Redscan – The Red Team Web Application Scanner

A PROJECT REPORT

Submitted by

Sumon Nath (20MEI10014)
Shraddha Pandey (20MEI10029)
Srishti Singh (20MEI10072)
Anurag Gupta (20MEI10077)

in partial fulfillment for the award of the degree
of

INTEGRATED MASTER OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING
(SPECIALIZATION IN CYBER SECURITY IN ASSOCIATION WITH VIRTUSA)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

DECEMBER 2021

VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled **"RED SCAN – THE RED TEAM WEB APPLICATION SCANNER"** is the Bonafide work of "**SUMON NATH (20MEI10014), SHRADDHA PANDEY (20MEI10029), SRISHTI SINGH (20MEI10072), ANURAG GUPTA (20MEI10077)"** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
DR. H. AZATH, PROGRAM CHAIR
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
DR. H. AZATH, PROGRAM CHAIR
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 17/12/2021.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr. Shishir Kumar Shandilya**, Head of the Department, School of Computing Science and Engineering (Specialization in Cyber Security and Digital Forensics) for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide **Dr. H. AZATH**, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
ARP	Address Resolution Protocol
CSR	Certificate Signing Request
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DoD	Department of Defense
DoS	Denial of Service
DSS	Digital Signature Standard
EAL	Evaluation Assurance Level
GIG	Global Information Grid
HTTP	Hypertext Transfer Protocol
HUMINT	Human Intelligence
IA	Information Assurance
IV	Initialization Vector
LAN	Local Area Network
MAC	Mandatory Access Control
MAC	Media Access Control
MAC	Message Authentication Code
NSA	National Security Agency
PBX	Private Branch Exchange
PC	Personal Computer
ROM	Read-Only Memory
SSID	Service Set Identifier
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TPM	Trusted Platform Module
URL	Uniform Resource Locator

LIST OF FIGURES AND GRAPHS

LIST OF TABLES

ABSTRACT

Part of the challenge with business cybersecurity is maintaining and using the full set of tools necessary for keeping up with the changing threat landscape. As IoT botnets, crypto mining malware, and other emerging threats evolve, it is increasingly unrealistic for organizations to keep up on their own. Being prepared remains critically important to maintaining business operations and productivity, however. By selecting a comprehensive, proactive security and remediation service and planning ahead, you can be reasonably assured that your business will meet any security challenges it might face. Therefore, we planned to build a tool that is useful for the organization in order to scan web application and discover the vulnerabilities that are present in there system so that they can take further action to mitigate it. Overall, the script works as automated tool as a red team works to find the loopholes of their respected services. It will offer to the organization a better understanding of its assets, security flaws and overall risk, reducing the likelihood that a cybercriminal will breach its systems and catch the business off guard. It overcomes the many limitations incorporated in the attendance.

0

TABLE OF CONTENTS

CHAPTER – 1

PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

Organizations need a Web application scanning solution that can scan for security loopholes in Web-based applications to prevent would-be hackers from gaining unauthorized access to corporate information and data. Web applications are proving to be the weakest link in overall corporate security, even though companies have left no stone unturned in installing the better-known network security and anti-virus solutions. Quick to take advantage of this vulnerability, hackers have now begun to use Web applications as a platform for gaining access to corporate data; consequently, the regular use of a web application scanner is essential.

"Redscan – The Red Team Web Application Scanner" here, starting from terminologies of our project. The "Redscan" means ease the challenge of facing the latest threats alone and discover our range of specialist security services to reduce the burden and enhance your organization's cyber resilience. Similarly, we worked with python script and prepared a code or frame work which includes many certain libraries and modules to process the vulnerability scanning work. Modules like Burp Suite, Nmap, Nikto, dnsmap, amass, Golismerp, WebDAV and SSLyze and approximate 40 tools. This provides to scan the web application, find out the weak points of application and instant remediation to secure it.

1.2 Motivation for The Work

Today in this cyber world, many kind web services are running to gain a large market access for earning profit by providing certain service to client. Another face of this is web services are at the ease of data leak by their opponents, insider treats, and hackers. They need security to protect their data by adding different counter measures to it. So, at this stage they have a different wing of team who analyzes what are treats to their web application. And these threats are called bug by attacker side, many of these web applications have some bug bounty program which provides certain gifts, certificate and rewards to pentester. Which motivate us to work on a framework and Skelton model to quick access to vulnerabilities of application.

