PUSL3190 Computing Individual Project

Project Proposal

The Basic Security Test Application

Supervisor: Dr. Pabudi Abeyrathne

Name: Pasidu A Liyanage
Plymouth Index Number: 10899322
Degree Program: Bsc(Hons) Computer Security

Table of Contents

Chapter 01

1.0 Problem Statement

1.1 Identify the problem

IT auditing firms are responsible for ensuring that the systems they audit are secure and free of vulnerabilities that could expose businesses to cyberattacks or system failures. In this function, auditors must look for flaws in a variety of systems, including networks, software, databases, and hardware components. However, the current state of cybersecurity auditing technologies and processes makes it impossible for these specialists to conduct quick and reliable vulnerability assessments. The necessity to provide consistent, detailed security assessments while meeting accurate timelines puts additional strain on audit teams. A simplified, effective technique for finding and fixing vulnerabilities is critical to maintaining the security of business systems. As a result, the dearth of effective technologies that combine speed, simplicity, and accuracy creates a huge gap that IT audit firms must fill.

1.2 Context

IT auditors are responsible for conducting vulnerability assessments, which frequently involve scanning large, complex networks and systems. This is often a multi-step procedure requiring knowledge of multiple security testing approaches, including penetration testing, network analysis, and system configuration checks. Unfortunately, as many of the procedures involved are manual, these processes take a long time. Each stage consumes time, increasing the overall duration of the audit. When it comes to discovering and patching vulnerabilities, time is sometimes of the essence, since unchecked systems leave themselves open to possible threats. As a result, there is an increasing demand for a solution that enables IT auditors to speed up their examinations while maintaining accuracy and depth.

1.3 Challenges

One of the most significant issues that IT auditors face is the inaccuracies associated with manual testing and report generation. In many cases, auditors must perform independent checks on various parts of the system (such as checking for open ports, checking firewall rules, and scanning for software vulnerabilities) before collecting their results into a report [1]. This manual procedure presents various risks, including human mistake and delayed reporting, which can lead to incomplete audits. Furthermore, preparing a full, detailed report following each audit can be time-consuming and difficult. Furthermore, the complexity of current security tools affects IT auditors' tasks.

While sophisticated tools like Nessus and OpenVAS exist, they can have steep learning curves and require extensive setup before they can be utilized successfully . As a result, there is an obvious need for a solution that simplifies both the testing and reporting processes while removing unnecessary complexi

1.4 Impact without a solution

The lack of an effective solution for vulnerability scanning and audit report generation can have major consequences for companies and their IT audit teams. Audits take longer to perform without a simplified solution, which increases the amount of time important system vulnerabilities stay unpatched. The greater chances that hackers or malicious individuals may exploit mistakes. Furthermore, auditors may overlook critical security issues due to limitations

on time or the huge amount of data supplied by complex technologies. As a result, companies may pass audits while aware that their systems still contain severe vulnerabilities.If security events occur following an incomplete audit, this may result in regulatory noncompliance, fines, or reputational damage. In the long run, failing to detect and resolve risks quickly can destroy trust between companies and their customers or partners. As a result, it is vital to install a solution that allows IT auditors to conduct thorough, timely assessments of system vulnerabilities while limiting the risk of oversight and ensuring that firms maintain strong cybersecurity defenses.

Chapter 02

2.0 Project Description

2.1 Introduction to the project

The Basic Security Test Application is a project designed to simplify the process of identifying and controlling system vulnerabilities for IT audit firms. The purpose is to provide a simple, efficient tool for IT auditors to perform security assessments on various company systems without having to rely on unnecessarily complex software. This tool will assist auditors in evaluating the security posture of networks, databases, and applications by running many tests concurrently and creating automatic reports based on the results.

By automating critical steps in the auditing process, the Basic Security Test Application will dramatically minimize the time necessary to conduct system tests. It will also increase the audit process's accuracy and completeness by requiring no personal interaction to perform many security checks. The application is intended to give an intuitive, user-friendly interface that enables auditors to use the tool with little training or technical knowledge. This will allow IT auditors to concentrate on the analysis and recommendations for system improvements rather than learning complex software tools.

