5-2023

# A Survey and Comparative Study on Vulnerability Scanning Tools

Cassidy Khounborine
*University of Arkansas, Fayetteville*

# A Survey and Comparative Study on Vulnerability Scanning Tools

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering

College of Engineering

University of Arkansas

Fayetteville, AR

April, 2023

by

Cassidy Khounborine

## Abstract

Vulnerability scanners are a tool used by many organizations and developers as part of their vulnerability management. These scanners aid in the security of applications, databases, networks, etc. There are many different options available for vulnerability scanners that vary in the analysis method they encompass or target for which they scan, among many other features. This thesis explores the different types of scanners available and aims to ease the burden of selecting the ideal vulnerability scanner for one’s needs by conducting a survey and comparative analysis of vulnerability scanners. Before diving into the vulnerability scanners available, background information is provided regarding the types of testing a vulnerability scanner may use as well as the types of vulnerability scanners out there. This thesis highlights application scanners, database scanners, and network-based scanners as those were the types of vulnerability scanners primarily found in the survey. This thesis also compares the accuracy of two network scanners—OpenVAS and Nessus—when scanning the same target and discusses the results and their implications.

# Acknowledgment

First and foremost, I would like to give thanks to my honors thesis advisor, Dr. Qinghua Li, in the Department of Computer Science and Computer Engineering. Without his assistance and involvement throughout this process, this thesis could not have been accomplished. Thank you for all your support and guidance through this past year.

I wish to also express my gratitude to my committee members, Dr. Matthew Patitz and Dr. Kevin Jin in the Department of Computer Science and Computer Engineering, for agreeing to be on my committee despite their busy schedules. I have not had the opportunity to work with Dr. Kevin Jin, but I have previously worked with Dr. Matthew Patitz in his research lab, an experience which I am in most grateful for as it was my first glimpse into the world of academic research. Additionally, this work is supported in part by the NSF under award number 1751255.

# Contents

# 1 Introduction

Vulnerability scanners are a testing tool utilized by many organizations and developers that identify coding flaws and misconfigurations in the codebases. These scanners usually rely on a database that consists of known vulnerabilities such as the National Vulnerability Database (NVD) by the National Institute of Standards and Technology (NIST) [26]. The NVD is comprised of several hundreds of thousands of common vulnerabilities and exposures (CVE), each of which is an entry that is assigned an identification number—usually related to the year it was found—a description of the vulnerability, and known software affected by it. Each CVE is also assigned a common vulnerability scoring system (CVSS) score which details how severe the vulnerability is. Most vulnerability scanners cross reference their findings against the NVD and include CVEs and CVSS scores, if possible, when generating the vulnerability report. However, vulnerability scanners do take it one step further than just running against a vulnerability database by also pulling in information from various sources such as from the documentation sites of different software.

Vulnerability scanners are an important tool to aid in preventative measures from cybersecurity threats and attacks. These scanners are particularly useful when integrating with third-party providers or open-source code. Focusing in on open-source code, due to the nature of open-source projects, there is no guarantee that the best security practices were implemented by every developer contributing to the project. As such, it can be difficult to find vulnerabilities in the code or to verify that there are no vulnerabilities without the use of a vulnerability scanner.

Vulnerability scanners are not solely limited to just scanning code. There are a wide range of vulnerability scanners out there, created for different scanning targets with different features. It can be challenging to determine which scanner is the best for one's needs. To aid in this endeavor, this thesis explores 35 different vulnerability scanners available and their features. Before discussing these scanners and their features, background information is provided in relation to the various types of scanners and the types of testing methods these scanners implement for better understanding of the survey portion of this thesis.

In addition to the survey, this thesis conducts a comparative study between two scanners to further assist with selecting the best vulnerability scanner for one’s needs. As there are many scanners available with the same scanning target and some of the same features one may be looking for, it is important to factor in other aspects to make a more well-rounded decision such as the accuracy of these scanners. To test the accuracy of these scanners, two scanners were selected to scan the same target. From this, 100 vulnerabilities were selected from the combined pool of vulnerabilities found by these scanners to be analyzed for accuracy. As seen later in this thesis, some vulnerabilities were found to be a false positive, in which case this thesis then discusses the implications of this result.

This thesis is organized as follows. Chapter 2 introduces background information. Chapter 3 presents the survey of 35 tools. Chapter 4 describes the comparative study over two scanning tools. Chapter 5 concludes this thesis and discusses future work.

# 2 Background

The survey focuses on different features associated with the vulnerability scanners such as the type of scanner, scanning targets, and different vulnerabilities tested. In order to understand the different vulnerability scanners available, it’s important to provide information on some of the features associated with the scanners.

## 2.1 Types of Testing

Vulnerability scanners incorporate or extend upon different forms of penetration testing, namely white box testing and black box testing. Penetration testing is a simulated cyberattack against the computer system that checks for exploitable vulnerabilities. White box testing can be defined as testing from the view of a developer, where the tester has full access and complete knowledge regarding the target being tested as well as features the target has. This form of testing results in more vulnerabilities being found since the tester has complete access to the codebase. However, it can be more time consuming since there is a lot of information at hand. In contrast, black box testing is a form of external testing from the view of an attacker, where the tester has no knowledge of the technologies or frameworks the target was built on, only provided with the target URL.

Along with penetration testing, there is also authenticated and unauthenticated testing. Authenticated testing occurs when the tester has access to a user account and is able to log in as the user and view vulnerabilities from a trusted user’s perspective. Unauthenticated occurs when the tester does not have access to a trusted user’s account and views vulnerabilities from an attacker’s perspective.

## 2.2 Types of Scanners

### 2.2.1 Application Scanners

As the name suggests, application scanners detect software vulnerabilities and/or misconfigurations in an application. Application scanners utilize different analysis methods such as software composition analysis (SCA). Scanners for web applications in particular use static analysis, often referred to as static application security testing (SAST), and dynamic code analysis, otherwise known as dynamic application security testing (DAST). Throughout the thesis, each name for these methods will be used interchangeably. All of these methods have their own merits and pitfalls, but it is ideal to use all when possible to ensure the security of an application. The following section provides more in-depth information regarding each analysis method as well as some advantages and disadvantages.

## SOFTWARE COMPOSITION ANALYSIS

Software composition analysis analyzes third-party open source code for vulnerabilities. It's important to note that SCA tools do not perform static or dynamic analysis of the code within the third-party components themselves. They analyze the list of dependencies, cross reference them against their list of known vulnerable dependencies and report the matching dependencies. In comparison to SAST or DAST tools, SCA tools achieve fairly accurate results. Software composition analysis is often partially covered as part of SAST tools, but they are not necessarily complete or exhaustive when compared to the standalone SCA tool.

## WEB APPLICATION SCANNERS

Under the umbrella of application scanners, there are web application scanners. These scanners detect vulnerabilities in web applications and often look for and find the most common security vulnerabilities found in these applications. Some of these common vulnerabilities as stated by OWASP [2] include:

- *Cross-site scripting (XSS):* a security vulnerability where malicious scripts are injected into trusted sites. This usually occurs when an attacker uses a web application to send a browser side script that contains malicious code to an unsuspecting user [14].
- *SQL injections:* a security vulnerability that allows an attacker to insert an SQL query via input data from the client to an application [15].
- *File inclusion:* a vulnerability where an attacker is able to access, view, and include files located in the server's file system [16].
- *Carriage return line feed (CRLF) injections*: a vulnerability with the carriage return and linefeed characters where they are injected via input data to alter the way an application works or to confuse an administrator [17].
- *Lightweight Directory Access Protocol (LDAP) injection:* a vulnerability that occurs when the user input is not properly sanitized, allowing an attacker the possibility to modify LDAP statements via local proxy [1].
- *Server-side request forgery (SSRF):* a vulnerability that allows an attacker to "send a request from the back end of the software to another server or local service" [18].
- *XML External Entity (XXE):* a vulnerability that occurs when a "web application or API accepts unsanitized XML data and its back end XML parser is configured to allow external XML entity parsing" [19].