1.3 Ideology of Project

To develop a tool where we can successfully scan all types of vulnerabilities that is present in a web application and tell the threat level of that vulnerability and provide the solution for mitigation. It will offer to the organization a better understanding of its assets, security flaws and overall risk, reducing the likelihood that a cybercriminal will breach its systems and catch the business off guard.

Clearly, Web applications are the biggest Achilles heel in an organization's security strategy. They are much more difficult to protect than traditional applications that reside behind a firewall. Web application security needs to be stringently checked using an automated Web application security scanner.

1.4 Techniques Used

We included following sub division in our project to be executed well.

Classifies the Vulnerability's Severity

Links the vulnerability with threat level and remediation database

Help Context

Initializing the idle loader/spinner class

Command that is used to initiate the tool (with parameters and extra params)

Tool Responses (Begins) [Responses + Severity (c - critical | h - high | m - medium | l - low | i - informational) + Reference for Vuln Definition and Remediation]

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

Vulnerabilities and Remediation/tool fix

Shuffling Scan Order (starts)

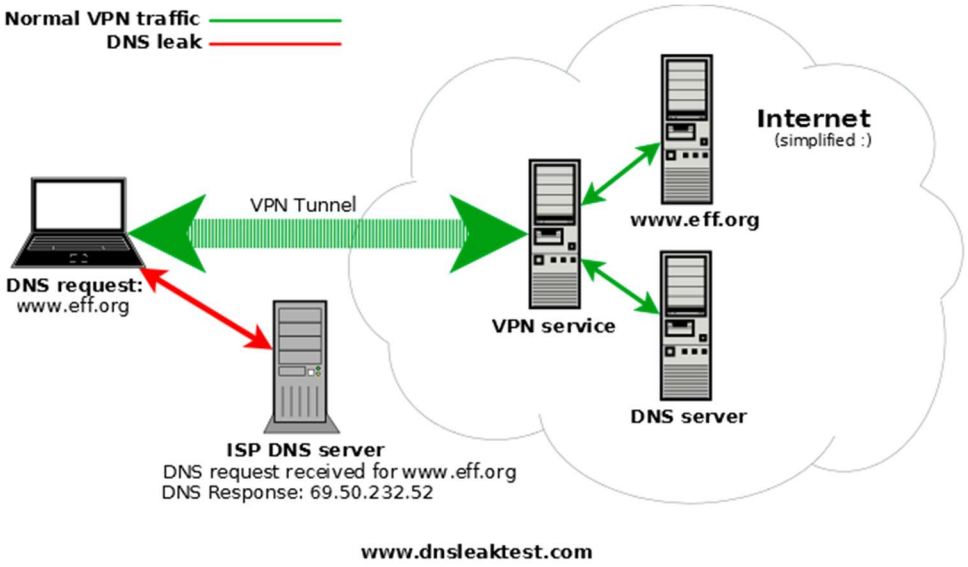
Report and documentation phase

1.5 Problem Statement

Generally, due to the following issues faced by web application or web services:

- ✓ DNS issues and network connectivity

An essential element of successful web traffic management is DNS queries, which is why an issue with these systems can result in a plethora of issues. Without the proper protection, faulty DNS queries can prevent visitors from reaching your website, while also causing errors, 404s, and incorrect pathways.



- ✓ Slow servers and loading time

If your servers are particularly slow, they could be hosted using a shared account, which means that your site is sharing the server with hundreds, possibly thousands of other websites. You can address this common roadblock by checking with your hosting company to determine whether or not the site is hosted on a dedicated server.

- ✓ Poorly written code

Another web application performance problem that many face is with poorly written code, which could refer to inefficient code, memory leaks, or synchronization issues. Your application could also deadlock due to ineffectual algorithms, as well as the performance degradation of a web application.

- ✓ Lack of load balancing

Slow response times can also be caused by poor load distribution. When new site visitors are assigned incorrectly, it can drown out your servers even if the system is under capacity. Such an issue can cause a slow response time, especially if your site is receiving too many requests.

- ✓ Traffic spikes

Spikes happen, especially during a marketing promotion with videos, and a company may not be prepared for the extra traffic. This issue can also cause your servers to slow down, hindering the performance of your site and harming your brand.

- ✓ Specific HTML title tags

Even the name of your website can affect its performance as HTML title tags are essential to its success. These tags sum up the entire content of your website or web page to major search engines such as Google.