The tool's basic function is to detect vulnerabilities such as open ports, obsolete software versions, weak configurations, and potential points of entry for malicious actors. After the tests are completed, the system will automatically provide a full report outlining the vulnerabilities detected, as well as recommendations for how to remedy them. The report can then be simply distributed to clients, resulting in faster decision-making and more effective vulnerability management.

2.2 Features

Multiple Security Tests Performed Concurrently:

The application will allow IT auditors to execute many security tests together. Port scanning, for example, can be used in combination with vulnerability detection algorithms that look at program versions and security configurations. The application will save time by running many tests concurrently, allowing auditors to do their tasks more quickly and efficiently.

This function is especially beneficial for large systems with numerous components, where manual testing could take hours or even days. Auditors may gain a comprehensive perspective of the system's security in significantly less time by running multiple tests at the same time.

Furthermore, the application will handle test priority, ensuring that key areas are addressed first.

Automatic Report Generation:

One of the primary advantages of the Basic Security Test Application is its ability to generate results automatically whenever tests are performed. These reports will provide a summary of the vulnerabilities discovered, the severity of each issue, and recommendations for resolution. The automatic report creation option will save auditors a substantial amount of time when compared to manually compiling test findings into a cohesive document.

Reports will be configurable, allowing auditors to adjust the content to individual client requirements. For example, some clients may need comprehensive technical explanations, while others may prefer high-level overviews. The ability to generate reports automatically will help to assure consistency and professionalism in the auditing process, consequently improving the overall quality of service given by IT audit firms.

Simple and Intuitive User Interface:

The application's design focuses simplicity and employ. Many existing security products have complex interfaces that require extensive technical knowledge to use successfully. In contrast, the Basic Security Test Application will have a simple, user-friendly interface that will allow auditors to quickly explore the tool, configure tests, and analyze results.

The interface will be designed to guide users through the procedure step by step, decreasing the possibility of errors and ensuring that even auditors with low technical skills can use the tool comfortably.

2.3 Target Audience

The main audience for this project is IT audit firms and their auditors, who are in the role of analyzing the security posture of various systems and assuring compliance with security standards. The application is intended to assist auditors execute their tasks more efficiently, allowing them to immediately discover weaknesses and prepare reports for customers.

The application is ideal for businesses that do frequent system audits or require a simple, dependable solution for conducting basic security evaluations. IT auditors that work with a variety of clients and systems will benefit greatly from an easy-to-use program that provides quick results.

2.4 Project Objectives

1. **Quick, Automated Method to Scan for System Vulnerabilities:**

The major goal of the project is to develop a program that can quickly and automatically scan company systems for vulnerabilities. Traditional vulnerability scanning approaches frequently rely on manual steps, which can be slow and costly. By automating the procedure, the Basic Security Test Application enables auditors to complete audits in a fraction of the time while maintaining accuracy.

Automation will ensure that all critical sections of the system are properly checked for flaws, lowering the possibility of human mistake and detecting vulnerabilities as soon as feasible.

2. **Provide an easy-to-use interface for IT auditors with minimal training:**

A primary goal is to build the program with usability in mind. Many current security tools are extremely complex, needing significant instruction before clients can utilize them effectively. The Basic Security Test Application, on the other hand, will be created with an intuitive user interface that will allow auditors to quickly learn how to use the tool without the need for specialist training.

This will make the tool more accessible to a large variety of users, allowing even individuals with less technical knowledge to conduct successful vulnerability assessments.

3. **Allow multiple security tests to run simultaneously:**

Another key goal of the project is to enable IT auditors to execute various security tests simultaneously. This will significantly prevent the time needed to complete audits. Instead, the system will automatically execute numerous checks at once, ensuring that all important aspects of the system's security are evaluated simultaneously.

This capability will be especially useful for large or complex systems, where performing individual tests can take a long time. The tool's concurrent testing feature will increase audit efficiency and assist IT auditors in meeting tight deadlines.

