maintained.

The practical implementation of the chosen scenario will involve monitoring and calculation of three key elements. The first one is the total count of vulnerabilities detected during the scanning process. The second aspect is the number of assets that were targeted by the scanner during the vulnerability assessment. Finally, the time required to perform the vulnerability scan will also be closely monitored and recorded.

5.9 Vulnerability Scanning Workflow

The Vulnerability Scanning Workflow depicted below illustrates the process used for conducting vulnerability scanning.

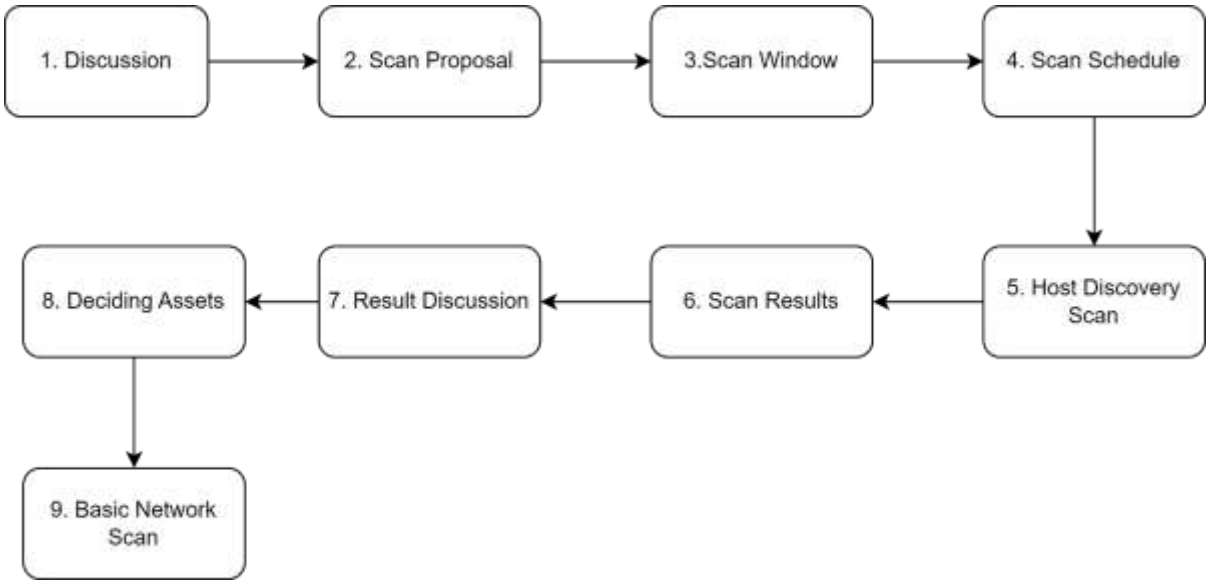

Figure 10. Diagram of the workflow for the vulnerability scanning procedure. This figure is self-drawn.

As illustrated in the above Figure 10, initially, a Discussion stage was undertaken where potential scenarios for placing the Nessus Scanner within the internal network were designed, taking into account their respective advantages and disadvantages. Following this, a scanning proposal document was prepared, which included all relevant details about the different scenarios. The proposal was then presented to higher management, and after a meeting, the first scenario from the proposal was chosen.

Then during the Scan Window stage, the number of targeted VLANs was determined. Then, in the Scan Schedule stage, it was decided that the scan would take place after office hours. The Host Discovery Scan was configured and executed in the Host Discovery Scan stage. After the scan, the results were analysed based on device category, ownership, location, and open ports. The information security team discussed the results with the network team. During the asset selection process, detailed discussions were held between the information security team and the network team about the devices that were scanned, including their category, ownership, and location. The network team explained their procedure for conducting thorough research and gathering information on the devices. Suspicious ports were used as a criterion for selecting assets. In cases where the exact device information was

unknown or there was contradictory information during the information gathering stage, the Basic Network Scan was performed on the selected assets.

6 Implementation of the experiment

From one of the several possible scenarios, Scenario 1, was selected and carried out during the experiment. The VLANs included in a larger network with the IP address range of 10.184.X.X, capable of accommodating up to 4094 hosts. To maintain confidentiality for Sanoma Media, the network address of the larger network is not disclosed. The network infrastructure used in the experiment is identical to that of Scenario 1.

6.1 Deployment of Nessus Scanner

To make the experiment feasible, the initial step was to set up and set the configurations for the Nessus scanner on a server that runs on the Linux operating system. The machine designated for installing Nessus Scanner was equipped with two Intel Xeon E5-2640 v3 processors, each containing eight cores capable of reaching a maximum turbo speed of 3.40 GHz, with a base speed of 2.60 GHz. Additionally, the server was fitted with two 64GB DDR4 RAMs. Its storage capacity was initially 2TB, but it could be expanded up to 16TB. The server was running on a Linux-based operating system, specifically Ubuntu 20.04.

After deciding the server and the operating system used in the server Nessus Scanner was installed in the server. The steps to install Nessus Scanner were:

1. Download the Nessus Scanner version from Tenable website.
2. Then in the command line, the Nessus Scanner installation command was executed. The command was:

```
dpkg -i Nessus-<version number>-debian6_amd64.deb
```

3. Then in command line, the Nessus Scanner daemon start command was executed. The command was:

```
systemctl start nessusd
```

6.2 Linking Nessus Scanner with Tenable.io

Once Nessus Scanner was installed on the Ubuntu-based server, it was connected to Tenable.io by following steps:

1. Log in to Tenable.io account.

2. From the "Scanners" tab on the left-hand side of the screen, use the "Add Scanner" button.

3. Select "Nessus" as the type of scanner which will be added.

4. Following the on-screen instructions to download, Tenable Nessus Scanner connector is installed.

5. The Connector is launched and the API Key of Tenable.io is provided which will link the Nessus Scanner to Tenable.io.

6.3 Performing Host Discovery Scan

Once the Nessus Scanner was connected to Tenable.io, a Host Discovery Scan was set up and executed. The following steps were taken to configure the Host Discovery Scan:

1. To set up the Host Discovery Scan, the initial step is to navigate to the "Scans" section and select the "Create Scan" tab located in the top right corner of the page, as demonstrated in Figure 11.

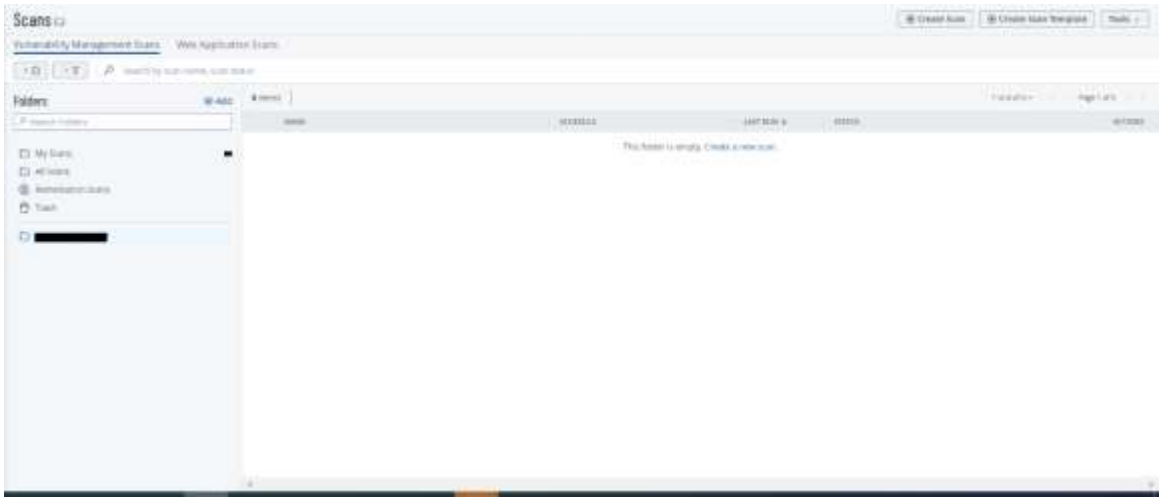

Figure 11. An image of the 'Scan' section within the software. (c) Tenable, used with permission.

