

WEB VULNERABILITY SCANNER WITH MULTI-TOOL INTEGRATION

A PROJECT REPORT

Submitted by

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

(Specialization in Cyber Security and Digital Forensics)

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING VIT
AMARAVATHI UNIVERSITY**

LIST OF ABBREVIATIONS

1	NMAP	NETWORK MAPPER
2	DNS RECON	Domain Name System Reconnaissance
3	XSS	Cross-Site Scripting
4	HTML5	Hypertext Markup Language 5
5	SQL	Structured Query Language
6	SSH PROTOCOL	Secure Shell Protocol
7	SQLI	Structured Query Language
8	IPV6	Internal Protocol Version 6
9	TCP	Transmission Control Protocol

ABSTRACT

Web vulnerabilities can readily be exploited by malicious users to steal data, compromise user identities, get access to confidential files or information, spam the site, inject scripts, or even seize control of the server. Web applications are constantly developed and launched to help cater to our growing needs as we continue to use the internet. Some companies may not have the knowledge or resources to follow proper SDLC (Software Development Life Cycle) best practices, which means that lapses in security can harm the stability of the web application when they are launched. This framework allows you to perform automated vulnerability scans for Windows, iOS, and Android devices. You can use this tool if you are performing penetration testing and various types of analysis on your applications. ect on the safety of some web applications. If an application or web service is compromised then that could spell disaster for the company that created it. Scenarios like this make it necessary for organizations to have web application security testing and assessment tools available to them.

Kali offers a range of different vulnerability assessment tools that will help you to identify potential risks and vulnerabilities before they become a problem. But still, It is quite a fuss for a pentester to perform binge-tool-scanning (running security scanning tools one after the other) sans automation. Unless you are a pro at automating stuff, it is a herculean task to perform a binge scan for every engagement. We normally use scanning tools such as Nmap, the harvester, and DNS recon for vulnerability scanning in websites, so now our tool integrates 10's of tools in kali Linux to perform website scanning at a time. These scanning tools discover vulnerabilities, effectively judge false positives, collectively correlate results, and save precious time; all these under one roof. At the end of this project, we learned a lot of tools in kali Linux that are used for web vulnerabilities, and as we mentioned we were able to achieve most of our motive we tested this scanner tool on a few websites and were able to find vulnerabilities present in the particular website and we able to fix them as well

Table of Contents

Chapter No.	Title	Page No.
1	CHAPTER-1: PROJECT DESCRIPTION AND OUTLINE 1.1 Introduction 1.2 Motivation for the work 1.3 Objective of the work 1.4 Proposed Work 1.5 Features	6-8
2	CHAPTER-2: RELATED WORK INVESTIGATION 2.1 Introduction 2.2 Existing Approaches/Methods/Limitations/Research papers.	8-11

3	<p>CHAPTER-3: MODULES AND EXPLANATION</p> <p>3.1 Introduction 3.2 Color module 3.3 Scanners 3.4 Vulnerability Severity 3.5 Tool Status 3.6 Tool Response 3.7 Remediation Module 3.8 Flowchart & Execution</p>	12-15
4	<p>CHAPTER-4: CONCLUSIONS AND RECOMMENDATION</p> <p>4.1 Outline 4.2 Limitation/Constraints of the System 4.3 Future Enhancements 4.4 Inference 4.5 References</p>	16-19

Chapter 1: PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

A website vulnerability is a flaw or misconfiguration in the coding of a website or web application that allows an attacker to take control of the site and, potentially, the hosting server. The majority of flaws are exploited using automated tools like vulnerability scanners and botnets. Cybercriminals create specific tools that trawl the internet for popular and published vulnerabilities in certain platforms, such as WordPress or Joomla.

Once discovered, these flaws are used to steal data, disseminate malicious information, or insert defacement and spam into the vulnerable site. According to SiteLock data, websites are subjected to an average of 22 attacks per day, or over 8,000 attacks per year. Web vulnerabilities can readily be exploited by malicious users to steal data, compromise user identities, get access to confidential files or information, spam the site, inject scripts, or even seize control of the server. attacks can cause significant damage to a company's reputation and financial status. Understanding and eliminating website vulnerabilities is crucial for every firm that maintains a website or web application.