The way in which these scanners analyze a web application can be divided into two different methods: static analysis and dynamic code analysis.

SAST is a form of white box testing that helps identify security issues in the source code. The scanning target for these scanners can vary. They may scan the source code or the binary code. However, more often than not, the source code is the primary scanning target. Scanners utilizing this method can be used early in the software development cycle as they do not require a working application to run. When utilized in the initial stages of development, it quickly identifies security vulnerabilities and allows developers to resolve these issues without breaking builds and before passing on these vulnerabilities to the final release of an application. Some SAST tools also provide real-time feedback to developers as they code allowing them to fix any security issues before moving on to the next stage of development. It also ensures that the security of an application is not an afterthought. In comparison to code reviews performed by humans, SAST tools are able to quickly scan millions of lines of code in a matter of a few minutes. They also provide complete code coverage. To get the most out of SAST scanners, it’s important to frequently scan an application whether that’s daily, weekly, monthly, or before a new release.

In contrast, DAST is a form of black box testing that allows developers to scan a running application and identify vulnerabilities. Dynamic code analysis does not require access to the source code to run, only a running application. DAST tools work by bombarding a running application with potentially malicious inputs to common vulnerabilities and analyzes the application’s response to them. Examples of these inputs and the associated vulnerabilities are shown in Table 1.

| **Inputs** | **Vulnerability** |
|---|---|
| **SQL queries** | SQL injections |
| **Long input strings** | Buffer overflow vulnerabilities |

| | |
|---|---|
| **Negative and large positive numbers** | Integer underflow and overflow vulnerabilities |
| **Unexpected input data** | No set particular vulnerability, but aims to exploit assumptions made by developers |

*Table 1: Potentially malicious inputs and their associated vulnerability often used in dynamic code analysis to test an application.*

If the application has a negative response to these inputs, such as crashing, the DAST tools record the identified vulnerability. Since DAST tools scan while the application is running, they are able to identify runtime vulnerabilities that SAST tools are unable to such as configuration errors or memory allocation errors. Due to the nature of DAST tools, dynamic code analysis is usually used towards the end of the software development lifecycle during the testing phase. This may cause remediation for vulnerabilities found to be pushed into the next cycle.

### 2.2.2 Network-based Scanners

Network-based scanners identify and detect possible security attacks and vulnerable systems on networks. With these scanners, unknown or unauthorized devices and systems on a network can be identified. They can also aid in determining if there are unknown perimeter points on a network. Examples of this include unauthorized remote access servers or connections to insecure networks [7].

Network-based scanners have a variety of different scans that can be performed to test networks. Important vulnerability assessments to include are a brute force scan, credentialed scan, and exploit scan. A brute force scan uses a default, dictionary, or custom list created by system administrators that contain common, unsecure, and weak passwords—birthdays, “password”, etc.—to check for weak passwords.

### 2.2.3 Database Scanners

Database scanners are used to identify vulnerabilities in a database and to determine whether the information stored in the database is protected from attacks. These scanners look both internally and externally for possible security risks. Internally, they look for possible configurations that can be exploited. Externally, they perform different security testing techniques to ensure the security of a database. As one of the most common attacks against a database is an SQL injection, SQL injection tests can be performed to test the database’s security. An important attack to test for is password cracking. As the name suggests, password cracking is an attack where an intruder uses a password-cracking tool or tries to guess a username/password combination to gain access.

# 3 Survey of Scanning Tools

The survey of scanning tools was compiled by referencing sites [12, 28-32] that listed various vulnerability scanners and selecting 35 of them. The name of each vulnerability scanner contains a hyperlink to the website where it is available for download and installation and/or for more information about it. The “type” column refers to how the vulnerability scanner is available, either free and/or as an open source project or commercially. As the name suggests, the “scanning target” column contains information regarding the scanning target of each vulnerability scanner such as a web application or source code. From there, the columns refer to different common features of the vulnerability scanners found as well as common vulnerabilities found by scanners. The documentation provided for each scanner aided in determining whether a particular scanner contained a certain feature or tested for a certain vulnerability.

| Symbol | Meaning |
|---|---|
| **Y** | Yes, the vulnerability scanner has this feature or finds/tests for this vulnerability. |
| **N** | No, the vulnerability doesn't have this feature or finds/tests for this vulnerability. |
| **-** | This feature does not apply to this vulnerability scanner. |
| **?** | Inconclusive whether or not this vulnerability scanner has this feature or finds/tests for this vulnerability. |

*Table 2: Key explaining the different symbols used in the survey tab.*

| Name | Type | Scanning Target | SAST | SCA | DAST |
|---|---|---|---|---|---|
| `Trivy` | Open source | Application container | ? | ? | ? |
| `Clair` | Open source | Application container | Y | N | N |
| `Anchore` | Open source | Application container (Docker) | Y | N | N |
| `Sqlmap` | Open source | Database/web application | N | N | Y |
| `Wapiti` | Open source | Web application | N | N | Y |
| `VisualCodeGrepper` | Open source | Source code | Y | N | N |
| `Brakeman` | Open source | Ruby on Rails applications | Y | N | N |
| `Bandit` | Open source | Python source code | Y | N | N |
| `Secure Programming Lint (SPLINT)` | Open source | C source code | Y | N | N |

| | | | | | |
|---|---|---|---|---|---|
| **UNO** | Open source | C source code | Y | N | N |
| **Checkstyle** | Open source | Java source code | Y | N | N |
| **Nessus** | Commercial | Network | - | - | - |
| **OpenVAS** | Open source | Network | - | - | - |
| **OWASP Zed Attack Proxy (ZAP)** | Open source | Web application | N | N | Y |
| **Burp Suite** | Commercial | Web application | N | N | Y |
| **Vega** | Open source | Web application | ? | ? | ? |
| **Nikto2** | Open source | Web application | ? | ? | ? |
| **Nexpose** | Commercial | Network | - | - | - |
| **OpenSCAP** | Open source | Application containers, database, network infrastructure, hosts | ? | ? | ? |
| **Wireshark** | Open source | Network/packet | - | - | - |
| **Aircrack-Ng** | Open source | WiFi network | - | - | - |
| **VAF** | Open source | Web application | N | N | Y |
| **Nmap** | Open source | Network | - | - | - |
| **Detectify** | Commercial | Web application | Y | N | Y |
| **Metasploit** | Open source, commercial | Network | - | - | - |
| **ThreatMapper** | Open source | Hosts, application containers | N | N | Y |
| **Watchdog** | Open source | Network | N | N | Y |
| **Snyk** | Commercial | Source code, open source dependencies, container (images), infrastructure as code configurations | Y | Y | N |
| **Black Duck** | Commercial | Application, containers (third-party open source code) | N | Y | N |
| **Mend (formerly WhiteSource)** | Commercial | Source code/applications | Y | Y | N |
| **Arachni** | Open source | Web application | N | N | Y |
| **XssPy** | Open source | Web application | N | N | Y |
| **w3af** | Open source | Web application | N | N | Y |
| **Wfuzz** | Open source | Web application | N | N | Y |
| **Grabber** | Open source | Web application | N | N | Y |

