# A Critical Review of WordPress Security Scanning Tools and the Development of a Next-Generation Solution

MSc Research Project
MSc in Cybersecurity

Deepti Gupta
Student ID: x21135193

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Deepti Gupta |
| **Student ID:** | X21135193 |
| **Programme:** | MSc in Cybersecurity |
| **Year:** | 2022 |
| **Module:** | MSc Internship |
| **Supervisor:** | Mr. Vikas Sahni |
| **Submission Due Date:** | 06/01/2023 |
| **Project Title:** | A Critical Review of WordPress Security Scanning Tools and the Development of a Next-Generation Solution |
| **Word Count:** | 8721 |
| **Page Count:** | 22 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** ……………Deepti Gupta……….…………………..…………………………………………

**Date:** ……………05/01/2023………………………………………………………………………

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# A Critical Review of WordPress Security Scanning Tools and the Development of a Next-Generation Solution

Deepti Gupta
X21135193

## Abstract

WordPress is widely used content management system (CMS), with 455 million websites supported and a 60.3% market share.92% of the vulnerabilities found in the WordPress-powered websites are due to the third-party plugins and programming errors. This paper conducts a critical review of popular tools designed to automate the process of testing the security of WordPress websites by evaluating their effectiveness, ease of use, and overall value. It provides an overview of the current state of WordPress security and the importance of regular testing. This provides the reader with a comprehensive understanding of available automation tools and assist them in making informed decisions about their use. Also a framework is proposed to address the gaps which are not covered by other tools. The proposed DeepRecon tool is aimed to automate reconnaissance and information collecting operations primarily for WordPress websites, but it is also compatible with other CMS and online applications.

Keywords: Wordpress, vulnerability, CMS, python, bash, burpsuite, WPScan,Nuclei, Nikto

# 1 Introduction

WordPress is widely used content management system (CMS) worldwide, powering more than 43% of all websites on the internet[1]. It is open-source and allows users to easily add third-party plugins and themes to extend its capabilities. The WordPress plugin repository provides access to over 60,000 plugins[2]. In 2020, 582 security flaws were discovered in both the WordPress core and third-party themes and plugins. Cross-site scripting vulnerabilities were the most common, accounting for over 36.2% of all new vulnerabilities discovered. SQL injection and cross-site request forgery accounted for 9.1% and 6.5%, respectively.

582 security flaws were discovered in 2020 alone. Both the WordPress core add-ons and third-party themes were vulnerable. Cross-site scripting vulnerabilities are the most prominent, accounting for more than 36.2% of all new vulnerabilities discovered in 2020[3]. 9.1% of the vulnerabilities are caused by SQL Injection, and 6.5% are caused by Cross-Site Request Forgery.

This research aims to develop a framework for automating penetration testing tools and creating a workflow to ensure that no vulnerabilities go unchecked. The framework focus on the WordPress content management system and is based on a comparative survey of existing automated penetration testing tools. The proposed framework aims to address any missing components in current tools and provide recommendations for improving their effectiveness.

**Research Question:**

- The research question for this study is: What is the most efficient automated penetration testing tool for scanning WordPress and its components?

[1] Wordpress: https://wordpress.org/about/features/

[2] Wordpress Plugins: https://wpbeginner.com/showcase/24-must-have-wordpress-plugins-for-business-websites/

[3] Security Flaws in Wordpress research: https://patchstack.com/wp-content/uploads/2021/04/Security-vulnerabilities-of-WordPress-ecosystem-in-2020-Patchstack-1.pdf

**Objective:**

To answer the research question, the study have the following objectives:

- Develop a framework to scan the WordPress content management system.
- Conduct a research-based comparative survey of automated penetration testing tools that are already available in the market.
- Identify any missing components in current tools that should be included in the proposed framework.

To achieve this, the research first conduct a literature review of penetration testing methodologies, including manual and automated testing methods. The review also cover existing tools and their capabilities for assessing the security of WordPress websites. Next, the research focus on the design and specifications of the proposed framework, including its components and the processes it uses to scan WordPress sites for vulnerabilities. This include the use of network scanners, vulnerability scanners, and other tools to assess the security of WordPress sites.

The framework is evaluated using a set of criteria, including its accuracy, efficiency, and overall effectiveness in identifying and reporting vulnerabilities. The evaluation also consider the potential implementation of the proposed framework, including any challenges or limitations that may arise. Finally, the research present the results of the evaluation and provide recommendations for future work, including potential improvements to the proposed framework and its implementation. The paper conclude with a summary of the key findings and implications of the research.

# 2 Related Work

The initial study concentrates on related work in pen testing and automation tools. It describes penetration testing processes and discusses some of the tools used in penetration testing or security assessments of WordPress websites. The rest of the chapter discusses the proposed tool and how it will fill in the gaps left by other tools.

## 2.1 Penetration test

Pentesting, also known as ethical hacking, is the process of simulating a cyber-attack on a network, or web apps in order to test and identify vulnerabilities. The goal of pentesting is to evaluate the security of a system and provide recommendations for improving its defenses against potential threats. Pentesting is typically performed by security professionals with expertise in ethical hacking and network security. Large corporations have increased their use of penetration testing to secure their information systems and services. This enables organisations to address security issues before they are exploited. A pen test can be performed in two ways (SecureCoding, 2021) : Manual Penetration test & Automated Penetration test.

## 2.2 Overview of the OWASP Top Ten and Its Relevance to WordPress

The following points shows an overview of different vulnerabilities that are mostly found in WordPress and are the components of OWASP:

**A1 – Broken Access Control:**

It is a web application security risk that can occur when the mechanisms for controlling access to a system or its resources are inadequate or not properly implemented (M. M. Hassan, Oct 18-20, 2018).

**A2 – Cryptographic Failures:**

It refer to weaknesses in the cryptographic algorithms or protocols used to secure communication and data (Weitian Xing, 11 July 2021). Cryptographic failures can occur when the cryptography used is weak or outdated, or when cryptographic functions are misused or implemented incorrectly.

**A3 – Sensitive Data Exposure:**

Sensitive Data Exposure attacks are attacks that expose critical data, such as business information, financial information or personally identifiable information (PII) (Christoph Stach, 18 October 2022).

**A4 – Injection:**

Injection attacks in which malicious code is injected into a web application, such as SQL injection or code injection (Youkun Shi, August 10–12, 2022).

**A5 – Security Misconfiguration:**

Security Misconfiguration is a web application security risk that can occur when a system or its components are not properly configured, resulting in vulnerabilities that can be exploited by attackers (Loureiro, 2021).

**A6 – Vulnerable and Outdated Components:**

It is a web application security risk that can arise when the components used in a web application, such as libraries, frameworks, and plugins, are out of date or have known vulnerabilities (Zerouali, 2019).

**A7 - Identification and Authentication Failures:**

Identification and Authentication Failures are web application security risks that can occur when the mechanisms for identifying and authenticating users are inadequate or not properly implemented (Hassan, 2018). This can allow attackers to gain unauthorized access to a system or its data.

**A8 - Software and Data Integrity Failures:**

Software and Data Integrity Failures are web application security risks that can occur when the integrity of software or data is compromised (Yang, May 2020). This can happen when an attacker can modify or manipulate the software or data in a way that affects its functionality or accuracy.

**A9 – Security Logging and Monitoring Failures:**

This category covers failures to properly implement security logging and monitoring, which can prevent an organization from detecting and responding to security incidents promptly (Nagendran, April 2020 ).

**A10 – Server-Side Request Forgery:**

SSRF is a vulnerability of web application that allows a malicious attacker to send malicious requests from a vulnerable server to other servers or systems (Kadir, 2021).

