A Mini Project Abstract On

WEBSITE VULNERABILITY SCANNER WITH PATCH MANAGEMENT

Submitted to **CMR ENGINEERING COLLEGE**

BACHELOR OF TECHNOLOGY

IN

CSE-CYBER SECURITY

Submitted By

VINOD BIRADAR (218R1A6263)

N.SHIVA TEJA (218R1A6245)

VISHAL VISWAKARMA (228R5A6207)

Under the guidance of

Ch.Venkateswarlu

Assistant Professor, Department of CSE

Department CSE-Cyber Security

CMR ENGINEERING COLLEGE

CMR GROUP OF INSTITUTIONS
EXPLORE TO INVENT

UGC AUTONOMOUS

(Accredited by NBA Approved by AICTE,NEW DELHI, Affiliated to JNTU Hyderabad)

Kandlakoya, Medchal Road, Medchal, Malkajgiri Dist. Hyderabad-501 401)

2024-2025

Table of Contents

1.ABSTRACT

Websites experience multiple attacks per day. A website vulnerability is a weakness or misconfiguration in a website or web application code that allows an attacker to gain some level of

control of the site and possibly the hosting server. A website vulnerability is a software code flaw/ bug, or some other weakness in the website/ web application or its components and processes. Once found, these vulnerabilities are then exploited to steal data, distribute malicious content, or inject defacement and spam content into the vulnerable site. The main objective of this project is to design a website vulnerability scanner which scans vulnerabilities in the web application by taking url as input. It addresses and detect common vulnerabilities like XSS and SQL Injection vulnerabilities among others. All vulnerabilities are displayed and also provides information about risks and how to prevent them.

2. Introduction :

Vulnerabilities have been around for years, largely due to not validating form inputs, misconfigured web servers, and application design flaws, and they can be exploited to compromise the application's security. These vulnerabilities are not the same as other common types of vulnerabilities, such as network or asset. They arise because web applications need to interact with multiple users across multiple networks, and that level of accessibility is easily taken advantage of by hackers. In fact, 50% of all sites were vulnerable to at least one serious exploitable vulnerability throughout 2021, according to a new report by NTT Application Security. These creates a security problem for all business as well as government people. So, making of an affordable scanner is important. Scan for vulnerabilities in web applications and find SQL Injection, XSS, Server Side-Request Forgery, Directory Traversal, and others, plus web server configuration issues. Vulnerability scanning is commonly considered to be the most efficient way to check Your site against a huge list of known vulnerabilities - and identify potential weaknesses in the security of your applications. Vulnerability scanning can be used as part of a standalone assessment, or as part of a continuous overall security monitoring strategy.

3. LITERATURE SURVEY

RushabhTimbadia1,SurajPawar2,Devansh Sayani3,RikenShah4,BhavnaArora5

https://www.irjet.net/archives/V8/i5/I RJETV8I5588.pdf

- Research on Web Application Security Vulnerability Scanning Technology Bin Wang; Lu Liu;Feng Li; Jianye Zhang; Tao Chen; Zhenwan Zou https://ieeexplore.ieee.org/document/8997964

- Vulnerability Scanners-A Proactive Approach To Assess Web Application SecurityTY PY - 2014/03/27 VL - 4 DO - 10.5121/ijcsa.2014.4111 JO - International Journal on Computational Science & Applications

- Web Application Security Scanner for Prevention and Protection against Vulnerabilities Binny George1, Jenu Maria Scaria1, Jobin B1, Praseetha VM2 Some of the existing solutions include

Nessus vulnerability scanner :

Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer you have connected to a network

OpenVAS :

OpenVAS is a full-featured vulnerability scanner. Its capabilities include unauthenticated and authenticated testing, various high-level and low-level internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to

implement any type of vulnerability test. The scanner obtains the tests for detecting vulnerabilities from a feed that has a long history and daily updates.

Acunetix scanner :

Acunetix is an automated web application security testing tool that audits your web applications by checking for vulnerabilities like SQL Injection, Cross site scripting and other exploitable vulnerabilities.

4. System Analysis

4.1 Existing System

This work designs and implements WebCloud, a practical browser-side encryption solution, leveraging modern Web technologies. It solves all the above three problems while achievingseveral additional remarkable features: robust and immediate user revocation, fast data processing with offline encryption and outsourced decryption.As an interesting by-product, the design of WebCloud naturally embodies a dedicated and practical ciphertext-policy attributebased key encapsulation mechanism (CP-AB-KEM) scheme, which can be useful in other applications.Fine-Grained Access Control Mechanism with ABE. It is widelyaccepted that attribute-based encryption (ABE) is promising for fine-grained access control of data. However, we find that the existing ABE schemes suffer from high computational overhead, or some vital missing functionalities, e.g., inefficient dataencryption, robust and immediate user revocation, offline encryption and outsourced decryption simultaneously.

4.2 Proposing System

• Web-applications are often subject to attacks from hackers for a variety of reasons, from gathering data stored by the website to replacing files served with malicious ones. • The proposed system is a web vulnerability scanner, designed to help identify potential security weaknesses in websites. • It is built using Python and the Streamlit library to create a userfriendly interface. • When a user enters the URL of a website to scan, the system initiates the scanning process. It checks for various types of vulnerabilities that could potentially be exploited by attackers. • The vulnerabilities the system checks for include: SQL Injection Cross-Site Scripting (XSS):Cross-Site Request Forgery (CSRF) Server-Side Request Forgery (SSRF) Local File Inclusion (LFI) Remote Code Execution (RCE) • After scanning the website for these vulnerabilities, the system provides a summary of the results, categorizing each vulnerability with indicating its severity. • It also offers additional resources and links to learn more about each vulnerability category and how to prevent them .

4.3Requirement Analysis

The project involved analyzing the design of a few applications so as to make the application more user-friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reduce the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

4.4 Functional Requirements

The Functional requirements for a system describe the functionality or the services that the system is expected to provide.These are the statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situations.

4.5 Non-Functional Requirements

4.6 Feasibility Study

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

4.6.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organisation. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.6.2 TechnicalFeasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.6.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the

users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5. System Specifications :

5.1 Hardware Requirements

- **System:** Dual core cpu with 2.4 GHz clock speed.
- **Hard Disk:** 10 GB and more.
- **Monitor**: Colour Monitor.
- **Mouse:** Optical Mouse.
- **Ram:** 1gb and above.

5.2 Software Requirements

- **Operating system:** Windows 7 and above.
- **Coding Language:** Python.
- **Front-End:** Python.
- **Designing:**CLI.
- **Data Base:** SQLite.

6. System Design Unified Modelling Language Diagrams

UML is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems. UML was created by the Object Management Group (OMG) and the UML1.0 specification draft was proposed to the OMG in January 1997.

There are several types of UML diagrams and each one of them serves a different purposeregardless of whether it is being designed before the implementation or after (as part ofdocumentation). UML has a direct relation with object-oriented analysis and design.

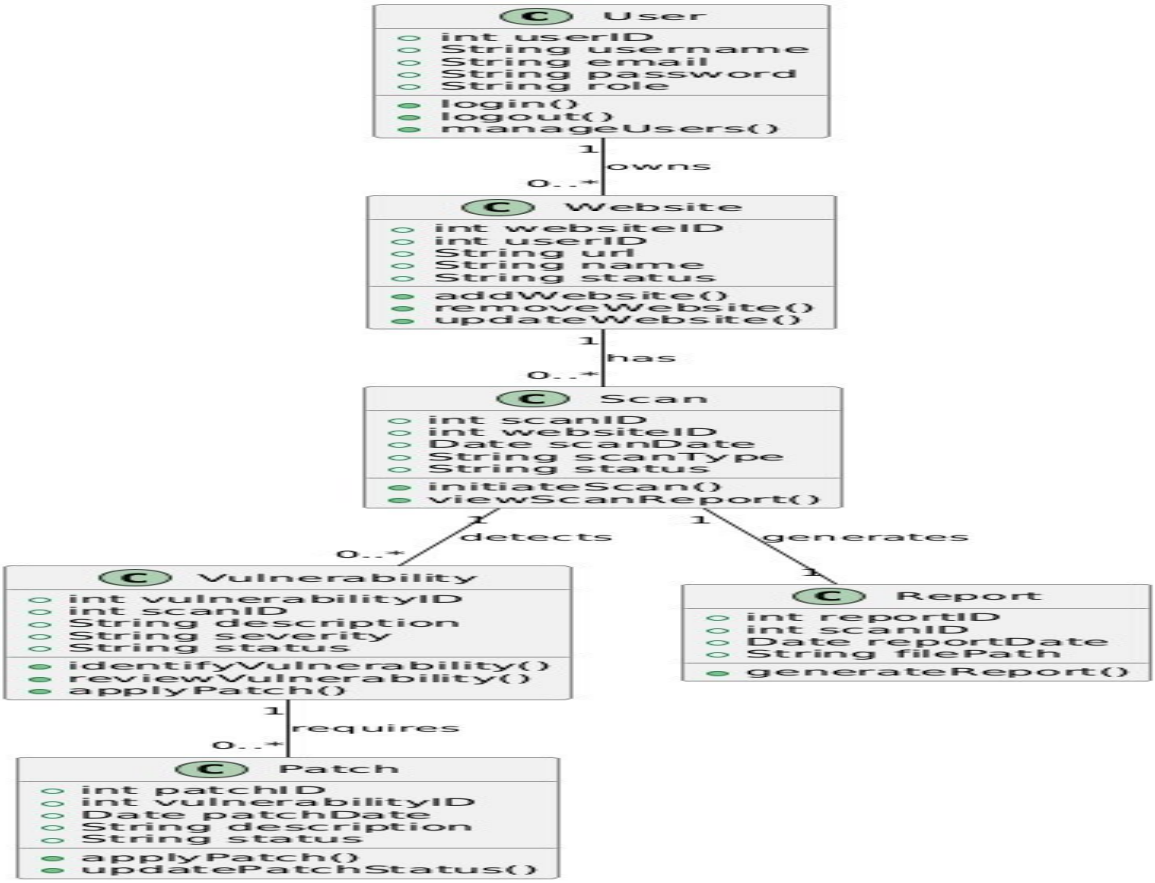
Aftersomestandardization, UML has becomeanOMGstandard.

