Karal, M. (2026). Nessus, Qualys ve OpenVAS Zafiyet Tarama Araçlarının İşletim Sistemleri ve Web Uygulamalarındaki Zafiyet Tespit Kabiliyetleri. EMO Bilimsel Dergi, 16(2), 103-112.

Makale Türü: Araştırma Makalesi
Makale Gönderim Tarihi: 20 Şubat 2026
Makale Kabul Tarihi: 28 Mayıs 2026

Scanning Capabilities of Nessus, Qualys and OpenVAS Vulnerability Scanner Tools for Vulnerable Operating Systems and Web Applications

Nessus, Qualys ve OpenVAS Zafiyet Tarama Araçlarının İşletim Sistemleri ve Web Uygulamalarındaki Zafiyet Tespit Kabiliyetleri

Mehmet Kara[1]

0000-0001-7312-0503

[1], Department of Software Engineering, Kocaeli Health and Technology University, Kocaeli, Türkiye
mehmet.kara@kocaelisaglik.edu.tr

Öz

Web uygulamaları, kullanıcı ve sistemler için ilk karşılama arayüzü olduğundan kritik atak yüzeyi olarak değerlendirilir. Bu yüzden bunların güvenlikleri kritik önemdedir. SQL enjeksiyon, siteler arası betik çalıştırma (Cross-Site Scripting-XSS) ve güvensiz kimlik doğrulama mekanizmaları gibi yaygın web uygulaması zafiyetleri, saldırganların hassas verilere ve iç ağdaki kritik servislere doğrudan erişim elde etmesine olanak sağlayabilmektedir. Bu nedenle, zafiyet tarama araçlarının web uygulaması güvenlik zafiyetlerini etkin bir şekilde tespit edip raporlayabilmesi büyük önem taşımaktadır. Zafiyet tarama araçları, bilgi sistemlerinde kullanılan yazılım ve donanımlardaki risklerin tespit edilmesinde yaygın olarak kullanılmaktadır. Ancak bilgi sistemlerde kullanılan yazılım ve donanımların çeşitliliği nedeniyle, zafiyet tarama ve tespit araçları birbirinden oldukça farklı yeteneklere sahiptir. Bu yetenek farklılıkları, zafiyet tarama araçlarının karşılaştırılmasını güçleştirmektedir. Bu çalışmada, genel bilgi sistem ve web uygulaması zafiyet tespiti için gerekli karşılaştırma kriterleri belirlendikten sonra, yaygın olarak kullanılan Nessus, Qualys ve OpenVAS zafiyet tarama araçları incelenmiştir. Elde edilen sonuçlar, Belli zafiyetleri içeren işletim sistemleri ve web uygulamaları üzerinde ilgili zafiyet tarama araçları çalıştırılarak performansları karşılaştırılmıştır.

Anahtar Kelimeler: Zafiyet tarayıcı, CVE, Nessus, Qualys, OpenVAS, Web uygulaması

Abstract

Web applications, which serve as the primary interface between users and organizational systems, represent a particularly critical attack surface that requires specialized attention to ensure their security. Common web application vulnerabilities, such as SQL injection, cross-site scripting (XSS), and insecure authentication mechanisms, can provide attackers with direct access to sensitive data and backend systems. Therefore, it is crucial for vulnerability scanning tools to effectively detect and report web application security flaws. Vulnerability scanning tools are widely used to detect risks in the software and hardware used in information systems. However, owing to the variety of software and hardware used in information systems, vulnerability scanning and detection tools have very different capabilities. These differences in capabilities make it difficult to compare the vulnerability scanning tools. In this study, after determining the necessary comparison criteria for general information system and web application vulnerability detection, the widely used Nessus, Qualys, and OpenVAS vulnerability scanning tools were examined. The results were compared by running the vulnerability scanning tools on operating systems and web applications known to contain vulnerabilities.

Keywords: Vulnerability Scanner, CVE, Nessus, Qualys, OpenVAS, Web Application

1. Introduction

Vulnerability scanning tools play a critical role in both personal and commercial cyber security contexts. These tools employ various techniques to identify security weaknesses in information systems and mitigate the risks associated with such vulnerabilities. Information systems are evaluated using testing approaches commonly classified as black-box, gray-box, and white box testing. In black-box testing, no prior information about the system is provided to the testing team; gray-box testing involves partial disclosure of system details; and white-box testing grants full access to system information. In all penetration testing scenarios, vulnerability scanning tools are an indispensable resource for security analysts. They deliver essential security-related data, including known vulnerabilities—often referenced through Common Vulnerabilities and Exposures (CVE) identifiers—misconfigurations, open ports, operating system fingerprints, and service detection results.

Beyond the provision of raw vulnerability data, these tools enable administrators to generate comprehensive reports that present a holistic view of system security. Such reports typically include the severity levels of the identified vulnerabilities, their sources, recommended mitigation strategies, and exploitability assessments.

Modern information system infrastructures incorporate a wide range of technologies and paradigms, including cloud computing, mobile computing, the Internet of Things (IoT), industrial control systems, wireless networks, and systems operating over the Transmission Control Protocol/Internet Protocol (TCP/IP) stack. This technological diversity significantly increases the complexity of the vulnerability assessment. Consequently, specialized vulnerability scanning tools have been developed to target specific domains, such as cloud environments, IoT ecosystems, mobile platforms, databases, and web applications. This study focuses on vulnerability scanning tools designed to provide broad coverage

across Information Technology (IT) systems and web applications.

In this study, three prominent vulnerability scanning tools—Nessus, OpenVAS, and Qualys—were comparatively evaluated using two systems with known security weaknesses.