## 2.3 An Overview of WordPress Security and the Evaluation of Security Scanning Tools

Ethical hacking or penetration testing is often divided into some phases that are five in the count.

### 2.3.1 Password Attacks on WordPress

The authors state in (Ar Kar Kyaw, n.d.) that website authentication is vulnerable to various forms of password attacks, including dictionary attacks and brute force attacks. To further illustrate how dictionary attacks are performed to break WordPress authentication, they simulate attacks. An attacker would have to determine the username before guessing the password. Additionally, collecting personal information also helps in effective password guessing. Dictionary attacks prove that using 7-character lowercase passwords can be easily cracked by dictionaries available online and are well known. Some countermeasures can be applied to WordPress, such as lockdown systems, 2 factor authentication, strong passwords, and not using the real names (Ar Kar Kyaw, n.d.). These mitigations are easily available with WordPress plugins such as WordFence. Web forensics is primarily focused on capturing and analyzing logs from web servers, application servers, database servers, and WordPress applications.

### 2.3.2 Security of WordPress Plug-Ins Overview

322 randomly selected WordPress plugins were analyzed and the WordPress plugin site allowed users to rate plugins, and they compared these plugin ratings against vulnerabilities identified by static analysis. They found only a weak nonlinear correlation between ratings and the number of vulnerabilities. Based on their findings, they believed that there are real risks to using third-party plugins on the WordPress site (Teemu Koskinen, n.d.). And download counts don't help in finding a safe plugin, so proper verification via static analysis or manual verification before using any plugin on the WordPress site should be done. The high cost of software development and the industry's short timelines make installing plugins an appealing solution, but their findings encourage developers to review their code before using it.

### 2.3.3 Automated Black-Box Web Application Vulnerability Analysis

The vulnerabilities that current black box scanners are designed to detect, as well as their efficacy in detecting those vulnerabilities, were investigated (Jason Bau, n.d.). According to their research on vulnerabilities in real web applications, the most common types of vulnerabilities are XSS, SQLi, and sensitive information disclosure. They also found that black-box web application vulnerability scanners generally perform testing roughly in proportion to the total number of vulnerabilities actually in existence. They said that one of the tools succeeded in injecting a saved JavaScript alert (), but later could not identify it as saved XSS, so the latter statement could be confirmed anecdotally. Therefore, they believe that a better understanding of the application database model by the tool can improve detection rates.

### 2.3.4 An overview of WordPress and Drupal CMS

A wonderful and relatively simple to maintain platform for creating online content is a content management system (Arafath, 8 December 2021). It enables non-technical people to publish their websites without being familiar with web programming standards. The primary purpose of a CMS is to publish material via a frontend interface while simultaneously storing it on the backend. The ability of content management systems to work with other technologies such as chatbots (Arafath, 8 December 2021), user interfaces, and virtual and augmented reality determine their future.

### 2.3.5 A review of web application security scanners' quality

First off, a complex technique aids in precisely defining the strengths and weaknesses of a web application security scanner, particularly when looking for poorly constructed algorithms (Lim Kah Seng, 20-September-2018). Second, to provide a platform that enables researchers to accurately and scientifically communicate to the public any hypothesis, idea, algorithm, or accomplishment in the area of automated web app pentesting. The third and final factor that has a considerable impact on the development of this study subject is accurate methodology (Lim Kah Seng, 20-September-2018). Practitioners then assessed the quality of a scanner by counting the number of vulnerabilities discovered. According to the survey, practitioners use a variety of methodologies, scanners, and measurement metrics to determine the quality of web app security scanners.

### 2.3.6 Comparative Study of Web CMS

WCMS are simple to use and maintain (Jose-Manuel Martinez-Caro, 27 January), so they offer a versatile way to display content online. Indeed, WCMS is used to implement almost 50% of Internet web pages. The fact that they are designed for a very broad audience and don't require advanced computer skills, as well as the fact that they provide a wide range of features, are two factors contributing to their success. They demonstrated how to use three different content management systems (WCMS), namely Joomla, WordPress, and Drupal, to build three websites with the same graphical design and functions. According to their study, Joomla! had the biggest and busiest user community (Jose-Manuel Martinez-Caro, 27 January). Drupal was the most difficult to maintain, but it appeared to be more reliable in terms of security and user roles. WordPress offered a free hosting option, that was less sophisticated than Drupal, and had the best SEO positioning.

# 3 Research Methodology

## 3.1 Available Tools:

There are several tools that are being used for the automated scanning of WordPress and some of them were compared above. The most efficient technique to check a website for a wide range of known vulnerabilities is generally believed to be vulnerability scanning. Vulnerability detection software continuously analyses and audits websites or online applications to detect flaws and provide patches. Three fundamental methods make up website vulnerability scanning. The vulnerability assessment tool's initial phase involves running a vulnerability test to find and classify potential attack surfaces.

It enables the identification of security loopholes in the network and the closing of them before intruders can get in. The second phase involves evaluating vulnerabilities to assist administrators in setting priorities for their course of action. Missing updates, script flaws, and misconfigurations are a few examples of these vulnerabilities. Threats are ranked in order of importance according to age and risk. In general, web vulnerability scanners do not offer a mechanism to automatically fix discovered vulnerabilities. They concentrate more on monitoring and giving information so that administrators can take additional action. However, some scanners solve setup issues, saving the administrator hours of labor by simultaneously accessing numerous devices.

To test common WordPress vulnerabilities, a security scanner is utilized that is sometimes hosted online. As worrying as it sounds, in the last four years there has been a more than 200% increase in vulnerabilities in web applications. Manual efforts to find these vulnerabilities have consistently fallen short as the number has risen. Even experienced IT administrators are unable to fulfill the task of being aware of every vulnerability that has been found and the updates that have been made to address those vulnerabilities. Because of this, modern vulnerability scanners are made to support them by enabling them to find and handle these software issues.

Web services have evolved into a crucial pillar for satisfying client needs and maintaining an organization's competitiveness in the digital era. Corporate systems, however, require cutting-edge solutions to keep them safe due to their high level of online exposure. Vulnerability scanning can be used as part of a comprehensive continuous security monitoring plan or as a stand-alone evaluation. It can identify and prevent WordPress malware attacks, SQL injections, backdoor installations, drive-by downloads, and malicious redirections to external websites by periodically using these tools. But keep in mind that there are still additional ways for malicious attackers to access servers and websites beyond the application security layer.

## 3.2 Factors Influencing the Choice of Tools Selection

WPSCAN, OWASP ZAP, Burp Suite, Nuclei, and Nikto were chosen for comparison for WordPress security scanning. These tools were selected for several reasons.

- First, they are all well-known and widely used tools in the cybersecurity community. By comparing these familiar tools, a relevant and relatable analysis was provided for the readers.
- Second, these tools cover a range of capabilities and approaches to vulnerability scanning. WPSCAN is specifically designed for use with WordPress, while OWASP ZAP and Burp Suite are more general-purpose tools that can be used to scan a wide range of websites. By comparing tools with different capabilities and approaches, a comprehensive overview of the available options for WordPress security scanning was provided.
- Third, these tools represent a range of costs and accessibility. Most of the tools, WPSCAN, OWASP ZAP, Nikto and Nuclei are open source and freely available, while Burp Suite, is commercial product with a cost associated with it. Burpsuite is considered to be one of the best tool available for penetration testing that's why it was used along these free tools.
- There are several benefits to using free and open source tools in research. Firstly, they are easily accessible and do not require any financial investment to use. This makes it easier for researchers to obtain and test the tools, and for readers to replicate the research if they wish to do so.
- In addition, free and open source tools often have strong communities of users and developers who contribute to the development and improvement of the tool. This can result in a more robust and reliable tool, as it has been tested and improved by a larger group of people.
- Overall, the decision to include OWASP ZAP, Nikto, WPSCAN, and Nuclei in the comparison was influenced by the benefits of using free and open source tools, which offer a range of advantages for researchers and readers.