*Table 3: Survey of vulnerability scanners with their respective type, scanning target, and tool classification (SAST, SCA, DAST).*

| Name | Authenticated Testing | Unauthenticated Testing | Supports many OS/multi-platform | Generate vulnerability reports |
|---|---|---|---|---|
| Trivy | ? | ? | Y | Y |
| Clair | - | - | Y | Y |
| Anchore | - | - | Y | Y |
| Sqlmap | ? | ? | Y | Y |
| Wapiti | ? | ? | Y | Y |
| VisualCodeGrepper | - | - | N | Y |
| Brakeman | - | - | Y | Y |
| Bandit | - | - | Y | Y |
| Secure Programming Lint (SPLINT) | - | - | Y | Y |
| UNO | - | - | Y | Y |
| Checkstyle | - | - | Y | Y |
| Nessus | Y | Y | Y | Y |
| OpenVAS | Y | Y | N | Y |
| OWASP Zed Attack Proxy (ZAP) | ? | ? | Y | Y |
| Burp Suite | ? | ? | ? | Y |
| Vega | ? | ? | Y | Y |
| Nikto2 | ? | ? | Y | Y |
| Nexpose | ? | ? | ? | Y |
| OpenSCAP | ? | ? | Y | Y |
| Wireshark | - | - | Y | Y |
| Aircrack-Ng | ? | ? | Y | Y |
| VAF | ? | ? | Y | Y |

| | | | | |
|---|---|---|---|---|
| Nmap | ? | ? | Y | Y |
| Detectify | Y | Y | ? | Y |
| Metasploit | ? | ? | Y | Y |
| ThreatMapper | ? | ? | Y | Y |
| Watchdog | ? | ? | N | Y |
| Snyk | - | - | Y | Y |
| Black Duck | ? | ? | ? | Y |
| Mend (formerly WhiteSource) | - | - | ? | Y |
| Arachni | ? | ? | Y | Y |
| XssPy | ? | ? | Y | Y |
| w3af | ? | ? | Y | Y |
| Wfuzz | ? | ? | Y | Y |
| Grabber | ? | ? | Y | Y |

*Table 4: Survey of vulnerability scanners with features related to testing, multi-platform, and vulnerability reports generated.*

| Name | XSS | SQL Injections | File Inclusion | CRLF | LDAP | SSRF | XXE |
|---|---|---|---|---|---|---|---|
| Trivy | Y | ? | ? | ? | ? | ? | ? |
| Clair | ? | ? | ? | ? | ? | ? | ? |
| Anchore | ? | ? | ? | ? | ? | ? | ? |
| Sqlmap | N | Y | N | N | N | N | N |
| Wapiti | Y | Y | Y | Y | ? | Y | Y |
| VisualCodeGrepper | Y | Y | Y | ? | ? | ? | ? |
| Brakeman | Y | Y | ? | ? | ? | ? | ? |
| Bandit | ? | Y | ? | ? | ? | ? | ? |
| Secure Programming Lint (SPLINT) | ? | ? | ? | ? | ? | ? | ? |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| UNO | ? | ? | ? | ? | ? | ? | ? |
| Checkstyle | ? | ? | ? | ? | ? | ? | ? |
| Nessus | ? | ? | ? | ? | ? | ? | ? |
| OpenVAS | Y | ? | ? | ? | ? | ? | ? |
| OWASP Zed Attack Proxy (ZAP) | Y | Y | Y | ? | ? | ? | ? |
| Burp Suite | Y | Y | ? | ? | ? | ? | ? |
| Vega | Y | Y | Y | ? | ? | ? | ? |
| Nikto2 | Y | Y | ? | ? | ? | ? | ? |
| Nexpose | ? | ? | ? | ? | ? | ? | ? |
| OpenSCAP | ? | ? | ? | ? | ? | ? | ? |
| Wireshark | ? | ? | ? | ? | ? | ? | ? |
| Aircrack-Ng | ? | ? | ? | ? | ? | ? | ? |
| VAF | ? | ? | ? | ? | ? | ? | ? |
| Nmap | ? | ? | ? | ? | ? | ? | ? |
| Detectify | Y | ? | ? | ? | ? | ? | ? |
| Metasploit | ? | ? | ? | ? | ? | ? | ? |
| ThreatMapper | ? | ? | ? | ? | ? | ? | ? |
| Watchdog | ? | ? | ? | ? | ? | ? | ? |
| Snyk | ? | ? | ? | ? | ? | ? | ? |
| Black Duck | ? | ? | ? | ? | ? | ? | ? |
| Mend (formerly WhiteSource) | ? | ? | ? | ? | ? | ? | ? |
| Arachni | Y | Y | Y | ? | ? | ? | ? |
| XssPy | Y | N | N | N | N | N | N |
| w3af | Y | Y | ? | ? | ? | ? | ? |
| Wfuzz | Y | Y | ? | ? | Y | ? | ? |
| Grabber | Y | Y | Y | ? | ? | ? | ? |

*Table 5: Survey of vulnerability scanners with different types of vulnerabilities they may detect or test for.*

As seen in *Table 3*, out of the 35 vulnerability scanners surveyed, 28 were open source and 8 were commercial with 1 scanner, Metasploit [55], being open source and available commercially with commercial support. There were 27 application scanners with scanning targets ranging from source code to application containers to web applications. Out of these application scanners, 10 used SAST as their analysis method, 11 used DAST, 1 scanner used both SAST and DAST (Detectify [54]), and 3 used SCA with 2 being used in conjunction with SAST. The 4 remaining application scanners were inconclusive in their analysis method. Aside from application scanners, there were 9 network and host scanners, 1 database scanner, and 1 scanner, OpenSCAP [49], being able to scan both networks and databases. Including OpenSCAP, 3 of these scanners—Sqlmap [36] and ThreatMapper [56]—overlapped with the application scanners. *Tables 6-8* were created to summarize these findings.

| **Scanner Availability** | |
|---|---|
| *Open source* | 28 |
| *Commercial* | 7 |
| *Open source and commercial* | 1 |
| **TOTAL** | **35** |

*Table 6: Number of scanners that are open source and available commercially.*

| Scanner Type | |
|---|---|
| *Application* | 24 |
| *Network, host* | 8 |
| *Application and database* | 1 |
| *Application, database, network, and host* | 1 |
| *Host and application* | 1 |
| **TOTAL** | **35** |

*Table 7: Number of scanners for each scanner type.*

| Application Analysis Method | |
|---|---|
| *SAST* | 8 |
| *SAST and SCA* | 2 |
| *DAST* | 11 |
| *SAST and DAST* | 1 |
| *SCA* | 1 |
| *Inconclusive* | 4 |
| **TOTAL** | **27** |

*Table 8: Number of analysis methods utilized by application scanners.*

As seen in *Table 4*, 3 scanners—Nessus [22], OpenVAS [21], and Detectify—supported authenticated and unauthenticated testing. Most of the scanners, 24 out of the 35, are multi-platform, supporting many different operating systems. For example, the OWASP ZAP [44] scanner is available for Windows, Linux, and macOS. They even have a cross platform package available as well. In the 11 remaining scanners, 3 scanners—VisualCodeGrepper [38], OpenVAS, and Watchdog [57]—are not multi-platform, and 8 scanners are inconclusive in their multi-platform capability. However, all 35 vulnerability scanners generate vulnerability reports.

As seen in *Table 5*, it is inconclusive for many of these scanners on whether they are able to find or test for these various vulnerabilities. However, out of the different vulnerabilities listed, XSS and SQL injections were the vulnerabilities most tested for by these scanners. For XSS, 15 scanners can detect this vulnerability, 1 scanner cannot, and it is inconclusive for the remaining 19 scanners. The one scanner that cannot detect XSS is Sqlmap, which was specifically designed to test for various SQL injection techniques. For SQL injections, 13 scanners can detect this vulnerability, 1 scanner cannot, and it is inconclusive for the remaining 21 scanners. The one scanner that cannot detect SQL injections is XssPy [62], which was design to only test for XSS vulnerabilities. For file inclusion, 6 scanners can detect this vulnerability, 2 scanners cannot (Sqlmap, XssPy), and the remaining 27 scanners are inconclusive for this vulnerability. There was only one scanner, Wapiti [41], that detected CRLF injections. For LDAP injections, there was also only one scanner that detected this vulnerability, Wfuzz [64]. For SSRF and XXE vulnerabilities, the only scanner that was able to identify these vulnerabilities was Wapiti.