OpenVAS is a freely available, open-source vulnerability scanning tool that performs network-based scanning when it is deployed on Linux platforms. Developed with strong community support, OpenVAS distinguishes itself from many commercial alternatives by offering comprehensive functionality at no additional cost. Although it provides fewer visual reporting features than some proprietary tools, its reports include port-based vulnerability classifications, Common Vulnerability Scoring System (CVSS) metrics, vulnerability names, detection reliability scores, impact analyses, detection methodologies, and remediation recommendations. Despite its relatively complex installation and usage processes, OpenVAS remains one of the most capable free vulnerabilities scanning solutions available [1].

Nessus is one of the most widely adopted vulnerability scanning tools in the cybersecurity domain. Although a limited community edition is available at no cost, the professional version enables more extensive and in-depth scanning capabilities. Nessus supports a broad range of scanning types, including basic and advanced network scans, web application assessments, mobile system scans and malware detection. In addition to identifying vulnerabilities, it is effective in detecting security misconfiguration. Its graphical user interface facilitates efficient IP- and network-based scanning of the QR code. The scan results include detailed vulnerability descriptions, remediation suggestions, risk assessments, references, and plug-in outputs. Nessus is particularly notable for its high detection accuracy and low false-positive rate [2], [3].

Qualys is a commercial, cloud-based vulnerability scanning solution recognized for its extensive CVE coverage, modular add-on architecture, and low false-positive rates. Its advanced reporting framework provides detailed vulnerability information, including identifiers, categories, associated CVEs, affected services and users, threat levels, remediation guidance, compatibility considerations, and exploitability metrics. QualysGuard Enterprise further enhances vulnerability assessment by assigning severity scores ranging from 1 to 5 and supplying supplementary insights related to verification, potential impact, and applicability of vulnerabilities.

To ensure the comparability of the scanning results, experiments were conducted on the Metasploitable II and Kevgir operating systems in accordance with predefined evaluation criteria. Metasploitable II is a deliberately vulnerable platform containing security flaws at both the operating system level and within hosted applications and services, and it is widely used in penetration testing. Similarly, Kevgir includes known vulnerabilities across its operating systems and application stacks. Prior to the vulnerability assessment, all vulnerable systems, web applications (such as DVWA), and critical services were updated to their latest available configurations.

The remainder of this paper is structured as follows: Section 2 reviews the related literature. Section 3 identifies the system and web application features required for effective vulnerability scanning. Section 4 presents a performance evaluation of the vulnerability scanners for system and web application assessments. Finally, Section 5 concludes the study and outlines the directions for future research.

2. Related Works

Vulnerability testing tools were primarily employed by security testing firms and individual attackers. However, with the widespread digitalization of business processes, organizations have increasingly adopted these tools for routine and systematic vulnerability assessments. This shift has led to the integration of additional capabilities into vulnerability scanning tools, including compliance with legal and regulatory requirements, support for periodic and automated scanning, and the ability to assess sensitive information across networks and systems. These functional enhancements are also reflected in the advanced reporting mechanisms provided by modern scanners.

In recent years, numerous studies have been published that compare the functionalities and effectiveness of different vulnerability scanning tools. Given the diversity of products and evaluation criteria used across these studies, it has become necessary to first define a set of common characteristics for vulnerability scanners. Accordingly, this study identifies shared evaluation properties and conducts a comparative analysis of Nessus, Qualys, and OpenVAS based on these criteria.

Kushe conducted a comparative study of the Nessus and Retina vulnerability scanning tools, focusing on scan duration, vulnerability detection capability, and update frequency. The results indicated that Nessus exhibited slower performance when web application scanning features were enabled, but achieved faster scan times when these features were disabled [2].

Qualys and Microsoft Defender Endpoint (MDE) vulnerability scanning tools were analyzed across three phases related to update responsiveness by Boonchuay et al. These phases examined the speed of vulnerability patch detection, the detection capability of the scanning tools, and the time required for patches to become available within the scanners. The authors concluded that the overall performance was suboptimal and emphasized the need for automated vulnerability scanning and patch verification processes [4].

Chalvatzis et al. compared Nessus, OpenVAS, and the Nmap Scripting Engine (NSE), focusing on unique and shared CVEs, risk assessment capabilities, and scripting support. The study concluded that all three tools were effective in risk assessment, noting that Nessus could have been considered the most effective option had it remained freely available. Nonetheless, the authors emphasized the significance of all three tools in vulnerability detection [5].

Muin et al. utilized Nessus to audit the network security of a university campus. The primary objective was to identify vulnerabilities in the university's website and to develop an application capable of monitoring system security based on the identified weaknesses [6].

Öztürk and Kara identified the fundamental features required for effective vulnerability scanning tools through a comparative analysis. Using Nessus, Qualys, and OpenVAS, they evaluated these features on the Metasploitable operating system and visualized the comparative results to demonstrate tool performance [7]. In this study, in addition to vulnerability assessments conducted on the Metasploitable II system, the vulnerable Kevgir operating system was also scanned and analyzed using the Qualys, Nessus, and OpenVAS vulnerability scanning tools. Furthermore, the web application vulnerability detection capabilities of these tools were also evaluated.

Recent studies have also explored the use of large language models for vulnerability detection. Fu et al. conducted a

comprehensive evaluation of ChatGPT across four common vulnerability-related tasks: function-level and line-level vulnerability prediction, vulnerability classification, severity estimation, and vulnerability repair [8]. Similarly, Zhang et al. compared ChatGPT with state-of-the-art language models specifically designed for software vulnerability analysis [9].

Zukran and Siraj proposed a methodology for comparing open-source automated vulnerability scanners based on detection coverage and precision rate, drawing on evaluation techniques. Vulnerabilities were deliberately injected into web applications and subsequently scanned. The study evaluated OWASP ZAP and Skipfish, concluding that OWASP ZAP outperformed Skipfish by approximately twofold in precision rate while achieving comparable detection coverage, particularly excelling in identifying high-severity vulnerabilities [10].