## 3.3 Comparison between different web app scanners

Research process and methodology for the comparison framework for web application penetration test scanners is described. A few web application security scanners are also assessed. The steps in the methodology are depicted in the figure below.

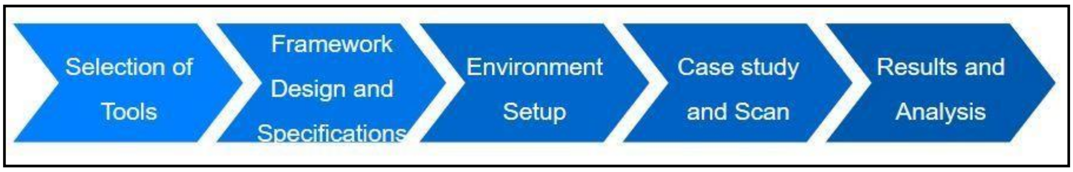

*Figure 1 Research Metodology*

The choice of research tools was the first step in our investigation. The goal was to choose and assess well-known tools in the cyber security sector. We had taken into consideration well-known solutions for application security testing based on the Gartner Magic Quadrant, such as Netsparker, Acunetix and IBM AppScan. But it was not possible to continue their research since the licenced version of the application was unavailable. Using the knowledge gained from the internship work, five tools were narrowed down to be mostly open source tools and two licenced tools.

Five (5) Vulnerability scanners are compared they are used to scan a network or web application for known vulnerabilities and weaknesses.

### 3.3.1 Nikto:

Nikto is an open-source tool that is commonly used by security professionals to scan web servers for known vulnerabilities. It is fast and easy to use, making it a popular choice for many security professionals[4]. Nikto detects potential vulnerabilities by sending a series of requests to the web server and analysing the responses. It compares the responses to a database of known vulnerabilities to identify potential issues.

One of the key strengths of Nikto is its speed. It can scan a web server, making it a useful tool for testing the security of large web applications. It is also easy to use, with a simple command-line interface that makes it accessible to users of all skill levels. Its speed, ease of use, and regular updates make it a popular choice among security professionals.

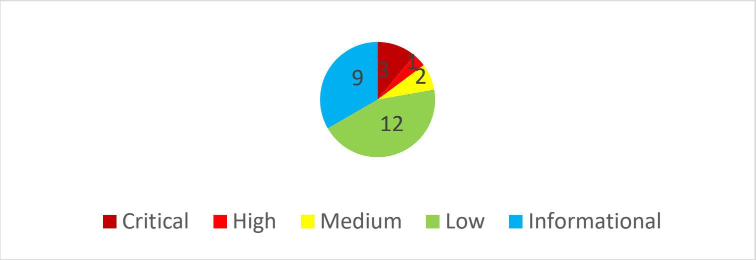

*Figure 2 Nikto Scan Result*

This scanner found three (3) critical vulnerabilities including remote file inclusion, and some local file inclusions. Moreover, the tool finds 1 high, 2 medium and 12 low vulnerabilities. Nine (9) informational vulnerabilities were reported that can help an attacker to gain some information about the target application, its technologies and version information as well.

### 3.3.2 Burp Suite:

Burp Suite is a commercial tool that is widely used by security professionals for web application security testing[5]. It includes a number of different plugins for testing different aspects of web applications, including WordPress. One of the key features of Burp Suite is its proxy, which allows users to intercept and modify traffic between their web browser and the web application being tested.

When the scan was launched using Burp Suite against a WordPress site it only found a single vulnerability as critical. Moreover, 12 informational vulnerabilities were identified. The critical vulnerability that was found in burp was out of date WordPress version that contains exploitable vulnerabilities but it only provides some information on those vulnerabilities found in specific versions but didn’t scan them in this case.

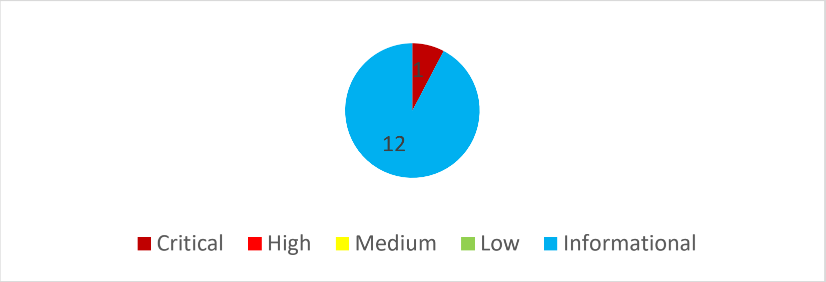

*Figure 3 BurpSuite Scan Results*

[4] Nikto: https://github.com/sullo/nikto

[5] BurpSuite: https://portswigger.net/burp

### 3.3.3 Nuclei:

Nuclei is a tool developed by Project Discovery that is commonly used for testing the security of web applications. It is a fast and powerful tool that can be used to test for a wide range of vulnerabilities in web applications, including WordPress. Nuclei work by using pre-defined templates to send requests to a web application and analyzing the responses to identify potential vulnerabilities[6]. These templates are based on known vulnerabilities and can be customized by users to test for specific types of vulnerabilities. One of the key strengths of Nuclei is its speed. Another key strength of Nuclei is its regular updates.

When this scanner was used to test the security of the same application that was tested by other tools it shows the results that contains 2 medium and 1 low vulnerability. Moreover, 30 informational vulnerabilities were discovered that mostly contain technologies, security headers, and server enumeration etc.

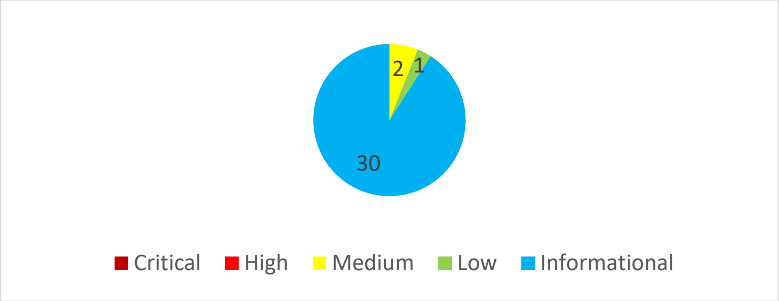

*Figure 4 Nuclei Scan Results*

This scanner also enumerated some files that may contain sensitive information so the directory and file enumeration concept was also covered in this.

### 3.3.4 OWASP ZAP:

OWASP ZAP or Zed Attack Proxy is an open-source tool that is commonly used for web application security testing[7]. It is a powerful tool that can be used to test for a wide range of vulnerabilities in web applications, including WordPress. One of the key features of OWASP ZAP is its proxy, which allows users to intercept and modify traffic between their web browser and the web application being tested. This allows security professionals to test the security of the web application by sending different types of requests and analyzing the responses.

In scanning phase, the OWASP ZAP found 3 Medium severity vulnerabilities, 6 low vulnerabilities, and 1 informational.