2. Then after reaching to “Create Scan” Section select Host Discovery. The illustration can be found on the following page, specifically in Figure 12.

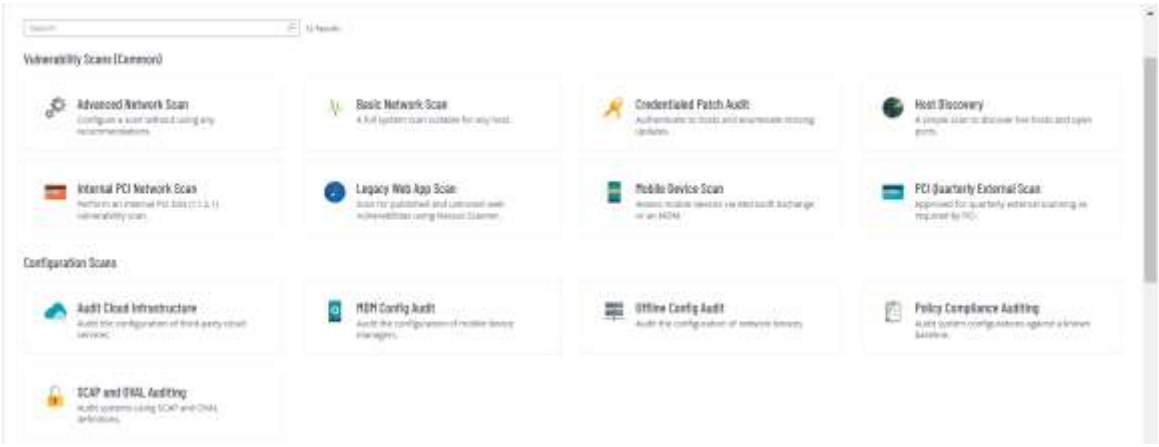

Figure 12. An image showing the 'Create Scan' section showcasing a variety of scan types. (c) Tenable, used with permission.

3. Once in the "Host Discovery Scan" section, provide a name for the scan to aid in identification, and then choose the "Scanner Type" as "Internal Scanner," as depicted in Figure 13 below.

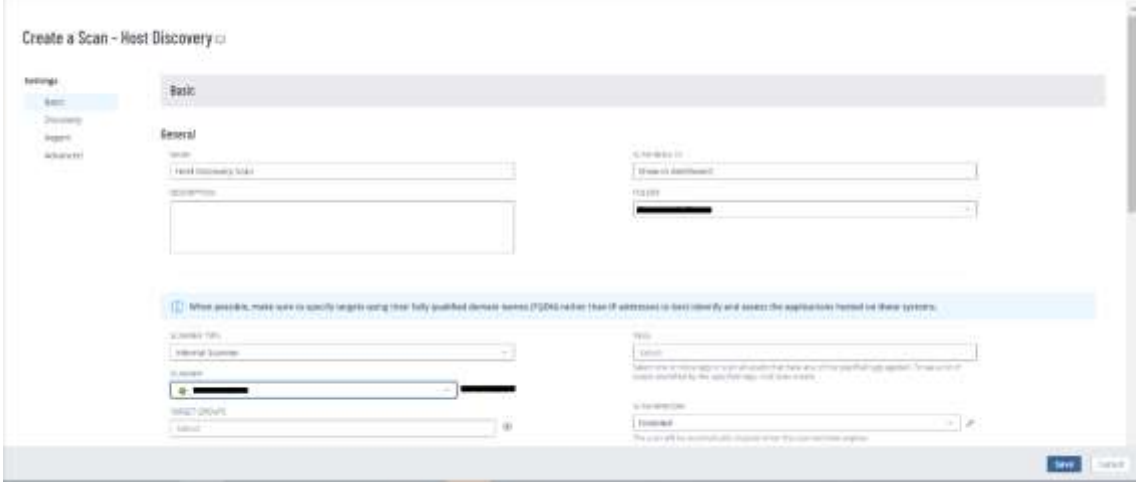

Figure 13. An image showing the 'Create a Scan – Basic Setting' section of Host Discovery Scan. (c) Tenable, used with permission.

4. Likewise, under the "Target" section, enter the network on which the Host Discovery Scan will be conducted. Next, set the scan schedule in the "Schedule" section, ensuring that the scan does not have an adverse impact on the Vlans. Finally, determine the "User Permissions" to grant Execute and Edit permissions to other users, as desired by the scan creator, as depicted in Figure 14 that is on the next page.

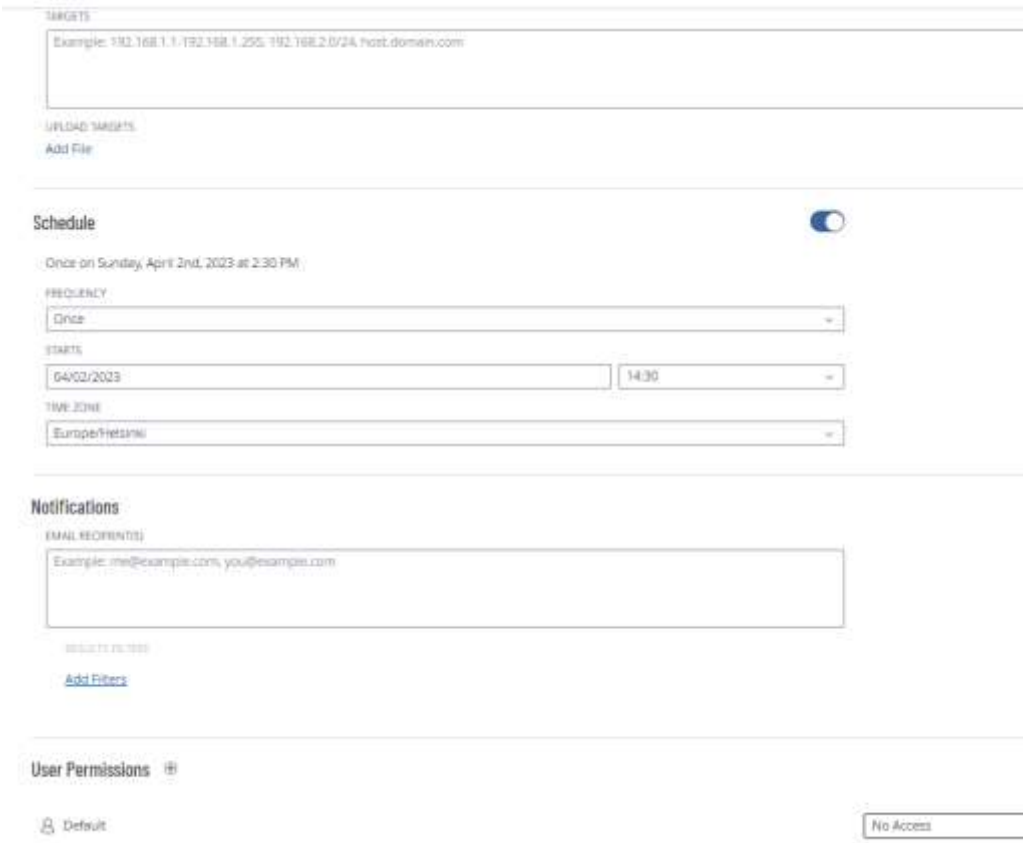

Figure 14. An image showing continuation of 'Create a Scan – Basic Setting' section of Host Discovery Scan. (c) Tenable, used with permission.

5. Regarding the discovery process, there are various options available in the "Discovery" settings. However, the most comprehensive option is the Port scan (all ports) as it provides detailed information about all the target office devices, such as their Operating System, IP address, open ports, and services running on those open ports. The "Discovery" settings are illustrated in Figure 15 below.

Figure 15. An image showing the 'Create a Scan – Discovery Setting' section of Host Discovery Scan. (c) Tenable, used with permission.