Moreira et al. developed an automated vulnerability detection system and validated its effectiveness across multiple vulnerable platforms and applications, demonstrating its adaptability and reliability in diverse testing environments [11]

Santoso et al. investigated the use of vulnerability assessment tools to support developers in identifying and addressing security flaws in both newly developed and existing websites. Their experimental system supported on-demand and scheduled vulnerability assessments and generated comprehensive reports upon completion. The findings highlight the system's contribution to improved vulnerability management and prioritized remediation strategies [12].

The Common Vulnerability Scoring System (CVSS) is a numerical framework used to represent the severity of vulnerabilities rather than direct risk assessment. Multiple versions exist, including CVSS v2.0, v3.x, and v4.0. CVSS scores constitute a core component of CVE records, as they quantify the criticality of vulnerabilities affecting information system assets [13]. Vulnerability scanning tools typically associate detected vulnerabilities with CVE identifiers. Due to variations in severity classification across tools, this study standardizes comparisons by categorizing vulnerabilities based on CVSS scores derived from CVE data, specifically utilizing CVSS v3.x.

Bhardwaj and Saraswat conducted a comparative evaluation of Qualys and Nessus across several dimensions, including scalability, accuracy, cloud compatibility, and cost. Their findings indicate that Qualys is better suited for large-scale, cloud-centric infrastructures, whereas Nessus performs more effectively in on-premises network audits and detailed scanning tasks. The study underscores the importance of a standardized vulnerability assessment methodology—ranging from initial planning to continuous monitoring—and discusses the trade-offs between commercial solutions and free, open-source alternatives [14].

Keijou A. and Bekaroo G. in their paper, critically reviewed and analyzed the main vulnerability scanners in the context of WLANs. As part of the research, four tools, such as the Nessus vulnerability scanner, OpenVAS, Nexpose, and GFI LanGuard were examined through comparative analysis, and it was concluded that each scanning tool handles different vulnerabilities, with the Nessus vulnerability scanning tool being faster and detecting more vulnerabilities. In addition, according to the article, it is also said that the necessity of WLAN security and vulnerability scanning tools may be insufficient due to situations such as old version protocols and the inability to detect fake access points [15].

Web applications are continuously evolving, and consequently, numerous empirical studies have been conducted to address a wide range of security vulnerabilities. Nevertheless, a critical analysis of existing work is necessary before proposing new vulnerability testing techniques. Rahman and Izurieta conducted a systematic mapping study to document state-of-the-art empirical research on web application security vulnerability detection, with the objective of providing a roadmap for synthesizing the reported findings. In their study, the existing literature was systematically reviewed using a systematic mapping methodology [16]

Previously, security testing companies and hackers mostly used vulnerability testing tools, but as companies computerize their business applications, they are also being purchased by companies and used for regular vulnerability testing. This situation has equipped vulnerability scanning tools with compliance with laws and regulations, periodic scanning capabilities, and the ability to scan sensitive information related to networks and systems. This is also reflected in reporting capabilities. In recent years, articles have been published in the literature comparing the capabilities of vulnerability scanning tools. Since the compared products and capabilities are different, firstly, common properties were defined for vulnerability scanners, then Nessus, Qualys, and OpenVAS were compared according to these properties.

3. Vulnerability Scanners and Their Capabilities

A wide range of metrics must be taken into account when comparing and evaluating vulnerability scanning tools, as each tool exhibits distinct characteristics and strengths. In certain contexts, factors such as cost-effectiveness and ease of use are prioritized, whereas in others, the breadth of detected vulnerabilities and the ability to identify unique or previously undiscovered weaknesses are of greater significance. In this study, a comparative framework was established in the form of a table that summarizes the general capabilities of vulnerability scanners and application scanning tools.

3.1 General Vulnerability Scanner Capabilities

Vulnerability scanning tools have evolved through the efforts of both commercial vendors and open-source development communities. Although these tools share several common functionalities, there is currently no universally accepted standard, framework, or guideline that clearly delineates the scope and capabilities of security testing tools. This lack of standardization complicates direct comparison across different solutions. To address this challenge, the present study first defines the core characteristics expected of an effective vulnerability scanning tool and subsequently evaluates widely used scanners according to these defined criteria.

The scope of this study is primarily limited to vulnerability detection capabilities. Other aspects, including performance metrics, system requirements, and user interface usability, are intentionally excluded from the evaluation.

Many metrics must be considered when comparing and evaluating vulnerability scanning tools. Each tool has something that makes it unique. In some cases, while free and easy use is important, the number of vulnerabilities found and the uniqueness of these vulnerabilities also come to the fore. In this study we created a table to compare general vulnerability and web application scanner capabilities. These capabilities take part in Table 1.

Vulnerability scanning tools have been developed over time by commercial companies and open-source developer communities. Despite certain similar features, there is no standard, framework, or guideline defining the boundaries of security testing tools. This makes comparing these tools difficult. This study first identifies the characteristics expected from a good vulnerability scanning tool and then compares commonly used vulnerability scanning tools based on these characteristics.

This study focused primarily on vulnerability detection capabilities. Performance requirements, system requirements, and user interface friendliness were not considered in this study.

The comparison features and statuses of Nessus, Qualys, and OpenVAS are included in Table 1.