The two broadest categories that compass all other types are:

1. **Behavioural UML diagram**
2. **Structural UML diagram.**

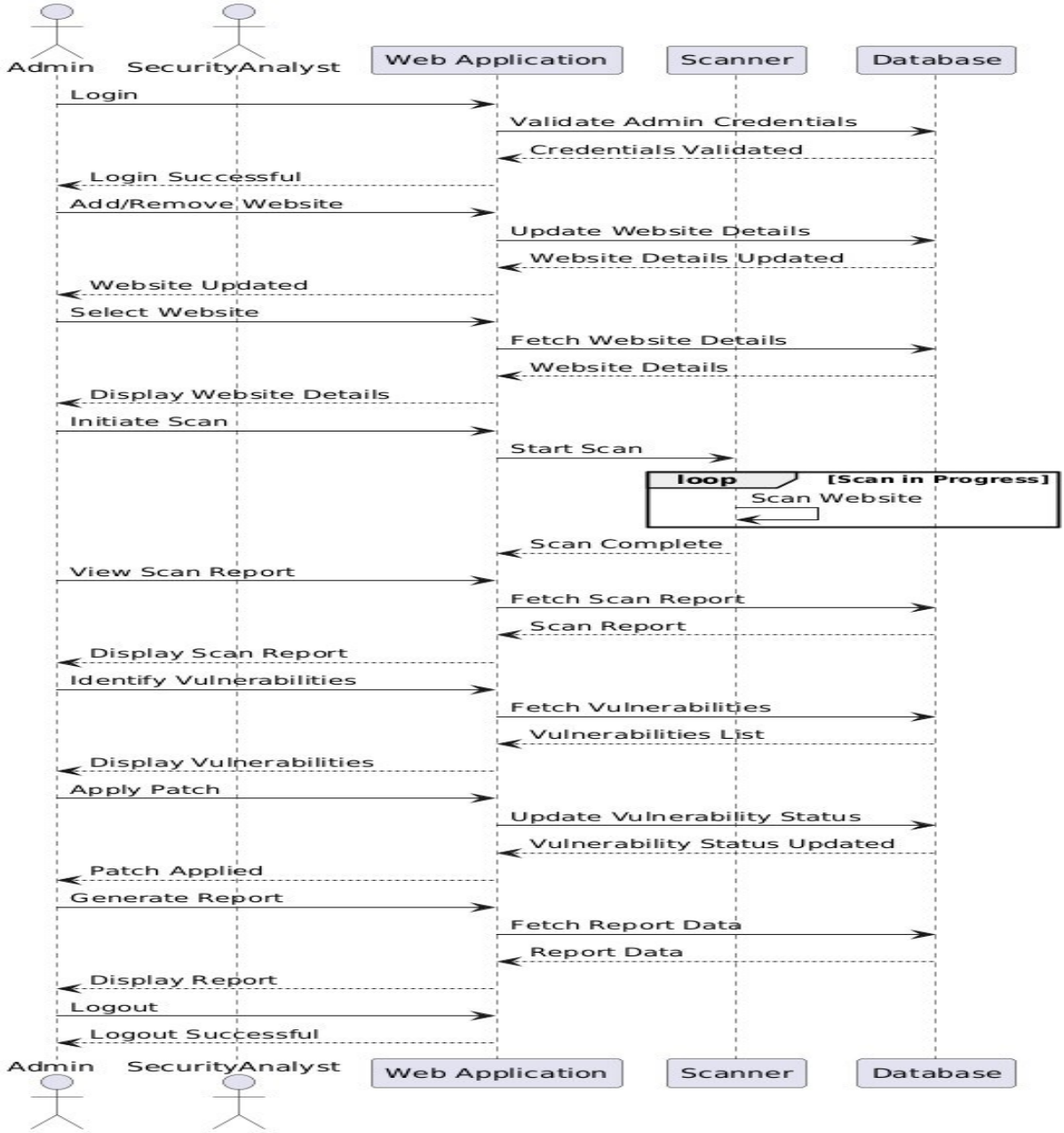
As it suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas others describe the behaviour of the system, its actors, and its building components.

Class diagram :



Class diagrams are a type of UML (Unified Modeling Language) diagram used in software engineering to visually represent the structure and relationships of classes in a system. UML is a standardized modeling language that helps in designing and documenting software systems. They are an integral part of the software development process, helping in both the design and documentation phases.

SEQUENCE DIAGRAM :

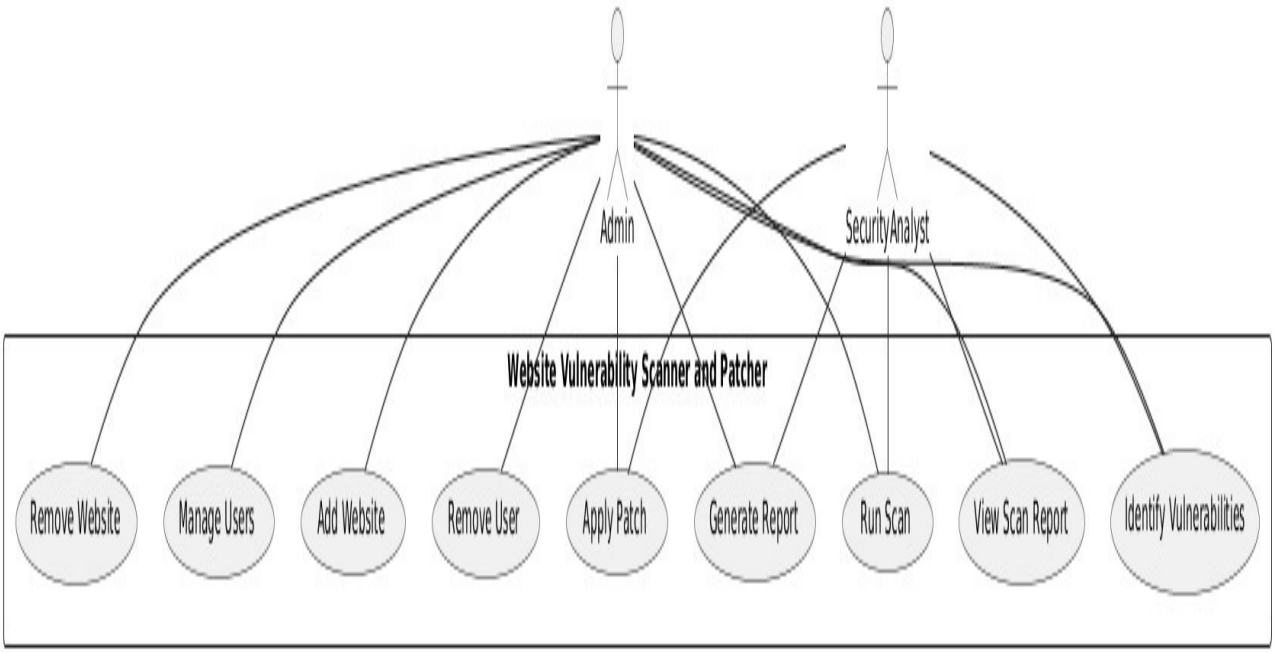


A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

ACTIVITY DIAGRAM :

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

USE CASE DIAGRAM :



A use-case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted

7. System Implementation

7.1 Modules

- User
- ADMIN
- SCAN
- PATCH
- REPORT

User

It defines the access rights of the cloud users. A volume can be created, if it has not exceeded its quota of the permitted volumes and user Authorization is an important security concern in cloud computing environments. a POST request from the authorized user on the volumes resource would

create a new volume. a DELETE request on the volume resource by an authorized user would delete the volume. if the user of the service is authorized to do so, and the volume is not attached to any instance.It aims at regulating the access of the users to system resources.

Admin

The cloud administrator using Keystone and users or user groups are assigned the roles in these projects. It defines the access rights of the cloud users in the project. A volume can be created, if the project has not exceeded its quota of the permitted volumes and a useris authorized to create a volume in the project. Similarly, a volume can be deleted, if the user of the service is authorized to do so, and the volume is not attached to any instance, i.e., its status is not in use.

SCAN:

- Refers to the process of systematically examining a system, network, or application to detect vulnerabilities, misconfigurations, or security weaknesses. Scanning tools (e.g., vulnerability scanners) are often used to detect potential issues like open ports, outdated software, or insecure settings.

PATCH:

- A patch is a software update released by developers to fix bugs, vulnerabilities, or security flaws. Patch management is the process of applying these updates to software or systems to ensure they remain secure and functional.

REPORT:

- After a scan or security assessment, a report is generated to provide detailed information about the findings. The report typically includes detected vulnerabilities, their severity, and recommendations for remediation. It serves as a reference for further action, like patching or fixing issues.