4. **Simplify audit documentation by creating detailed automated reports:**

The ultimate goal is to ensure that the application generates detailed, automatic reports following each audit. These reports will summarize the vulnerabilities discovered, prioritize them by severity, and make actionable suggestions for remedy. By automating this procedure, the technology will save auditors time while also ensuring that clients receive clear, professional reports following each audit.

2.5 Project Keywords

- Vulnerability Scanning
- IT Audit Automation
- Security Testing
- User-Friendly Interface
- Automated Reporting

Chapter 03

3.0 Research Gap and Literature Review

3.1 Literature Review

To better understand the current state of IT security testing tools and auditing processes, this literature review analyzes available technologies, their capabilities, and the challenges that IT auditors . The environment of information systems security auditing is continually changing as technology progresses and cyber attacks become more complex. The following overview takes on research papers to highlight the importance, technique, and problems of conducting effective information system security audits.

Security audits are crucial tools for firms to analyze their information security postures. emphasizes the need of regular audits in establishing a security baseline, allowing vulnerabilities to be detected and addressed before they are exploited [2].
Similarly, highlights that cyber security audits conduct a thorough evaluation of an organization's IT infrastructure, finding dangers and weak links in the system [3].

Focus on vulnerability audits:
Describe the importance of using various tools and strategies to achieve comprehensive coverage of all components inside an information system [4], [5]. The program makes it simple for auditors to understand how the tool increases productivity, accuracy, and reliability in finding vulnerabilities and emphasizing crucial areas.
IT audit processes include solutions that automate basic audit operations including scanning for open ports, firewalls, and logging to detect unauthorized connections [6], [7].
address an application module for wireless network security testing, which simulates potential attack pathways and assesses vulnerability [6].

The importance of port scan detection:
This reconnaissance phase includes methods like port scanning, in which attackers analyze TCP and UDP ports to find open communication channels and system weaknesses. Open ports, such as entry points in a building, can expose systems to unauthorized access, and systems with numerous open ports are generally more vulnerable. However, fewer open ports may represent a larger security risk in some configurations. [8]. This study underlines the significance of expansive testing for open ports on computer networks for assessing security [9].

Challenges and Technical Depth:
The specific nature of the skills required for these audits involves auditors having extensive technical knowledge.IS auditors require specialized abilities to effectively analyze technical observations. In order to accomplish this, the application should have a training module. Provides information and courses to help auditors improve their technical abilities in IS auditing. Includes case studies and examples that demonstrate frequent challenges in IS auditing and how to handle them.
Creates thorough audit reports that summarize results, make actionable recommendations, and identify areas for improvement. Management uses images (charts and charts) to convey data in a clear and effective manner [10], [11].

Modular framework for security scanning:
NSS's modular configuration offers focused scanning and auditing capabilities, making it both efficient and user-friendly. It attempts to identify system vulnerabilities in the same way as NSS can find and resolve exploits and information leaks [4].
Host Scan: Used to detect live hosts.
Port Scan: Detecting open and closed ports.
Pinging and NSLookup are used to test connectivity and resolve DNS information.

User Awareness and Training:
Several articles, notably, emphasize the crucial significance of user awareness and training in improving company safety [6], [9]. Regular discussions and presentations regarding security policies may significantly improve staff knowledge and compliance, lowering the likelihood of human errors that result in security breaches.

Integration with Governance:
Effective IS audits help to improve corporate governance by giving management with the insights it needs to make sound decisions. [3] explores how cyber risk management frameworks can bridge the gap between technical cybersecurity practices and executive governance, establishing a security culture throughout the firm.

Future Research Directions:
investigating the usefulness of future technologies, such as artificial intelligence and machine learning, in improving auditing processes and giving predictive capabilities against potential cyber threats [5], [11].

3.2 Research Gap

The focus is on risk scanning technologies, automated reporting systems and simple user-friendly interfaces designed to audit companies.

Nmap and OpenVAS are widely used in IT audits to identify network vulnerabilities. However, it demands a thorough understanding of command-line interfaces, which may be difficult for non-technical auditors. The project will provide a simple, streamlined tool that can perform multiple security checks (e.g. port scanning and vulnerability verification) simultaneously, allowing auditors to perform audits quickly without requiring extensive technical knowledge.
Common vulnerabilities can improve the basic security testing application's capacity to efficiently find security issues.
Gap detection: While tools like Nmap and OpenVAS provide extensive scanning capabilities, they are difficult for non-technical people to understand. Provides IT auditors with a simple application without overshadowing essential security information.