# 4 Comparative Analysis

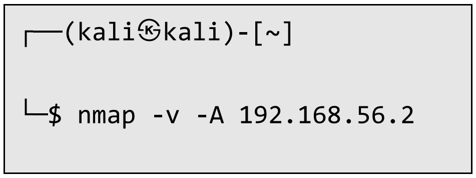
While the survey provides one with a better idea of which scanner suits their needs in terms of the scanning target and features it provides, one may still be stuck deciding between similar scanners. In which case, it can be useful to expand beyond these features and look at the accuracy of these scanners to determine which scanner is better. In general, having the accuracy of a vulnerability scanner is important to determine its usefulness and reliability.

After finding several different vulnerability scanners, two network scanners were selected to run tests with: OpenVAS and Nessus. OpenVAS was created by Greenbone Networks in 2006 [21] and was designed to work in a Linux environment. This scanner is a full-featured vulnerability scanner that allows testers to run both authenticated and unauthenticated testing. It checks visible ports and services it has access to for known exploits. Nessus was created by Tenable [22] and also allows for authenticated and unauthenticated testing among many other scans. Both these scanners support 212,000 CVEs [65-66]. However, OpenVAS does filter their results before reporting them back to the user. Issues found with the threat level “log”, “debug” or “false positive” are not reported to the user, and only results with a minimum quality of detection (QoD)—a percentage indicating the reliability of the vulnerability detected—of 70 are shown [27].

Before diving into the process of comparing the two network vulnerability scanners, it’s important to describe the software that was used to test and analyze these two scanners. During the experimental portion, VirtualBox and Metasploitable 2 were used. The VirtualBox software [25] was used to add a Kali Linux virtual machine in which OpenVAS could be installed. Nessus was also installed on this virtual machine. Metasploitable 2 is an Ubuntu Linux virtual machine that was intentionally designed to be vulnerable as a testing environment for penetration testers and security researchers [24]. During the analysis portion, the network scanner Nmap [23] and the NVD by NIST were used. Nmap assisted with discovering installed software on Metasploitable 2. The NVD assisted with discovering the common platform enumeration (CPE) names of installed software and as a source of information pertaining to each CVE that was discovered by the scanners. The NVD was also used as the ground truth during this experiment to verify the accuracy of these scanners.

## 4.1 Approach

Once all the software was downloaded and installed, the Kali Linux and Metasploitable 2 virtual machines were spun up to conduct the vulnerability scans. A credentialed scan was used by both OpenVAS and Nessus on Metasploitable 2 by providing the IP address of Metasploitable 2 as the scanning target and the credentials associated with the machine—both username and password being “msfadmin”. After obtaining the results of each scan, the network scanner Nmap was then employed. As Nmap was already installed on the Kali Linux virtual machine, the command in *Figure 1* was inputted into the terminal to receive a list of ports active on the host as well as names of various software and their version numbers, if applicable, on each port.



```
┌──(kali㉿kali)-[~]

└─$ nmap -v -A 192.168.56.2
```

*Figure 1: Nmap command used to discover open ports and the software installed on the Metasploitable 2 virtual machine.*

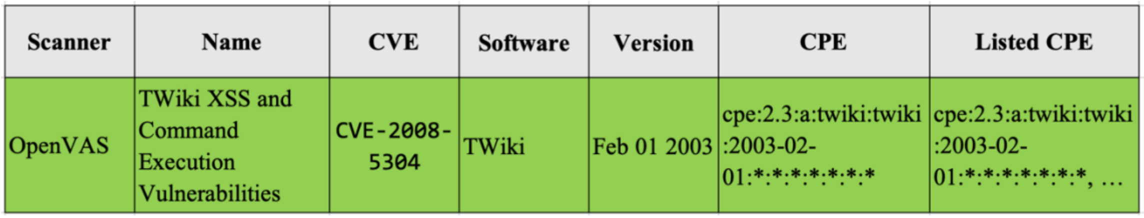
From there, the method of analyzing the found vulnerabilities consisted of determining the CPE name from the NVD based off the software name and version number. It is important to note that not all the vulnerabilities found were analyzed due to the large number of vulnerabilities and only those that returned a CVE associated with them were analyzed.

## 4.2 Results

In total, OpenVAS scan found 67 vulnerabilities, whereas the Nessus scan found 952 vulnerabilities on Metasploitable 2. Both scanners only found 15 vulnerabilities in common. These results indicate that the OpenVAS scan had false negatives, vulnerabilities not found or reported, as the number of vulnerabilities found by OpenVAS was much lower than Nessus, undermining the reliability of this scanner. The large discrepancy between the results of the two scanners could be attributed to the filtering OpenVAS does before reporting the vulnerabilities found as prior to this filtering, 571 vulnerabilities were found.

After receiving the vulnerabilities from each scanner, 100 were selected out of the combined vulnerabilities to be analyzed. These 100 vulnerabilities included the common vulnerabilities found by both scanners, all the OpenVAS vulnerabilities found, and the remaining vulnerabilities were selected at random out of the vulnerabilities found by Nessus. These 100 vulnerabilities and the results of the analysis were compiled into a table with column headings: scanner, name, CVE, software, version, CPE, and listed CPE. The "scanner" column lists the scanner that the vulnerability was found in and includes both if it was a shared vulnerability found. The "name" column lists the description given by the scanner for the vulnerability. The "CVE" column lists the CVE associated with the vulnerability and is hyperlinked to the NVD entry for it.

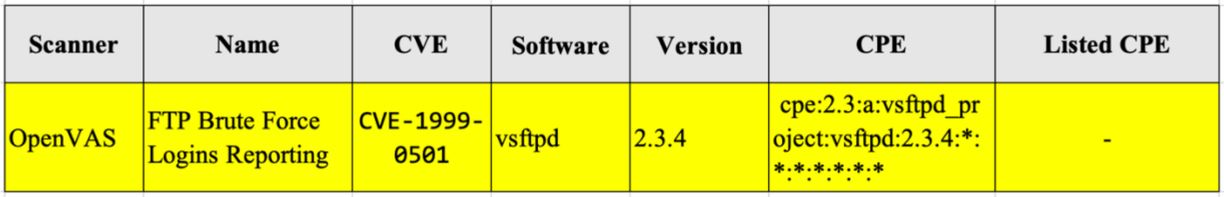
The “software” and “version” columns list the software and version numbers respectively that are associated with the vulnerability. The “listed CPE” column contains CPEs that were listed under the known affected software configurations in the CVE entry. Typically, the CVE listed several CPEs under this section, so “…” were used to denote that more CPEs were listed other than the one stated in this column. To indicate the accuracy, each vulnerability was highlighted a certain color. A vulnerability was deemed accurate and highlighted green if the CPE for the software was listed under known affected software configurations in the CVE associated with the vulnerability as shown in *Figure 3*.



| Scanner | Name | CVE | Software | Version | CPE | Listed CPE |
|---|---|---|---|---|---|---|
| OpenVAS | TWiki XSS and Command Execution Vulnerabilities | CVE-2008-5304 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:*, … |

*Figure 3: Accurate vulnerability result.*

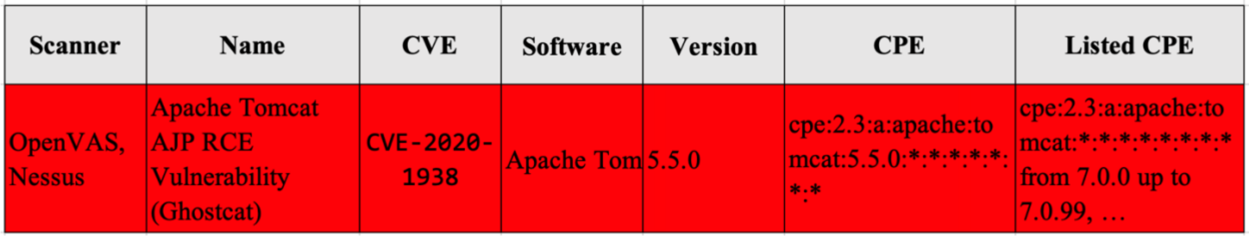
A vulnerability was deemed inconclusive and highlighted yellow if no CPEs were listed to check against under known affected software configurations in the associated CVE as shown in *Figure 4*.