6. Step 6 involves selecting the preferred report option in the "Report" section. Depending on the scenario, the user can choose from the four available options. In this case, "Display host respond to ping" and "Designate hosts by their DNS name" were selected, as it would be easier to identify active office devices in the results. The following Figure 16 below displays the options available in the "Report" section.

Figure 16. An image showing the 'Create a Scan – Report Setting' section of Host Discovery Scan. (c) Tenable, used with permission.

7. The "Advanced" settings section offers additional options for including in scans. Enabling the "Slow down the scan when network congestion is detected" option can be useful because the scanner will adjust the scan intensity by sending fewer scan packets to the target network when network congestion is detected. The network timeout is set to 5 seconds by default, and the maximum number of simultaneous checks per host is set to 5 to avoid network congestion. Additionally, the "Max simultaneous hosts per scan" value is set to 50, which means that the Nessus Scanner will scan up to 50 office devices in the given target network simultaneously with the Host Discovery Scan. Setting the "Max simultaneous hosts per scan" value can help reduce the probability of network congestion. The appropriate value for this setting depends on the person's expertise who is configuring the scan. Figure 17 on the subsequent page displays the section for "Advanced" settings.

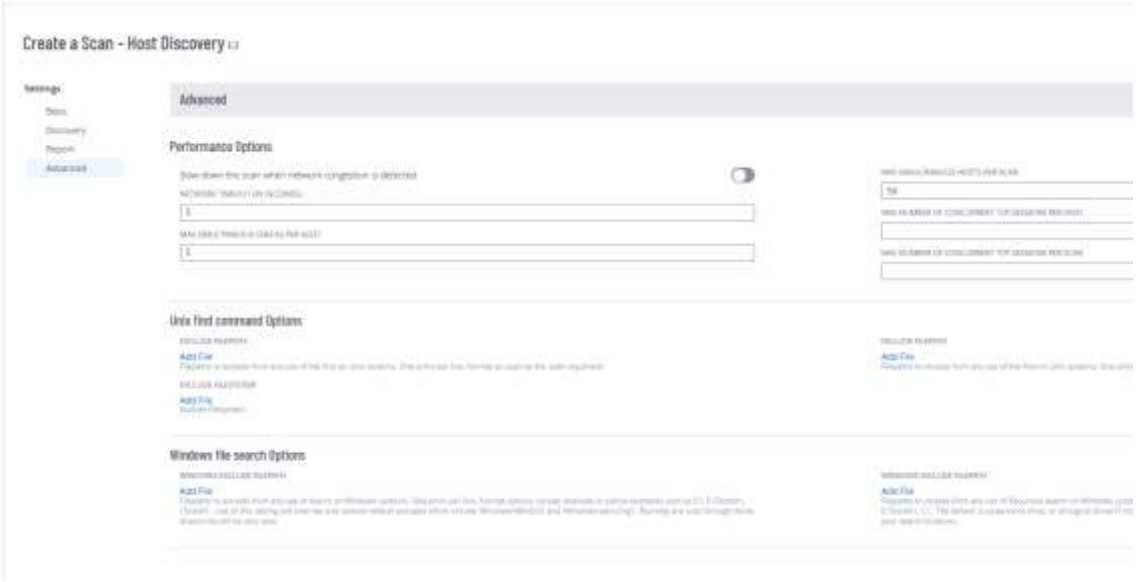

Figure 17. An image showing the 'Create a Scan – Advance Setting' section of Host Discovery Scan. (c) Tenable, used with permission.

After configuring the Host Discovery Scan, the Nessus Scanner will automatically start the scan as it was scheduled from before.

6.4 Performing Basic Network Scan

The following steps were taken to configure the Basic Network Scan:

1. To begin the process of setting up the Basic Network Scan, go to the "Scans" section and click on the "Create Scan" tab in the top right corner of the page, as shown in Figure 18 below.

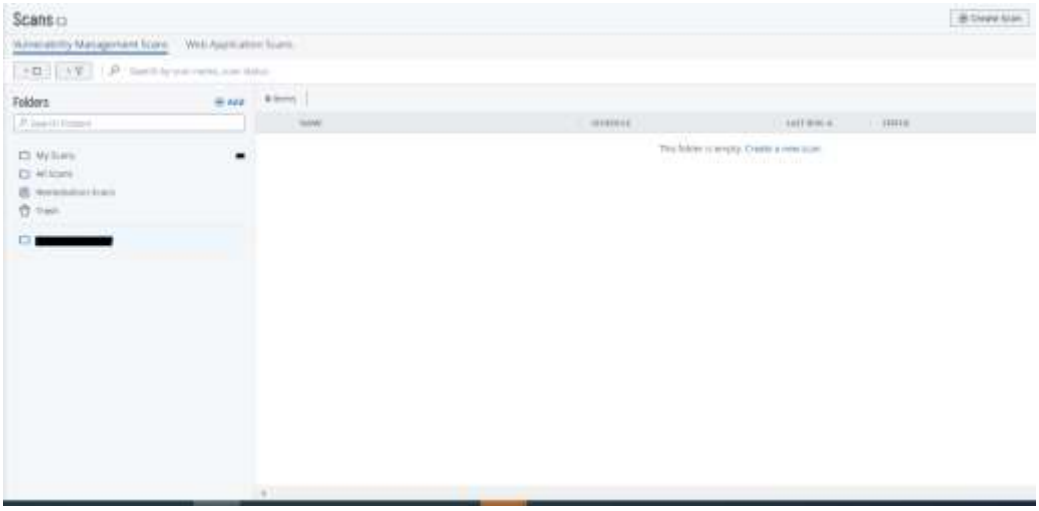

Figure 18. A picture displaying the 'Scan' section found in the software. (c) Tenable, used with permission.

2. Next, once you have arrived at the "Create Scan" section, you should choose the "Basic Network Scan" option. The step being referred to is demonstrated in Figure 19, as shown below.

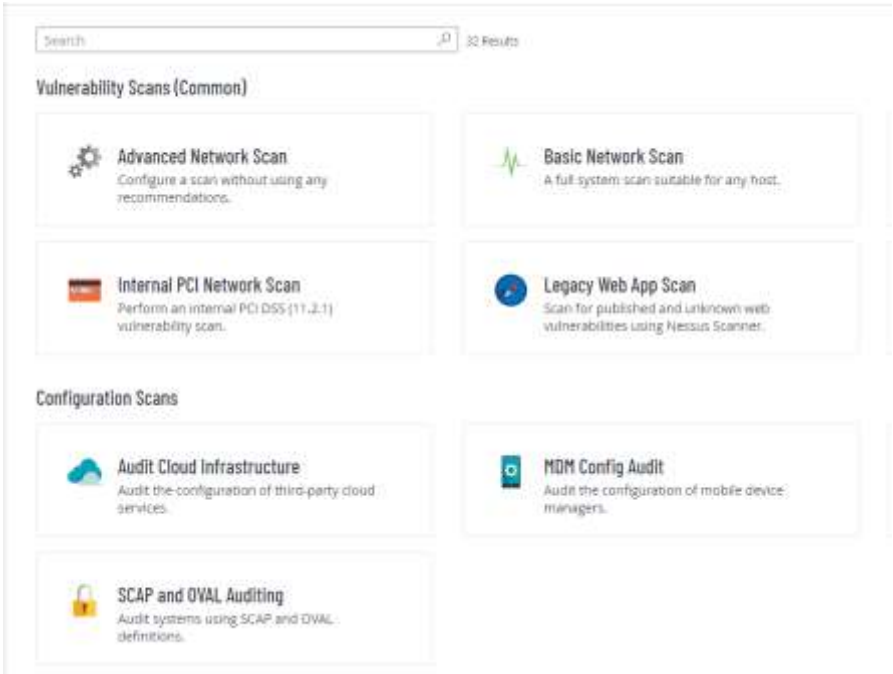

Figure 19. A picture displaying a range of scan types within the 'Create Scan' section. (c) Tenable, used with permission.