7.3 Technologies

7.3.1 Python

A general-purpose, interactive, object-oriented, and high-level interpreted programming language is called Python. Python is an interpreted language with a design philosophy that prioritizes code readability. For example, instead of using curly brackets or keywords to delimit code blocks,

Python uses whitespace indentation. Python also has a syntax that enables programmers to express concepts in less code than they might in languages like C++ or Java. It offers building blocks that make it possible to program clearly on both small and large dimensions. There are Python interpreters for many different operating systems. The community-based development model is shared by nearly all of Python's variant implementations, including Python, the standard implementation. The nonprofit Python Software Foundation is in charge of running Python. Python has an autonomous memory management system and a dynamic type system.It includes a sizable and thorough standard library and supports a variety of programming paradigms, including imperative, functional, procedural, and object-oriented.

Command Line Arguments

Many programs can be run to provide you with some basic information about how they should be run. Python enables you to do this with -h −

$ python -h usage: python [option] ... [-c cmd | -m mod | file

| -] [arg] ...

Options and arguments (and corresponding environment variables):

-c cmd: program passed in as string (terminates option list)

-d: debug output from parser (also PYTHONDEBUG=x)

-E: ignore environment variables (such as PYTHONPATH)

-h: print this help message and exit

The Project Structure

Client-Server Architecture:

Client-side interface for initiating scans and viewing reports.

Server-side logic for vulnerability scanning and report generation.

Integration with OWASP ZAP:

Utilize OWASP ZAP API for automated vulnerability scanning.

Develop custom logic to interpret scan results and generate reports.

Database Integration:

Store scan results and configuration settings in a relational database.

SQLite or PostgreSQL for lightweight data storage.

8.Source Code

```
import sys import
socket import
subprocess import
os import time
import signal
import random
import string
import threading
import re
from urllib.parse import urlsplit
# Scan Time Elapser intervals
= (
    ('h', 3600),
    ('m', 60),
    ('s', 1),
)
def display_time(seconds, granularity=3):
result = []     seconds = seconds + 1

    for name, count in intervals:
value = seconds // count         if
value:
            seconds -= value * count
result.append("{}{}".format(value, name))
return ' '.join(result[:granularity]) def
url_maker(url):
    if not re.match(r'http(s?)\:', url):
        url = 'http://' + url
parsed = urlsplit(url)     host
```

```
= parsed.netloc     if
host.startswith('www.'):
        host  =  host[4:]
return      host      def
check_internet():
    os.system('ping -c1 github.com > sa_net 2>&1')
if "0% packet loss" in open('sa_net').read():
        val = 1
else:
val = 0
    os.system('rm sa_net > /dev/null 2>&1')
return val
# Initializing the color module class class
bcolors:
    HEADER = '\033[95m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    BADFAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'
    BG_ERR_TXT = '\033[41m'  # For critical errors and crashes BG_HEAD_TXT = '\033[100m'
    BG_ENDL_TXT = '\033[46m'
    BG_CRIT_TXT = '\033[45m'
    BG_HIGH_TXT = '\033[41m'
    BG_MED_TXT = '\033[43m'
    BG_LOW_TXT = '\033[44m'
    BG_INFO_TXT = '\033[42m'
# Classifies the Vulnerability's Severity
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
elif val == 'm':
        result = bcolors.BG_MED_TXT+" medium "+bcolors.ENDC
elif val == 'l':
        result = bcolors.BG_LOW_TXT+" low "+bcolors.ENDC
else:
        result = bcolors.BG_INFO_TXT+" info "+bcolors.ENDC
return result
# Legends proc_high = bcolors.BADFAIL + "●" +
bcolors.ENDC proc_med = bcolors.WARNING + "●" +
bcolors.ENDC proc_low = bcolors.OKGREEN + "●" +
bcolors.ENDC
# Links the vulnerability with threat level and remediation database def
vul_remed_info(v1, v2, v3):
    print(bcolors.BOLD+"Vulnerability Threat Level"+bcolors.ENDC)
print("\t"+vul_info(v2)+" "+bcolors.WARNING + \
str(tool_resp[v1][0])+bcolors.ENDC)
    print(bcolors.BOLD+"Vulnerability Definition"+bcolors.ENDC)
print("\t"+bcolors.BADFAIL+str(tools_fix[v3-1][1])+bcolors.ENDC)
print(bcolors.BOLD+"Vulnerability Remediation"+bcolors.ENDC)
print("\t"+bcolors.OKGREEN+str(tools_fix[v3-1][2])+bcolors.ENDC)

# Scanner Help Context def
helper():
    print(bcolors.OKBLUE+"Information:"+bcolors.ENDC)     print("----------
-")
    print("\t./scanner.py example.com: Scans the domain example.com")
print("\t./scanner.py --update   : Updates the scanner to the latest version.")
print("\t./scanner.py --help     : Displays this help context.")
print(bcolors.OKBLUE+"Interactive:"+bcolors.ENDC)     print("------------")
    print("\tCtrl+C: Skips current test.")
print("\tCtrl+Z: Quits Scanner.")
```

```
   print(bcolors.OKBLUE+"Legends:"+bcolors.ENDC)
print("--------")     print("\t["+proc_high + \
       "]: Scan process may take longer times (not predictable).")
print("\t["+proc_med+"]: Scan process may take less than 10 minutes.")
print("\t["+proc_low+"]: Scan process may take less than a minute or two.")
print(bcolors.OKBLUE+"Vulnerability Information:"+bcolors.ENDC)
print("--------------------------")     print("\t" + \         vul_info(
          'c')+": Requires immediate attention as it may lead to compromise or service unavailability.")
print("\t" + \         vul_info(
          'h')+"    : May not lead to an immediate compromise, but there are high chances of probability.")
print("\t" + \         vul_info(
          'm')+"  : Attacker may correlate multiple vulnerabilities of this type to launch a sophisticated
attack.")     print("\t" + \         vul_info(
          'l')+"     : Not a serious issue, but it is recommended to attend the finding.")
print("\t" + \

       vul_info(
          'i')+"    : Not classified as a vulnerability, simply an useful informational alert to be
considered.\n") # Clears Line def clear():
   sys.stdout.write("\033[F")
sys.stdout.write("\033[K")
# Scanner Logo def
logo():
   print(bcolors.WARNING)
   print("""
O    O O O O O O O O O O    O O O O  O O O O O      O    O    O
O    O O       O   O   O      O         O O    O O   O
O    O O       O  O    O      O         O  O   O O   O
O  O  O O O O O   O O        O    O        O   O  O  O  O
O O O O O       O  O        O   O        O O O O  O   O O
O O  O O O       O   O        O  O        O     O O   O O
O    O O O O O O O O O O    O O O O  O O O O O O       O O    O

MINI PROJECT BY VINOD AND TEAM""")
```

```
   print(bcolors.ENDC)
# Initiliazing the idle loader/spinner class
class Spinner:     busy = False     delay =
0.10     @staticmethod     def
spinning_cursor():
       while 1:
          for cursor in ' ( * ) ( * ) ':
yield cursor  # ←↑↓→
          # for cursor in '←↑↓→': yield cursor
def __init__(self, delay=None):
       self.spinner_generator = self.spinning_cursor()
if delay and float(delay):

          self.delay = delay
def spinner_task(self):
try:
          while self.busy:
             # sys.stdout.write(next(self.spinner_generator))
print(bcolors.BG_ERR_TXT + \
                next(self.spinner_generator)+bcolors.ENDC, end=' ')
sys.stdout.flush()                time.sleep(self.delay)
sys.stdout.write('\b')                sys.stdout.flush()         except
(KeyboardInterrupt, SystemExit):
          # clear()
          print("\n\t" + bcolors.BG_ERR_TXT + \
             "Scanner received a series of Ctrl+C hits. Quitting..." + bcolors.ENDC)
sys.exit(1)     def start(self):
       self.busy = True
       threading.Thread(target=self.spinner_task).start()
def stop(self):         try:
          self.busy = False
time.sleep(self.delay)         except
(KeyboardInterrupt, SystemExit):
          # clear()
```

```
        print("\n\t" + bcolors.BG_ERR_TXT + \
            "Scanner received a series of Ctrl+C hits. Quitting..." + bcolors.ENDC)
sys.exit(1)
# End ofloader/spinner class #
Instantiating the spinner/loader class
spinner = Spinner()
# Scanners that will be used and filename rotation (default: enabled (1)) tool_names
= [
   ["host", "Host - Checks for existence of IPV6 address.", "host", 1],
   ["wp_check","WordPress Checker - Checks for WordPress Installation.","wget",1],
   ["uniscan","Uniscan - Checks for robots.txt & sitemap.xml","uniscan",1],
   ["wafw00f","Wafw00f - Checks for Application Firewalls.","wafw00f",1],
   ["nmap","Nmap - Fast Scan [Only Few Port Checks]","nmap",1],
   ["nmap_header","Nmap [XSS Filter Check] - Checks if XSS Protection Header is present.","nmap",1],
   ["nmap_sloris","Nmap [Slowloris DoS] - Checks for Slowloris Denial of Service
Vulnerability.","nmap",1],
   ["nmap_hbleed","Nmap [Heartbleed] - Checks only for Heartbleed Vulnerability.","nmap",1],
   ["nmap_poodle","Nmap [POODLE] - Checks only for Poodle Vulnerability.","nmap",1],
   ["nmap_ccs","Nmap [OpenSSL CCS Injection] - Checks only for CCS Injection.","nmap",1],
   ["nmap_freak","Nmap [FREAK] - Checks only for FREAK Vulnerability.","nmap",1],
   ["nmap_logjam","Nmap [LOGJAM] - Checks for LOGJAM Vulnerability.","nmap",1],
   ["nmap_telnet","Nmap [TELNET] - Checks if TELNET service is running.","nmap",1],
   ["nmap_ftp","Nmap [FTP] - Checks if FTP service is running.","nmap",1],
   ["nmap_stuxnet","Nmap [STUXNET] - Checks if the host is affected by STUXNET
Worm.","nmap",1],
   ["webdav","WebDAV - Checks if WEBDAV enabled on Home directory.","davtest",1],
        ["golismero_finger","Golismero - Does a fingerprint on the Domain.","golismero",1],
           ["uniscan_filebrute","Uniscan - Brutes for Filenames on the Domain.","uniscan",1],
           ["uniscan_dirbrute", "Uniscan - Brutes Directories on the Domain.","uniscan",1],
           ["uniscan_ministresser", "Uniscan - Stress Tests the Domain.","uniscan",1],
           ["uniscan_rfi","Uniscan - Checks for LFI, RFI and RCE.","uniscan",1],
           ["uniscan_xss","Uniscan - Checks for XSS, SQLi, BSQLi & Other Checks.","uniscan",1],
           ["nikto_xss","Nikto - Checks for Apache Expect XSS Header.","nikto",1],
           ["nikto_subrute","Nikto - Brutes Subdomains.","nikto",1],
```

```
        ["nikto_shellshock","Nikto - Checks for Shellshock Bug.","nikto",1],
        ["nikto_internalip","Nikto - Checks for Internal IP Leak.","nikto",1],
        ["nikto_putdel","Nikto - Checks for HTTP PUT DEL.","nikto",1],
        ["nikto_headers","Nikto - Checks the Domain Headers.","nikto",1],
        ["nikto_ms01070","Nikto - Checks for MS10-070 Vulnerability.","nikto",1],
        ["nikto_servermsgs","Nikto - Checks for Server Issues.","nikto",1],
        ["nikto_outdated","Nikto - Checks if Server is Outdated.","nikto",1],
        ["nikto_httpoptions","Nikto - Checks for HTTP Options on the Domain.","nikto",1],
["nikto_cgi","Nikto - Enumerates CGI Directories.","nikto",1],
        ["nikto_ssl","Nikto - Performs SSL Checks.","nikto",1],
        ["nikto_sitefiles","Nikto - Checks for any interesting files on the Domain.","nikto",1],
        ["nikto_paths","Nikto - Checks for Injectable Paths.","nikto",1],
        ["dnsmap_brute","DNSMap - Brutes Subdomains.","dnsmap",1],
        ["nmap_sqlserver","Nmap - Checks for MS-SQL Server DB","nmap",1],
        ["nmap_mysql", "Nmap - Checks for MySQL DB","nmap",1],
        ["nmap_oracle", "Nmap - Checks for ORACLE DB","nmap",1],
        ["nmap_rdp_udp","Nmap - Checks for Remote Desktop Service over UDP","nmap",1],
        ["nmap_rdp_tcp","Nmap - Checks for Remote Desktop Service over TCP","nmap",1],
        ["nmap_full_ps_tcp","Nmap - Performs a Full TCP Port Scan","nmap",1],
        ["nmap_full_ps_udp","Nmap - Performs a Full UDP Port Scan","nmap",1],
        ["nmap_snmp","Nmap - Checks for SNMP Service","nmap",1],
        ["aspnet_elmah_axd","Checks for ASP.net Elmah Logger","wget",1],
        ["nmap_tcp_smb","Checks for SMB Service over TCP","nmap",1],
        ["nmap_udp_smb","Checks for SMB Service over UDP","nmap",1],
        ["wapiti","Wapiti - Checks for SQLi, RCE, XSS and Other Vulnerabilities","wapiti",1],
        ["nmap_iis","Nmap - Checks for IIS WebDAV","nmap",1],
        ["whatweb","WhatWeb - Checks for X-XSS Protection Header","whatweb",1]]
# Command that is used to initiate the tool (with parameters and extra params) tool_cmd
= [
   ["host ",""],
   ["wget -O temp_wp_check --tries=1 ","/wp-admin"],
   ["uniscan -e -u ",""],
   ["wafw00f ",""],
   ["nmap -F --open -Pn ",""],
```