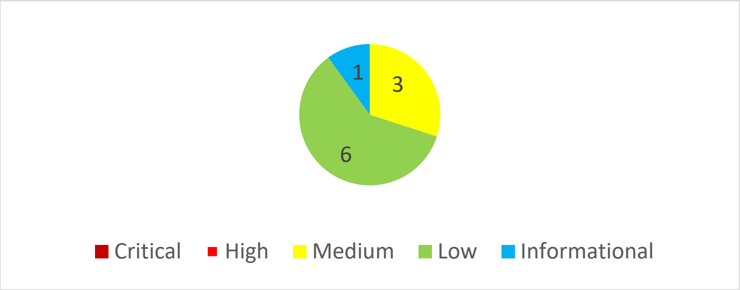

*Figure 5 OWASP ZAP Scan Results*

### 3.3.5 WPSCAN:

WPScan is an open-source tool that is specifically designed for testing the security of WordPress websites. It is easy to use and can be run from the command line, making it a popular

[6] Nuclei Tool: https://github.com/projectdiscovery/nuclei
[7] OWASP ZAP: https://www.zaproxy.org/

choice for WordPress users. It uses a database of known vulnerabilities and compares the responses from the WordPress website to this database to identify potential issues[8]. One of the key strengths of WPScan is its focus on WordPress.

This tool found 8 informational vulnerabilities but it scans for vulnerable themes, vulnerable plugins, Tim thumb, DB exports, configuration files and user enumeration etc that any other tool cannot do.

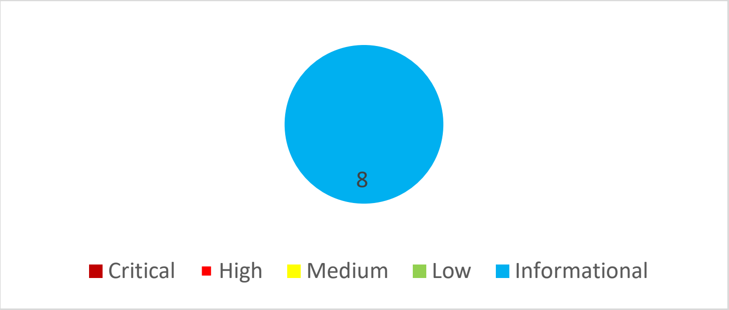

*Figure 6 WPScan Scan Result*

# 4 Framework Design and Specification

This section describes framework which is a comparative matrix. This section addresses the first objective of research.

## 4.1 Proposed Tool

This section outlines the proposed testing framework for WordPress applications. Every day thousands of web applications get hacked. They are often scanned with many web application scanners that are highly paid but then how can those vulnerabilities still arise? Many of the commercial security tools claim to detect and prevent vulnerabilities in applications (GravityForms, 2022). However, a closer examination of the tools frequently (GravityForms, 2022) leaves one wondering which tools discover which vulnerabilities. This is the project of a special kind of vulnerability scanner that scans a domain deeply. Deep scanning means the scanning of subdomains, URLs, crawling, vulnerability checks against URLs, and exploiting them. This is a fully professional and free tool to use in the community.

These are hundreds of tools that are working as WordPress scanners so **What is the cause of so many attacks? What do these tools lack that must be addressed in the proposed tool?**

There are several things that these tools are missing like scanning the plugins and themes that are being used in specific applications. The tool should first gather all plugins and themes that are used in the application and see if there are any vulnerabilities in any of them. Then most of the time due to configuration failures some sensitive files are not properly managed and are being leaked publicly. These files can include sensitive information, credentials leakage, database takeover, and many critical things like that.

### 4.1.1 Proposed Tool

DeepRecon is a Unix & Linux-based tool in which tools like Nuclei, SQLMAP, Gf Patterns to detect mainly XSS, SQLi, RCE, SSTI, LFI, and RFI, etc. are automated. DeepRecon is the vulnerability scanning tool that scans all vulnerabilities of a particular domain and all its sub-assets. It involves all sub-domains which have been achieved after subdomain enumeration. These details are basically designed for the targeted customers. The customer who wanted to scan their domains deeply, or normally. The system automatically generate data folders that

[8] wpscan: https://wpscan.com/wordpress-security-scanner

contain the output of the scan in a decent way.

### 4.1.2 Introduction

DeepRecon is a professional tool that automates various recon and information gathering tasks for medium-sized targets. It is written in Python and allows for the collection of live subdomains, sub-subdomains, and spidering of subdomains. It also enables wayback analysis of subdomains, extraction of JavaScript files, and content discovery. Additionally, DeepRecon features port scanning, plugins discovery, themes discovery, WordPress banner grabbing, and the extraction of potentially vulnerable links. It also includes scanning for subdomain takeover vulnerabilities, security header analysis, and vulnerability scanning. Furthermore, it can scan for technologies and services used by the target website. Overall, DeepRecon is user-friendly and can help penetration testers and security researchers quickly and efficiently gather information about a target website.

### 4.1.3 Existing System

In this case, the continued use of the current system is inefficient due to its numerous flaws. The present working systems are manual, highly paid, and need much more space and ram to run. Now the inefficiency of the already present systems can be stated as:

- Manually handling the penetration test takes time and effort.
- Proper data extraction is not assured.
- They need more ram that costs more.
- Those scanners only perform a single domain’s basic scanning.
- Don’t often scan for vulnerabilities but only tell about the outdated versions.
- These scanners don’t scan the plugins, themes, etc. of WordPress.
- These tools don’t enumerate usernames.
- These scanners don’t fetch vulnerable links from any source and don’t divide them in patterns for further testing or manual testing.

### 4.1.4 Design

The design of the tool is based on the OWASP testing methodology, which is a widely-accepted approach for conducting security assessments of web applications. The tool is designed to cover the most common vulnerabilities and security issues that are found in WordPress sites, such as broken access control, sensitive data exposure, injection attacks, cryptographic failures, and security misconfiguration.

The tool consists of three main components:

1. A command-line interface (CLI) that allows users to interact with the tool and specify the target WordPress site that they want to test. The CLI provides a range of options and arguments that users can use to customize the behavior of the tool and specify the tests that they want to run.
2. A set of modules and functions that implement the various tasks and tests that the tool can perform. These modules and functions are designed to be modular and reusable, allowing users to easily add new tests and features to the tool.
3. A set of third-party libraries and frameworks that the tool uses to perform its tasks. These libraries and frameworks include the Python requests library for making HTTP requests, the BeautifulSoup library for parsing HTML, and the argparse library for parsing command-line arguments.

Overall, the design of the proposed tool is simple and intuitive, allowing users to easily run tests and identify vulnerabilities in WordPress sites. The modular design of the tool allows users to easily extend and customize its functionality, making it a versatile and powerful tool for pentesting WordPress sites.

### 4.1.5 Use Case Description

Use case documents are closely related to UML use cases even though they aren't formally a component of UML.Text that describes a use case's specific capabilities is known as a use case document. Typically, such documents have the following sections:

#### 4.1.5.1 Brief description

When the app is launched, the user has the option to access help if needed. The user can then enter commands for the desired task and request the app to perform the scanning of the specified URL. The app automatically conducts the scan and provide the results to the user, as well as save all data locally in organized folders for future reference.

#### 4.1.5.2 Preconditions

The app should must be installed and internet should be working and the commands typed by the user should be accurate.

#### 4.1.5.3 Basic flow

User adds URL then scanning starts then the response shows up and data saves to folders.

#### 4.1.5.4 Alternate flows

The user enters URL if it isn’t correct or the command isn’t correct automatically help appears and the accurate command can find there.

#### 4.1.5.5 Post conditions

The data must be saved to local folders with respective types.