Periodic online vulnerability testing will allow you to address security flaws before they are exploited by cybercriminals.

1.2 Motivation

Our motivation to take up this project was by seeing the number of tools available in Kali Linux. it is quite a huge task for a pen tester to perform binge tool scanning (running security scanning tools one after another). So, with the help of amazing guidance from our project supervisor, we decided to bring this project to some light. This is a convenient tool as we have shown through the demo in our review II. Our web vulnerability scanner with multi-tool integration is suitable for solving this problem.

1.3 Objective

Our problem statement is mainly focused on website security and our objective is to solve this problem through automation, which entails running multiple scanning tools to discover vulnerabilities, effectively judge false positives, and collectively correlate results.

1.4 Proposed work

We normally use scanning tools such as Nmap, the harvester, and DNS recon for vulnerability scanning in websites, so now our tool integrates 10s of tools in kali Linux together to

perform website scanning at a time. These scanning tools discover vulnerabilities, effectively judge false positives, collectively correlate results, and save precious time; all these under one roof.

1.5 FEATURES

- One step installation.
- Executes a multitude of security scanning tools, does other custom-coded checks, and prints the results spontaneously.
- Some of the tools include Nmap, DNS Recon, Uniscan, Fierce, The Harvester Nikto, etc execute under one entity.
- Saves a lot of time, indeed a lot of time.
- Checks for the same vulnerabilities with multiple tools to help you zero in on false positives effectively.
- Extremely light-weighted and not process intensive.
- Legends to help you understand with multiple tools to help you zero in on false positives effectively.
- Critical, High, Medium, Low, and Informational classification of vulnerabilities.
- Vulnerability definitions guide you on what the vulnerability is and the threat it can pose.
- Remediation tells you how to plug/fix the found vulnerabilities.
- The executive summary gives you an overall context of the scan performed with critical, high, low, and informational issues discovered.
- A detailed comprehensive report in a portable document format at (*.pdf) with complete details of the scans and tools used.

Chapter 2:

Related Work Investigation

2.1 Introduction

There are no particular old versions for our project but still, there are a few tools present now we will see them and compare them with our work. Our web vulnerability scanner will simultaneously run a bunch of tools and will make it easy for a person using Kali Linux or cyber enthusiast, for learning or attacking a website, we normally need to use so many tools, it is a lot of time-consuming so to solve this problem we made this tool.

The ultimate goal of this program is to solve this problem through automation viz running multiple scanning tools to discover vulnerabilities, effectively judge false positives, collectively correlate results, and save precious time.

2.2 Existing Work

We will discuss a few scanning tools already available online which are made to be used in Kali Linux one such tool is ARACHNI. Arachni is a feature-full, modular, high-performance Ruby framework aimed at helping penetration testers and administrators evaluate the security of web applications.

- **METHODOLOGY:**

It is smart, it trains itself by monitoring and learning from the web application's behavior during the scan process and can perform meta-analysis using several factors to correctly assess the trustworthiness of results and intelligently identify false positives. Unlike other scanners, it considers the dynamic nature of web applications, can identify changes produced by traversing through the cyclomatic complexity paths of a web application and adjusts itself accordingly. This allows for the smooth handling of attack/input vectors that would otherwise be unnoticed by nonhumans.

It can also audit and examine client-side code, as well as enable highly complex web applications that leverage technologies like JavaScript, HTML5, DOM manipulation, and AJAX, thanks to its integrated browser environment.

Finally, it's flexible enough to handle a wide range of scenarios, from a simple command-line scanner to a worldwide high-performance grid of scanners, a Ruby library for scripted audits, and a multi-user multi-scan web collaboration. This is completely different from our tool, so by our research, we can confirm this is the first version and type of project which concentrated more on the integration of multiple tools.