Existing security testing applications provide perfect technical results but an automated reporting system aims to automatically identify problems and provide the necessary guidance in a short period of time.

By using these principles, the initial security testing application can significantly reduce audit timeframes and increase the efficiency of IT audit processes, providing auditors with a comprehensive security assessment in a fraction of the time required for sequential testing.

Gap detection: An automated reporting system detects hazards while providing simple, clear and actionable information. Reports should be simple to understand and directly applicable to an audit setting, with technical terminology understandable to auditors or clients.

The next goal of the project is to create a user-friendly interface that allows auditors to perform tests and generate reports with minimal effort. The interface will be user-friendly, reducing the need for extensive instructions and making the audit process more efficient.

Wireshark, for example, is an industry-standard network protocol analyzer, but its interface can be intimidating to new users with deep packet inspection.

Gap detection: IT auditors will benefit from a system with a simple interface that facilitates risk testing and reporting.

These systems automate report generation, reducing the need for manual report writing, which can be time-consuming and error-prone. Automated reports that prioritize risks based on severity and provide remediation recommendations have been shown to accelerate client decision-making and increase overall audit efficiency.

Chapter 04

4.0 Requirements Analysis

4.1 Knowledge Required

- **Basic Vulnerability Scanning Techniques:**

Vulnerability scanning is the process of detecting vulnerabilities in a system's security. These flaws could be the result of outdated security protocols, open ports, or misconfigurations. Common techniques include port scanning , service version detection (to discover obsolete or insecure software), and network scanning . Understanding these strategies enables you to create the backend logic that scans systems for these vulnerabilities. It is necessary to be familiar with tools such as Nmap and to understand basic networking concepts such as IP addresses and TCP/UDP protocols.

- **Familiarity with Programming**

The application's backend will be developed mainly using Python. You'll need to be able to develop efficient, clean code as well as understand fundamental cybersecurity concepts. Knowledge of Python libraries such as os for interfacing with the system, socket for network communications, and libraries for communicating with tools such as Nmap is necessary. You should also grasp topics like encryption, firewalls, authentication, and how different forms.

- **Learning about UI Design**

Auditors with a variety of technical expertise can simply use the application. UI design includes knowing the principles of good layout (where features such as buttons, input fields, and results are placed), usability (how easy a user can learn and use the system), and accessibility. Auditors, for example, should be able to run tests with just a few clicks and view results that are clearly displayed.

4.2 Required Devices

Network Test Environment:

Running the security tests requires a controlled environment . This environment simulates realworld network infrastructures, allowing you to test various vulnerability scenarios such as open ports, Firewalls that have not been configured properly, Services and applications are outdated. Why It's Special: Simulating real-world network environments. It is critical for ensuring that the Basic Security Test Application appropriately detects vulnerabilities in actual network configurations.

Safe Testing: A managed network ensures that you do not mistakenly disrupt live systems while testing.

4.3 Tools and Technologies

Development tools: It contains a variety of libraries for networking, data processing, and creating reports, making it suitable for this project. You'll use Python to handle backend logic, perform security tests, analyze results, and integrate with vulnerability assessment programs. Its simplicity enables you to quickly design and iterate on the application.

Scanning Libraries for Vulnerability Detection: Nmap is an useful open-source program for network scanning and vulnerability identification. It assists auditors in identifying open ports, the services that run on those ports, and any potential vulnerabilities linked with them. The application will incorporate Nmap using Python libraries like python-nmap or by performing Nmap commands from within the Python script. You might also look into other scanning libraries, such as OpenVAS or Nikto, depending on the specific needs of your audit. These libraries will enable the program to automatically perform scans and display the results to the user.

Tools for Creating Automated Reports: Once the security tests are performed, the application must create reports. Python's FPDF library allows you to programmatically generate PDF reports including tables, pictures, and text with structure. This is essential for producing professional-looking audit reports that auditors may submit to their clients. The reports will include information about the tests performed, vulnerabilities discovered, and recommendations for mitigation. FPDF allows you to generate a PDF directly from the program, without the need for additional applications.