| Scanner | Name | CVE | Software | Version | CPE | Listed CPE |
|---|---|---|---|---|---|---|
| OpenVAS | FTP Brute Force Logins Reporting | CVE-1999-0501 | vsftpd | 2.3.4 | cpe:2.3:a:vsftpd_project:vsftpd:2.3.4:*:*:*:*:*:*:* | - |

*Figure 4: Inconclusive vulnerability result.*

A vulnerability was deemed inaccurate and highlighted red if the CPE for the software was not listed under known affected software configurations in the associated CVE as shown in *Figure 5*.



| Scanner | Name | CVE | Software | Version | CPE | Listed CPE |
|---|---|---|---|---|---|---|
| OpenVAS,<br>Nessus | Apache Tomcat AJP RCE Vulnerability (Ghostcat) | CVE-2020-1938 | Apache Tom | 5.5.0 | cpe:2.3:a:apache:tomcat:5.5.0:*:*:*:*:*:* | cpe:2.3:a:apache:tomcat:*:*:*:*:*:*:*:* from 7.0.0 up to 7.0.99, … |

*Figure 5: Inaccurate vulnerability result.*

As the table comprised of all the vulnerabilities analyzed is quite extensive, the full table can be found in the appendix.

As there were 6 vulnerabilities that could not be verified as accurate or inaccurate, these vulnerabilities are not included in calculating the accuracy of these two scanners. Out of the 25 vulnerabilities found for OpenVAS with a CVE, 21 vulnerabilities were an accurate result, making the accuracy of OpenVAS 84%. Out of the 79 vulnerabilities selected from Nessus, 64 vulnerabilities were an accurate result, giving Nessus an accuracy of 81%. The difference in the accuracy rating between these two scanners is fairly slim with only a 3% difference. However, it's important to note that the Nessus accuracy statistic may be higher or lower depending on the full analysis of the vulnerabilities found. Factoring in other aspects such as that OpenVAS is open source and Nessus is commercial, one may be more inclined to choose OpenVAS if they do not wish to invest in a vulnerability scanner. However, OpenVAS is not multi-platform, only available on Linux environments, so it may not be a viable option for some. Factoring in other aspects such as that Nessus was able to detect more vulnerabilities than OpenVAS, one may be inclined to choose Nessus instead. Regardless, these results indicate that both scanners reported false positives which further undermines their reliability. From this study, it can be concluded that while vulnerability scanners do help with vulnerability management in a system, network, application, etc., they are not the sole solution for its security. It is crucial to not solely rely on these scanners as false negatives and positives are possible as seen by the results of this study.

# 5 Conclusion & Future Work

A vulnerability scanner is an important and useful tool that aids in maintaining the security of an application, network, and/or database. There are many different types of scanners such as an application scanner, a network scanner, and a database scanner among others, and these all incorporate different forms of testing—white box, black box, authenticated, unauthenticated—to identify vulnerabilities. It can be challenging to figure out which scanner available best fits one's needs. Therefore, a survey was conducted to ease that burden and offer a variety of options that highlight features or tested vulnerabilities one might prefer. A study was also conducted on two network scanners—OpenVAS and Nessus—to compare the accuracy of these scanners, where it was found that these scanners were fairly accurate but perhaps not ideal.

This work could be extended upon by analyzing all the vulnerabilities Nessus found rather than just a portion for a more accurate analysis. It would be worthwhile to inspect the coverage of each vulnerability scanner as well. This could be done by looking at the list of CVEs related to each CPE for the software installed on the Metasploitable 2 virtual machine and examining it against the vulnerabilities found by the scanners for that particular CPE. Expanding from network scanners and analyzing the accuracy and coverage of other types of scanners such as application and database scanners may provide more insight to ideal vulnerability scanners as well.

# 6 Appendix

## 6.1 OpenVAS and Nessus Comparative Analysis Table

| Scanner | Name | CVE | Software | Version | CPE | Listed CPE |
|---|---|---|---|---|---|---|
| OpenVAS, Nessus | rLogin Service Detection / The rlogin service is running | CVE-1999-0651 | | | | - |
| OpenVAS, Nessus | rexecd Service Detection / The rexec service is running | CVE-1999-0618 | | | | - |
| OpenVAS, Nessus | ICMP Timestamp Request Remote Date Disclosure | CVE-1999-0524 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:-:*:*:*:*:*:*:*, … |
| OpenVAS | TWiki XSS and Command Execution Vulnerabilities | CVE-2008-5304 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:*, … |
| OpenVAS | TWiki XSS and Command Execution Vulnerabilities | CVE-2008-5305 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:*:*:*:*:*:*:*:* up to 4.2.3 (including), … |
| OpenVAS, Nessus | Apache Tomcat AJP RCE Vulnerability (Ghostcat) | CVE-2020-1938 | Apache Tomcat | 5.5.0 | cpe:2.3:a:apache:tomcat:5.5.0:*:*:*:*:*:*:* | cpe:2.3:a:apache:tomcat:*:*:*:*:*:*:*:* from 7.0.0 up to 7.0.99, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| OpenVAS | DistCC RCE Vulnerability (CVE-2004-2687) | CVE-2004-2687 | Samba | 3.0.20 | cpe:2.3:a:samba:samba:3.0.20:-:*:*:*:*:*:* | cpe:2.3:a:samba:samba:*:*:*:*:*:*:*:* up to 2.18.3 (including), … |
| OpenVAS, Nessus | PHP-CGI-based setups vulnerability when parsing query string parameters from php files. | CVE-2012-1823 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| OpenVAS, Nessus | PHP-CGI-based setups vulnerability when parsing query string parameters from php files. | CVE-2012-2311 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| OpenVAS | PHP-CGI-based setups vulnerability when parsing query string parameters from php files. | CVE-2012-2336 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| OpenVAS | PHP-CGI-based setups vulnerability when parsing query string parameters from php files. | CVE-2012-2335 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.3.12:*:*:*:*:*:*:*,<br>cpe:2.3:a:php:php:5.4.2:*:*:*:*:*:*:* |
| OpenVAS | FTP Brute Force Logins Reporting | CVE-1999-0501 | vsftpd | 2.3.4 | cpe:2.3:a:vsftpd_project:vsftpd:2.3.4:*:*:*:*:*:*:* | - |
| OpenVAS | FTP Brute Force Logins Reporting | CVE-1999-0507 | vsftpd | 2.3.4 | cpe:2.3:a:vsftpd_project:v | - |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | sftpd:2.3.4:*:*:*:*:*:*:*:* | |
| OpenVAS | FTP Brute Force Logins Reporting | CVE-1999-0508 | vsftpd | 2.3.4 | cpe:2.3:a:vsftpd_project:vsftpd:2.3.4:*:*:*:*:*:*:*:* | - |
| OpenVAS | SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability | CVE-2014-0224 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:*:*:*:*:*:*:*:* up to 0.9.8za (excluding), … |
| OpenVAS | TWiki Cross-Site Request Forgery Vulnerability - Sep10 | CVE-2009-4898 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:*:*:*:*:*:*:*:* up to 4.3.1 (including), … |
| OpenVAS | Anonymous FTP Login Reporting | CVE-1999-0497 | | | | - |
| OpenVAS | TWiki < 6.1.0 XSS Vulnerability | CVE-2018-20212 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:6.0.2:*:*:*:*:*:*:* |
| OpenVAS | jQuery < 1.9.0 XSS Vulnerability | CVE-2012-6708 | jQuery | 1.3.2 | cpe:2.3:a:jquery:jquery:1.3.2:*:*:*:*:*:*:* | cpe:2.3:a:jquery:jquery:*:*:*:*:*:*:*:* up to 1.9.0 (excluding) |
| OpenVAS | TWiki Cross-Site Request Forgery Vulnerability | CVE-2009-1339 | TWiki | Feb 01 2003 | cpe:2.3:a:twiki:twiki:2003-02-01:*:*:*:*:*:*:* | cpe:2.3:a:twiki:twiki:*:*:*:*:*:*:*:* up to 4.3.0 (including), … |
| OpenVAS | Samba MS-RPC Remote Shell Command Execution Vulnerability | CVE-2007-2447 | Samba | 3.0.20 | cpe:2.3:a:samba:samba:3.0.20:*:*:*:*:*:*:* | cpe:2.3:a:samba:samba:3.0.20:*:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | - Active Check | | | | | |
| OpenVAS, Nessus | SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption) | CVE-2016-0800 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:1.0.1:*:*:*:*:*:*:*, cpe:2.3:a:openssl:openssl:1.0.2:*:*:*:*:*:*:*, … |
| OpenVAS, Nessus | SSL/TLS: SSLv3 Protocol CBC Cipher Suites Information Disclosure Vulnerability (POODLE) | CVE-2014-3566 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:*, … |
| OpenVAS | /doc directory browsable | CVE-1999-0678 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:-:*:*:*:*:*:*:*, … |
| OpenVAS | SSL/TLS: Renegotiation DoS Vulnerability | CVE-2011-1473 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:*:*:*:*:*:*:*:* up to 0.9.8k (including), … |
| OpenVAS, Nessus | SSL/TLS: Report Weak Cipher Suites | CVE-2015-4000 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:*:*:*:*:*:*:*:* up to 1.0.1m (including), … |
| OpenVAS | jQuery < 1.6.3 XSS Vulnerability | CVE-2011-4969 | jQuery | 1.3.2 | cpe:2.3:a:jquery:jquery:1.3.2:*:*:*:*:*:*:* | cpe:2.3:a:jquery:jquery:*:*:*:*:*:*:*:* up to 1.6.2 (including), … |
| OpenVAS, Nessus | SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade | CVE-2015-0204 | OpenSSL | 0.9.8g | cpe:2.3:a:openssl:openssl:0.9.8g:*:*:*:*:*:*:* | cpe:2.3:a:openssl:openssl:*:*:*:*:*:*:*:* up to 0.9.8zc (including), … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Issue (FREAK) | | | | | |
| OpenVAS | phpMyAdmin 'error.php' Cross Site Scripting Vulnerability | CVE-2010-4480 | phpMyAdmin | 3.3.1 | cpe:2.3:a:phpmyadmin:phpmyadmin:3.3.1.0:*:*:*:*:*:*:* | cpe:2.3:a:phpmyadmin:phpmyadmin:3.3.8.1:*:*:*:*:*:*:*, cpe:2.3:a:phpmyadmin:phpmyadmin:3.3.9.0:*:*:*:*:*:*:* |
| OpenVAS, Nessus | Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability | CVE-2012-0053 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:*:*:*:*:*:*:*:* from (including) 2.2.0 up to 2.2.22 (excluding), … |
| OpenVAS, Nessus | SMTP Service STARTTLS Plaintext Command | CVE-2011-0411 | Postfix | 2.5.1 | cpe:2.3:a:postfix:postfix:2.5.1:*:*:*:*:*:*:* | cpe:2.3:a:postfix:postfix:2.5.1:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-0420 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.3.5:*:*:*:*:*:*:* |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2006-7243 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / | CVE-2010-4697 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | | | | | |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2010-4698 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-0421 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-0441 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.3.5:*:*:*:*:*:*:* |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-0708 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1072 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:pear:*:*:*:*:*:*:*:* versions up to 1.9.1 (inclusive), … |