3. After accessing the "Basic Network Scan" section, it is required to give the scan a name that will help with identification. Additionally, select "Internal Scanner" as the "Scanner Type," as shown in Figure 20 below.

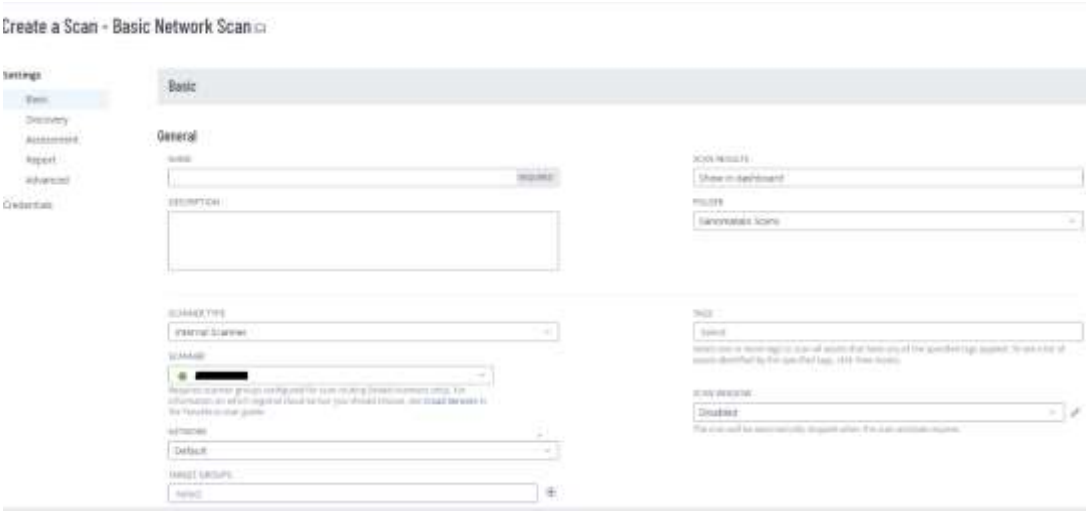

Figure 20. An image showing the 'Create a Scan – Basic Setting' section of Basic Network Scan. (c) Tenable, used with permission.

4. In a similar manner, in the "Target" section, input the IP address of the device where the Basic Network Scan will take place. Then, in the "Schedule" section, set a scan schedule that won't negatively affect the Vlans. Lastly, the user should decide on the "User Permissions" they want to give other users, such as Execute and Edit permissions, as per their preferences. Figure 21 below shows the depiction of this step.

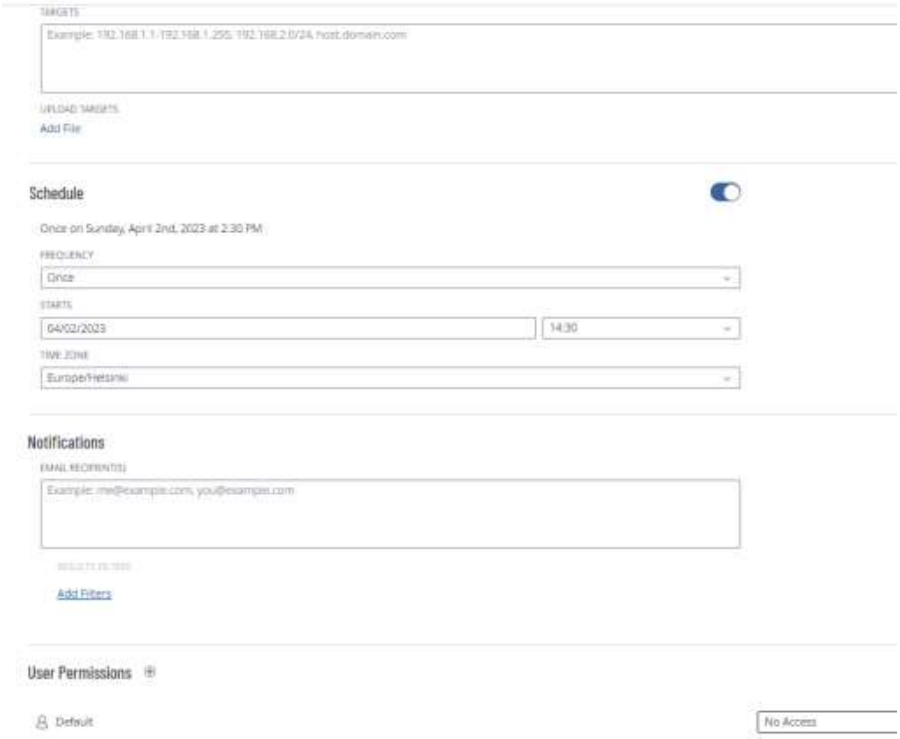

Figure 21. An image showing continuation of 'Create a Scan – Basic Setting' section of Basic Network Scan. (c) Tenable, used with permission.

5. When it comes to the discovery process, the "Discovery" settings offer several choices. Among these options, the most extensive one is the Port scan (all ports) since

it gives a thorough overview of all the target office devices. The available options in the "Discovery" settings are displayed in Figure 22 below.

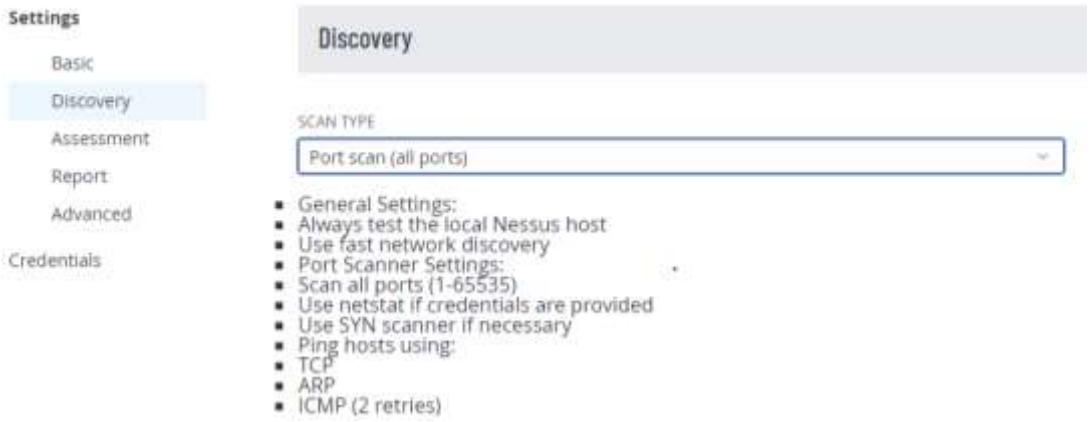

Figure 22. An image showing the 'Create a Scan – Discovery Setting' section of Basic Network Scan. (c) Tenable, used with permission.

6. To configure the Assessment Setting, choose "Override normal accuracy" and then select "Avoid potential false alarms" from the dropdown options. This option is preferred as it allows skipping any false alarms that may be present. In the “General” setting, select "Only use credentials provided by the user" to ensure that the scan functions at its best capacity by utilizing the device's credentials. For a demonstration of the Assessment Setting, refer to Figure 23 below.