["nmap -p80 --script http-security-headers -Pn ",""],

["nmap -p80,443 --script http-slowloris --max-parallelism 500 -Pn ",""],

["nmap -p443 --script ssl-heartbleed -Pn ",""],

["nmap -p443 --script ssl-poodle -Pn ",""],

["nmap -p443 --script ssl-ccs-injection -Pn ",""],

["nmap -p443 --script ssl-enum-ciphers -Pn ",""],

["nmap -p443 --script ssl-dh-params -Pn ",""],

["nmap -p23 --open -Pn ",""],

["nmap -p21 --open -Pn ",""],

["nmap --script stuxnet-detect -p445 -Pn ",""],

["davtest -url http://",""],

["golismero -e fingerprint_web scan ",""],

["uniscan -w -u ",""],

["uniscan -q -u ",""],

["uniscan -r -u ",""],

["uniscan -s -u ",""],

["uniscan -d -u ",""],

["nikto -Plugins 'apache_expect_xss' -host ",""],

["nikto -Plugins 'subdomain' -host ",""],

["nikto -Plugins 'shellshock' -host ",""],

["nikto -Plugins 'cookies' -host ",""],

["nikto -Plugins 'put_del_test' -host ",""],

["nikto -Plugins 'headers' -host ",""],

["nikto -Plugins 'ms10-070' -host ",""],

["nikto -Plugins 'msgs' -host ",""],

["nikto -Plugins 'outdated' -host ",""],

["nikto -Plugins 'httpoptions' -host ",""],

["nikto -Plugins 'cgi' -host ",""],

["nikto -Plugins 'ssl' -host ",""],

["nikto -Plugins 'sitefiles' -host ",""],

["nikto -Plugins 'paths' -host ",""],

["dnsmap ",""],

["nmap -p1433 --open -Pn ",""],

```
        ["nmap -p3306 --open -Pn ",""],
        ["nmap -p1521 --open -Pn ",""],
        ["nmap -p3389 --open -sU -Pn ",""],
        ["nmap -p3389 --open -sT -Pn ",""],
        ["nmap -p1-65535 --open -Pn ",""],
        ["nmap -p1-65535 -sU --open -Pn ",""],
        ["nmap -p161 -sU --open -Pn ",""],
        ["wget -O temp_aspnet_elmah_axd --tries=1 ","/elmah.axd"],
        ["nmap -p445,137-139 --open -Pn ",""],
        ["nmap -p137,138 --open -Pn ",""],
["wapiti "," -f txt -o temp_wapiti"],
        ["nmap -p80 --script=http-iis-webdav-vuln -Pn ",""],
        ["whatweb "," -a 1"]
]
# Tool Responses (Begins) [Responses + Severity (c - critical | h - high | m - medium | l - low | i - informational) + Reference for Vuln Definition and Remediation] tool_resp = [
   ["Does not have an IPv6 Address. It is good to have one.", "i", 1],
   ["WordPress Installation Found. Check for vulnerabilities corresponds to that version.","i",2],
   ["robots.txt/sitemap.xml found. Check those files for any information.","i",3],
   ["No Web Application Firewall Detected","m",4],
   ["Some ports are open. Perform a full-scan manually.","l",5],
   ["XSS Protection Filter is Disabled.","m",6],
   ["Vulnerable to Slowloris Denial of Service.","c",7],
   ["HEARTBLEED Vulnerability Found with Nmap.","h",8],
   ["POODLE Vulnerability Detected.","h",9],
   ["OpenSSL CCS Injection Detected.","h",10],
   ["FREAK Vulnerability Detected.","h",11],
   ["LOGJAM Vulnerability Detected.","h",12],
   ["Telnet Service Detected.","h",13],
   ["FTP Service Detected.","c",14],
   ["Vulnerable to STUXNET.","c",15],
   ["WebDAV Enabled.","m",16],
        ["Found some information through Fingerprinting.","l",17],
        ["Open Files Found with Uniscan.","m",18],
```

["Open Directories Found with Uniscan.","m",19],

["Vulnerable to Stress Tests.","h",20],

["Uniscan detected possible LFI, RFI or RCE.","h",21],

["Uniscan detected possible XSS, SQLi, BSQLi.","h",22],

["Apache Expect XSS Header not present.","m",6],

["Found Subdomains with Nikto.","m",23],

["Webserver vulnerable to Shellshock Bug.","c",24], #40

["Webserver leaks Internal IP.","l",25], #41

["HTTP PUT DEL Methods Enabled.","m",26], #43

["Some vulnerable headers exposed.","m",27], #43

["Webserver vulnerable to MS10-070.","h",28], #44

["Some issues found on the Webserver.","m",36], #

["Webserver is Outdated.","h",29], #45

["Some issues found with HTTP Options.","l",26],

["CGI Directories Enumerated.","l",19], #26

["Vulnerabilities reported in SSL Scans.","m",37], #26

["Interesting Files Detected.","m",18], #25

["Injectable Paths Detected.","l",30], #46

["Found Subdomains with DNSMap.","m",23],

["MS-SQL DB Service Detected.","l",31], #47

["MySQL DB Service Detected.","l",31], #47

["ORACLE DB Service Detected.","l",31], #47

["RDP Server Detected over UDP.","h",32], #48

["RDP Server Detected over TCP.","h",32], #48

["TCP Ports are Open","l",5], #8

["UDP Ports are Open","l",5], #8

["SNMP Service Detected.","m",33], #49

["Elmah is Configured.","m",34], #50

["SMB Ports are Open over TCP","m",35], #51

["SMB Ports are Open over UDP","m",35], #51

["Wapiti discovered a range of vulnerabilities","h",36],#

["IIS WebDAV is Enabled","m",10], #16

["X-XSS Protection is not Present","m",6] #12

]