- **LIMITATIONS:**

There are a few limitations but this tool doesn't use the basic tools used in Kali Linux, and also as per our work we will provide the vulnerability and the risk factor, as well as remediation for that particular vulnerability and the program used to write the code, is different, and basic difference between our scanning tool and Arachni I train itself by monitoring and learning from the web application's behavior during the scan process and can perform meta-analysis using several factors to correctly assess the trustworthiness of results and intelligently identify false positives but our scanner it trains itself by monitoring and learning from the web application's behavior during the scan process and can perform meta-analysis using several factors to correctly assess it.

Here we will compare few of the research papers with our work along with their classification and statistics.

Source Information	International Journal on Computational Science & Applications 4(1) DOI:10.5121/ijcsa.2014.4111	Cryptography and Security- https://doi.org/10.48550/arXiv.1706.08017
Research topic/question	Vulnerability Scanners- A proactive Approach to Assess Web Application Security	Web Vulnerability Scanners: A Case Study

Methodology	This paper focus on usage of various vulnerability scanners and their related methodology to detect the various vulnerabilities	Acunetix is an automated web Vulnerability scanner which scans any web application or websites that use HTTP or HTTPS protocols and are accessible through a web browser
Findings	Able to identify various vulnerabilities available in the web applications or remote host across the network and identified new mechanisms that can be deployed to secure the network.	After the scanning process completes each vulnerability detected provides additional description, impact and useful tips to fix the vulnerability .
Limitations	Much like our project it doesn't use all the required tools and can't provide you remediation for problem	Acunetix allows multiple scans simultaneously, but this may require more time for completing the entire scanning process
Areas for future research	This paper will completely address various tools used for scanning and their comparative study	Further technical studies can be done to compare different vulnerability scanners , their effectiveness and their peculiar strengths , which in turn would help developers choose an appropriate WVS for each web application.

Chapter 3: Modules and Explanations

3.1 INTRODUCTION

Nmap is one of the scanning tools along with wapiti which is a "black-box" vulnerability scanner, scanning the pages of the deployed web application, extracting links and forms and attacking the scripts, sending payloads, and looking for error messages, special strings, or abnormal behaviors.

We use different modules such as color modules class which is used to represent Headers and also, Warnings, Fail classification of vulnerabilities. We also use Docker, which is an opensource containerization platform used for developing, deploying, and managing applications in lightweight virtualized environments called containers.

MODULES

3.2 COLOR MODULE

This Module helps us to allocate different colors for each section like headers, vulnerabilities, warnings, etc.

Initializing the color module class bcolors:

HEADER = '\033[95m' – PURPLE

WARNING = '\033[93m' – YELLOW

FAIL = '\033[91m' - RED

3.3 SCANNERS

In this module, we will have multiple tools and scanners and their implementation.

["Nmap", Nmap - Fast Scan [Only Few Port Checks]",

```
"nmap",1],
["the Harvester", "The Harvester - Scans for emails using
Google's passive search.", "the Harvester" ,1],
["dnsrecon" , "DNSRecon - Attempts
Multiple Zone Transfers on Nameservers.", "dnsrecon",1],
```

3.4 VULNERABILITY SEVERITY

This module describes the severity of Vulnerability as Critical High, Low, and Medium.

```
def vul_info(Val): result
= " if Val ==
'c':
result = bcolors.BG_CRIT_TXT+" critical "+bcolors.ENDC elif
val == 'h':
result = bcolors.BG_HIGH_TXT+" high
"+bcolors.ENDC else: return result.
```

3.5 TOOL RESPONSE MODULE

In This module, we get responses about the severity and based on scanners we employed on the website

```
["Does not have an IPv6 Address. It is good to
have one.", "i",],

["No Web Application Firewall` Detected" , "m" ,],
```

```
["Some ports are open. Perform a full- scan manually.",  
"I"],
```

3.6 TOOL STATUS

In This module, we will obtain the tool status once it gets obtained responses about the severity.

```
["has IPv6" ,1,proc_low, "< 15s" , "ipv6" , ["not found" ,  
"has IPv6"]],
```

```
["tcp open" ,0,proc_med, "< 2m" , "Nmap open"  
,"Failed to resolve"]],
```

```
["No emails found" ,1,proc_med, "< 3m" , "harvester"  
,"No hosts found" , "No emails found"]],
```

3.7 REMEDIATION

This module will provide remediation for the vulnerabilities found by each tool under the scanners module.