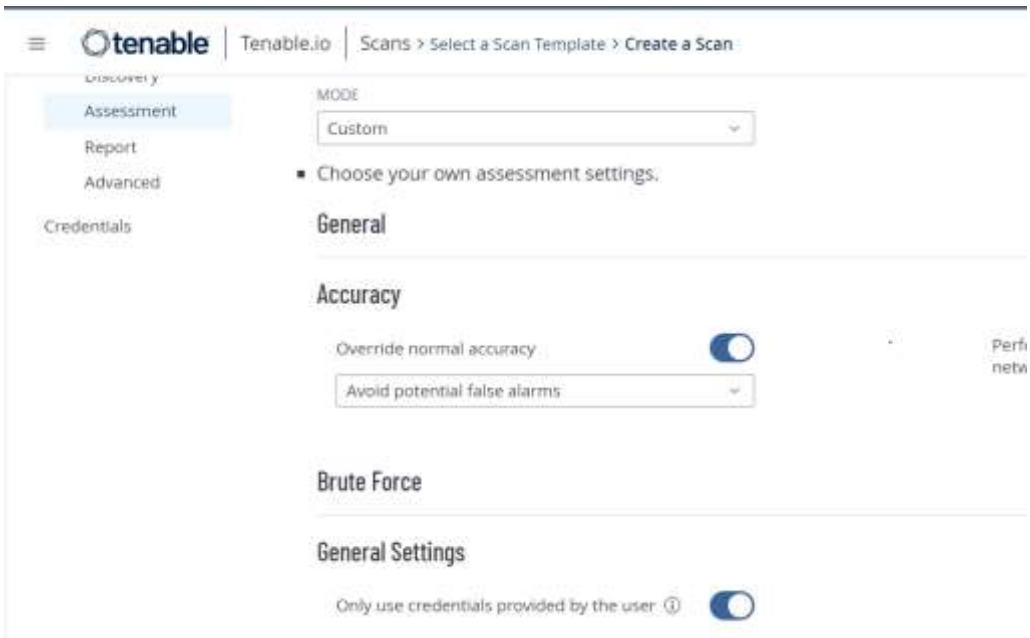

Figure 23. An image showing the 'Create a Scan – Assessment Setting' section of Basic Network Scan. (c) Tenable, used with permission.

7. Select the "Scan low bandwidth links" option for the "Mode" setting in the Advanced Setting, as depicted in Figure 24 below. This option is preferred to prevent any negative impact on the selected device or the network where it is located due to the scan.

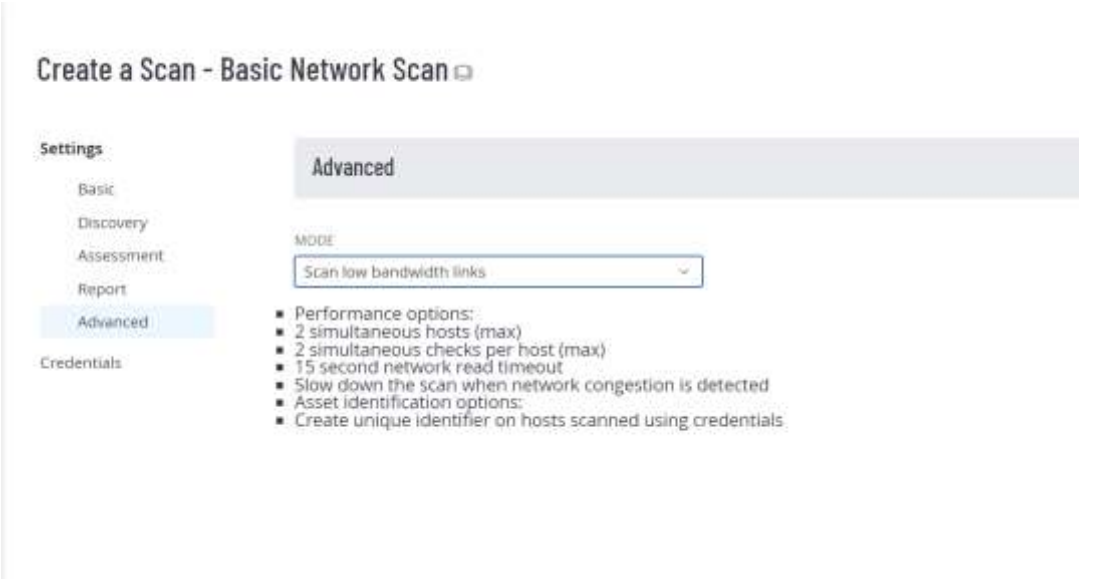

Figure 24. An image showing the 'Create a Scan – Advance Setting' section of Basic Network Scan. (c) Tenable, used with permission.

8. Figure 25 below demonstrates the process of providing the required device credentials for the Basic Network Scan in the Credential section. This enables the scan to conduct a thorough investigation.

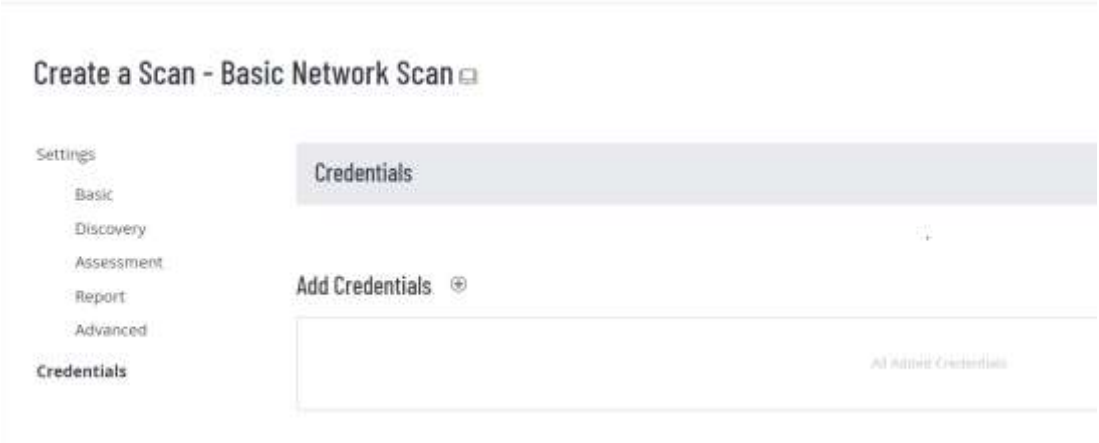

Figure 25. An image showing the 'Create a Scan – Credential Setting' section of Basic Network Scan. (c) Tenable, used with permission.

7 Analysis of the Results

7.1 Security Analysis of Host Discovery Scan

The Host Discovery Scan took 90 minutes to complete, during which it scanned a total of 4094 hosts across all the vlans in the super net. The scan produced numerous results, including the identification of the operating system running on each host and the open ports on hosts. To identify the open ports the scanner used Netstat Ports Scanner (SSH), Nessus SYN Scanner, and Nessus SNMP Scanner plugins. Then in order to identify the operating system the scanner used OS-Identification plugin. The scan results for the VLANS belonging to the super net are presented below.

7.1.1 OS – Information

The following table shows the operating systems which were identified during the Host Discovery Scan by the Nessus Scanner.

Table 2. List of operating systems identified by Host Discovery Scan.

Operating System
CISCO IOS
VMware ESXi
HP Switch
Polycom Teleconferencing Device
Ubuntu 20.04
Not Found

During the assessment, it was found that seven network devices were running on the CISCO IOS operating system. Additionally, there were two devices that were using the VMware ESXi virtual machine to deploy other operating systems. However, due to security measures in place, the Nessus Scanner was unable to obtain information about these operating systems. The scanner did, however, recognize VMware ESXi as an operating system. Another operating system identified by the scanner was the HP Switch, which is a proprietary OS of switch. Furthermore, the scanner was able to determine that four IP phones were in use based on their operating systems being identified as Polycom Teleconferencing Devices. Lastly, the

scanner successfully identified the operating system of the device on which it was installed, which was Ubuntu 20.04.

The Nessus Scanner also provided information that the operating system of 165 hosts could not be identified. Upon further analysis and discussion, it was determined that approximately 65 of these devices were Wireless Access Points as port 22, which is commonly used for ssh, was open. This port is open to allow for centralized maintenance of the Wireless Access Points. Additionally, it was concluded that there were at least 100 active workstations in the network. This determination was made after a technical security discussion with the Information Security team, as the devices were successfully pinged by the Nessus Scanner but the scanner was unable to obtain operating system information for the devices. The reason for this was that the workstations can be behind firewalls, which resulted in the scanner's scanning packets being dropped.

7.1.2 Common Opened Ports

The following Ports were observed to be open:

Table 3. List of common open ports identified by Host Discovery Scan.