| | | | | | | |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1092 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1144 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:pear:*:*:*:*:*:*:*:* versions up to 1.9.1 (inclusive), … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1148 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1153 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1464 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : | CVE-2011-1466 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | php5 vulnerabilities (USN-1126-1) | | | | | |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1467 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1468 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1469 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1470 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 / 11.04 : php5 vulnerabilities (USN-1126-1) | CVE-2011-1471 | PHP | 5.2.4 | cpe:2.3:a:php:php:5.2.4:*:*:*:*:*:*:* | cpe:2.3:a:php:php:*:*:*:*:*:*:*:* up to 5.2.2 (including), … |

| | | | | | | |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 6.06 LTS / 8.04 LTS / 9.10 / 10.04 LTS / 10.10 : samba vulnerability (USN-1075-1) | CVE-2011-0719 | Samba | 3.0.20 | cpe:2.3:a:samba:samba:3.0.20:*:*:*:*:*:*:* | cpe:2.3:a:samba:samba:3.0.20:*:*:*:*:*:*:*, … |
| Nessus | Samba Badlock Vulnerability | CVE-2016-2118 | Samba | 3.0.20 | cpe:2.3:a:samba:samba:3.0.20:*:*:*:*:*:*:* | cpe:2.3:a:samba:samba:*:*:*:*:*:*:*:* from 3.6.0 (including) up to 4.2.10 (excluding), … |
| Nessus | Ubuntu 6.06 LTS / 7.10 / 8.04 LTS : linux-source-2.6.15/22, linux vulnerabilities (USN-714-1) | CVE-2008-5079 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:, … |
| Nessus | Ubuntu 6.06 LTS / 7.10 / 8.04 LTS : linux-source-2.6.15/22, linux vulnerabilities (USN-714-1) | CVE-2008-5134 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2007-5925 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:mysql:mysql:*:*:*:*:*:*:*:* up to 5.1.23_bk (including) |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : postgresql- | CVE-2012-0866 | PostgreSQL | 8.3 | cpe:2.3:a:postgresql:postgresql:8.3:*:*:*:*:*:*:* | cpe:2.3:a:postgresql:postgresql:8.3:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 8.3, postgresql-8.4, postgresql-9.1 vulnerabilities (USN-1378-1) | | | | | |
| Nessus | ISC BIND Service Downgrade / Reflected DoS | CVE-2020-8616 | ISC Bind | 9.4.2 | cpe:2.3:a:isc:bind:9.4.2:*:*:*:*:*:*:* | cpe:2.3:a:isc:bind:*:*:*:*:*:*:*:* from 9.0.0 (including) up to 9.11.18 (including), … |
| Nessus | ISC BIND Denial of Service | CVE-2020-8617 | ISC Bind | 9.4.2 | cpe:2.3:a:isc:bind:9.4.2:*:*:*:*:*:*:* | cpe:2.3:a:isc:bind:*:*:*:*:*:*:*:* from 9.0.0 (including) up to 9.11.18 (including), … |
| Nessus | ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS | CVE-2020-8622 | ISC Bind | 9.4.2 | cpe:2.3:a:isc:bind:9.4.2:*:*:*:*:*:*:* | cpe:2.3:a:isc:bind:*:*:*:*:*:*:*:* from 9.0.0 (including) up to 9.11.21 (including), … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : postgresql-8.3, postgresql-8.4, postgresql-9.1 vulnerabilities (USN-1789-1) | CVE-2013-1899 | Ubuntu | 8.0.4 | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:* | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:*, .. |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : postgresql-8.3, postgresql-8.4, | CVE-2013-1900 | Ubuntu | 8.0.4 | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:* | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:*, .. |