- ✓ Failing to optimize bandwidth Usage

When developing and testing a site, businesses often rely on a local network environment. This may not seem like an issue at first because adding visual, audio, video or other high-volume data may not affect your local network.

1.6 Objective of The Work

Our objective is to provide a common platform that scan any website such as example.com, web application scan automatically and with great accuracy web applications to give greater visibility of vulnerabilities present in different types of assets; as well as a context that helps you prioritize the correction of faults. Also show the threat level of vulnerability, define the vulnerability and also provide vulnerability remediation. At the end extracted report in the separately.

1.7 Organization of the Project

For giving our project, a shape of automated vulnerability scanning tool we included the following python libraries given below.

# Importing the libraries	
	import sys
	import socket
	import subprocess
	import os
	import time
	import signal
	import random
	import string
	import threading
	import re
	from urlparse import urlsplit

Table1.1

1.8 Summary

Part of the challenge with business cybersecurity is maintaining and using the full set of tools necessary for keeping up with the changing threat landscape. As IoT botnets, crypto mining malware, and other emerging threats evolve, it is increasingly unrealistic for organizations to keep up on their own. Being prepared remains critically important to maintaining business operations and productivity, however. By selecting a comprehensive, proactive security and remediation service and planning ahead, you can be reasonably assured that your business will meet any security challenges it might face. Therefore, we planned to build a tool that is useful for the organization in order to scan web application and discover the vulnerabilities that are present in there system so that they can take further action to mitigate it.

CHAPTER – 2

RELATED WORK INVESTIGATION

2.1 Introduction

For making our plan a success our team visited many sources where we searched for sub modules of every tool nikto, Nmap, uniscan, etc. And found the best of file and inserted in our code. For more coding frequency we visited GitHub sources for making the code. And our team each member has their own responsibility data collector and manager, beta tester, collaborator and script editor.

2.2 Core area of the project

The main core area is to combine all data and script and its sub class together and run it. Which covers the automated testing phase of project. the script works as automated tool as a red team works to find the loopholes of their respected services. It will offer to the organization a better understanding of its assets, security flaws and overall risk.

2.3 Existing Approaches/Methods

1.2(Vulnerability Scanning)

1. Nikto2

Nikto2 is an open-source vulnerability scanning software that focuses on web application security. Nikto2 can find around 6700 dangerous files causing issues to web servers and report outdated servers based versions. On top of that, Nikto2 can alert on server configuration issues and perform web server scans within a minimal time.

Nikto2 doesn't offer any countermeasures for vulnerabilities found nor provide risk assessment features. However, Nikto2 is a frequently updated tool that enables a broader coverage of vulnerabilities.

2. Netsparker

Netsparker is another web application vulnerability tool with an automation feature available to find vulnerabilities. This tool is also capable of finding vulnerabilities in thousands of web applications within a few hours.

Although it is a paid enterprise-level vulnerability tool, it has many advanced features. It has crawling technology that finds vulnerabilities by crawling into the application. Netsparker can describe and suggest mitigation techniques for vulnerabilities found. Also, security solutions for advanced vulnerability assessment are available.

3. OpenVAS

OpenVAS is a powerful vulnerability scanning tool that supports large-scale scans which are suitable for organizations. You can use this tool for finding vulnerabilities not only in the web application or web servers but also in databases, operating systems, networks, and virtual machines.

OpenVAS receives updates daily, which broadens the vulnerability detection coverage. It also helps in risk assessment and suggests countermeasures for the vulnerabilities detected.

4. W3AF

W3AF is a free and open-source tool known as Web Application Attack and Framework. This tool is an open-source vulnerability scanning tool for web applications. It creates a framework which helps to secure the web application by finding and exploiting the vulnerabilities. This tool is known for user-friendliness. Along with vulnerability scanning options, W3AF has exploitation facilities used for penetration testing work as well.

Moreover, W3AF covers a high-broaden collection of vulnerabilities. Domains that are attacked frequently, especially with newly identified vulnerabilities, can select this tool.

5. Arachni

Arachni is also a dedicated vulnerability tool for web applications. This tool covers a variety of vulnerabilities and is updated regularly. Arachni provides facilities for risk assessment as well as suggests tips and countermeasures for vulnerabilities found.