Common Opened Ports
22/tcp/ssh
443/tcp/www
135/tcp/epmap
68/udp
161/udp/snmp
445/tcp/cifs

Like 22/tcp/ssh is the default port for the Secure Shell (SSH). This protocol is used for secure remote access to servers and computers, providing encrypted communication between the client and the server. It enables users to safely log in to a remote machine and perform various commands. Port 443/tcp/www is used for HTTPS traffic, which encrypts communication between a client and web server for secure web browsing. . Port 135 is utilized by the Endpoint Mapper Protocol (EPMAP) in the TCP protocol suite, which is employed by Microsoft Windows OS for discovering Distributed Component Object Model (DCOM) services on remote servers. Port 68 is designated for the Bootstrap Protocol (BOOTP) in the

User Datagram Protocol (UDP) suite of protocols. BOOTP is a networking protocol that allows networked computers to obtain essential configuration information, such as IP address, subnet mask, and default gateway, from a server known as the BOOTP server. Typically, BOOTP is utilized during the boot-up process of a computer to acquire essential network settings to enable communication with other devices on the network.

Port 161/udp was also open in the some of the devices. This port is used to run the Simple Network Management Protocol (SNMP) for managing and monitoring network devices. Another port that was found to be open was 445/tcp/cifs. This port is specifically assigned to the Common Internet File System (CIFS) protocol, which provides a way to share and remotely access files over the internet. CIFS is essentially a version of the Server Message Block (SMB) protocol that operates over the Transmission Control Protocol (TCP).

7.1.3 Unknown Open Ports

During the Host Discovery Scan, the Nessus Scanner detected 46 open ports in the devices which were not recognized or identified. It is necessary to conduct a deeper investigation into the reason why these ports were opened. This investigation process should began with a discussion with the network security team to understand the purpose behind opening these ports. If, during the discussion, it is determined that a specific unknown port is being opened by a legitimate software or service, then the investigation for that port is concluded. If that is not the situation, then it is recommended that the owner of the respective device shut down the software or service that is responsible for opening the port. During the discussion with the network team, it was essential to determine the owner of the device so that they can be contacted to shut down the software or service. If the device in question is a network device, then the network team is the responsible authority to be contacted. However, if it is a server, the owner of the server should be contacted. This applies similarly to workstations and other types of devices as well. If it is determined that no software is running on a specific open port, then the firewall logs can be reviewed to determine if the port needs to be open for internet access or not. This task is carried out by the network team who then use a centralized management software to close the open port. When performing the investigation table below can be used to identify commonly known ports that can be considered malicious.

Table 4. List of common known malicious open ports.

Commonly Known Malicious Open Ports
10101/tcp
8012/tcp
12345/tcp
5966/tcp
7961/tcp
6965/tcp
5001/tcp
31335/tcp

According to research, certain ports have been associated with the setup of Trojans. For example, port 5001 may be used for this purpose, as well as port 12345/tcp. Port 31335/tcp is not a reserved port and can be exploited by attackers for Trojan attacks. Port 10101/tcp has been linked to the Brain Spy Trojan [36]. Additionally, based on research and analysis, port 8012/tcp may be used by attackers to remotely control a device, which is referred to as a Backdoor Ptakks attack [37]. Finally, ports 5966/tcp, 7961/tcp, and 6965/tcp are also susceptible to being used for Backdoor Trojan attacks [38].

7.2 Security Analysis of Basic Network Scan

Following the completion of a Host Discovery Scan, several devices were identified as having unknown open ports that were deemed to be potentially malicious. This raised concerns about the nature of the port and prompted the need for a Basic Network Scan to be conducted on those specific devices. Among these devices were access points, IP phones, and a switch. The Basic Network Scan scanned eight devices which were selected. The total time taken by the scanner to scan the eight devices was 113 minutes. The scan was able to find out eight port vulnerabilities in total. The reason it took so much time because the devices were present in different vlans of the super net which were situated in different floors, so sending scan packets and retrieving the scan result packets took time. Then another reason was that the network was busy at the time the scan took place, as the network devices had to perform several packet forwarding while the scan was being conducted. After conducting the Basic Network Scan, a total of 30 vulnerabilities were detected.

The discovery of these 30 vulnerabilities indicates that there are actions that may be necessary to address these issues and enhance the network's overall security posture. Out of the total 30 vulnerabilities detected, 3 were categorized as low-risk, 26 were classified as medium-risk, and one vulnerability was identified as high-risk. This means that the majority of the vulnerabilities found during the Basic Network Scan were classified as medium-risk, indicating that they have the potential to cause significant harm to the network if exploited by a malicious actor. The identification of one high-risk vulnerability suggests that immediate attention and remediation are necessary to prevent potential security breaches and protect the network from harm. Then about 10 vulnerabilities among the total detected vulnerabilities required software updates to be applied. This means that some of the vulnerabilities discovered during the Basic Network Scan were likely caused by outdated or unpatched software or software components. The remaining vulnerabilities were attributed to unknown open ports. Securing these open ports is crucial for preventing unauthorized access to the network and mitigating the risks posed by these vulnerabilities.

Once the vulnerabilities were identified using the Basic Network Scan, they were analyzed and prioritized based on their level of risk. The highest-risk vulnerability was addressed and resolved first, followed by the medium-risk vulnerabilities, and finally the low-risk vulnerabilities. This approach ensured that the vulnerabilities are addressed promptly, reducing the risk of a potential security breach. By prioritizing and addressing vulnerabilities based on their level of risk, the organization would effectively manage their resources and prioritize their efforts to enhance the overall security. Furthermore, vulnerabilities resulting from outdated or unpatched software and software components were resolved by applying the latest patches and security fixes, which helped to address the vulnerabilities and prevent potential security threats. Finally, vulnerabilities that were caused by unknown open ports were resolved by closing those ports.

The specific vulnerabilities of Sanoma Media are not disclosed because of confidentiality concerns. However, for the purpose of comparing practical and theoretical vulnerabilities listed, hypothetical practical vulnerabilities from the Tenable vulnerability database are included in the following part of this chapter.

The following table below shows the example port level vulnerabilities along with their severity level that how critical these vulnerabilities are:

Table 5. List of example practical port vulnerabilities along with their severity level.

Example Practical Port Vulnerability	**Severity Level**
EFTP- Remote DoS	Medium
Sensitive Information Disclose	Medium
Remote Code Execution vulnerability in RDP	Critical
DOS – EXIM<4.96	High
Jenkin < 2.46.2/2.57	Critical
PHP version unsupported	Critical
SSL/TLS Diffie-Hellman Modulus < 1024 bits	Low
Bit Torrent Detection	Low

The table presented above illustrates various example practical vulnerabilities that are related to ports. For example, one vulnerability with a medium level of criticality, called EFTP-Remote DoS, results from the lack of a carriage return when sending data, causing the EFTP service to crash. Another medium criticality vulnerability, referred to as Sensitive Information Disclose, arises when a confidential document or file is mistakenly included in the web root. The third vulnerability, a Remote Code Execution vulnerability, is classified as critical due to its severity. This vulnerability permits an attacker to inject a harmful dll file into the current user's working directory. When the user opens the file, the attacker gains control of the user's system, allowing for arbitrary code execution.

Another vulnerability that may be present in hosts is DOS-EXIM<4.96. This vulnerability is considered highly severe and results from a flaw in the configuration files of older EXIM versions, which attackers can use to launch a Denial of Service attack. EXIM functions as a message transfer agent. In addition, there was a significant vulnerability identified in Jenkins, known as Jenkins < 2.46.2/2.57, which occurs in the Jenkin.java file, the source java file of Jenkins. Exploiting this vulnerability enables attackers to create an untrusted serialized java signed object, which they can then use to bypass block list protection mechanisms and execute arbitrary code on the target system.