| | | | | | | |
|---|---|---|---|---|---|---|
| | postgresql-9.1 vulnerabilities (USN-1789-1) | | | | | |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : postgresql-8.3, postgresql-8.4, postgresql-9.1 vulnerabilities (USN-1789-1) | CVE-2013-1901 | Ubuntu | 8.0.4 | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:* | cpe:2.3:o:canonical:ubuntu_linux:8.04:-:lts:*:*:*:*:*, .. |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : perl vulnerability (USN-1770-1) | CVE-2013-1667 | Perl | 5.8.8 | cpe:2.3:a:perl:perl:5.8.8:*:*:*:*:*:*:*:* | cpe:2.3:a:perl:perl:5.8.8:*:*:*:*:*:*:* |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : apache2 vulnerabilities (USN-1368-1) | CVE-2011-3607 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : apache2 vulnerabilities (USN-1368-1) | CVE-2011-4317 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : apache2 vulnerabilities (USN-1368-1) | CVE-2012-0021 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:2.2.17:*:*:*:*:*:*:*, … |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : apache2 vulnerabilities (USN-1368-1) | CVE-2012-0031 | Apache | 2.2.8 | cpe:2.3:a:apache:http_server:2.2.8:*:*:*:*:*:*:* | cpe:2.3:a:apache:http_server:*:*:*:*:*:*:*:*:* from 2.2.0 (including) up to 2.2.22 (excluding) |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : postgresql-8.3, postgresql-8.4, postgresql-9.1 vulnerabilities (USN-1378-1) | CVE-2012-0867 | PostgreSQL | 8.3 | cpe:2.3:a:postgresql:postgresql:8.3:*:*:*:*:*:*:* | cpe:2.3:a:postgresql:postgresql:8.4:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : postgresql-8.3, postgresql-8.4, postgresql-9.1 vulnerabilities (USN-1378-1) | CVE-2012-0868 | PostgreSQL | 8.3 | cpe:2.3:a:postgresql:postgresql:8.3:*:*:*:*:*:*:* | cpe:2.3:a:postgresql:postgresql:8.3:*:*:*:*:*:*:*, …. |
| Nessus | Ubuntu 8.04 LTS : linux vulnerabilities (USN-1390-1) | CVE-2011-1476 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*:* up to 2.6.38.8 (including), … |
| Nessus | Ubuntu 8.04 LTS : linux vulnerabilities (USN-1390-1) | CVE-2011-1477 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*:* up to 2.6.39 (inclusive), … |
| Nessus | Ubuntu 8.04 LTS : linux vulnerabilities (USN-1390-1) | CVE-2011-2182 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*:* up to 2.6.39 (inclusive), … |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 8.04 LTS : linux vulnerabilities (USN-1390-1) | CVE-2011-4324 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* up to 2.6.28.10 (inclusive), … |
| Nessus | Ubuntu 8.04 LTS : linux vulnerabilities (USN-1390-1) | CVE-2012-0028 | Linux | 2.6.24 | cpe:2.3:o:linux:linux_kernel:2.6.24:*:*:*:*:*:*:* | cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* up to 2.6.27.62 (inclusive), … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2008-3963 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2008-4098 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2008-4456 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.45:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, | CVE-2008-7247 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | | | | | |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2009-2446 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2009-4019 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2009-4030 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:mysql:mysql:5.1.23:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2009-4484 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:*:*:*:*:*:*:*:* from 5.0.0 (including) up to 5.0.90 (excluding), … |

| | | | | | | |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2010-1621 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:mysql:mysql:*:*:*:*:*:*:*:* u pto 5.1.45 (inclusive), … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2010-1626 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:mysql:mysql:*:*:*:*:*:*:*:* up to 5.1.45 (inclusive), … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2010-1848 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2010-1849 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.0.51:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.04 / 11.10 / 12.04 LTS : bind9 | CVE-2012-5166 | ISC Bind | 9.4.2 | cpe:2.3:a:isc:bind:9.4.2:*:*:*:*:*:*:* | cpe:2.3:a:isc:bind:9.4.2:*:*:*:*:*:*:*, … |

| | | | | | | |
|---|---|---|---|---|---|---|
| | vulnerability (USN-1601-1) | | | | | |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2008-5983 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:*:*:*:*:*:*:*:* up to 2.6.6 (excluding), … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2010-1634 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:*:*:*:*:*:*:*:* from 2.5.0 (including) up to 2.5.6 (excluding), … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2010-2089 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:*:*:*:*:*:*:*:* from 2.5.0 (including) up to 2.5.6 (excluding), … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2010-3493 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:3.1:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2011-1015 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:3.0:*:*:*:*:*:*:* |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2011-1521 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2011-4940 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:*:*:*:*:*:*:*:* up to 2.5.6 (inclusive), … |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2011-4944 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:2.6.1:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2012-0845 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS : python2.5 vulnerabilities (USN-1613-1) | CVE-2012-0876 | Python | 2.5.2 | cpe:2.3:a:python:python:2.5.2:*:*:*:*:*:*:* | cpe:2.3:a:python:python:*:*:*:*:*:*:*:* from 2.6.0 (including) up to 2.6.8 (excluding), … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : perl vulnerabilities (USN-1643-1) | CVE-2011-2939 | Perl | 5.8.8 | cpe:2.3:a:perl:perl:5.8.8:*:*:*:*:*:*:* | cpe:2.3:a:perl:perl:5.8.8:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 11.10 / 12.04 LTS / 12.10 : perl vulnerabilities (USN-1643-1) | CVE-2012-5195 | Perl | 5.8.8 | cpe:2.3:a:perl:perl:5.8.8:*:*:*:*:*:*:* | cpe:2.3:a:perl:perl:5.12.0:*:*:*:*:*:*:*, … |
| Nessus | Ubuntu 8.04 LTS / 10.04 LTS / 10.10 / 11.04 / 11.10 : mysql-5.1, mysql-dfsg-5.0, mysql-dfsg-5.1 vulnerabilities (USN-1397-1) | CVE-2012-0117 | mySQL | 5.0.51a | cpe:2.3:a:oracle:mysql:5.0.51a:*:*:*:*:*:*:* | cpe:2.3:a:oracle:mysql:5.5.0:*:*:*:*:*:*:*, … |

# 7 References

[1] OWASP. (n.d.). *LDAP Injection*. OWASP. Retrieved from `https://owasp.org/www-community/attacks/LDAP_Injection`.

[2] OWASP. (n.d.). *Vulnerabilities*. OWASP. Retrieved from `https://owasp.org/www-community/vulnerabilities/`.

[3] Georgieva, E. (2022, October 9). *Black Box Penetration Testing: When Do You Need One?* PurpleSec. Retrieved from `https://purplesec.us/learn/black-box-penetration-testing/`.

[4] Synopsys. (n.d.). *Static Application Security Testing*. Synopsys. Retrieved from `https://www.synopsys.com/glossary/what-is-sast.html`.

[5] Check Point. (n.d.). *What is Dynamic Code Analysis?* Check Point. Retrieved from `https://www.checkpoint.com/cyber-hub/cloud-security/what-is-dynamic-code-analysis/`.

[6] Synopsys. (n.d.). *Black Duck Software Composition Analysis*. Synopsys. Retrieved from `https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html`.

[7] Balbix. (n.d.). *What are vulnerability scanners*. Balbix. Retrieved from `https://www.balbix.com/insights/what-to-know-about-vulnerability-scanning-and-tools/`.

[8] RSI Security. (2021, November 9). *7 Types of Vulnerability Scanners*. RSI Security. Retrieved from `https://blog.rsisecurity.com/7-types-of-vulnerability-scanners/`.

[9] RSI Security. (2020, August 28). *Top 5 Types of Penetration Testing*. RSI Security. Retrieved from `https://blog.rsisecurity.com/top-5-types-of-penetration-testing/`.

[10] Georgieva, E. (2022, November 2). *White Box Penetration Testing: When Do You Need One?* PurpleSec. Retrieved from `https://purplesec.us/learn/white-box-penetration-testing/`.

[11] TutorialsPoint. (n.d.). *Database Testing – Security.* TutorialsPoint. Retrieved from `https://www.tutorialspoint.com/database_testing/database_testing_security.htm#`.

[12] Sharma, L. (2023, March 16). *10 DevSecOps to Know as a Developer or Sysadmin.* Geekflare. Retrieved from `https://geekflare.com/devsecops-tools/`.