4.4 Learning Requirements

Exploring Different Network Security Protocols:

Because your application will evaluate system vulnerabil ities, you must understand the underlying network protocols that control how data is tra nsported across networks. For example:

- TCP/IP is the primary communication protocol for most networks, and understnding its structure will aid in identifying common vulnerabilities.
- SSL/TLS protocols are used to protect data in transit across the internet. Understanding how these protocols perform is critical for developing tests that determine whether systems use encryption properly.
- SSH is a secure mechanism for access remote systems. Familiarity with SSH will help you to scan for vulnerabilities in its configuration.

Chapter 05

5.1 system development

5.1.1 Backend development

This phase consists of coding the application's fundamental functions. For your project, this involves developing the vulnerability scanning tool, integrating other libraries, and processing the results.

- Vulnerability Scanner Development: Writing Python programming to automate vulnerability tests.
- Integration of libraries: To find vulnerabilities, use tools such as Nmap or other security libraries.
- Backend Logic: Developing the logic that controls the scanning process and returns results.
- Outcome: A functional backend that manages security tests and data for reporting.

5.1.2 Frontend development

Create the graphical user interface that will allow IT auditors to interact with the system. The goal is to make the interface easy to use, allowing users to start scans and read reports with little effort.

- Designing UI Layout: Specifying how elements like as buttons and forms will appear and function.
- Usability testing ensures that the interface is simple to use and understand.
- Outcome: A clean, simple, and functional interface that allows IT auditors to execute scans and evaluate results.

5.2 System Design

This stage focuses on defining the system's overall structure, as well as how different components interact. The Basic Security Test Application requires planning out how the vulnerability scanner, reporting system, and user interface (UI) will work together.

- Scanning Engine: The core module responsible for security scans such as port scanning and vulnerability detection.
- Automated Report Generator: This component generates and formats reports based on scan results.
- User Interface (UI): A simple, easy-to-use interface that enables IT auditors to interact with the system.

5.3 Automated System

This phase focuses on creating automated reports using the scanning data. The reports will summarize found vulnerabilities and provide critical information to IT auditors. PDF Generation: Use libraries to convert results into professional, downloadable PDF reports.
Outcome: A fully operational reporting system that generates professional reports for each audit.

5.5 Testing and evaluation

After creating the basic features, it is necessary to extensively test each component to ensure that the system functions as designed.

- Unit testing involves testing individual components such as the scanning engine and report generator.
- Integration testing ensures that all components (backend, frontend, and reporting) perform easily together.
- Performance testing involves evaluating how the application handles several concurrent scans and huge datasets.
- Bug Fixing involves debugging and addressing any difficulties that appear while testing.
- Outcome: A reliable, bug-free system that performs well under a variety of testing situations.

5.6 Deployment and Implementation

Making the application ready for use in the real world. This step involves final testing in the target environment, application deployment, and any necessary end-user documentation.

- Final User Testing: IT auditors can test the application in real-life situations.

Chapter 06

6.1 project timeline

1. Planning **(October - November)**

Activities include setting project goals and parameters, conducting preliminary research and literature analyzing, and compiling a list of the necessary tools and resources. Conduct a literature review on relevant cybersecurity frameworks, vulnerability scanning tools, and IT auditing procedures.Research and System Design.

Outcome: This phase gives you a foundation of information, allowing you to determine the project's goals and ensure your solution meets the specific demands of IT auditors.

2. System Design **(November - December)**

Activities include designing and developing the system architecture, as well as providing simple user interface designs. System architecture takes two weeks, while wireframe design takes four weeks.

- Determine the overall structure of the application.
- Divide the system into basic components such as the scanning tool, automatic reporting and user interface.
- Choose the technologies and libraries (such as Python, FPDF, and Nmap) that will be used to build each module.