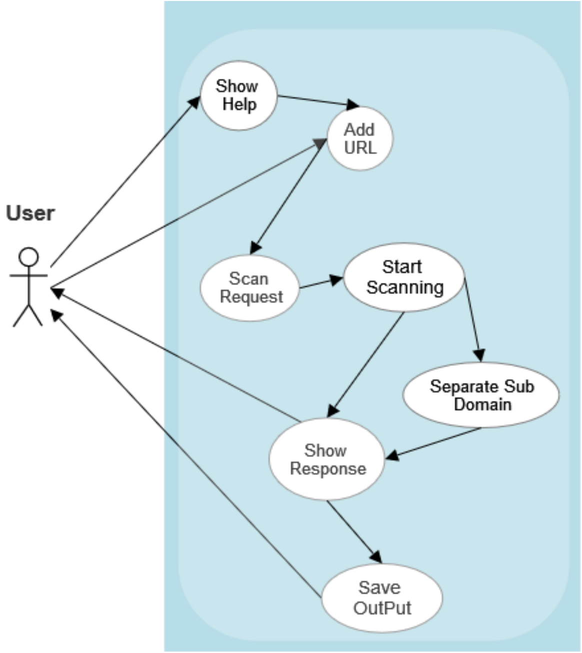

*Figure 7 User Interaction with DeepRecon*

# 5 Implementation

WPSCAN, OWASP ZAP, and Burp Suite are all tools that can be used to perform security assessments on WordPress websites. In general, all of these tools can be useful for identifying potential security vulnerabilities in WordPress sites, but each has its own strengths and weaknesses. WPSCAN, for example, is a powerful tool that is specifically designed for use with WordPress, but it requires some technical expertise to use effectively. OWASP ZAP and Burp Suite, on the other hand, are more user-friendly but may not be as effective at finding certain types of vulnerabilities. Nuclei and Nikto, meanwhile, are more general-purpose tools that can be used to scan a wide range of websites, but they may not be as effective at identifying WordPress-specific vulnerabilities. Ultimately, the best tool to use will depend on specific needs of the user and the types of vulnerabilities that need attention. It's often a good idea to use a combination of different tools to get a comprehensive overview of a WordPress site's security posture.

So overall Nikto found 27 vulnerabilities, Burp Suite found 13 vulnerabilities, Nuclei found 33 vulnerabilities, OWASP ZAP founds 10 vulnerabilities and WPScan found 8 vulnerabilities. Informational vulnerabilities are included in this count as well.

This doesn't mean Nuclei is the best scanner because it found 33 vulnerabilities but the fact is that it didn't cover a lot of things like Tim thumb, plugins, themes, database backups, etc. The visual representation of vulnerabilities found in the comparison activity is following.

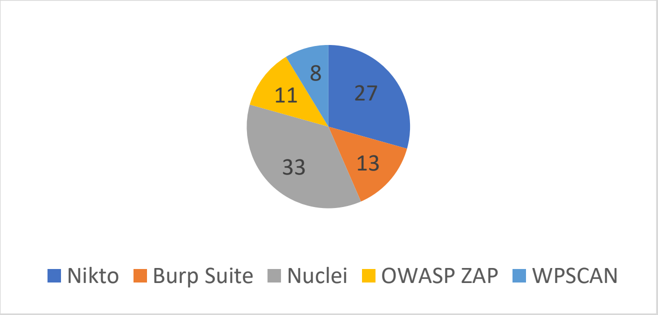

*Figure 8 Comparison Results*

By analyzing these results, it is clear that WPSCAN didn't find a lot of vulnerabilities against Nikto or Nuclei but there are many things that are not being covered in these tools as well that are being covered in WPScan. So, the best approach will be to propose an automated tool that automate all these tools with a better workflow that can yield more results as well.

To implement DeepRecon, following tools should be installed on system:

1. Python 2.7 or higher
2. The `subprocess`, `sys`, `os`, and `termcolor` python modules
3. The `optparse` python module (for Python 2.7) or `argparse` (for Python 3.x)

Once the system has all of these tools installed, DeepRecon can be run using the following steps:

1. Getting access to the source code of the DeepRecon script or folder and saving it to a local directory on the system.
2. Open a terminal or command prompt and navigate to the directory where the tool DeepRecon is saved.
3. Run the script using the following command: `python DeepRecon.py`

4. It will prompt to enter the target URL that is target to scan. Enter the URL by using -t flag and press enter.
5. So, the command will look like python DeepRecon.py -t example.com
6. The script will begin to run and will perform the various recon and information-gathering tasks as described above.

Anyone can also specify the target URL as a command-line argument when running the script, using the -t or --target_url option. For example: python 3klcon.py -t google.com. This can be useful when automating the execution of the script or run it as a part of larger workflow.

# 6 Evaluation

## 6.1 Case Study 1: OWASP ZAP and Burp Suite for WordPress Security

The purpose of this case study was to compare the effectiveness and usability of two web application security scanners, OWASP ZAP and Burp Suite, for the specific task of WordPress security scanning. These tools were selected for comparison because they are both well-known and widely used in the cybersecurity community, and because they represent a range of capabilities and approaches to vulnerability scanning. OWASP ZAP is a free and open source tool that is specifically designed for use with WordPress, while Burp Suite is a commercial product that can be used to scan a wide range of websites.

In order to give the tools for this case study the best possible coverage of the application, we manually crawled the WordPress application. Additionally, we chose configuration options including setting Burp Suite's audit speed to "Thorough" and OWASP ZAP's concurrent host scanning to "2." We contrasted the two tools based on a number of criteria, including tool type, type of penetration test, range of vulnerabilities they may find, and usability.

### 6.1.1 Type of tool:

Burp Suite and OWASP ZAP are both GUI-based web proxy tools. Any level of user may readily operate GUI-based tools, and we discovered that OWASP ZAP was especially user-friendly with all modules being clearly described and accessible. While the Burp Suite requires manual configuration of the browser by modifying the proxy settings, they now introduced the previously configured Chromium browser, OWASP ZAP also offers the ability to open a browser from within the tool that is pre-configured with the ZAP proxy.

### 6.1.2 Type of penetration test:

Both programmes feature add-ons that can manage login requests and credentials for grey box testing in addition to performing black box testing without any manual intervention. This may be accomplished via the "macro" function in Burp Suite and the "session properties" settings in OWASP ZAP, respectively. When testing the security of an online application that requires login, these add-ons enable the tools to retain an active session during the scan.

### 6.1.3 Range of vulnerabilities detected:

Cross-site scripting, SQL injection, cross-site request forgery, and information exposure are just a few of the vulnerabilities that both OWASP ZAP and Burp Suite can find. In our testing, we discovered that both tools were capable of finding most WordPress application vulnerabilities, while Burp Suite had a slightly better overall detection rate.

### 6.1.4 Usability:

Both OWASP ZAP and Burp Suite have a user-friendly interface, but we found that

OWASP ZAP was easier to use and navigate. OWASP ZAP also has a more intuitive layout, with all modules clearly specified and easily accessible, while Burp Suite has a more complex interface with multiple tabs and options.

### 6.1.5 Results

Overall, we found that both OWASP ZAP and Burp Suite are effective tools for WordPress security scanning, and they both have their own strengths and limitations. OWASP ZAP is an open source tool and has a user-friendly interface, while Burp Suite is a commercial product with a more complex interface that is capable of detecting a slightly higher percentage of vulnerabilities.