[13] NIST. (n.d.). *Database Scanning Tools.* NIST. Retrieved from `https://www.nist.gov/itl/ssd/software-quality-group/database-scanning-tools`.

[14] OWASP. (n.d.). *Cross Site Scripting (XSS).* OWASP. Retrieved from `https://owasp.org/www-community/attacks/xss/`.

[15] OWASP. (n.d.). *SQL Injection*. OWASP. Retrieved from `https://owasp.org/www-community/attacks/SQL_Injection`.

[16] Invicti. (n.d.). *Local file inclusion.* Invicti. Retrieved from `https://www.invicti.com/learn/local-file-inclusion-lfi/`.

[17] Invicti. (n.d.). *CRLF injection.* Invicti. Retrieved from `https://www.invicti.com/learn/crlf-injection/`.

[18] Invicti. (n.d.). *Server-side request forgery (SSRF).* Invicti. Retrieved from `https://www.invicti.com/learn/server-side-request-forgery-ssrf/`.

[19] Invicti. (n.d.). *XML external entity (XXE).* Invicti. Retrieved from `https://www.invicti.com/learn/xml-external-entity-xxe/`.

[21] Greenbone OpenVAS. (n.d.). *Greenbone OpenVAS*. Greebone OpenVAS. Retrieved from `https://www.openvas.org/`.

[22] Tenable. (n.d.). *Nessus*. Tenable. Retrieved from `https://www.tenable.com/products/nessus`.

[23] Nmap.org. (n.d.). *News*. Nmap.org. Retrieved from `https://nmap.org/`.

[24] Rapid7. (n.d.). *Metasploitable 2.* Rapid7. Retrieved from `https://docs.rapid7.com/metasploit/metasploitable-2/`.

[25] Oracle. (n.d.). *VirtualBox.* Oracle. Retrieved from `https://www.virtualbox.org/`.

[26] NIST. (n.d.). *National Vulnerability Database.* NIST. Retrieved from `https://nvd.nist.gov/`.

[27] Greenbone Security Manager. (n.d.). *21.19 Quality of Detection (QoD).* Greenbone Security Manger. Retrieved from `https://docs.greenbone.net/GSM-Manual/gos-20.08/en/glossary.html#quality-of-detection-qod`.

[28] Shankdhar, P. (2020, July 13). *14 best open-source web application vulnerability scanners [updated for 2020].* Infosec. Retrieved from `https://resources.infosecinstitute.com/topic/14-popular-web-application-vulnerability-scanners/`.

[29] Kumar, C. (2022, September 6). *12 Open Source Web Security Scanner to Find Vulnerabilities.* Geekflare. Retrieved from `https://geekflare.com/open-source-web-security-scanner/`.

[30] Aqua. (n.d.). *Open Source Vulnerability Scanning: Methods and Top 5 Tools.* Aqua. Retrieved from `https://www.aquasec.com/cloud-native-academy/vulnerability-management/open-source-vulnerability-scanning/`.

[31] Snyk. (n.d.). *7 Reasons to use an open source vulnerability scanner*. Snyk. Retrieved from `https://snyk.io/series/open-source-security/open-source-vulnerability-scanners/`.

[32] Breachlock. (2023, March 6). *Top 5 open-source tools for network vulnerability scanning.* Breachlock. Retrieved from `https://www.breachlock.com/resources/blog/top-5-open-source-tools-for-network-vulnerability-scanning/`. Breachlock. Retrieved from `https://www.breachlock.com/resources/blog/top-5-open-source-tools-for-network-vulnerability-scanning/`.

[33] Aqua. (n.d.). *Aqua Trivy: Vulnerability and Misconfiguration Scanning*. Retrieved from `https://www.aquasec.com/products/trivy/`.

[34] Github. (n.d.). *Clair*. Github. Retrieved from `https://github.com/quay/clair`.

[35] Github. (n.d.). *Anchore Engine*. Github. Retrieved from `https://github.com/anchore/anchore-engine`.

[36] Sqlmap. (n.d.). *Introduction*. Sqlmap. Retrieved from `https://sqlmap.org/`.

[37] Wapiti. (2023, January 16). *Wapiti.* Wapiti. Retrieved from `https://wapiti-scanner.github.io/`.

[38] Github. (n.d.). *VCG (VisualCodeGrepper)*. Github. Retrieved from `https://github.com/nccgroup/VCG`.

[39] Github. (n.d.). *Brakeman.* Github. Retrieved from `https://github.com/presidentbeef/brakeman`.

[40] Github. (n.d.). *Bandit*. Github. Retrieved from `https://github.com/PyCQA/bandit`.

[41] Splint. (n.d.). *Splint.* Splint. Retrieved from `https://splint.org/`.

[42] Github. (n.d.). *UNO*. Github. Retrieved from `https://github.com/nimble-code/Uno`.

[43] Checkstyle. (2023, March 25). *Checkstyle.* Checkstyle. Retrieved from `https://checkstyle.org/`.

[44] OWASP. (n.d.). *OWASP Zed Attack Proxy (ZAP).* OWASP. Retrieved from `https://www.zaproxy.org/`.

[45] PortSwigger. (n.d.). *What do you want to do with Burp Suite?* PortSwigger. Retrieved from `https://portswigger.net/burp`.

[46] Subgraph. (n.d.). *Vega helps you find and fix cross-site scripting (XSS), SQL injection, and more.* Subgraph. Retrieved from `https://subgraph.com/vega/`.

[47] Github. (2022, October 8). *Home.* Github. Retrieved from `https://github.com/sullo/nikto/wiki`.

[48] Rapid7. (n.d.). *Nexpose Vulnerability Scanner.* Rapid7. Retrieved from `https://www.rapid7.com/products/nexpose/`.

[49] OpenSCAP. (n.d.). *Tools*. OpenSCAP. Retrieved from `https://www.open-scap.org/`.

[50] Wireshark. (n.d.). *Download Wireshark.* Wireshark. Retrieved from https://www.wireshark.org/download.html.

[51] Aircrack-ng. (n.d.). *Downloads.* Aircrack-ng. Retrieved from https://www.aircrack-ng.org/downloads.html.

[52] Github. (n.d.). *vaf*. Github. Retrieved from https://github.com/d4rckh/vaf.

[53] Gaucher, R. (n.d.). *Grabber*. Retrieved from http://rgaucher.info/beta/grabber/.

[54] Detectify. (n.d.). *Complete External Attack Surface Management for AppSec & ProdSec teams*. Detectify. Retrieved from https://detectify.com/.

[55] Rapid7. (n.d.). *Get Metasploit.* Rapid7. Retrieved from https://www.metasploit.com/download.

[56] Deepfence. (n.d.). *ThreatMapper*. Deepfence. Retrieved from https://deepfence.io/threatmapper/.

[57] Github. (n.d.). *Watchdog.* Github. Retrieved from https://github.com/flipkart-incubator/watchdog.

[58] Snyk. (n.d.). *Developer loved, Security trusted.* Snyk. Retrieved from https://snyk.io/.

[59] Mend.io. (n.d.). *Shift Left. Secure at Scale.* Mend.io. Retrieved from https://www.mend.io/.

[60] Arachni. (n.d.). *Arachni.* Arachni. Retrieved from https://www.arachni-scanner.com/.

[61] Github. (n.d.). *XssPy – Web Application XSS Scanner.* Github. Retrieved from https://github.com/faizann24/XssPy.

[62] W3af. (n.d.). *SQL injection, Cross-Site scripting and much more*. W3af. Retrieved from https://w3af.org/.

[63] Github. (n.d.). *Wfuzz – The Web Fuzzer*. Github. Retrieved from https://github.com/xmendez/wfuzz.

[64] Greenbone Security Manager. (n.d.). *CVEs 212793 of 212793*. Greenbone Security Manager. Retrieved from `https://secinfo.greenbone.net/cves`.

[65] Tenable. (n.d.). *CVEs.* Tenable. Retrieved from `https://www.tenable.com/cve`.