Table 1. Comparison of Nessus, Qualys, and OpenVAS vulnerability scanning tools [7]

Features	**NESSUS**	**QUALYS**	**OpenVAS**
CVE's and Plugins Support	All NVD Database CVE's (307K)	More than 300K vulnerabilities	More than 256K CVEs
Scanning Methods	Network and agent-based scanning	Network-based scanning	Network and agent-based scanning
Report Techniques and Formats	HTML, CSV, PDF, XML	HTML, PDF, CSV, XML, MHT	HTML, CSV, PDF
User Interfaces	Graphic interfaces and command prompt	Web-based interfaces	Graphic interfaces and command prompt
Customizable Reports	Detailed Reports can be managed for usage.	Support customizable reports and manageable for usage.	Detailed reports, but customization is limited
Credentials	Supports credentials from local files, credential managers, Active Directory, and LDAP.	Supports credentials from local files, credential managers, Active Directory, and LDAP.	Only supports credentials from local files.
Vulnerability Scanning Supports	Cloud, IoT, ICS, Mobile, IoT, General, Web Applications	Cloud, IoT, ICS, Mobile, IoT, General, Web Applications	Authentication test, high and low-level internet, and industrial protocol test
Vulnerability Exploitation Info	Yes	Yes	No
Vulnerability Scoring Approach	CVSS, EPSS, VRP	Special	Special
Vulnerability Screening with compliance, regulatory and standard requirements	Provides compliance policy templates (CIS Benchmarks, DISA STIGs, PCI DSS, HIPAA, ISO 27001, GDPR, etc.) and audit information for operating system databases, network devices	Host-based report, vulnerability-based report, executive report, PCI DSS, NIST 800-53, HIPAA, ISO 27001, CIS benchmarks compliance	Provides executive report, technical report, but not comprehensive compliance reports.

Vulnerabilities affecting information systems—such as those cataloged under Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), and VulDB—are maintained in varying formats by different organizations. Among these repositories, the CVE catalog managed by MITRE is the most widely adopted. Vulnerability scanning tools typically report identified weaknesses using CVE reference identifiers. Each CVE entry includes structured information such as the CVE identifier, affected product and version, vulnerability type, references, descriptive details, and associated organizational metadata.

CVE records are maintained within the National Vulnerability Database (NVD), which additionally incorporates the Common Vulnerability Scoring System (CVSS) to quantify vulnerability severity. Vulnerability scanning tools commonly assign severity scores based on CVSS versions 2.0, 3.x, and, more recently, 4.0.

In terms of vulnerability coverage, Nessus and Qualys are capable of assessing approximately 300,000 known vulnerabilities, whereas OpenVAS supports testing for approximately 256,000 CVEs. Compared to the other tools, OpenVAS places greater emphasis on open-source software and network-level vulnerabilities. All three tools are also capable of identifying system configuration weaknesses; however, while Nessus explicitly presents vendor- and product-level vulnerability metadata within its interface, such information is not consistently exposed in OpenVAS and Qualys.

Regarding reporting capabilities, Nessus and Qualys support a wide range of output formats, including Hypertext Markup Language (HTML), Comma-Separated Values (CSV), Portable Document Format (PDF), and Extensible Markup Language (XML). In contrast, OpenVAS offers a more limited set of reporting options in terms of both format diversity and report content. Nessus further distinguishes itself by providing reports at multiple levels of granularity, such as asset-based, operating system–based, technical, detailed, and administrator-oriented reports. Additionally, both Nessus and Qualys offer extensive compliance and regulatory reporting aligned with standards such as CIS Benchmarks, DISA STIGs, PCI DSS, HIPAA, ISO/IEC 27001, and the General Data Protection

Regulation (GDPR), whereas OpenVAS provides minimal support for compliance reporting.

Vulnerability scanning can be conducted either through network-based techniques or via agents installed directly on target systems. Accordingly, the agent-based and network-based scanning capabilities of each tool were evaluated. Qualys enables scanning through a vendor-provided virtual appliance, while Nessus and OpenVAS are installed and executed as native applications on host systems. All three tools support deployment across commonly used operating systems, including Windows, Linux, and macOS.

User interaction with vulnerability scanners is typically facilitated through command-line interfaces, web-based dashboards, or graphical user interfaces. Each of the evaluated tools supports both command-line and web-based management. Nessus and OpenVAS are deployed as web applications hosted on the operating system, whereas Qualys operates through a cloud-managed interface accessed via a virtual machine.

Credentialed scanning represents a critical capability in vulnerability assessment, as it allows scanners to authenticate using predefined or default credentials and service information, such as usernames and passwords, Simple Network Management Protocol (SNMP), Secure Shell (SSH), and Active Directory credentials. Nessus, Qualys, and OpenVAS all support credentialed scanning and can perform more comprehensive assessments when authentication data is available. By default, Qualys and OpenVAS can attempt predefined credentials during scans, whereas Nessus requires credential information to be explicitly provided by the user.

While Nessus and Qualys generate reports with extensive depth, categorization, and visual representation, OpenVAS produces comparatively fewer report types with limited visualization capabilities.

A key distinguishing feature of modern vulnerability scanning tools is their ability to assess diverse IT environments through modular plugin architectures. Nessus and Qualys support scanning across a broad spectrum of systems, including traditional IT infrastructures, web applications, mobile platforms, cloud environments, Internet of Things (IoT) devices, malware, and ransomware. They also provide compliance assessments for cloud environments, Payment Card Industry Data Security Standard (PCI DSS), and organizational policy enforcement. In contrast, OpenVAS is limited to scanning computer systems and web applications and does not offer compliance assessment capabilities.

Finally, each tool categorizes vulnerabilities using distinct severity classification schemes. Nessus classifies vulnerabilities as Critical, High, Medium, Low, or Informational; Qualys assigns severity levels ranging from 1 to 5; and OpenVAS categorizes findings as High, Medium, Low, or Log/Registration. To enable objective comparison across tools, all identified vulnerabilities in this study were normalized according to their corresponding CVSS scores.

Some vulnerabilities have been grouped under the heading "Multiple Vulnerabilities." This is a common occurrence in vulnerability scanning tools. A scan was performed after installing the web plugin in Nessus. It was observed that Nessus could perform detailed checks once the necessary plugins for the target system were installed. The tools were not evaluated based on speed.