3. Phase of Development **(November - December)**

This phase focuses on developing your system design into a working application. You'll start by developing the application's basic functionality, making sure that the main modules are correctly integrated. Create the vulnerability scanning module, which will enable the program to perform a variety of security tests. Create the user interface, making it simple and intuitive for IT auditors. Implement an automatic reporting system that generates PDFs based on test results [1].

4. Testing **(December-January)**

Thoroughly test the program to ensure that it functions properly, discovers vulnerabilities, and generates accurate reports. Tasks include doing functional and nonfunctional testing, debugging errors, and improving the system based on feedback from early tests. Evaluate the application's performance in various network situations. Outcome: A stable and optimized program that is free of major problems and ready for usage in the real world.

5. Implementation **(January-February)**

Finalise the application for use, optimize it for efficiency, and make it ready for use.
Tasks: Fine-tune the system based on testing results, improve the automatic reporting feature, and guarantee that all security tests run concurrently avoiding problems with performance.
Outcome: A fully functional, user-friendly application that works the needs of IT auditors.

6. Phase of Documentation **(February-March)**

Write the final report and provide project documentation.

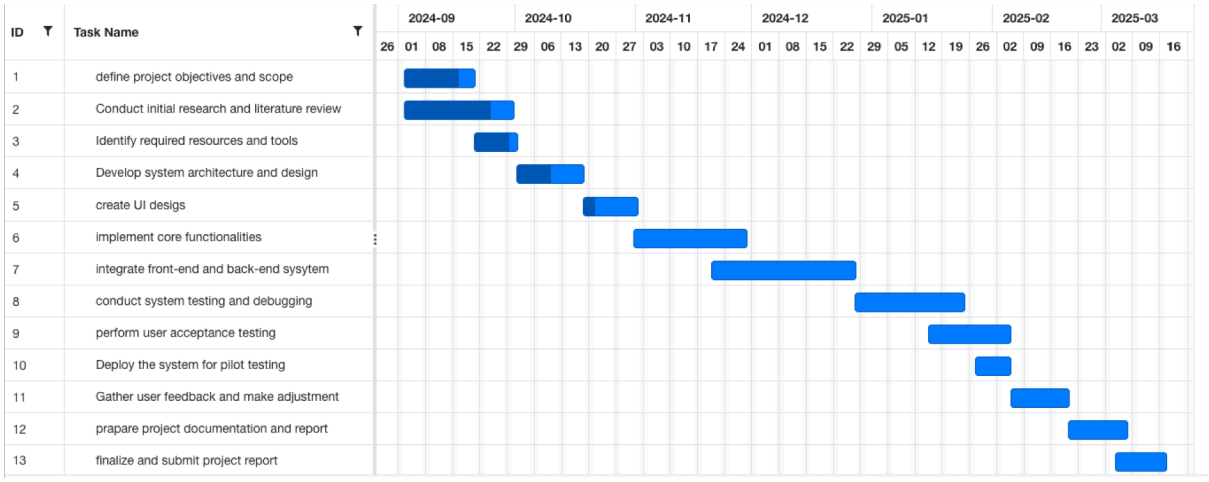
Documentation Create a final project report that includes technical documentation and screenshots of the user interface. Ensure that all stages of the development process are well explained.

Outcome: A complete project report outlining your efforts and explaining how the program works.

7. Phase of Submission

The successful submission of the final year project marks the end of the development and evaluation proces.

6.2 Gantt chart



Reference

[1] S. Kamara, S. Fahmy, E. Schultz, F. Kerschbaum, and M. Frantzen, "Analysis of Vulnerabilities in Internet Firewalls," Mar. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404803003109

[2] E. C. Lo and M. Marchand, "SECURITY AUDIT: A CASE STUDY," Niagara Falls, ON, Canada, May 2004. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1344989

[3] K. D. Jadhav, "THE ROLE OF CYBER SECURITY AUDITS THE ROLE OF CYBER SECURITY AUDITS IN MANAGING COMPANY SYSTEMS AND APPLICATIONS," Jan. 2023. [Online]. Available: https://www.researchgate.net/publication/367559332_THE_ROLE_OF_CYBER_SECURITY_AUDITS