Tool Responses (Ends)

Tool Status (Response Data + Response Code (if status check fails and you still got to push it + Legends + Approx Time + Tool Identification + Bad Responses) tool_status

= [

["has IPv6", 1, proc_low, " < 15s", "ipv6", ["not found", "has IPv6"]], ["wp-login",0,proc_low," < 30s","wpcheck",["unable to resolve host address","Connection timed out"]],

["[+]",0,proc_low," < 40s","robotscheck",["Use of uninitialized value in unpack at"]],

["No WAF",0,proc_low," < 45s","wafcheck",["appears to be down"]],

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

["XSS filter is disabled",0,proc_low," < 20s","nmapxssh",["Failed to resolve"]],

["VULNERABLE",0,proc_high," < 45m","nmapdos",["Failed to resolve"]],

["VULNERABLE",0,proc_low," < 30s","nmap1",["Failed to resolve"]],

["VULNERABLE",0,proc_low," < 35s","nmap2",["Failed to resolve"]],

["VULNERABLE",0,proc_low," < 35s","nmap3",["Failed to resolve"]],

["VULNERABLE",0,proc_low," < 30s","nmap4",["Failed to resolve"]],

["VULNERABLE",0,proc_low," < 35s","nmap5",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmaptelnet",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmapftp",["Failed to resolve"]],

["open",0,proc_low," < 20s","nmapstux",["Failed to resolve"]],

["SUCCEED",0,proc_low," < 30s","webdav",["is not DAV enabled or not accessible."]],

["No vulnerabilities found",1,proc_low," < 15s","golism10",["Cannot resolve domain name","No vulnerabilities found"]],

["[+]",0,proc_med," < 2m","uniscan2",["Use of uninitialized value in unpack at"]],

["[+]",0,proc_med," < 5m","uniscan3",["Use of uninitialized value in unpack at"]],

["[+]",0,proc_med," < 9m","uniscan4",["Use of uninitialized value in unpack at"]],

["[+]",0,proc_med," < 8m","uniscan5",["Use of uninitialized value in unpack at"]],

["[+]",0,proc_med," < 9m","uniscan6",["Use of uninitialized value in unpack at"]],

["0 item(s) reported",1,proc_low," < 35s","nikto1",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto2",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto3",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto4",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto5",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto6",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto7",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto8",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto9",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto10",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto11",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto12",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto13",["ERROR: Cannot resolve hostname","0 item(s) reported","No web server found","0 host(s) tested"]],

["0 item(s) reported",1,proc_low," < 35s","nikto14","ERROR: Cannot resolve hostname , 0 item(s) reported"],

["#1",0,proc_high," < 30m","dnsmap_brute",["[+] 0 (sub)domains and 0 IP address(es) found"]],

["open",0,proc_low," < 15s","nmapmssql",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmapmysql",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmaporacle",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmapudprdp",["Failed to resolve"]],

["open",0,proc_low," < 15s","nmaptcprdp",["Failed to resolve"]],

["open",0,proc_high," > 50m","nmapfulltcp",["Failed to resolve"]],

["open",0,proc_high," > 75m","nmapfulludp",["Failed to resolve"]],

["open",0,proc_low," < 30s","nmapsnmp",["Failed to resolve"]],

["Microsoft SQL Server Error Log",0,proc_low," < 30s","elmahxd",["unable to resolve host address","Connection timed out"]],

["open",0,proc_low," < 20s","nmaptcpsmb",["Failed to resolve"]],

["open",0,proc_low," < 20s","nmapudpsmb",["Failed to resolve"]],

["Host:",0,proc_med," < 5m","wapiti",["none"]],

["WebDAV is ENABLED",0,proc_low," < 40s","nmapwebdaviis",["Failed to resolve"]],

["X-XSS-Protection[1",1,proc_med," < 3m","whatweb",["Timed out","Socket error","X-XSSProtection[1"]]

]

Vulnerabilities and Remediation tools_fix

= [

[1, "Not a vulnerability, just an informational alert. The host does not have IPv6 support. IPv6 provides more security as IPSec (responsible for CIA - Confidentiality, Integrity and Availablity) is incorporated into this model. So it is good to have IPv6 Support.",

"It is recommended to implement IPv6. More information on how to implement IPv6 can be found from this resource. https://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco-on-cisco/IPv6Implementation_CS.html"],

[2, "It is not bad to have a CMS in WordPress. There are chances that the version may contain vulnerabilities or any third party scripts associated with it may possess vulnerabilities","It is recommended to conceal the version of WordPress. This resource contains more information on how to secure your WordPress Blog. https://codex.wordpress.org/Hardening_WordPress"],

[3, "Sometimes robots.txt or sitemap.xml may contain rules such that certain links that are not supposed to be accessed/indexed by crawlers and search engines. Search engines may skip those links but attackers will be able to access it directly.", "It is a good practice not to include sensitive links in the robots or sitemap files."],

[4, "Without a Web Application Firewall, An attacker may try to inject various attack patterns either manually or using automated scanners. An automated scanner may send hordes of attack vectors and patterns to validate an attack, there are also chances for the application to get DoS`ed (Denial of Service)", "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. This resource contains information on web application firewalls that could suit your application. https://www.gartner.com/reviews/market/web-application-firewall"],

[5, "Open Ports give attackers a hint to exploit the services. Attackers try to retrieve banner information through the ports and understand what type of service the host is running","It is recommended to close the ports of unused services and use a firewall to filter the ports wherever necessary. This resource may give more insights. https://security.stackexchange.com/a/145781/6137"],

[6, "As the target is lacking this header, older browsers will be prone to Reflected XSS attacks.", "Modern browsers does not face any issues with this vulnerability (missing headers). However, older browsers are strongly recommended to be upgraded."],

[7, "This attack works by opening multiple simultaneous connections to the web server and it keeps them alive as long as possible by continously sending partial HTTP requests, which never gets completed. They easily slip through IDS by sending partial requests.", "If you are using Apache Module, `mod_antiloris` would help. For other setup you can find more detailed remediation on this resource. https://www.acunetix.com/blog/articles/slow-http-dos-attacks-mitigate-apache-http-server/"],

[8, "This vulnerability seriously leaks private information of your host. An attacker can keep the TLS connection alive and can retrieve a maximum of 64K of data per heartbeat.", "PFS (Perfect Forward Secrecy) can be implemented to make decryption difficult. Complete remediation and resource information is available here. http://heartbleed.com/"],

[9, "By exploiting this vulnerability, an attacker will be able gain access to sensitive data in a n encrypted session such as session ids, cookies and with those data obtained, will be able to impersonate that particular user.", "This is a flaw in the SSL 3.0 Protocol. A better remediation would be to disable using the SSL 3.0 protocol. For more information, check this resource. https://www.uscert.gov/ncas/alerts/TA14-290A"],

[10, "This attacks takes place in the SSL Negotiation (Handshake) which makes the client unaware of the attack. By successfully altering the handshake, the attacker will be able to pry on all the information that is sent from the client to server and vice-versa","Upgrading OpenSSL to latest versions will mitigate this issue. This resource gives more information about the vulnerability and the associated remediation. http://ccsinjection.lepidum.co.jp/"],

[11, "With this vulnerability the attacker will be able to perform a MiTM attack and thus compromising the confidentiality factor.", "Upgrading OpenSSL to latest version will mitigate this issue.

Versions prior to 1.1.0 is prone to this vulnerability. More information can be found in this resource. https://bobcares.com/blog/how-to-fix-sweet32-birthday-attacks-vulnerability-cve-2016-2183/"],

[12, "With the LogJam attack, the attacker will be able to downgrade the TLS connection which allows the attacker to read and modify any data passed over the connection.", "Make sure any TLS libraries you use are up-to-date, that servers you maintain use 2048-bit or larger primes, and that clients you maintain reject Diffie-Hellman primes smaller than 1024-bit. More information can be found in this resource. https://weakdh.org/"],

[13, "Through this deprecated protocol, an attacker may be able to perform MiTM and other complicated attacks.", "It is highly recommended to stop using this service and it is far outdated. SSH can be used to replace TELNET. For more information, check this resource https://www.ssh.com/ssh/telnet"],

[14, "This protocol does not support secure communication and there are likely high chances for the attacker to eavesdrop the communication. Also, many FTP programs have exploits available in the web such that an attacker can directly crash the application or either get a SHELL access to that target.", "Proper suggested fix is use an SSH protocol instead of FTP. It supports secure communication and chances for MiTM attacks are quite rare."],

[15, "The StuxNet is level-3 worm that exposes critical information of the target organization. It was a cyber weapon that was designed to thwart the nuclear intelligence of Iran. Seriously wonder how it got here? Hope this isn't a false positive Nmap ;)", "It is highly recommended to perform a complete rootkit scan on the host. For more information refer to this resource. https://www.symantec.com/security_response/writeup.jsp?docid=2010-071400-3123-99&tabid=3"],

[16, "WebDAV is supposed to contain multiple vulnerabilities. In some case, an attacker may hide a malicious DLL file in the WebDAV share however, and upon convincing the user to open a perfectly harmless and legitimate file, execute code under the context of that user","It is recommended to disable WebDAV. Some critical resource regarding disbling WebDAV can be found on this URL. https://www.networkworld.com/article/2202909/network-security/-webdav-is-bad---says-securityresearcher.html"],

[17, "Attackers always do a fingerprint of any server before they launch an attack. Fingerprinting gives them information about the server type, content- they are serving, last modification times etc, this gives an attacker to learn more information about the target","A good practice is to obfuscate the information to outside world. Doing so, the attackers will have tough time understanding the server's tech stack and therefore leverage an attack."],