Another vulnerability among the list of Port vulnerabilities pertains to the use of an Unsupported PHP version, which is classified as critical. Additionally, there is a vulnerability in SSL/TLS caused by the use of a modulus in Diffie Hellman that is less than 1024 bits. This flaw could make it easier for attackers to break into Diffie Hellman and gain access to the public and private keys. Lastly, there is a low criticality vulnerability associated with the use of BIT torrent on a particular port. This vulnerability is significant according to the organization's policy because users who employ BitTorrent may be vulnerable to malicious code injection while downloading files, potentially leading to system hijacking and exposure of sensitive data.

The table presented below illustrates the remedies for the example practical port vulnerabilities.

Table 6. List of example practical ports vulnerabilities along with their solution.

Example Practical Port Vulnerability	**Solution**
EFTP- Remote DoS	Upgrade to the latest version of EFTP
Sensitive Information Disclose	Remove static file from webroot
Remote Code Execution vulnerability in RDP	Port closure
DOS – EXIM<4.96	Upgrade latest version of Exim
Jenkin < 2.46.2/2.57	Upgrade to latest version of Jenkin
PHP version unsupported	Upgrade to latest version of PHP
SSL/TLS Diffie-Hellman Modulus < 1024 bits	Reconfigure Service
Bit Torrent Detection	Port closure

To address the first vulnerability, EFTP-Remote DoS, it is necessary to upgrade to a latest version of EFTP. The next vulnerability involves Sensitive Information Disclosure, which can be prevented by removing unnecessary files, including static files. Another solution is to utilize a password in the Webroot to restrict the addition of only essential files to the Webroot directory. For the Remote Code Execution vulnerability in Remote Desktop Protocol (RDP), the best course of action is to close the port immediately and set up a firewall rule to block incoming and outgoing traffic to any port running RDP.

To address the DOS-EXIM vulnerability, it is recommended to upgrade to the latest version of EXIM, which is 4.96 or a more recent release. The same approach applies to Jenkins, version 2.57 or newer is preferred. Additionally, for unsupported versions of PHP, it was necessary to install the current supported version of PHP.

In terms of protocols, it has been observed that the modulus used in the Diffie Hellman algorithm of SSL/TLS is less than 1024 bits. To address this vulnerability, it is recommended to use unique moduli that are at least 2048 bits or larger, and then reconfigure the service. Then in case of BitTorrent, the best solution recommended is to close the port on which BitTorrent is running.

Upon analyzing, it was seen that three of the example practical port vulnerabilities are linked to theoretical vulnerabilities. For example, EFTP-Remote DOS vulnerability was linked to Denial of Service because if the EFTP server receives a large request with thousands of characters, it will crash, and the server memory will be entirely overwritten. Similarly, Remote Code Execution vulnerability in RDP was associated with the theoretical Remote Code Execution because the aim is to gain access to all users who use Remote Desktop Protocol through a malicious server created by the attacker, who then remotely controls the affected systems. Lastly, DOS-EXIM < 4.96 vulnerability was comparable to Denial of Service because as an error exits in the regex handler, it can be exploited by the attacker to cause a crash.

7.3 Suggestion to improve Scan Speed

There are several scanning parameters present in the Nessus Scanner. Some of these parameters are really important and when these parameters are tweaked correctly then it can increase the speed of the scan and hence improve the vulnerability scanning.

The first parameter is “Scan Type” and in this parameter it is recommended to select the Internal Scanner. The reason to select Internal Scanner option is that when an organization selects its own scanner to scan the results the scan time will decrease as compared to when scanning is done with the help of cloud scanners that will require more time to scan the internal assets as they are connected to the Internet. Then when it comes to writing the IP ranges in the “Target Group” that will be scanned it is recommended to specify a particular VLAN rather than giving the network address of the super net which contains all the VLANs. The reason is that when the scanner focuses and perform a scan on a particular VLAN, scan results will be shown faster.

Also, it is suggested to set the scan schedule in the “Schedule” option present in the scan setting. The reason to do so is that if the volume of network traffic within the organization reduces during a specific period of time, it will enhance the scanning efficiency.

Vulnerability Scanning efficiency also decreases when the false positive results occur. So to avoid such kind of situation it is suggested to enable the “Use fast network discovery”. The reason is that when the option is enabled only those hosts which are active will respond to ping request sent by Nessus Scanner and this will decrease the amount of false positives which hence decreases scan time. When selecting the “Ping Method” method option it is recommended to select “Default Ping Option” as compared to the “UDP option” because UDP ping increases the scan time because the scanner will be waiting for response in the form error message. If there is no error during the communication between the scanner and target host, the scanner will not receive any form of error message. Then if the network team have information about which services are running on the devices’ on which ports, it would be an easy to mention the port ranges in the “Port Scan Range” so that only those ports are scanned. So it is suggested to contact the network team for such kind of information. The suggestion is to prepare a scanning proposal in which the security team of the organization already knows what type of scan to carry and which assets to scan. As result the vulnerability scanning efficiency will also increase because the scope of the scan has decreased.

8 Conclusion

The objective of this master's thesis was to enhance the vulnerability scanning process of the internal network and determine the ideal positioning of the scanner. Initially, a thorough analysis of the organization's internal network was done, followed by gathering both practical and theoretical knowledge of Tenable products.

Subsequently, multiple scenarios were formulated to determine the optimal placement of the scanner within the internal network for vulnerability scanning. Among these scenarios, a one scenario was chosen because it met the organization's business and financial requirements. Once the scenario was selected, it was implemented. Next, a Host Discovery Scan was conducted on the network, and subsequently, certain devices with unclear ports opened were singled out. Finally, a Basic Network Scan was performed on the selected devices. Three parameters were examined during the execution of the scans. In case of Host Discovery Scan it analysis was done on the time it took to complete . On the other hand, in case of the Basic Network Scan the number of vulnerabilities discovered, the severity level associated with the discovered vulnerabilities, the duration of the scan, and the number of assets targeted by the scan was examined.

In addition, a security analysis was conducted by comparing the practical vulnerabilities with the theoretical vulnerabilities. Afterwards, recommendations were provided on how to speed up a scan. Despite the security measures in place, vulnerabilities were still identified. This indicates that vulnerabilities can exist despite the level of security. Also, it is crucial to identify and mitigate vulnerabilities promptly based on their level of severity.

Then before implementing a new vulnerability management tool, it is recommended to discuss its technical, business, and financial aspects with security professionals within the organization. When performing scans, it is suggested that security experts ensure that the performance of the network and devices is not negatively impacted. Finally, conducting regular scans of the network and its assets is recommended to maintain security at all times.

8.1 Challenges and Limitation

During the research study, several challenges and limitations were encountered. These are summarized as follows:

- The initial obstacle was to thoroughly review the extensive documentation for Tenable's vulnerability products, such as Nessus Scanner and Tenable.io. While examining the documents, some definitions were unclear and could not be found online for further clarification and comprehension.

- Extensive meetings and scrutiny were conducted to choose a specific scenario for implementing the Nessus Scanner due to the high costs associated with its hardware and licensing.

- Another obstacle was determining the appropriate time frame for the Nessus Scanner to conduct designated scans on the target vlans. Conducting scans during peak hours could potentially disrupt network traffic and host devices, as well as generate multiple scanning traffic for a specific target device.

- The Nessus Scanner occasionally produced false positives, and it was necessary to consult with other team members to verify whether the false positive was accurate or simply an error.

- Another obstacle encountered was the need to gain a comprehensive understanding of the internal network. This difficulty was resolved through scheduling meetings with the network team to gain a deeper understanding of the infrastructure.

8.2 Future Work

The current research is concentrated on vulnerability scanning of the internal network, like in internal devices and network devices such as switches. In the future, this research could be expanded to include servers, firewalls, and routers. Additionally, it could be extended to the DMZ section of an organization, which contains publicly available services.

References

[1] M. Krishna and D. Seidman, “Why do information security vulnerabilities exist?,” Quora.https://www.quora.com/Why-do-information-security-vulnerabilities-exist (accessed May 03, 2023).