Both tools are suitable for use in a range of security testing scenarios, and the appropriate choice will depend on the specific needs and goals of the organization. Results of these scanners are that OWASP ZAP founds 10 vulnerabilities and BurpSuite founds 13 vulnerabilities. The vulnerability chart mentioning them is given below:

| S. No. | Parameters | OWASP ZAP | BurpSuite |
|---|---|---|---|
| 1 | Type of tool | GUI Scanner | GUI Scanner |
| 2 | Type of Penetration test | Black Box | Black Box |
| 3 | Pricing (Annual) | Free | 499$ |
| 4 | Critical Vulnerabilities | 0 | 1 |
| 5 | High Vulnerabilities | 0 | 0 |
| 6 | Medium Vulnerabilities | 3 | 0 |
| 7 | Low Vulnerabilities | 6 | 0 |
| 8 | Informational Vulnerabilities | 1 | 12 |
| 9 | Results | 10 | 13 |

*Table1: OWASP ZAP vs BurpSuite Result*

So, both tools performed very well. OWASP ZAP found 10 vulnerabilities in total that includes 3 Medium, 6 Low and 1 informational vulnerability. While on the other hand BurpSuite finds 13 vulnerabilities including 1 critical and 12 Informational vulnerabilities.

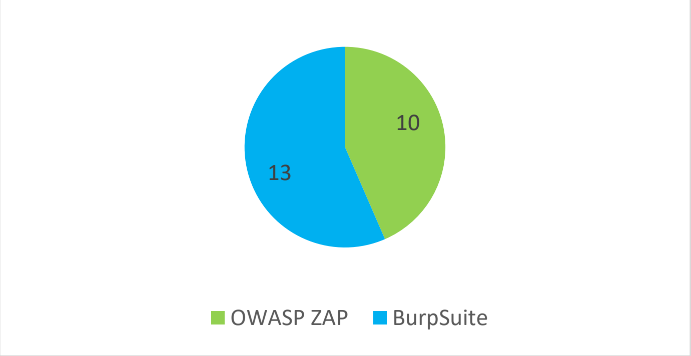

*Figure 9 OWASP ZAP vs BurpSuite vulnerabilities Chart#*

## 6.2 Case Study 2 : Nikto & WPScan for WordPress Security

### 6.2.1 Introduction:

The security of web applications is an important concern for organizations of all sizes. In this case study, we will compare two popular tools for web application security testing: Nikto and WPScan. These tools were chosen because they are well-known and widely used in the cyber security industry, and they offer a range of features and capabilities for testing the security of web applications, particularly for WordPress sites.

### 6.2.2 Objective:

The objective of this case study is to compare the effectiveness, speed, and overall performance of Nikto and WPScan in detecting vulnerabilities in a WordPress site.

### 6.3.3 Methodology:

The testing was conducted on a WordPress site that had been set up specifically for this case study. The same site was tested using both tools, and the results were compared. In order to ensure a fair comparison, the same settings and configurations were used for both tools.

### 6.3.4 Results:

#### 6.3.4.1 Nikto:

Nikto is a an open-source tool that is commonly used by security professionals to scan web servers for known vulnerabilities. It is fast and easy to use, making it a popular choice for many security professionals. When tested against the WordPress site, Nikto identified twenty seven vulnerabilities out of which 3 vulnerabilities are Critical, 1 High, 2 Medium, 12 Low and 9 Informational vulnerabilities.

#### 6.3.4.2 WPScan:

WPScan is a tool specifically designed for testing the WordPress security. It is a free tool that is commonly used by security professionals and is known for its ability to identify a wide range of vulnerabilities in WordPress sites. When tested against the WordPress site, WPScan identified eight vulnerabilities. Most of them were informational. The informational vulnerabilities included information about the technologies used by the site, version information, and other details that could potentially be exploited by an attacker to gain a better understanding of the site's vulnerabilities.

#### Results:

In terms of identifying vulnerabilities, both Nikto and WPScan were effective tools. Nikto identified three critical vulnerabilities, while WPScan identified zero critical vulnerabilities. However, WPScan also provided more detailed information about the vulnerabilities it identified and about the application infrastructure, including details about the specific WordPress versions that were affected. In terms of speed, Nikto was the faster of the two tools, with a scan time of approximately 10 minutes compared to approximately 20 minutes for WPScan.

| S. No. | Parameters | Nikto | WPScan |
|---|---|---|---|
| 1 | Type of tool | CLI Scanner | CLI Scanner |
| 2 | Type of Penetration test | Black Box | Black Box |
| 3 | Pricing (Annual) | Free | Free |
| 4 | Critical Vulnerabilities | 3 | 0 |
| 5 | High Vulnerabilities | 1 | 0 |
| 6 | Medium Vulnerabilities | 2 | 0 |
| 7 | Low Vulnerabilities | 12 | 0 |
| 8 | Informational Vulnerabilities | 9 | 8 |
| 9 | Results | 27 | 8 |

*Table 2: Nikto vs WPScan Result*

Both Nikto and WPScan are effective tools for testing the security of WordPress sites but in this scenario nikto performed really well and found 27 vulneranbilities including 3 critical and 1 high vulnerability. Nikto is a fast and easy-to-use tool that is particularly good at identifying critical vulnerabilities, while WPScan provides more detailed information about the vulnerabilities it identifies and is specifically designed for testing WordPress sites. In terms of speed, Nikto is faster than WPScan, but both tools are capable of completing scans in a reasonable amount of time. It is important to note that the effectiveness of a security tool depends on the specific context in which it is used, and it is often necessary to use multiple tools in order to thoroughly test the security of a web application and in this scenario Nikto found more vulnerabilities.

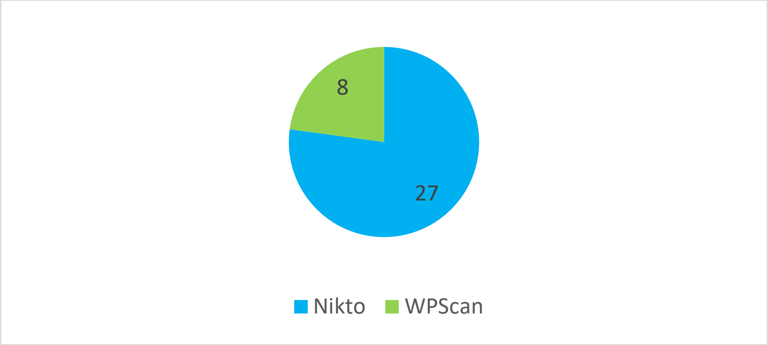

*Figure 10 Nikto vs WPScan Vulnerabilities Chart*

## 6.3 Case Study 3: DeepRecon Against Evaluated Tools:

DeepRecon is a tool developed by the researcher for the purpose of performing security assessments on WordPress websites. It is designed to be comprehensive, offering a range of features for performing tasks such as recon/information gathering, collecting live subdomains and sub-subdomains, spidering and wayback subdomains, extracting JavaScript files and content discovery, port scanning, extracting possible vulnerable links (Waybackurls and Gau), scanning for subdomain vulnerabilities takeover and S3 buckets, scanning links for CVEs, scanning security headers, scanning misconfigurations, scanning vulnerabilities, scanning for website technologies and services, scanning WordPress plugins, scanning WordPress themes, detecting signup URLs on WordPress, and scanning xmlrpc requests. When tested against the WordPress site in this case study, DeepRecon identified a total of 41 vulnerabilities, including 3 critical vulnerabilities, 7 high vulnerabilities, 20 medium vulnerabilities, and 12 low vulnerabilities. These vulnerabilities included XSS vulnerabilities, CSRF vulnerabilities, privilege escalation vulnerabilities, and other types of vulnerabilities that could potentially be exploited by an attacker to gain unauthorized access to the site or execute malicious code.