[18, "Attackers may find considerable amount of information from these files. There are even chances attackers may get access to critical information from these files.", "It is recommended to block or restrict access to these files unless necessary."],

[19, "Attackers may find considerable amount of information from these directories. There are even chances attackers may get access to critical information from these directories.", "It is recommended to block or restrict access to these directories unless necessary."],

[20, "Attackers mostly try to render web applications or service useless by flooding the target, such that blocking access to legitimate users. This may affect the business of a company or organization as

well as the reputation","By ensuring proper load balancers in place, configuring rate limits and multiple connection restrictions, such attacks can be drastically mitigated."],

[21, "Intruders will be able to remotely include shell files and will be able to access the core file system or they will be able to read all the files as well. There are even higher chances for the attacker to remote execute code on the file system.", "Secure code practices will mostly prevent LFI, RFI and RCE attacks. The following resource gives a detailed insight on secure coding practices. https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices"],

[22, "Hackers will be able to steal data from the backend and also they can authenticate themselves to the website and can impersonate as any user since they have total control over the backend. They can even wipe out the entire database. Attackers can also steal cookie information of an authenticated user and they can even redirect the target to any malicious address or totally deface the application.", "Proper input validation has to be done prior to directly querying the database information. A developer should remember not to trust an end-user's input. By following a secure coding methodology attacks like SQLi, XSS and BSQLi. The following resource guides on how to implement secure coding methodology on application development. https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices"],

[23, "Attackers may gather more information from subdomains relating to the parent domain. Attackers may even find other services from the subdomains and try to learn the architecture of the target. There are even chances for the attacker to find vulnerabilities as the attack surface gets larger with more subdomains discovered.", "It is sometimes wise to block sub domains like development, staging to the outside world, as it gives more information to the attacker about the tech stack. Complex naming practices also help in reducing the attack surface as attackers find hard to perform subdomain bruteforcing through dictionaries and wordlists."],

[24, "Attackers exploit the vulnerability in BASH to perform remote code execution on the target. An experienced attacker can easily take over the target system and access the internal sources of the machine","This vulnerability can be mitigated by patching the version of BASH. The following resource gives an indepth analysis of the vulnerability and how to mitigate it. https://www.symantec.com/connect/blogs/shellshock-all-you-need-know-about-bash-bug-vulnerability https://www.digitalocean.com/community/tutorials/how-to-protect-your-server-against-the-shellshockbash-vulnerability"],

[25, "Gives attacker an idea on how the address scheming is done internally on the organizational network. Discovering the private addresses used within an organization can help attackers in carrying out network-layer attacks aiming to penetrate the organization's internal infrastructure.", "Restrict the banner information to the outside world from the disclosing service. More information on mitigating this vulnerability can be found here. https://portswigger.net/kb/issues/00600300_private-ip-addressesdisclosed"],

[26, "There are chances for an attacker to manipulate files on the webserver.", "It is recommended to disable the HTTP PUT and DEL methods incase if you don't use any REST API Services. Following resources helps you how to disable these methods. http://www.techstacks.com/howto/disable-http-methods-in-tomcat.html https://docs.oracle.com/cd/E19857-01/820-5627/gghwc/index.html https://developer.ibm.com/answers/questions/321629/how-to-disable-http-methods-head-put-deleteoption/"],

[27, "Attackers try to learn more about the target from the amount of information exposed in the headers. An attacker may know what type of tech stack a web application is emphasizing and many other information.", "Banner Grabbing should be restricted and access to the services from outside would should be made minimum."],

[28, "An attacker who successfully exploited this vulnerability could read data, such as the view state, which was encrypted by the server. This vulnerability can also be used for data tampering, which, if

successfully exploited, could be used to decrypt and tamper with the data encrypted by the server.", "Microsoft has released a set of patches on their website to mitigate this issue. The information required to fix this vulnerability can be inferred from this resource. https://docs.microsoft.com/en-us/securityupdates/securitybulletins/2010/ms10-070"],

[29, "Any outdated web server may contain multiple vulnerabilities as their support would've been ended. An attacker may make use of such an opportunity to leverage attacks.", "It is highly recommended to upgrade the web server to the available latest version."],

[30, "Hackers will be able to manipulate the URLs easily through a GET/POST request. They will be able to inject multiple attack vectors in the URL with ease and able to monitor the response as well","By ensuring proper sanitization techniques and employing secure coding practices it will be impossible for the attacker to penetrate through. The following resource gives a detailed insight on secure coding practices. https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices"],

[31, "Since the attacker has knowledge about the particular type of backend the target is running, they will be able to launch a targetted exploit for the particular version. They may also try to authenticate with default credentials to get themselves through.", "Timely security patches for the backend has to be installed. Default credentials has to be changed. If possible, the banner information can be changed to mislead the attacker. The following resource gives more information on how to secure your backend. http://kb.bodhost.com/secure-database-server/"],

[32, "Attackers may launch remote exploits to either crash the service or tools like ncrack to try brute-forcing the password on the target.", "It is recommended to block the service to outside world and made the service accessible only through the a set of allowed IPs only really neccessary. The following resource provides insights on the risks and as well as the steps to block the service. https://www.perspectiverisk.com/remote-desktop-service-vulnerabilities/"],

[33, "Hackers will be able to read community strings through the service and enumerate quite an information from the target. Also, there are multiple Remote Code Execution and Denial of Service vulnerabilities related to SNMP services.", "Use a firewall to block the ports from the outside world. The following article gives wide insight on locking down SNMP service. https://www.techrepublic.com/article/lock-it-down-dont-allow-snmp-to-compromise-network-security/"],

[34, "Attackers will be able to find the logs and error information generated by the application. They will also be able to see the status codes that was generated on the application. By combining all these information, the attacker will be able to leverage an attack.", "By restricting access to the logger application from the outside world will be more than enough to mitigate this weakness."],

[35, "Cyber Criminals mainly target this service as it is very easier for them to perform a remote attack by running exploits. WannaCry Ransomware is one such example.", "Exposing SMB Service to the outside world is a bad idea, it is recommended to install latest patches for the service in order not to get compromised. The following resource provides a detailed information on SMB Hardening concepts. https://kb.iweb.com/hc/en-us/articles/115000274491-Securing-Windows-SMB-and-NetBios-NetBTServices"],

[36, "Particular Scanner found multiple vulnerabilities that an attacker may try to exploit the target.", "Refer to RS-Vulnerability-Report to view the complete information of the vulnerability, once the scan gets completed."],

[37, "SSL related vulnerabilities breaks the confidentiality factor. An attacker may perform a MiTM attack, intrepret and eavesdrop the communication.", "Proper implementation and upgraded version of SSL and TLS libraries are very critical when it comes to blocking SSL related vulnerabilities."],