3.2 Web Application Vulnerability Scanning Capabilities

Organizations are increasingly transitioning their processes from manual to electronic environments, a shift enabled by software. This transformation necessitates the adoption of a secure Software Development Lifecycle (SDLC) methodology or framework. Web applications, a specific type of internetwork application, play a crucial role in digital transformation due to their accessibility from anywhere in the world and their ability to function without requiring a dedicated client application.

The main features of web applications differ significantly from those of desktop applications. Key features of web applications include heterogeneous technologies, a large user base, heterogeneous execution environments, concurrent transactions, state changes, operating system diversity, multi-tier architecture, and a fast maintenance rate[17] . While some of these features provide significant convenience, others pose potential threats to the sensitive data that is processed, stored, and transmitted.

Web applications are accessed through web browsers in an online environment. Unlike desktop applications, where files are stored locally, in web applications, files are stored on a web server. Web applications operate on a client-server model, which consists of two primary components: the client-side and the server-side [18]. The general architecture of a web application is shown in Figure 1.

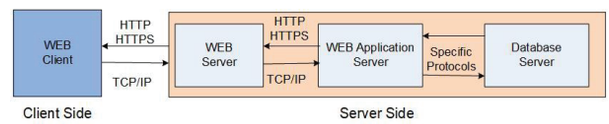

Figure 1. Web Architecture

Static pages and applications can be accessed via a web browser. While static web pages are read-only, web applications offer interactivity, enabling users to create, edit, manipulate, or delete data and content. If a web application server is used, this means that data is stored and processed. The data is also kept on the database server. Therefore, if the user will work on data, there will be a web application server and a database server connected to it.

In the context of web-based applications, user interaction with the program is inherently constrained. Failure to implement the requisite security protocols within the designated interface parameters for the user and throughout the development process can result in the emergence of critical vulnerabilities within the program. The following list contains the most commonly observed security vulnerabilities in web applications, along with explanations of these vulnerabilities. It is imperative that these vulnerabilities be checked during the design, coding, publication, and post-publication stages of web application development.

Known Vulnerabilities: This feature displays the quantity of vulnerabilities that the application is capable of identifying. This field indicates the number of vulnerabilities identified by the security scanning tool.

Authentication, Authorization, and Password Management: Authentication can be defined as the process of verifying the identity of a user. This is the initial step in any security process. The authentication of an individual's identity can be facilitated through the use of a password, a one-time password, or biometric data. The implementation of two or more of these factors in unison is referred to as multi-factor authentication. Authorization is defined as the process of granting a user or service access to a specific resource or function. It delineates the privileges (i.e., read, write, delete, modify, etc.) with which the user and service can access

resources. The password must be stored and processed in the system securely to verify the user's identity. It is evident that all vulnerability scanning tools endeavor to utilize the provided credentials; however, they are incapable of executing sophisticated attacks, such as brute force or dictionary attacks.

Input Validation: Input validation is defined as the process of verifying that input from a client or environment is in the expected format before its utilization. Failure to perform effective input validation can lead to vulnerabilities such as cross-site scripting (XSS), SQL injection, malicious file uploads, parser injections, Unicode attacks, and memory overflows. It has been demonstrated that both Nessus and Qualys possess the capability to execute input validation checks. However, it should be noted that OpenVAS is only capable of conducting a limited number of checks.

Session Management: Session management is defined as the process of tracking all actions performed by a user in a web application from the moment they log in through a session ID. The session ID is employed for the purposes of user authentication and the execution of authorization verifications. It is imperative that the session be securely established, managed, and terminated. Inadequate session security can result in a number of potential issues, including session hijacking, session fixation, and man-in-the-middle (MitM) attacks. Each security vulnerability scanning tool is designed to identify session-related vulnerabilities by verifying the integrity of Transport Layer Security (TLS) connections and the parameters of secure communication algorithms.

Secure Communication: In the event that sensitive data is transmitted via a web application, it is imperative that all transmitted data be encrypted. Following the identification of the service on the target system, critical parameters are examined. These parameters include encryption, hashing, protocol version, and secure access. The purpose of this examination is to ensure that access to the service is secure. All vulnerability scanning tools perform secure communication checks.

Data Protection: Web applications are responsible for the storage of a considerable amount of sensitive data, with a particular emphasis on identity verification information. This data must be safeguarded following its designated privacy level. OpenVAS and Qualys have been implemented to regulate access to sensitive data. It should be noted that Nessus does not currently possess the capacity to test stored data.

Business Logic: Business logic vulnerabilities are flaws in the design and implementation of an application that allow an attacker to elicit unintended behavior. This enables attackers to manipulate legitimate functionality to achieve a malicious goal. Each scanner does not include test steps for business logic errors.

Error Handling: Handling error messages improperly can lead to security issues in web applications. The most common problem is displaying detailed internal error messages, such as stack traces, database dumps, and error codes, to the user (i.e., hacker). These messages reveal application details that should never be disclosed and can lead to hacking. OpenVAS does not control error messages. Nessus evaluates the content of versions, updates, and default messages but does not check general error messages. Similarly, Qualys evaluates the content of versions, updates, and default messages and displays error messages in reports, but does not check the security of general error message content.

Compliance Standard: Organizations must comply with various laws and regulations to ensure information security. A vulnerability scanning tool must be able to test for compliance with certain standards and regulations. Qualys and Nessus perform compliance checks for standards such as the OWASP Top 10 and the GDPR. However, Nessus does not perform OpenVAS compliance checks.

Security Misconfiguration: A vulnerability scanning tool should detect and report misconfigurations, such as default passwords and unencrypted communication. All vulnerability scanning tools perform configuration checks.