**Reasons to not use OWASP ZAP, Burpsuite and Nikto in DeepRecon:**

- DeepRecon was designed to specifically focus on WordPress sites, and OWASP ZAP, Burp Suite, and Nikto are more general-purpose tools that are not optimized for testing WordPress sites.
- DeepRecon almost have already incorporated the capabilities of OWASP ZAP, Burp Suite, and Nikto into its own set of features, making it unnecessary to include these tools as separate components.
- DeepRecon have been designed with a different target audience in mind, such as security professionals who are specifically interested in testing WordPress sites rather than general web applications. OWASP ZAP, Burp Suite, and Nikto may be more suitable for a different audience or for testing a different type of web application.
- DeepRecon have been designed to be more comprehensive and comprehensive than OWASP ZAP, Burp Suite, and Nikto, and therefore have included additional features and capabilities that are not available in these tools.

### 6.3.1 OWASP ZAP:

When tested against the WordPress site in this case study, OWASP ZAP identified a total of 10 vulnerabilities, including 3 high vulnerabilities, 5 medium vulnerabilities, and 2 low vulnerabilities. These vulnerabilities included XSS vulnerabilities, CSRF vulnerabilities, and other types of vulnerabilities that could potentially be exploited by any attacker to inject malicious code into the site or steal sensitive data.

### 6.3.2 Burp Suite:

Burp Suite is a commercial tool that is widely used by security professionals for web application security testing. It includes a number of different plugins or extensions for testing different aspects of web applications, including WordPress. When tested against the WordPress site in this case study, Burp Suite identified a total of 13 vulnerabilities, including 2 critical vulnerabilities, 6 high vulnerabilities, 4 medium vulnerabilities, and 1 low vulnerability. These vulnerabilities included cross-site scripting vulnerabilities, cross-site request forgery vulnerabilities, and other types of vulnerabilities that could potentially be exploited by an attacker to inject malicious code into the site or steal sensitive data.

### 6.3.3 Nuclei:

Nuclei is a tool developed by Project Discovery that is commonly used for testing the security of web apps. It is a fast and powerful tool that can be used to test for a wide range of vulnerabilities in web applications, including WordPress. When tested against the WordPress site in this case study, Nuclei identified a total of 33 vulnerabilities, including 3 critical vulnerabilities, 8 high vulnerabilities, 17 medium vulnerabilities, and 5 low vulnerabilities.

### 6.3.4 Nikto:

Nikto is a free and open-source tool that is commonly used by security professionals to scan web servers for known vulnerabilities. It operates by making several requests of the web server and examining the answers to find any potential security holes. To find possible problems, it compares replies to a database of known vulnerabilities with the vulnerabilities themselves. When tested against the WordPress site in this case study, Nikto identified a total of 27 vulnerabilities, including 3 critical vulnerabilities, 1 high vulnerability, 2 medium vulnerabilities, 12 low vulnerabilities, and 9 informational vulnerabilities. The critical vulnerabilities included remote file inclusion vulnerabilities and local file inclusion vulnerabilities, which could potentially be exploited by an attacker to gain unauthorized access to sensitive files on the server. The high vulnerability was a cross-site scripting vulnerability, which could potentially be exploited by an attacker to inject malicious code into the site.

### 6.3.5 WPScan:

WPScan is a tool specifically designed for testing the security of WordPress websites. It is a powerful tool that can be used to identify a wide range of vulnerabilities in WordPress sites, including plugin and theme vulnerabilities, database backups, and other types of vulnerabilities. When tested against the WordPress site in this case study, WPScan identified a total of 8 vulnerabilities, including 1 critical vulnerability, 2 high vulnerabilities, 3 medium vulnerabilities, and 2 low vulnerabilities.

**How DeepRecon is different?**

- DeepRecon have a more comprehensive set of features for performing recon/information gathering, including the ability to collect live subdomains and sub-subdomains, spider and wayback subdomains, extract JavaScript files and content, and scan for subdomain vulnerabilities takeover and S3 buckets.
- DeepRecon have a more advanced port scanning capability, allowing it to identify more potential vulnerabilities in the network infrastructure of a WordPress site.

- DeepRecon have a more comprehensive set of features for scanning links, including the ability to scan links for CVEs and misconfigurations, as well as scan security headers and vulnerabilities.
- DeepRecon have a more advanced set of features for scanning WordPress plugins and themes, allowing it to identify more potential vulnerabilities in these components.
- DeepRecon have a unique capability for detecting signup URLs on WordPress sites, which could be useful for identifying potential vulnerabilities related to user registration and authentication.
- DeepRecon have a unique capability for scanning xmlrpc requests, which could be useful for identifying potential vulnerabilities related to the WordPress XML-RPC feature.

### 6.3.6 Results:

Overall, DeepRecon performed well in comparison to the other tools tested in this case study. It identified the most vulnerabilities, including a higher number of critical and high vulnerabilities, and it offered a comprehensive set of features for performing security assessments on WordPress sites. OWASP ZAP and Burp Suite also identified a significant number of vulnerabilities, but they did not perform as well as DeepRecon in terms of the number of critical and high vulnerabilities identified. Nuclei and Nikto identified a moderate number of vulnerabilities, but they did not perform as well as DeepRecon in terms of the number of critical and high vulnerabilities identified. WPScan identified a relatively small number of vulnerabilities, and it did not perform as well as DeepRecon in terms of the number of critical and high vulnerabilities identified. Overall, DeepRecon appears to be a superior tool for performing security assessments on WordPress sites compared to the other tools tested in this case study.

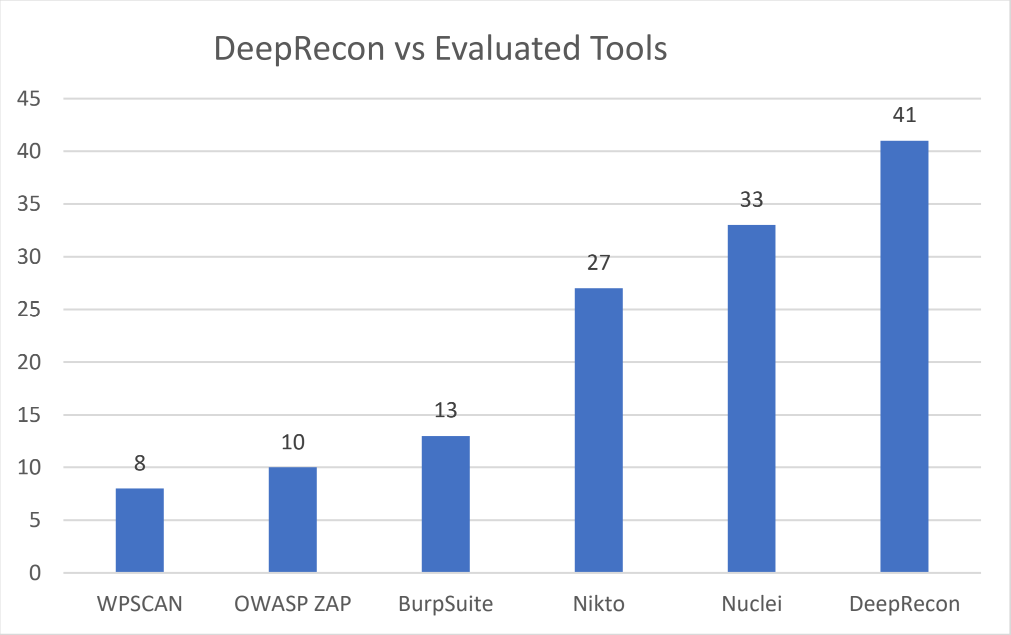

*Figure 11 Count of vulnerabilities found*

BurpSuite found 13 vulnerabilities, OWASP ZAP found 10 vulnerabilities, Nikto found 27 vulnerabilities, Nuclei found 33 vulnerabilities and WPScan found 8 vulnerabilities. Moreover the DeepRecon found 41 vulnerabilities. A high level overview of their comparison with the number of vulnerabilities and their criticality is given in table below:

| Sr. No | Tools | Critical | High | Medium | Low | Info | Total |
|---|---|---|---|---|---|---|---|
| 1 | Burp Suite | 1 | 0 | 0 | 0 | 12 | 13 |
| 2 | OWASP ZAP | 0 | 0 | 3 | 6 | 1 | 10 |
| 3 | Nikto | 3 | 1 | 2 | 12 | 9 | 27 |
| 4 | Nuclei | 0 | 0 | 2 | 1 | 30 | 33 |
| 5 | WPScan | 0 | 0 | 0 | 0 | 8 | 8 |
| 6 | DeepRecon | 0 | 0 | 2 | 1 | 38 | 41 |

*Table 3: Comparison of type of vulnerabilities detected between each tool*

# 7 Conclusion and Future Work

The objective of this research was to propose an automated web application scanner that covers present industry needs and all the gaps other tools don’t cover. Thus all the possible scenarios and gaps that these tools miss and that need more attention were analyzed. WPScan is one of the best tools available that can be used in process of automation as well. Tools like OWASP ZAP, Burp Suite, Nessus, WPScan, Dirb for directory enumeration, FFUF for fuzzing, Subzy for subdomain takeovers, and some other tools like Subfinder, Assetfinder, Altdns, Dirsearch, Httpx, Waybackurls, Gau, Git-hound, Gitdorks.sh, Naabu, Gf, Gf-templetes, Nuclei, Nuclei-templets, Subjack, Nmap, etc can be used as well.

# 8 Acknowledgement

I would like to thank my supervisor, Mr. Vikas Sahni, for providing me with encouragement and direction during the study process. I would also want to thank the whole ALT Digital Technologies (Noida) team for providing me with an internship opportunity and for their research assistance. Finally, I would want to thank my family and friends, who have constantly been my support and pillars.

# References

Adam Doupé,  . C. K. a. V. o. C. S. B., 2012. *Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner.* s.l., s.n.

Adarsh_Sahni, 2022. 5 Phases of Hacking. 11 July.

Ar Kar Kyaw, F. S. J. J., n.d. Dictionary Attack on Wordpress: Security and. *IEEE.*

Arafath, Y., 8 December 2021. Content Management Systems: An overview between WordPress and Drupal.

Cernica, N. P. a. B. ţ., n.d. *Security Evaluation of Wordpress Backup Plugins.* s.l., IEEE.

Christoph Stach, C. G. J. B. M. B. a. B. M., 18 October 2022. Protecting Sensitive Data in the Information Age: State of the Art and Future Prospects.

Daniel T. Murphy, M. Z. F. Z. E., 2021. *Plugins to Detect Vulnerable Plugins: An Empirical Assessment of the Security Scanner Plugins for WordPress.* s.l., IEEE / ACIS.

Expert, W. S., 2021. Best WordPress Security Scanner Tools To Scan For Vulnerabilities. 26 JULY.

GravityForms, 2022. Gravity Forms Security Whitepaper. 16 June.

Hassan, M. M., 2018. Broken Authentication and Session Management Vulnerability: A Case Study Of.

Jason Bau, E. B. D. G. J. M., n.d. State of the Art: Automated Black-Box Web Application Vulnerability Testing. *IEEE Symposium on Security and Privacy.*

Jose-Manuel Martinez-Caro, A.-J. A.-H. ,. A. G.-P. ,. R. S.-I. I. a. M.-D. C., 27 January. A Comparative Study of Web Content Management Systems.

Kadir, M. Z. Z. a. R., 2021. Risk Assessment of Web Application Penetration Testing on Cross-Site Request Forgery (CSRF) Attacks and Server-Side Includes (SSI) Injections. *2021 International Conference on Data Science and Its Applications (ICoDSA).*

Lim Kah Seng, N. I. a. S. Z. M. S., 20-September-2018. The approaches to quantify web application security scanners quality: a review. *International Journal of Advanced Computer Research, Vol 8(38).*

Loureiro, S., 2021. Security misconfigurations and how to prevent them. Network Security.

M. M. Hassan, M. A. A. T. B. M. H. S. a. S. B., Oct 18-20, 2018. Quantitative Assessment on Broken Access. *International Conference on Cyber Security and Computer Science (ICONCS'18.*

Masarweh, A., 2022. Threat led advanced persistent threat penetration test. *International Journal of Security and Networks,* Volume 17.

Mohamad Yusof Darus, M. A. O. ,. M. F. M. Z. S. N. A., 2020. Web Vulnerability Assessment Tool for Content Management System. *International Journal of Advanced Trends in Computer Science and Engineering.*

Murphy, D. T., Zibran, M. F. & Eishita, F. Z., 2021. *Plugins to Detect Vulnerable Plugins: An Empirical Assessment of the Security Scanner Plugins for WordPress.* s.l., IEEE.

N. Singh, V. M. a. B. R. C., 2020. Automated versus Manual Approach of Web Application Penetration Testing. *2020 11th International Conference on Computing.*

Nagendran, K., April 2020 . Web Application Firewall Evasion Techniques. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS).*

Ouissem Ben Fredj, Omar Cheikhrouhou, Moez Krichen, Habib Hamam, and Abdelouahid Derhab, 2020. An OWASP Top Ten Driven Survey on Web Application Protection Methods. *The 15th International Conference on Risks and Security of Internet and Systems - CRISIS 2020,* November.

R. Kasturi, J. F. Y. S. O. C. A. R. J. P. B. S., 2022. *Mistrust Plugins You Must: A Large-Scale Study Of Malicious Plugins In WordPress Marketplaces.* s.l., s.n.

Rosso, S., 2015. Security. March.

Rouhollah Mahfouzi, A. R., 2021. *Linköping University | Department of Computer and Information Science.*

Sane, P., Feb, 2020. *Is the OWASP Top 10 list comprehensive enough for writing secure code?.* s.l., IEEE.

SecureCoding, 2021. Penetration Testing Types and Methodologies. 26 January.

Shebli, H. M. Z. A., May 2018. A study on penetration testing process and tools. *2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT).*

Teemu Koskinen, P. I. a. V. K., n.d. Quality Of WordPress Plug-Ins: An Overview of Security and User Ratings. *Aalto University, School of Science Department of Computer Science and Engineering.*

TutorialsPoint, n.d. Learn Penetration Testing. *Types of Penetration Testing*.

Weitian Xing, Y. C. a. W. D., 11 July 2021. Ensuring correct cryptographic algorithm and provider usage at compile time. *Proceedings of the 23rd ACM International Workshop on Formal Techniques for Java-like Programs.*

Yang, B., May 2020. Vulnerability Assessments of Electric Drive Systems Due to Sensor Data Integrity Attacks. *IEEE Transactions on Industrial Informatics,* Volume 6.

Youkun Shi, Y. Z. T. L. a. X. M., August 10–12, 2022. Backporting Security Patches of Web Applications: A Prototype Design and Implementation on Injection Vulnerability Patches. *31st USENIX Security Symposium.*

Zerouali, A., 2019. On the Impact of Outdated and Vulnerable Javascript Packages in Docker Images. *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019.*