]

```
# Tool Set tools_precheck
= [     ["wapiti"],
["whatweb"], ["nmap"],
["golismero"], ["host"],
["wget"], ["uniscan"],
["wafw00f"], ["dirb"],
["davtest"],
["theharvester"],
["xsser"], [
        "dnsrecon"], ["fierce"], ["dnswalk"], ["whois"], ["sslyze"], ["lbd"], ["golismero"], ["dnsenum"],
["dmitry"], ["davtest"], ["nikto"], ["dnsmap"]
]
# Shuffling Scan Order (starts)
scan_shuffle = list(zip(tool_names, tool_cmd, tool_resp, tool_status)) random.shuffle(scan_shuffle)
tool_names, tool_cmd, tool_resp, tool_status = list(zip(*scan_shuffle)) #
Cross verification incase, breaks.
tool_checks = (len(tool_names) + len(tool_resp) + len(tool_status)) / 3
# Shuffling Scan Order (ends)
# Tool Head Pointer: (can be increased but certain tools will be skipped)
tool = 0 # Run Test runTest = 1
# For accessing list/dictionary elements arg1 = 0,arg2 =
1,arg3 = 2,arg4 = 3,arg5 = 4,arg6 = 5 # Detected
Vulnerabilities [will be dynamically populated] sa_vul_list
= list() sa_vul_num = 0 sa_vul = 0
# Total Time Elapsed
sa_total_elapsed = 0 #
Tool Pre Checker
sa_avail_tools = 0 #
Checks Skipped
sa_skipped_checks = 0
if len(sys.argv) == 1:
   logo()
helper() else:
```

```
    target = sys.argv[1].lower()     if target == '--update' or
target == '-u' or target == '--u':         logo()

        print("Scanner is updating....Please wait.\n")
spinner.start()
        # Checking internet connectivity first...
sa_internet_availability = check_internet()         if
sa_internet_availability == 0:
            print("\t" + bcolors.BG_ERR_TXT + \
                "There seems to be some problem connecting to the internet. Please try again or later." +
bcolors.ENDC)
            spinner.stop()
sys.exit(1)
        cmd = 'sha1sum scanner.py | grep .... | cut -c 1-40'
oldversion_hash = subprocess.check_output(cmd, shell=True)
oldversion_hash = oldversion_hash.strip()         os.system(
            'wget -N https://github.com/Malwareman007/Scanner-and-
Patcher/tree/main/Full%20Scanner/Web_scan.py -O Web_scan.py > /dev/null 2>&1')
newversion_hash = subprocess.check_output(cmd, shell=True)         newversion_hash
= newversion_hash.strip()         if oldversion_hash == newversion_hash:
clear()
            print("\t" + bcolors.OKBLUE + \
                "You already have the latest version of Scanner." + bcolors.ENDC)
else:
            clear()
            print("\t" + bcolors.OKGREEN + \
                "Scanner successfully updated to the latest version." + bcolors.ENDC)
spinner.stop()         sys.exit(1)     elif target == '--help' or target == '-h' or target
== '--h':
        logo()
helper()
sys.exit(1)     else:

        target = url_maker(target)
```

```
    os.system('rm te* > /dev/null 2>&1')  # Clearing previous scan files
os.system('clear')
    os.system('setterm -cursor off')
logo()
    print(bcolors.BG_HEAD_TXT + \
        "[ Checking Available Security Scanning Tools Phase... Initiated. ]"+bcolors.ENDC)
unavail_tools = 0       unavail_tools_names = list()       while (sa_avail_tools <
len(tools_precheck)):
        precmd = str(tools_precheck[sa_avail_tools][arg1])
try:
            p = subprocess.Popen([precmd], stdin=subprocess.PIPE,
                                 stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True)
output, err = p.communicate()              val = output + err            except:
            print("\t"+bcolors.BG_ERR_TXT+"Scanner was terminated abruptly..."+bcolors.ENDC)
sys.exit(1)            if "not found" in str(val):
            print("\t"+bcolors.OKBLUE + \
              tools_precheck[sa_avail_tools][arg1]+bcolors.ENDC + \
bcolors.BADFAIL+"...unavailable."+bcolors.ENDC)              for
scanner_index, scanner_val in enumerate(tool_names):                  if
scanner_val[2] == tools_precheck[sa_avail_tools][arg1]:
                  # disabling scanner as it's not available.
                  scanner_val[3] = 0
unavail_tools_names.append(
tools_precheck[sa_avail_tools][arg1])
unavail_tools = unavail_tools + 1           else:
            print("\t"+bcolors.OKBLUE + \
              tools_precheck[sa_avail_tools][arg1]+bcolors.ENDC + \
bcolors.OKGREEN+"...available."+bcolors.ENDC)           sa_avail_tools = sa_avail_tools + 1
clear()
    unavail_tools_names = list(set(unavail_tools_names))
if unavail_tools == 0:
        print("\t"+bcolors.OKGREEN + \
          "All Scanning Tools are available. All vulnerability checks will be performed by
Scanner."+bcolors.ENDC)       else:
```

```
        print("\t"+bcolors.WARNING+"Some of these tools "+bcolors.BADFAIL + \
str(unavail_tools_names)+bcolors.ENDC+bcolors.WARNING + \
            " are unavailable. Scanner can still perform tests by excluding these tools from the tests. Please install these tools to fully utilize the functionality of Scanner."+bcolors.ENDC)
print(bcolors.BG_ENDL_TXT + \
        "[ Checking Available Security Scanning Tools Phase... Completed. ]"+bcolors.ENDC)
print("\n")
      print(bcolors.BG_HEAD_TXT+"[ Preliminary Scan Phase Initiated... Loaded "+str(
tool_checks)+" vulnerability checks.  ]"+bcolors.ENDC)        # while (tool < 1):
while(tool < len(tool_names)):
        print("["+tool_status[tool][arg3]+tool_status[tool][arg4]+"] Deploying "+str(
            tool+1)+"/"+str(tool_checks)+" | "+bcolors.OKBLUE+tool_names[tool][arg2]+bcolors.ENDC, end=' ')            if tool_names[tool][arg4] == 0:
            print(bcolors.WARNING+"...Scanning Tool Unavailable. Auto-Skipping Test..."+bcolors.ENDC)
            sa_skipped_checks = sa_skipped_checks + 1
tool = tool + 1                continue
spinner.start()            scan_start = time.time()
        temp_file = "temp_"+tool_names[tool][arg1]
cmd = tool_cmd[tool][arg1]+target + \
tool_cmd[tool][arg2]+" > "+temp_file+" 2>&1"
try:
            subprocess.check_output(cmd, shell=True)
except KeyboardInterrupt:
            runTest = 0
except:
            runTest = 1            if runTest == 1:
spinner.stop()                scan_stop = time.time()
elapsed = scan_stop - scan_start                sa_total_elapsed =
sa_total_elapsed + elapsed
print(bcolors.OKBLUE+"\b...Completed in " + \
display_time(int(elapsed))+bcolors.ENDC+"\n")
clear()
```

```
            sa_tool_output_file = open(temp_file).read()                if
tool_status[tool][arg2] == 0:                    if tool_status[tool][arg1].lower()
in sa_tool_output_file.lower():
                    #print "\t"+ vul_info(tool_resp[tool][arg2]) + bcolors.BADFAIL +" "+
tool_resp[tool][arg1] + bcolors.ENDC
                    vul_remed_info(
                       tool, tool_resp[tool][arg2], tool_resp[tool][arg3])
sa_vul_list.append(
                       tool_names[tool][arg1]+"*"+tool_names[tool][arg2])
else:
                if any(i in sa_tool_output_file for i in tool_status[tool][arg6]):
                    m = 1  # This does nothing.
else:
                    #print "\t"+ vul_info(tool_resp[tool][arg2]) + bcolors.BADFAIL +" "+
tool_resp[tool][arg1] + bcolors.ENDC
                    vul_remed_info(
                       tool, tool_resp[tool][arg2], tool_resp[tool][arg3])
sa_vul_list.append(
                       tool_names[tool][arg1]+"*"+tool_names[tool][arg2])
else:
            runTest = 1
            spinner.stop()
scan_stop = time.time()                elapsed
= scan_stop - scan_start
            sa_total_elapsed = sa_total_elapsed + elapsed
print(bcolors.OKBLUE+"\b\b\b\b...Interrupted in " + \
display_time(int(elapsed))+bcolors.ENDC+"\n")                clear()
            print("\t"+bcolors.WARNING + \
               "Test Skipped. Performing Next. Press Ctrl+Z to Quit Scanner." + bcolors.ENDC)
sa_skipped_checks = sa_skipped_checks + 1            tool = tool+1
      print(bcolors.BG_ENDL_TXT + \
         "[ Preliminary Scan Phase Completed. ]"+bcolors.ENDC)
print("\n")
```

```
##################### Report & Documentation Phase ############################
print(bcolors.BG_HEAD_TXT + \
        "[ Report Generation Phase Initiated. ]"+bcolors.ENDC)
if len(sa_vul_list) == 0:
        print("\t"+bcolors.OKGREEN+"No Vulnerabilities Detected."+bcolors.ENDC)
else:           with open("SA-Vulnerability-Report", "a") as report:
while(sa_vul < len(sa_vul_list)):                   vuln_info = sa_vul_list[sa_vul].split('*')
report.write(vuln_info[arg2])
                report.write("\n------------------------\n\n")
temp_report_name = "temp_"+vuln_info[arg1]                  with
open(temp_report_name, 'r') as temp_report:
                    data = temp_report.read()
report.write(data)
report.write("\n\n")
temp_report.close()                 sa_vul
= sa_vul + 1
            print("\tComplete Vulnerability Report for "+bcolors.OKBLUE+target+bcolors.ENDC+"
named "+bcolors.OKGREEN + \
                "`SA-Vulnerability-Report`"+bcolors.ENDC + \
" is available under the same directory Scanner resides.")
report.close()
      # Writing all scan files output into SA-Debug-ScanLog for debugging purposes.
for file_index, file_name in enumerate(tool_names):          with open("SA-Debug-
ScanLog", "a") as report:               try:                    with
open("temp_"+file_name[arg1], 'r') as temp_report:
                    data = temp_report.read()
report.write(file_name[arg2])
                    report.write("\n------------------------\n\n")
report.write(data)                      report.write("\n\n")
temp_report.close()             except:                 break
report.close()
      print("\tTotal Number of Vulnerability Checks       : " + \
```

```
    bcolors.BOLD+bcolors.OKGREEN+str(len(tool_names))+bcolors.ENDC)
print("\tTotal Number of Vulnerability Checks Skipped: " +
bcolors.BOLD+bcolors.WARNING+str(sa_skipped_checks)+bcolors.ENDC)
    print("\tTotal Number of Vulnerabilities Detected    : " + \
      bcolors.BOLD+bcolors.BADFAIL+str(len(sa_vul_list))+bcolors.ENDC)
print("\tTotal Time Elapsed for the Scan             : "+bcolors.BOLD + \
bcolors.OKBLUE+display_time(int(sa_total_elapsed))+bcolors.ENDC)        print("\n")
    print("\tFor Debugging Purposes, You can view the complete output generated by all the tools named
" + \
       bcolors.OKBLUE+"`SA-Debug-ScanLog`"+bcolors.ENDC+" under the same directory.")
print(bcolors.BG_ENDL_TXT + \
       "[ Report Generation Phase Completed. ]"+bcolors.ENDC)
os.system('setterm -cursor on')
    os.system('rm te* > /dev/null 2>&1')  # Clearing previous scan files
```

9. Testing

UNIT TESTING:

Unit testing is like checking each piece of a puzzle to make sure it fits perfectly before putting the whole puzzle together. For a website vulnerability scanner, this means testing each small part of the scanner individually to make sure it works correctly.

- **Accuracy:** Helps make sure that the scanner finds vulnerabilities correctly and doesn't miss any issues.
- **Reliability:** Ensures that the scanner works well and doesn't break when changes are made.
- **Efficiency:** Makes it easier to find and fix problems early, before they become big issues.

INTEGRATION TESTING:

Integration testing checks how well different parts of a vulnerability scanner work together as a complete system.

It's Important:

- **To Ensure Everything Works Together:** Even if each part works fine alone, they need to function well together to be effective.
- **To Find Issues in Interaction:** This testing helps identify problems that occur when different parts of the scanner are used together.
- Integration testing ensures that all parts of the vulnerability scanner work together smoothly. It helps make sure the scanner finds and reports security issues accurately when used as a complete tool.