Arachni is a free and open-source vulnerability tool that supports Linux, Windows, and macOS. Arachni also assists in penetration testing by its ability to cope up with newly identified vulnerabilities.

6. Acunetix

Acunetix is a paid web application security scanner (open-source version also available) with many functionalities provided. Around 6500 vulnerabilities scanning range is available with this tool. In addition to web applications, it can also find vulnerabilities in the network as well.

Acunetix provides the ability to automate your scan. Suitable for large scale organizations as it can handle many devices. HSBC, NASA, USA Air force are few industrial giants who use Arachni for vulnerability tests.

7. Nmap

Nmap is one of the well-known free and open-source network scanning tools among many security professionals. Nmap uses the probing technique to discover hosts in the network and for operating system discovery.

This feature helps in detecting vulnerabilities in single or multiple networks. If you are new or learning with vulnerabilities scanning, then Nmap is a good start.

8. OpenSCAP

OpenSCAP is a framework of tools that assist in vulnerability scanning, vulnerability assessment, vulnerability measurement, creating security measures. OpenSCAP is a free and open-source tool developed by communities. OpenSCAP only supports Linux platforms.

OpenSCAP framework supports vulnerability scanning on web applications, web servers, databases, operating systems, networks, and virtual machines. Moreover, they provide a facility for risk assessment and support to counteract threats.

9. GoLismero

GoLismero is a free and open-source tool used for vulnerability scanning. GoLismero focuses on finding vulnerabilities on web applications but also can scan for vulnerabilities in the network as well. GoLismero is a convenient tool that works with results provided by other vulnerability tools such as OpenVAS, then combines the results and provides feedback.

GoLismero covers a wide range of vulnerabilities, including database and network vulnerabilities. Also, GoLismero facilitates countermeasures for vulnerabilities found.

10. Intruder

Intruder is a paid vulnerability scanner specifically designed to scan cloud-based storage. Intruder software starts to scan immediately after a vulnerability is released. The scanning mechanism in Intruder is automated and constantly monitors for vulnerabilities.

Intruder is suitable for enterprise-level vulnerability scanning as it can manage many devices. In addition to monitoring cloud-storage, Intruder can help identify network vulnerabilities as well as provide quality reporting and suggestions.

11. Comodo HackerProof

With Comodo Hackerproof you will be able to reduce cart abandonment, perform daily vulnerability scanning, and use the included PCI scanning tools. You can also utilize the drive-by attack prevention feature and build valuable trust with your visitors. Thanks to the benefit of Comodo Hackerproof, many businesses can convert more visitors into buyers. Buyers tend to feel safer when making a transaction with your business, and you should find that this drives your revenue up. With the patent-pending scanning technology, SiteInspector, you will enjoy a new level of security.

12. Aircrack

Aircrack also is known as Aircrack-NG, is a set of tools used for assessing the WiFi network security. These tools can also be utilized in network auditing, and support multiple OS's such as Linux, OS X, Solaris, NetBSD, Windows, and more.

The tool will focus on different areas of WiFi security, such as monitoring the packets and data, testing drivers and cards, cracking, replying to attacks, etc. This tool allows you to retrieve the lost keys by capturing the data packets.

13. Retina CS Community

Retina CS Community is an open-source web-based console that will enable you to make a more centralized and straightforward vulnerability management system. Retina CS Community has features like compliance reporting, patching, and configuration compliance, and because of this, you can perform an assessment of cross-platform vulnerability. The tool is excellent for saving time, cost, and effort when it comes to managing your network security. It features an automated vulnerability assessment for DBs, web applications, workstations, and servers. Businesses and organizations will get complete support for virtual environments with things like virtual app scanning and vCenter integration.

14. Microsoft Baseline Security Analyzer (MBSA)

An entirely free vulnerability scanner created by Microsoft, it's used for testing your Windows server or windows computer for vulnerabilities. The Microsoft Baseline Security Analyzer has several vital features, including scanning your network service packets, checking for security updates or other windows updates, and more. It is the ideal tool for Windows users. It's excellent for helping you to identify missing updates or security patches. Use the tool to install new security updates on your computer. Small to medium-sized businesses find the tool most useful, and it helps save the security department money with its features. You won't need to consult a security expert to resolve the vulnerabilities that the tool finds.