"It is recommended to implement IPv6. More information on how to implement IPv6 can be found in this resource. “Web Application Firewalls offer great protection against common web attacks like XSS, SQLi, etc. They also provide an additional line of defense to your security infrastructure.

3.8 Flowchart & Execution

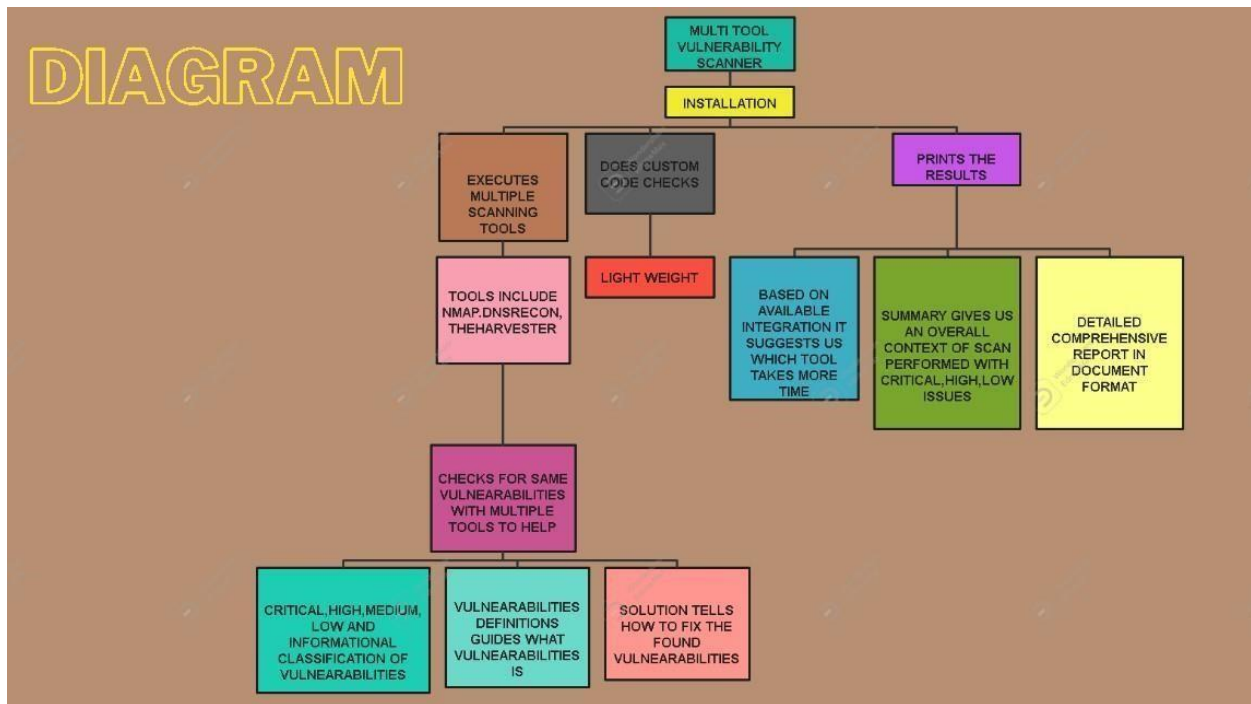


Figure - 1 Web Vulnerability Scanner

```
Terminal Emulator
Use the command line

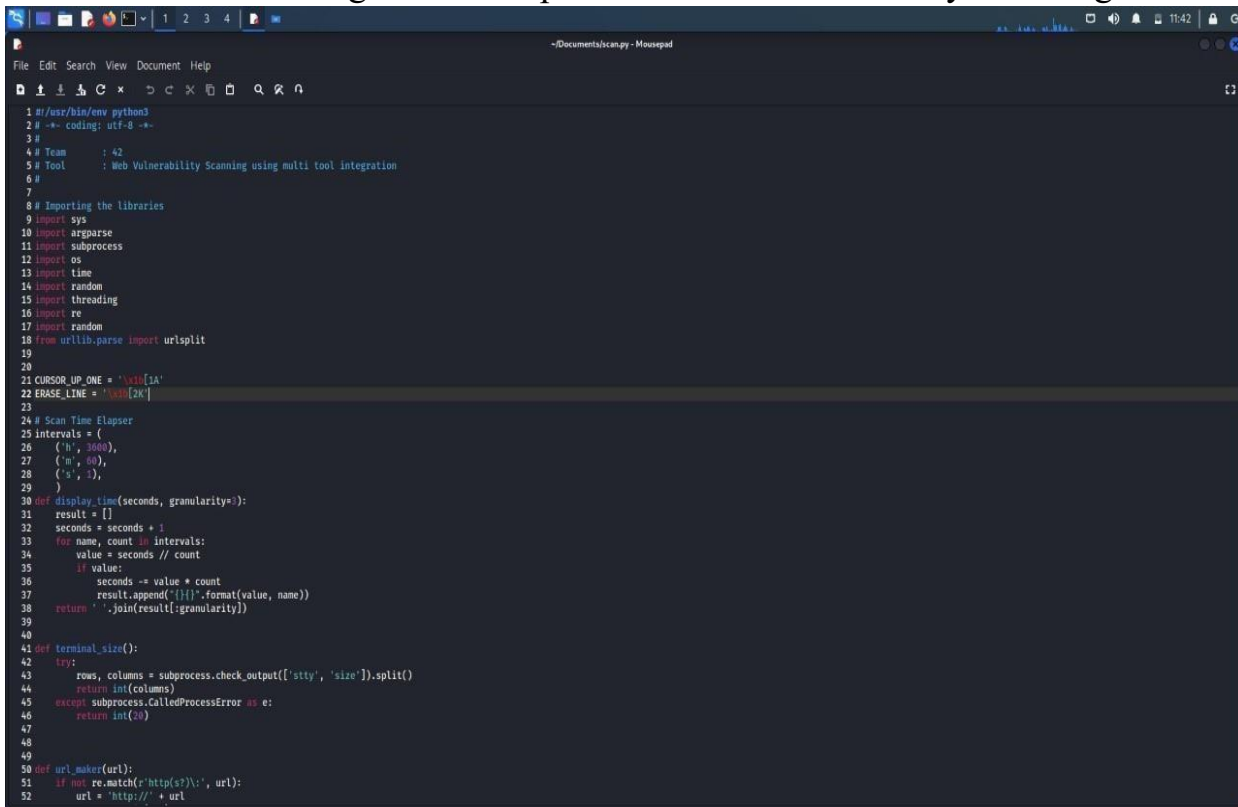
kali@kali:~/rapidscan

Checking Available Security Scanning Tools Phase... Initiated. ]
msfutil... available.
whatweb... available.
nmap... available.
golismere... unavailable.
host... available.
wget... available.
uniscan... unavailable.
wafwaf... available.
dirb... available.
dastext... available.
thomharvester... available.
xsser... unavailable.
dnsrecon... available.
fierce... available.
dnswalk... unavailable.
whois... available.
sslyze... available.
lbd... available.
golismere... unavailable.
dnscat... available.
dmtry... available.
dastext... available.
nikto... available.
dnsmap... unavailable.
amass... available.
Some of these tools ['golismere', 'xsser', 'dnsmap', 'dnswalk', 'uniscan'] are unavailable or will be skipped. RapidScan will still perform the rest of the tests. Install these tools to fully utilize the functionality of RapidScan.

Checking Available Security Scanning Tools Phase... Completed. ]

Preliminary Scan Phase Initiated... Loaded 40 vulnerability checks.
< 45s] Deploying 1/80 | Golismere - BruteForces for certain Files on the Domain.
Warning Tool Unavailable. Skipping Test...
< 20s] Deploying 2/80 | Checks for SMB Service over UDP
n Completed in 4s
< 20s] Deploying 3/80 | Nmap [STUNNET] - Checks if the host is affected by STUNNET Worm.
n Completed in 3s
< 35s] Deploying 4/80 | Dirb - Brutes the target for Open Directories.
n Completed in 32s
< 2s] Deploying 5/80 | Nmap - Fast Scan [Only Few Port Checks]
```