COMPONENT TESTING :

Component testing checks if each individual part of the vulnerability scanner works correctly on its own.

It's Important:

- **To Ensure Each Part Works Well:** Each component (like scanning, analyzing, or reporting parts) needs to function properly by itself.
- **To Catch Issues Early:** Finding and fixing problems in each part before they affect the whole system.

Summary:

- Component testing focuses on making sure each individualpart of the vulnerability scanner works correctly by itself. It helps identify and fix issues in each component before they affect the entire scanner.

FUNCTIONAL TESTING :

- Functional testing checks if the website vulnerability scanner does what it's supposed to do. It focuses on verifying that all the features and functions work correctly according to their requirements.

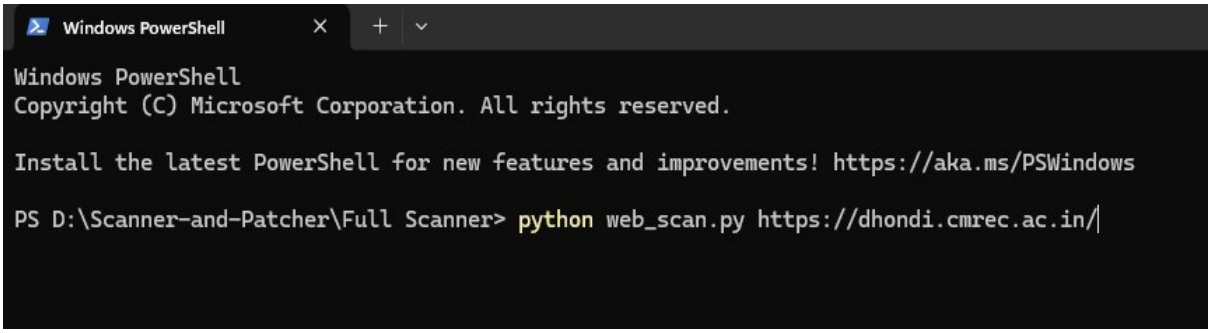
Why It's Important :

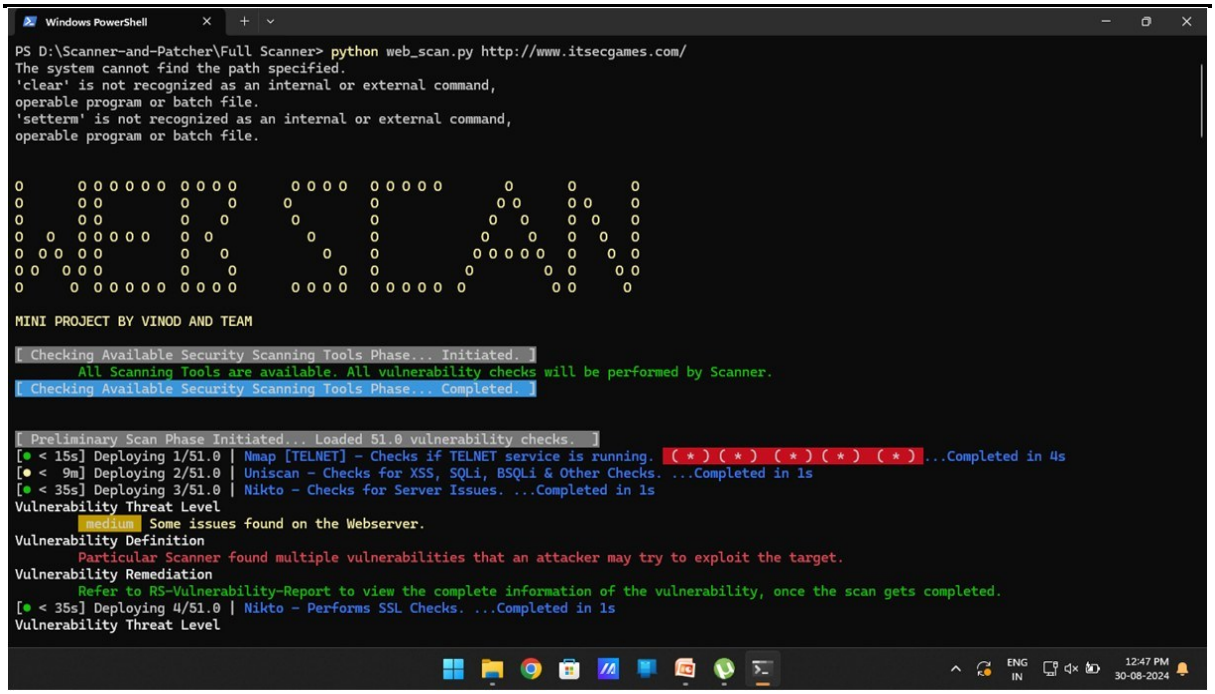
- **To Ensure All Features Work:** Confirms that every function of the scanner performs as expected.
- **To Validate Requirements:** Ensures the scanner meets the specified requirements and performs its tasks correctly.

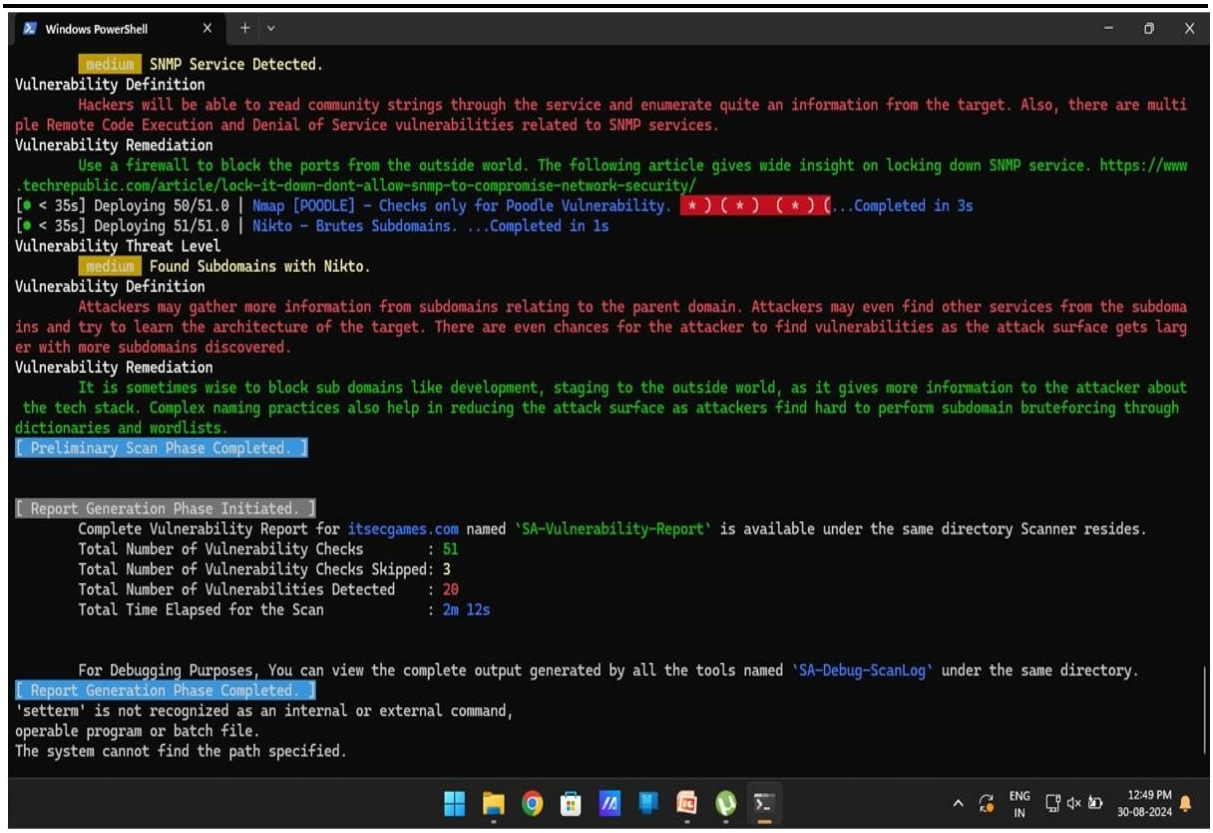
Summary:

- Functional testing makes sure that the website vulnerability scanner performs all its intended tasks correctly. It checks each feature to ensure it works as expected and meets the requirements.

10. Outputs:







11. Conclusion :

This tools is very helpful for finding vulnerabilities present in the Website .

Applications :

* A web application scanner explores a web application by crawling through its web pages and examines it for security vulnerabilities, which involves generation of malicious inputs and evaluation of application's responses.

* These scanners are automated tools that scan web applications to lookforsecurity vulnerabilities. They test web applications for common security problems such as cross-site scripting (XSS), SQL injection, and cross-site request forgery (CSRF).

* This scanner uses different tools like nmap, dnswalk, dnsrecon, dnsenum, dnsmap etc in order to scan ports, sites, hosts and network to find vulnerabilites like OpenSSL CCS Injection, Slowloris, Denial of Service, etc.

12. References

- https://ieeexplore.ieee.org/document/8997964

- https://www.irjet.net/archives/V8/i5/I RJETV8I5588
- Research on Web Application Security Vulnerability Scanning Technology Bin Wang; Lu Liu;Feng Li; Jianye Zhang; Tao Chen; Zhenwan Zou https://ieeexplore.ieee.org/document/8997964

Mini Project Co-OrdinatorInternal Guides