15. Nexpose

Nexpose is an open-source tool that you can use for no cost. Security experts regularly use this tool for vulnerability scanning. All the new vulnerabilities are included in the Nexpose database thanks to the GitHub community. You can use this tool with the Metasploit Framework, and you can rely on it to provide a detailed scanning of your web application. Before generating the report, it will take various elements into account.

Vulnerabilities are categorized by the tool according to their risk level and ranked from low to high. It's capable of scanning new devices, so your network remains secure. Nexpose is updated each week, so you know it will find the latest hazards.

16. Nessus Professional

Nessus is a branded and patented vulnerability scanner created by Tenable Network Security. Nessus will prevent the networks from attempts made by hackers, and it can scan the vulnerabilities that permit remote hacking of sensitive data. The tool offers an extensive range of OS, Dbs, applications, and several other devices among cloud infrastructure, virtual and physical networks. Millions of users trust Nessus for their vulnerability assessment and configuration issues.

17. SolarWinds Network Configuration Manager

SolarWinds Network Configuration Manager has consistently received high praise from users. The vulnerability assessment tool features that it includes addresses a specific type of vulnerability that many other options do not, such as misconfigured networking equipment. This feature sets it apart from the rest. The primary utility as a vulnerability scanning tool is in the validation of network

equipment configurations for errors and omissions. It can also be used to check device configurations for changes periodically.

2.4 Pros and cons

Benefits of vulnerability scanning tools

Fast results

The major advantage of an automated scanning tool is that it generates a result relatively quickly. That way, you can get a picture of your security whenever you want it.

Repeatable

An automated vulnerability scan is easy to repeat. You decide whether you want to run a scan daily, weekly or monthly and get an update on changes and vulnerabilities detected.

User-friendly

Most vulnerability scanning tools have a clear interface and are therefore easy to use. As such, the barrier to system administrators and others using them is low. It should be noted, however, that the results of the tools contain fairly specialist details. This means that a security specialist is still needed to interpret the findings and take action.

Constant monitoring

A vulnerability scanning tool can also be deployed effectively for constant monitoring, for instance if a lot of deployments are performed. Moreover, it offers system administrators continuous insight into the status of the infrastructure.

Drawbacks of vulnerability scanning tools

A vulnerability scanning tool will not find nearly all vulnerabilities

Because a vulnerability scanning tool also misses vulnerabilities, you have no guarantee that your systems are not vulnerable. This is one of the biggest limitations of all scanning tools, because there can still be vulnerabilities that hackers can exploit. There are two possible reasons for this:

- The scanner is not aware of the vulnerability, for example because it has only just been discovered.
- The vulnerability is too complex to be found by an automated tool because the attack is not trivial to automate.

Constant updates required

In order to ensure that the most recent vulnerabilities are found, you need to make sure the tool is continually updated.

False positives

Particularly if you have a large IT infrastructure, lots of servers and services, it can be hard to understand the impact of the findings/vulnerabilities of the scanning tool. As a result, you will often be faced with false positives. If you are not specialised in security, recognising them is a challenge,

which makes interpreting the results a time-consuming business. Moreover, if false positives are not filtered out, the tool does not get smarter and will continue to generate false results.

Implications of vulnerability unclear

If a vulnerability is found, it is sometimes difficult to assess what it means for business operations. What will be the impact on different departments, employees and processes? An automated tool will not tell you this and a system administrator will typically be more focused on the technical aspect of the vulnerability.

2.5 Issues/observations from investigation

We have many tools that are used for scanning the vulnerabilities scanning in web application but complete solution is not there that can provide to organization and scan all type of vulnerabilities in one go. Every tool is made for scanning a specific vulnerability that used to perform that task only and nothing more.

2.6 Summary

There are several commercial and free vulnerability scanners available on the market – here is a list of the most popular tools:

AppScan (IBM), Burp Suite (PortSwiger), Dnsmap, Amass, usiscan (HP), Sslyze, xsser and many more

But they are designed are designed to scan specific vulnerability as a result organization's need many tools to scan all types of vulnerabilities in a web application, that is time consuming and tend to many attacks like zero day and many more.

So, organizations need a Web application scanning solution that can scan for security loopholes in Web-based applications to prevent would-be hackers from gaining unauthorized access to corporate information and data and save their time.

CHAPTER - 3

REQUIREMENT ARTIFACTS

3.1 Introduction

To run the vulnerability scanning tools there are certain requirements which want to be fulfilled like the machine on which the python script is going to run have the newly updated version of python library to be successfully get the outcome.