File Upload: Depending on the application's requirements, the user may be asked to upload files. In such cases, the file type and size must be verified. The vulnerability scanning tool must be able to do so. Each vulnerability scanning tool checks for uncontrolled file vulnerabilities.

Web Server Vulnerabilities: Web application security refers to the security of applications that run on a web server. Vulnerabilities in the web server can affect the application. Therefore, a vulnerability scanning tool must be able to detect vulnerabilities in the web server. All vulnerability scanning tools check for web server vulnerabilities.

The checks performed by each tool for web application security requirements are provided in Table 2.

"Limited" means that the vulnerability scanner supports this feature, but only the basic features. For example, the security scanner tries the default username and password, but does not support brute force or dictionary attacks.

Qualys and OPENVAS appear to have better web security scanning capabilities than Nessus.

Table 2. Analysis of Nessus, Qualys, and OpenVAS Web Application vulnerability scanning capability

	NESSUS	**QUALYS**	**OpenVAS**
# Vulnerability	9190 Vulnerability		
Authentication and Authorization	Limited	Limited	Limited
Input Validation	Yes	Yes	Limited
Session Management	No	Yes	Yes
Security Communication	No	Yes	Yes
Data Protection	No	Yes	Yes
Business Logic	No|	No	No
Error Handling	No	Limited	Limited
Compliance Standard	No	Yes	Yes
Security Misconfiguration	Yes	Yes	Yes
File Upload	No		
Web Server Vulnerabilities	Yes (1967)	Yes	Yes

4. Performance Evaluations of Vulnerability Scanners

Vulnerability scanning tests were carried out on virtual machines deployed in VMware Workstation Pro 17, which was installed on a Windows 10 Professional host system. The corresponding test architecture is illustrated in Figure 2.

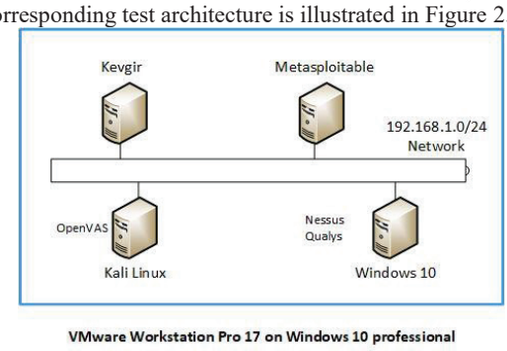

Figure 2. Test Architecture

Each vulnerability scanning tool was configured to maximize detection performance using the same system information. The most recent versions of the web applications were installed on the vulnerable operating system to ensure that the scan results reflected current conditions. In addition, all available scanning modules (e.g., web applications, databases, virtualization, and services) were enabled to utilize the full capabilities of each tool. During the experiments, Nessus version 10.11.0, the cloud-based Qualys VMDR Scanner version 12.17.41-1, and OpenVAS version 23.32.0 were utilized with their most up-to-date vulnerability signature databases.

4.1. System Vulnerability Scanning

Nessus, Qualys, and OpenVAS classify vulnerabilities using different severity schemes. Nessus assigns findings to Critical, High, Medium, Low, and Informational categories, whereas Qualys uses a 1–5 scale, where 1 represents low-severity issues and 5 indicates the most severe vulnerabilities. OpenVAS categorizes vulnerabilities as High, Medium, Low, and Log.

To enable meaningful comparison across tools, all detected vulnerabilities were normalized into Critical, High, Medium, Low, and Informational categories based on CVSS v3.0 scores.

In the Nessus configuration, potentially disruptive tests that could disable the operating system were excluded. Additionally, each scan was conducted only after updating the vulnerability scanners with the most recent plugin sets.

OpenVAS findings are reported together with a detection quality rating. Accordingly, only vulnerabilities with a detection quality score of 70% or higher were included in the table.

To maximize coverage, all plugins were enabled in each vulnerability scanning tool prior to the scans.

Figure 3 presents the vulnerabilities identified on the Metasploitable machine using all three scanners. The results indicate that Nessus detected the highest number of vulnerabilities.

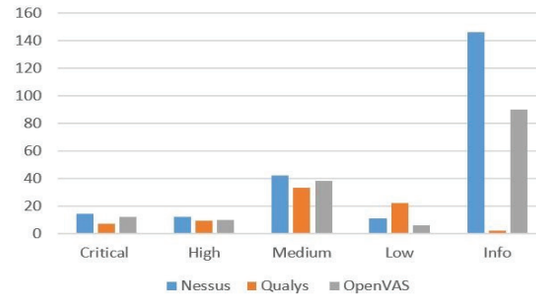

Figure 3. Metasploitable Vulnerability Scan Results [7]

In addition to verified findings, Qualys also reports potential vulnerabilities. In this scan, Qualys identified 11 Critical, 40 High, 110 Medium, and 31 Low potential vulnerabilities on the Metasploitable system. These potential findings require expert validation; if confirmed, they should be incorporated into the final vulnerability list.

Figure 4 illustrates the vulnerabilities detected on the Kevgir operating system using all three vulnerability scanning tools. The results show that Nessus identified the highest number of vulnerabilities.

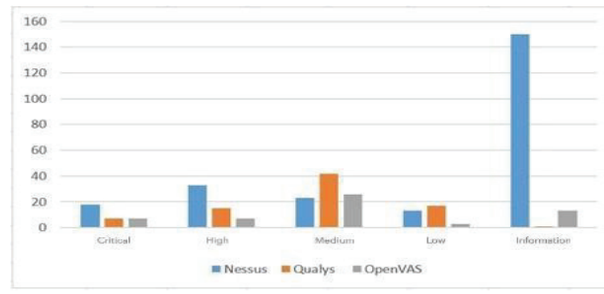

Figure 4. Kevgir Vulnerability Scan Results

In addition to the vulnerabilities that Qualys has identified as verified, it also identifies potential vulnerabilities. In this context, it has identified 11 Critical, 50 High, 64 Medium, and 6 Low potential vulnerabilities on Kevgir. These should be verified after expert control, and necessary actions should be taken. While Nessus and Qualys were able to find current vulnerabilities, OpenVAS was able to detect more old vulnerabilities. This is thought to be because the vulnerability database is updated later. After all, it is open source.