[4] G. Murali, M. Pranavi, Y. Navateja, and K. Bhargavi, "NETWORK SECURITY SCANNER," 2011. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/90636070/2af5cc555cae555660bb4226fab380a43594-libre.pdf?1662272260=&response-content-disposition=inline%3B+filename%3DNetwork_security_scanner.pdf&Expires=1730206193&Signature=JXcwMhMP5aGKSFIRlA03XQmd--AeCJvEeeji9C3Sm76svApm~d-yaAif7MZnAU9SirtkTktlpyJ8ZjUrgmdlzFDyh50YpRfAC2ILLx7suGETBBY0eJhX5Gg2iZ6uwuxYutWHZfUHSJgL1HmpGJfTZAM7rkieuNaZYohC8MZoNLCkdyyt3b1YySaB7TBSzaSUwYLsp33k1u4y~NHyUas~FfU-ry8k4jjEGpFH~pcIusDMOdsBh6OWyFENMjPeXENllyyX0OvWA8P4eC5stKRlHBO1nWQxRA3di94Y1J9y24ck8EAJdgLyVD650rmeOwgbBj3IKacbakgDgFhp1SPluQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[5] A.-M. Suduc, M. Bîzoi, and F. Gheorghe FILIP, "Audit for Information Systems Security," 2010. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8fe27c55f5298a4a67a8204f4596b3609440556e

[6] A. Aarthy Devi, A. K. Mohan, and M. Sethumadhavan, "Wireless Security Auditing: Attack Vectors and Mitigation Strategies," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 674–682. doi: 10.1016/j.procs.2017.09.153.

[7] Bayu Rima Aditya; Ridi Ferdiana; Paulus Insap Santosa, *Toward Modern IT Audit–Current Issues And Literature Review*. Yogyakarta, Indonesia: IEEE, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8528627

[8] P. Boyanov, "A COMPREHENSIVE SCANNING FOR OPEN, CLOSED AND FILTERED PORTS IN THE COMPUTER SYSTEMS AND NETWORKS," *Original Contribution Journal scientific and applied research*, vol. 23, 2022.

[9] Jayant Gadge and Anish Anand Patil, *Port Scan Detection*. New Delhi, India: I E E E, 2008. Accessed: Feb. 02, 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4772622

[10] V. N. Sayankar and V. N. Sayankar, "A Review on Information Systems Audit," *Research J. Engineering and Tech*, vol. 4, no. 3, Sep. 2013, [Online]. Available: https://www.researchgate.net/publication/381653679_A_Review_on_Information_Systems_Audit

[11] S. Rao Vemula, "Automating Security Testing: Strategies for Vulnerability Scanning, Penetration Testing, and Compliance Checks," *International Research Journal of Engineering and Technology (IRJET) e-International Research Journal of Engineering and Technology*, Jul. 2024, [Online]. Available: www.irjet.net

[12] S. Rao Vemula, "Automating Security Testing: Strategies for Vulnerability Scanning, Penetration Testing, and Compliance Checks," *International Research Journal of Engineering and Technology (IRJET) e-International Research Journal of Engineering and Technology*, Jul. 2024, [Online]. Available: www.irjet.net

[13] J. P. Seara and C. Serrão, "Intelligent System for Automation of Security Audits (SIAAS)," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 11, no. 1, Oct. 2024, doi: 10.4108/eetsis.3564.

[14] T. Vieira and C. Serrão, "Web Applications Security and Vulnerability Analysis Financial Web Applications Security Audit-A Case Study," Dec. 2016. [Online]. Available: https://repositorio.iscte-iul.pt/bitstream/10071/12991/5/Web-Applications-Security-and-Vulnerability-Analysis-Financial-Web-Applications-Security-Audit%E2%80%93A-Case-Study.pdf

[15] A. Ziro, S. Toibayeva, S. Gnatyuk, A. Imanbayev, M. Iavich, and Z. Zhaybergenova, "Research of the Information Security Audit System in Organizations," in *SIST 2023 - 2023 IEEE International Conference on Smart Information Systems and Technologies, Proceedings*, Astana, Kazakhstan: IEEE, May 2023, pp. 440–444. doi: 10.1109/SIST58284.2023.10223557.