3.2 Hardware and Software requirements

- ✓ Machine: Window 7 and above
- ✓ Clock Speed: 500 MHZ OR higher
- ✓ System Memory: 512 and above
- ✓ Hard Disk Space: 20 GB and above
- ✓ Any OS (Preferred for Linux Distro's for quick result), Python IDE

3.3 Specific Project requirements

There are such requirements mainly we need a window7 or above, 4 core above CPU for faster operations, some integrated libraries and updated programming language which may be not present in normal compiler so we preferred to Linux disrto terminal for it.

3.3.1 Data requirement

10 GB and above hard disk space must be provided to virtual machine. To store the result and for performance appraisals.

3.3.2 Functions requirement

Functions we mainly required an updated version library and VM.

3.3.3 Performance and security requirement

For user security and non-encryption work we recommend user to continuously changing his Ip of VM.

3.3.4 Look and Feel Requirements

We, for making the tool looks professional gives it some good checkpoints tool logo and help context. Where user can see how to work.

3.4 Summary

For quick results it is suggested to go with Linux Distros because it has frequently updated requirements to python scripts. As we well know with the fact that Linux command line environment is very much strong and supported with all the updated resources, modules and libraries. Otherwise on running in windows at some time it may face error because of library.

CHAPTER - 4

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal

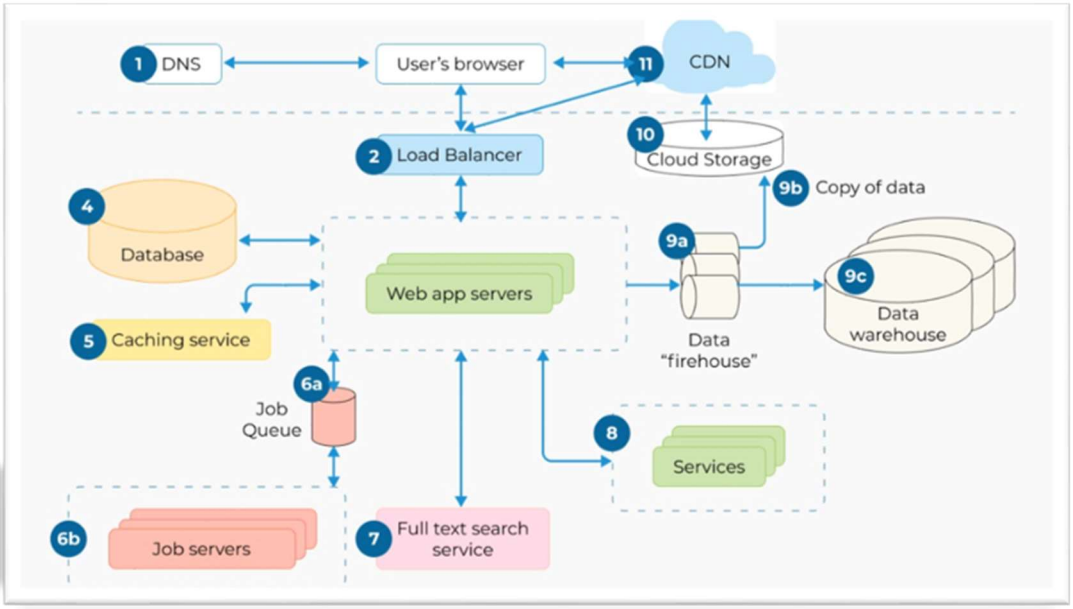
Built a common platform for all the vulnerability to be executed so we used python programming and added different modules in it. Like burp suite Nmap, nikto, dnsmap, amass and approximate 40 tools. It consist7000+ modulation and autosave result will be performed. In case of some of the tools that mentioned above not present even that it will perform rapid scan. To show the vulnerability threat level (Low, medium or high).

4.2 Functional modules design and analysis

```
import sys
import socket
import subprocess
import os
import time
import signal
import random
import string
import threading
import re
from urlparse import urlsplit
```

4.3 Software Architectural designs

It is a comprehensive framework that describes its form and a structure its components and how they fit together. Architectural design is a software component that can be something as simple as program module, but it can also be extended to include database and Middleware that enables the configuration of a network client and server.