Figure - 2: Snap Shot on Web Vulnerability Scanning



```
1 #!/usr/bin/env python3
2 #-*- coding: utf-8 -*-
3 #
4 # Team      : 42
5 # Tool      : Web Vulnerability Scanning using multi tool integration
6 #
7
8 # Importing the libraries
9 import sys
10 import argparse
11 import subprocess
12 import os
13 import time
14 import random
15 import threading
16 import re
17 import random
18 from urllib.parse import urlsplit
19
20
21 CURSOR_UP_ONE = '\x1b[1A'
22 ERASE_LINE = '\x1b[2K'
23
24 # Scan Time Elapser
25 intervals = (
26     ('h', 3600),
27     ('m', 60),
28     ('s', 1),
29 )
30
31 def display_time(seconds, granularity=1):
32     result = []
33     for name, count in intervals:
34         value = seconds // count
35         if value:
36             seconds -= value * count
37             result.append('{}:{:}'.format(value, name))
38     return ' '.join(result[:granularity])
39
40
41 def terminal_size():
42     try:
43         rows, columns = subprocess.check_output(['stty', 'size']).split()
44         return int(columns)
45     except subprocess.CalledProcessError as e:
46         return int(80)
47
48
49
50 def url_maker(url):
51     if not re.match(r'http(s)?\:\/', url):
52         url = 'http://' + url
```

Figure 3. Code Snippets

CHAPTER-4: CONCLUSIONS & RECOMMENDATIONS

4.1 Outline

The vulnerability scanning application crawls the website and then runs automated checks for common or known online vulnerabilities. It accomplishes this by launching a series of fake attacks and then assessing the consequences. This scanner automates the process of security scanning by using multiple available Linux security tools and some custom scripts.

4.2 Limitation/Constraints of the System

There are some key limitations of tools that are working and during a scan, we can see some tools are showing unavailable, still, there is work need to be done related to specifying the tools and making sure data is correct. There is still a long way to go for it as it does not completely perfect while showing a vulnerability as there are inbuilt integrated modules.

4.3 Future Enhancements

The ultimate goal of this program is to solve this problem through automation; viz. Running multiple scanning tools to discover vulnerabilities, effectively judge false positives, collectively correlate results, and save precious time; all these under one roof. During this project, we have done a lot of research about the tools used in this scanner. We can integrate it with other such tools to make it more comprehensive and increase the user base.

4.4 Inference

In the end, it is inferred that we have made an effort on the following points:

- A description of the background and context of the project and its relation to work already done in the area.
- Made a statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features of tools in detail.
- Finally, the system is implemented and tested.

4.5 References

- [1] : Harrell, Christopher R., et al. "Vulnerability Assessment, Remediation, and Automated Reporting: Case Studies of Higher Education Institutions." 2018 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2018.
- [2] : Wang, Yien, and Jianhua Yang. "Ethical hacking and network defense:

Choose your best network vulnerability scanning tool." 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2017.

[3] : Holm, Hannes, and Teodor Sommestad. "Saved: Scanning, vulnerabilities, exploits, and detection." MILCOM 2016-2016 IEEE Military Communications Conference. IEEE, 2016

[4] : Appiah, Vincent, et al. "Survey of Websites and Web Application Security Threats Using Vulnerability Assessment." (2018).

[5] : Aarya, P. S., et al. "Web Scanning: Existing Techniques and Future." 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2018.

[6]: Yadav, Ravinder, and Aakash Goyal. "Web Application Security." International Journal of Computer Science and Mobile Technology-Vol1 Tccru 10 (2014): 349-355.