[2] A. MURRAY, “The National Vulnerability Database Explained,” Mend, Dec. 18, 2018.https://www.mend.io/resources/blog/the-national-vulnerability-database-explained/ (accessed May 03, 2023).

[3] Balbix, “What is a vulnerability? Examples, Types, Causes,” Balbix, Sep. 23, 2022. https://www.balbix.com/insights/what-is-a-vulnerability/ (accessed Feb. 01, 2023).

[4] Arcserv, “Is Unpatched Software a Security Risk?,” Arcserve, Jan. 17, 2019. https://www.arcserve.com/blog/unpatched-software-security-risk (accessed Feb. 01, 2023).

[5] Reciprocity, “Security Misconfigurations: Definition, Causes, and Avoidance Strategies,” Risk Optics, Jul. 28, 2022. https://reciprocity.com/blog/security-misconfigurations-how-to-avoid-them/ (accessed Feb. 01, 2023).

[6] SailPoint, “How Compromised Credentials Lead to Data Breaches,” SailPoint, Sep. 30, 2020. https://www.sailpoint.com/identity-library/how-compromised-credentials-lead-to-data-breaches/ (accessed Feb. 01, 2023).

[7] Y. Guez, “6 encryption mistakes that lead to data breaches,” Crypteron, Oct. 20, 2016. https://www.crypteron.com/blog/the-real-problem-with-encryption/ (accessed Feb. 02, 2023).

[8] C. Stouffer, “What is a zero-day exploit?,” us.norton.com, Sep. 03, 2021. https://us.norton.com/blog/emerging-threats/how-do-zero-day-vulnerabilities-work (accessed Feb. 02, 2023).

[9] C. Signing, “Common Software Vulnerabilities in 2022 - Ways to Prevent Them,” Code Signing Store. https://codesigningstore.com/common-software-vulnerabilities (accessed Mar. 01, 2023).

[10] H. Edimo, “Five OS Vulnerabilities,” Medium, Feb. 23, 2021. https://laki-edimo.medium.com/five-os-vulnerabilities-1d9fafb1c87b (accessed Mar. 01, 2023).

[11] NIST, “NVD - Vulnerabilities,” Nist.gov, 2019. https://nvd.nist.gov/vuln (accessed Mar. 01, 2023).

[12] D. Schrader, “Common Open Port Vulnerabilities List,” https://blog.netwrix.com/, Aug. 04, 2022. https://blog.netwrix.com/2022/08/04/open-port-vulnerabilities-list/ (accessed Mar. 01, 2023).

[13]Balbix, “What is Vulnerability Scanning,” Balbix, Jan. 24, 2020. https://www.balbix.com/insights/what-is-vulnerability-scanning/ (accessed Feb. 02, 2023).

[14] N. James, “Top 7 Vulnerability Management Providers With Key Factors - Astra Security Blog,” www.getastra.com, Dec. 05, 2022. https://www.getastra.com/blog/security-audit/vulnerability-management-providers/ (accessed Feb. 03, 2023).

[15] S. Basu, “5 Vulnerability Scanning Types: A Thorough Exploration,” www.getastra.com, Mar. 28, 2022. https://www.getastra.com/blog/security-audit/vulnerability-scanning-types/ (accessed May 03, 2023).

[16] D. Hansen, “5 Reasons Why Every Company Needs A Vulnerability Assessment,” blog.freedmaxick.com, Aug. 25, 2017. https://blog.freedmaxick.com/summing-it-up/5-reasons-why-every-company-needs-a-vulnerability-assessment (accessed May 03, 2023).

[17] R. Singh, “Reasons Why Every Business Needs a Routine Vulnerability Assessment | Indusface Blog,” Indusface, Sep. 29, 2020. https://www.indusface.com/blog/reasons-why-every-business-needs-a-routine-vulnerability-assessment/ (accessed May 03, 2023).

[18] Tenable, “About Tenable,” Tenable®, Oct. 15, 2010. https://www.tenable.com/about-tenable/about-us (accessed Feb. 03, 2023).

[19] TrustRadius, “Tenable.io Reviews & Ratings 2023,” TrustRadius. https://www.trustradius.com/products/tenable-io/reviews?qs=pros-and-cons#comparisons (accessed Feb. 03, 2023).

[20] Tenable, “Nessus Product Family,” Tenable®, May 15, 2019. https://www.tenable.com/products/nessus (accessed Jan. 19, 2023).

[21] Tenable, “Nessus 10.4.x User Guide,” Mar. 2023. Accessed: Jan. 21, 2023. [Online]. Available: https://docs.tenable.com/nessus/10_4/Content/PDF/Nessus_10_4.pdf

[22] Tenable, “NNM Deployment Guide,” Tenable, Feb. 2023. Accessed: Jan. 23, 2023. [Online]. Available: https://docs.tenable.com/nnm/deployment/Content/PDF/NNMDeploymentGuide.pdf

[23] Tenable, “Tenable.io User Guide,” Tenable, Apr. 2023. Accessed: Jan. 25, 2023. [Online]. Available: https://docs.tenable.com/tenableio/Content/PDF/Tenableio_User_Guide.pdf

[24] Tenable, “Tenable.sc 6.0.x User Guide,” Tenable, Mar. 2023. Accessed: Feb. 27, 2023. [Online]. Available: https://docs.tenable.com/tenablesc/6_0/Content/PDF/Tenablesc_UserGuide.pdf

[25] Tenable, “Tenable.io,” Tenable®, Jan. 20, 2017. https://www.tenable.com/products/tenable-io (accessed Jan. 29, 2023).

[26] S. Bairwa, B. Mewara, and J. Gajrani, “Vulnerability scanners-a proactive approach to assess web application security,” International Journal on Computational Science & Applications, vol. 1, pp. 1–12, 2014.

[27] H. Al-Alami, A. Hadi, and H. Al-Bahadili, “Vulnerability scanning of IoT devices in Jordan using Shodan,” presented at the 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), Amman, Jordan, 2017, vol. 2nd, pp. 1–6.

[28] Y. Wang and J. Yang, “Ethical hacking and network defense: choose your best network vulnerability scanning tool,” presented at the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 2017, pp. 110–113.

[29] R. Kushe, “Comparative Study of Vulnerability Scanning tools: Nessus Vs Retina,” Security & Future, vol. 1, Art. no. 2, 2017.

[30] C. R. Harrell, M. Patton, H. Chen, and S. Samtani, “Vulnerability assessment, remediation, and automated reporting: Case studies of higher education institutions,” presented at the 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), Miami, FL, USA, 2018, pp. 148–153.

[31] M. Qasaimeh, A. Shamlawi, and T. Khairallah, “Black box evaluation of web application scanners: Standards mapping approach,” Journal of Theoretical and Applied Information Technology, vol. 96, Art. no. 14, 2018.

[32] A. M. Ugur, E. Altuncu, and K. Bicakci, “A first look at the usability of openvas vulnerability scanner,” presented at the Workshop on Usable Security and Privacy (USEC 2019) - NDSS, San Diego, CA, USA, 2019, pp. 1–11.

[33] R. Amankwah, J. Chen, P. K. Kudjo, and D. Towey, “An empirical comparison of commercial and open-source web vulnerability scanners,” Software: Practice and Experience, vol. 50, Art. no. 9, 2020.

[34] Y. Wang, Y. Bai, L. Li, X. Chen, and A. Chen, “Design of network vulnerability scanning system based on NVTs,” presented at the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 1774–1777.

[35] S. Pandey and A. Chaudhary, “Vulnerability scanning,” TechRxiv (IEEE), India, 2022.

[36] “Port 10101 (tcp/udp),” SpeedGuide. https://www.speedguide.net/port.php?port=10101 (accessed Apr. 24, 2023).

[37] Z. Ware, “Ptakks «Spyware, Malware and Adware Encyclopedia | Cyberlab Technologies,” Ptakks. https://cyberlab.com/spyware-blog/ptakks (accessed Apr. 24, 2023).

[38] “Speed Guide,” Speed Guide. https://www.speedguide.net/port.php (accessed Apr. 24, 2023).