4.4 Subsystem services

We provided following services during the scan:

Tool Responses (Begins) [Responses + Severity (c - critical | h - high | m - medium | l - low | i - informational) + Reference for Vuln Definition and Remediation]

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

Vulnerabilities and Remediation/tool fix

Shuffling Scan Order

4.5 User Interface designs

Interface design creates an effective communication medium between a human and a computer. In this project it's the communication between Administrators and station in charge design. Since this project requires database, the client machine requires proper connection with the server machine.

4.6 Summary

It will offer to the organization a better understanding of its assets, security flaws and overall risk, reducing the likelihood that a cybercriminal will breach its systems and catch the business off guard.

CHAPTER - 5

TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Outline

We used certain techniques for implementing our plan in python scripts. To do this we have imported following libraries import sys, socket, subprocess, os, time, signal, random, string, threading, re and urlsplit for making it to do vulnerability assessment. We have included database working of different modules in it like burp suite, dnsmap, nikto, Nmap, amass, etc. It has 7000+ modulation and autosave result.
The program first check for host, if host is down then remote router reports that it has no route to the destination, while using the modules recognition added and after that the same will check for response and starts scans using ack and TCP. It will show results it will show that level and remediation vulnerability type.

5.2 Technical coding and code solutions

Classifies the Vulnerability's Severity and links the vulnerability with threat level and remediation database to the Help Context. Now, Initializing the idle loader/spinner class and the command that is used to initiate the tool (with parameters and extra params).

Tool Responses (Begins) [Responses + Severity (c - critical | h - high | m - medium | l - low | i - informational) + Reference for Vuln Definition and Remediation]
Tool Status (Response Data + Response Code (if status check fails and you still got to push it + Legends + Approx. Time + Tool Identification + Bad Responses)

- Vulnerabilities and Remediation
- Shuffling Scan Order (starts)
- report and documentation phase

5.3 Working Layout of Forms

On running the program file using ". /scan.py (or) python scan.py" command in terminal will show the tool logo and some codes to implement and information about the level of treat by color dots. And several scans start the level of threat and remediation are shown. By the end of scan will get total vulnerability processed, skipped and detected. The same file also available in the file section where we have saved the scan.py file.

5.4 Prototype submission

We showed our second final prototype in Review –2 (26/11/2021) was working well. And the same above-mentioned process happening well planned.

5.5 Test and validation

We tested and validated our python scripts code in kali Linux terminal where it is working fine. While using “python scan.py”. We are getting the output as well. Also, we have done many websites testing vulnerability scanning to its mitigation.

5.6 Performance Analysis

While going through scanning part its takes time according to web service type suppose a website where it has inside subpages and more components in it so in this case scanning time may increase. All types of vulnerabilities SQL Injection, Cross Site Scripting, Broken Authentication and Session Management, Insecure Direct Object References, Cross Site Request Forgery, Security Misconfiguration, Insecure Cryptographic Storage, Failure to restrict URL Access, Insufficient Transport Layer Protection, Unvalidated Redirects and Forwards and many more are detected well. There are no defects in final design of the script by which the scanning process differ.

5.7 Summary

Overall, the script works as automated tool as a red team works to find the loopholes of their respected services. We successfully created a platform for scanning with minimum wastage of time.

CHAPTER - 6

PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

To scan for security loopholes in Web-based application. It will save the time of the organization by scan all type of vulnerabilities in a web application in minimum amount of time. It will also help in mitigate the vulnerability as we are also providing the solution for mitigation. It will help penetration testing.

6.2 key implementations in outline of the System

First, we will the clone the code from the GitHub repository. After that install the tool. Run the command python scan.py. It will show the help menu of the tool. For scanning the website run the command./scan.py <domain name> It will start the scanning process •After finishing the result it will save the result separate file in the same directory.

6.3 Significant project outcomes

The project provides much security. The simplicity and friendliness are the advantages of this project. The tool is made user friendly to the maximum so that anyone can run the tool. This project manages all the details without any risk. All the objectives were met with satisfaction. The performance of the tools found to be satisfactory.

6.4 Project applicability on Real-world applications

In real world where there is software, there are bugs –including security issues. With limited information available at all stages of the development lifecycle, the only practical way to test the security of an entire web application is from the outside in. Static code testing is a great way to minimize the number of vulnerabilities that are introduced in new code, but once a large application is up and running with all its modules, dependencies, and configuration parameters, nobody can be completely sure what's going on. There are many complex web applications which is why testing is a practical necessity.