4.2. Web Application Vulnerability Scanning

During the web application security scan, each security scanner was configured to detect the maximum number of vulnerabilities. In this context, the Nessus and Qualys web application security testing features were activated. It was observed that activating the plugins related to the counter system in Nessus resulted in the detection of more configuration and update vulnerabilities.

The vulnerabilities detected when Metasploit computers containing vulnerabilities were scanned with the same security testing tools are shown in Figure 5.

	Critical	High	Medium	Low
Nessus	1	5	26	5
Qualys	0	5	19	13
OpenVAS	0	6	28	4

Figure 5. Metasploitable Web Application Vulnerability Scan Results

Although the number of vulnerabilities found on the Metasploit machine was relatively similar, Nessus was observed to detect more critical vulnerabilities. Unlike Qualys, it provides a list of potential vulnerabilities in addition to the vulnerabilities it has identified. For the Metasploit machine, it identified 4 critical, 21 high, 44 medium, and 8 low vulnerabilities.

The vulnerabilities detected when the Kevgir computer, which contained vulnerabilities, was scanned with vulnerability scanning tools are shown in Figure 6.

	Critical	High	Medium	Low
Nessus	10	13	32	6
Qualys	2	12	29	12
OpenVAS	0	10	22	1

Figure 6. Kevgir Web Application Vulnerability Scan Results

According to these results, Nessus was found to have more web application vulnerabilities than the others in the Kevgir scan. Qualys followed, while OpenVAS detected fewer vulnerabilities than the other two. The lack of a module specifically targeting web applications in OpenVAS is believed to be a contributing factor. Unlike other scanners, Qualys provides a list of potential vulnerabilities in addition to the vulnerabilities it has identified. For the Metasploit machine, it identified 53 critical, 53 high and 1 medium potential vulnerabilities.

If the potential vulnerabilities presented by Qualys are to be further evaluated, false positive vulnerabilities should be ignored. In OpenVAS, under the "Quality of Detection" heading for vulnerabilities, a rating between 1 and 100 indicates how accurate the detection is.

5. Conclusion and Future Works

The use of electronic services (e.g., e-commerce, e-banking, e-government, and e-health) has become increasingly widespread due to advantages such as global accessibility, low cost, reduced labor requirements, and ease of use. Web applications, play a crucial role in digital transformation due to their accessibility from anywhere in the world and their ability to function without requiring a dedicated client application. However, despite the operational benefits of delivering services through computer-based systems, this shift also introduces significant security risks. Therefore, it is critical for organizations to identify these risks, implement appropriate countermeasures, and reduce and manage them to acceptable levels.

Vulnerability scanning tools are widely employed to detect and monitor security risks in the software and hardware components of information systems. Although such tools were previously used primarily by penetration testing firms and malicious actors, they are now commonly adopted by organizations to continuously monitor vulnerabilities and implement necessary security controls.

Web applications have become increasingly prevalent; therefore, in addition to being a feature of general-purpose vulnerability scanning tools, there are also tools specifically designed to scan vulnerabilities in web applications. In this study, the web application scanning capabilities of general-purpose vulnerability scanning tools were evaluated.

In this study, the key features required for an effective vulnerability scanning tool and Web application scanning tool were identified through an analysis of the relevant literature and widely used vulnerability scanners.

The vulnerability scanning tool identified features are as follows:

- Assessing vulnerabilities according to internationally recognized criteria
- Providing sufficient and actionable information regarding the identified vulnerabilities
- Detecting active vulnerabilities
- Offering an effective reporting infrastructure tailored to different user roles and output formats
- Ensuring compliance with multiple standards, regulations, and cloud governance requirements
- Performing scans without harming the target systems
- Supporting credential-based scanning by allowing the input of valid system credentials
- Enabling access through multiple interfaces, including web-based dashboards, graphical user interfaces, and command-line tools

The web application vulnerability scanning tool identified features are as follows:

- Testing authentication and authorization mechanism
- Checking input validations
- Session management control
- Testing security communication
- Checking data protection mechanism
- Testing error handling mechanism
- Checking suitability with compliance and standard
- Detecting security misconfiguration
- Checking uploaded file type and size
- Scanning web server vulnerabilities

Based on these criteria, the Nessus, Qualys, and OpenVAS vulnerability scanning tools were employed to assess the Metasploitable and Kevgir operating systems, as well as a web application. Among the evaluated tools, Nessus identified the highest number of vulnerabilities. Qualys ranked second, followed by OpenVAS. Unlike the other scanners, Qualys provides a list of potential vulnerabilities in addition to the confirmed findings it detects. In contrast, OpenVAS was observed to identify fewer up-to-date vulnerabilities.

In terms of reporting capabilities, OpenVAS offered fewer reporting options compared to the other tools. Nessus and Qualys were more effective in generating reports aligned with various standards and regulatory requirements.

Web application security scanning results indicated that Qualys supported a wider range of control features than the other tools. However, Nessus identified the highest number of web vulnerabilities. When the findings reported by Qualys under the potential vulnerabilities category were included, Qualys detected a vulnerability count comparable to Nessus. In contrast, OpenVAS identified fewer up-to-date vulnerabilities than the other scanners.

This study included operating system-, host-, and web application-based scans. Future research may extend the evaluation to additional infrastructures, including cloud environments, mobile devices, embedded systems, and industrial control systems such as SCADA.

6. References

[1] S. Pandey and A. Chaudhary, “Vulnerability scanning,” 2023. doi: 10.36227/techrxiv.20317194.v1.

[2] R. Kushe, "Comparative study of vulnerability scanning tools: Nessus vs Retina," *INTERNATIONAL SCIENTIFIC JOURNAL "SECURITY & FUTURE,"* vol. 1, no. 2, pp. 69–71, 2017.

[3] T. Singh and A. Kumar, "Analyzing security and privacy issues for multi-cloud service providers using Nessus," in *2023 5th International Conference on Electrical, Computer and Communication Technologies, ICECCT 2023*, Inc., doi: 10.1109/ICECCT56650.2023.10179727.

[4] K. Boonchuay, W. Siripaktanakon, O. Sangpetch, and A. Sangpetch, "Software vulnerability assessment: Vendor, Scanner, and User Analysis," in *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, IEEE Computer Society, 2022, pp. 214–221. doi: 10.1109/CloudCom55334.2022.00038.

[5] I. Chalvatzis, D. A. Karras, and R. C. Papademetriou, "Evaluation of security vulnerability scanners for small and medium enterprises business networks resilience towards risk assessment," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019. doi: 10.1109/ICAICA.2019.8873438.

[6] M. A. Muin, Kapti, and T. Yusnanto, "Campus website security vulnerability analysis using Nessus," *International Journal of Computer and Information System (IJCIS) Peer Reviewed-International Journal*, vol. 03, no. 2, pp. 79–82, 2020, doi: 10.29040/ijcis.v3i2.72.

[7] C. Ozturk and M. Kara, "5th International Eurasian Conference on Science, Engineering and Technology (EurasianSciEnTech 2024)," in *Practical Comparison of Nessus, Qualys and OpenVAS Vulnerability Scanner Tools*, Ankara:, Jun. 2024, pp. 710–716.

[8] M. Fu, C. K. Tantithamthavorn, V. Nguyen, and T. Le, "ChatGPT for Vulnerability Detection, Classification, and Repair: How Far Are We?," in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, IEEE Computer Society, 2023, pp. 632–636. doi: 10.1109/APSEC60848.2023.00085.

[9] C. Zhang, H. Liu, J. Zeng, K. Yang, Y. Li, and H. Li, "Prompt-Enhanced Software Vulnerability Detection Using ChatGPT," in *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*, ACM, Jan. 2024, pp. 276–277. doi: 10.1145/3639478.3643065.

[10] B. Zukran and M. M. Siraj, "Performance Comparison on SQL Injection and XSS Detection using Open Source Vulnerability Scanners," in *2021 International Conference on Data Science and Its Applications, ICoDSA 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 61–65. doi: 10.1109/ICoDSA53588.2021.9617484.

[11] D. Moreira, J. P. Seara, J. P. Pavia, and C. Serrão, "Intelligent Platform for Automating Vulnerability Detection in Web Applications," *Electronics (Basel).*, vol. 14, no. 1, p. 79, Dec. 2024, doi: 10.3390/electronics14010079.

[12] B. J. Santoso, R. M. Ijtihadie, and G. N. S. Aryawan, "Vulnerability Data Assessment and Management Based on Passive Scanning Method and CVSS," in *2023 14th International Conference on Information and Communication Technology and System, ICTS 2023*, pp. 325–330. doi: 10.1109/ICTS58770.2023.10330884.

[13] A. Balsam, M. Nowak, M. Walkowski, J. Oko, and S. Sujecki, "Analysis of CVSS Vulnerability Base Scores in the Context of Exploits' Availability," in *2023 23rd International Conference on Transparent Optical Networks (ICTON)*, IEEE, Jul. 2023, pp. 1–4. doi: 10.1109/ICTON59386.2023.10207394.

[14] R. Bhardwaj and D. Saraswat, "A Comparative Review of Vulnerability Assessment Tools: Insights from Qualys and Nessus," *International Journal of Engineering and Techniques*, vol. 11, no. 5, pp. 135–140, Sep. 2025.

[15] A. Kejiou and G. Bekaroo, "A Review and comparative analysis of vulnerability scanning tools for wireless LANs," in *Proceedings - 3rd International Conference on Next Generation Computing Applications, NextComp 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/NextComp55567.2022.9932245.

[16] K. Rahman and C. Izurieta, "A Mapping Study of Security Vulnerability Detection Approaches for Web Applications," in *Proceedings - 48th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 491–494. doi: 10.1109/SEAA56994.2022.00081.

[17] D. R. Lakshmi and S. S. Mallika, "A Review on Web Application Testing and its Current Research Directions," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 4, p. 2132, Aug. 2017, doi: 10.11591/ijece.v7i4.pp2132-2141.

[18] M. Aydos, Ç. Aldan, E. Coşkun, and A. Soydan, "Security testing of web applications: A systematic mapping of the literature," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6775–6792, Oct. 2022, doi: 10.1016/j.jksuci.2021.09.018.

Özgeçmişler

Mehmet Kara received his B.Sc. degree in electronics and communication engineering from Yildiz Technical University, Türkiye, and his M.Sc. and Ph.D. degrees in electronics and communication engineering from Kocaeli University, Türkiye. He has held technical and research positions at Kocaeli University, Armada Bilgisayar, TÜBİTAK BİLGEM, Sistem Bilgisayar, and UN Women. He is currently an assistant professor in the Department of Computer Engineering at Kocaeli Health and Technology University, Türkiye.

His research interests include cybersecurity, secure network design and deployment, network security assessment and testing, cybersecurity education and training, perimeter security technologies (including intrusion detection and prevention systems, firewalls, VPNs, routers, and switches), cloud security, IoT security, and secure software development.