6.5 Inference

It will provide security analyst with all the necessary security issues and its solution to prevent by the hackers. It provides the user with all the necessary privileges to access and modify the data intended for them. It doesn't entirely replace the existing system but it mostly automize the Scanning Process and all the data used.

CHAPTER-7

CONCLUSIONS AND RECOMMENDATION

7.1 Outline

In the end we successfully developed the tool that is useful for the organization in order to scan web application and discover the vulnerabilities that are present in there system so that they can take further action to mitigate it. For testing purpose, we tested secure website to most vulnerable website such as DVWA to check whether it is working fine or not and we found positive result.

7.2 Limitation/Constraints of the System

The project objective has to be achieved pertaining to the Time Constraint and Monetary constraint applied in accordance with the defined functionality of the system. We cannot think there is such limitation. But, yes at the time detecting new type of treat where user is necessary to testing manually then that could be one limitation suppose Log4j CVE2021 –4428 here some extra action taken from user side. Therefore, it may be a limitation for our tool.

7.3 Future Enhancements

However, features that are not included in the system can be considered as future enhancements. The limiting areas of the project contributing for enhancement thus are as follows, namely,

- ✓ We can plan to include newly generated CVE by automated tools script.
- ✓ We can plan to make it more performance strategic.
- ✓ We can plan to escape from time bonding while scanning can be lower by introducing some more variables.
- ✓ We can plan make an API version of it.
- ✓ We can plan to further enhancement in the code if required.

7.4 Inference

As a result of our work, we came with a conclusion as our tool "Redscan- The Red Team Web Application Scanner" is successfully working all the process like scan many types of vulnerabilities that is present in a web application and tell the threat level of that vulnerability and provide the solution for mitigation. It will offer to the organization a better understanding of its assets, security flaws and overall risk, reducing the likelihood that a cybercriminal will breach its systems and catch the business off guard. It overcomes the many limitations incorporated in the attendance.

- → Easy implementation environment
- → Generate report flexibly

Here we are attaching the link our project python code for scan.py file. Where we can copy code and use it Linux Debian.

Webtor/scan.py at main · edudsph/Webtor · GitHub

Appendix – A

Functionality Testing

Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing, etc.

Check out all the links:

- ✓ Test the outgoing links from all the pages to the specific domain under test.
- ✓ Test all internal links.
- ✓ Test links jumping on the same page.
- ✓ Test links are used to send emails to admin or other users from web pages.
- ✓ Test to see if there are any orphan pages.
- ✓ Finally, link checking includes, check for broken links in all the above-mentioned links.

Test forms on all pages:
Forms are an integral part of any website. Forms are used for receiving information from users and to interact with them. So, what should be checked in these forms?

- ✓ First, check all the validations on each field.
- ✓ Check for default values in the fields.
- ✓ Wrong inputs in the forms to the fields in the forms.
- ✓ Options to create forms, if any, form delete, view or modify the forms.

Let's take an example of the search engine project currently I am working on. In this project we have advertiser and affiliate signup steps. Each sign-up step is different but it's dependent on the other steps.

So the sign up flow should be executed correctly. There are different field validations like email Ids, User financial info validations, etc. All these validations should get checked in manual or automated web testing.

Cookie Testing:

Cookies are small files stored on the user machine. These are basically used to maintain the session – mainly the login sessions. Test the application by enabling or disabling the cookies in your browser options.

Test if the cookies are encrypted before writing to the user machine. If you are testing session cookies (i.e., cookies that expire after the session ends) check for login sessions and user stats after the session ends. Check the effect on application security by deleting the cookies. (I will soon write a separate article on cookie testing as well)

Validate your HTML/CSS:

If you are optimizing your site for Search engines then HTML/CSS validation is the most important one. Mainly validate the site for HTML syntax errors. Check if the site is crawlable to different search engines.

Database Testing:

Data consistency is also very important in a web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality.Check if all the database queries are executed correctly, data is retrieved and also updated correctly. More on database testing could be a load on DB, we will address this in web load or performance testing below.

In testing the functionality of the websites, the following should be tested:

Links
i. Internal Links
ii. External Links
iii. Mail Links
iv. Broken Links

Forms
i. Field validation
ii. Error message for wrong input
iii. Optional and Mandatory fields

Database
Testing will be done on database integrity.

Appendix – B

Usability Testing

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

- Ease of learning
- Navigation
- Subjective user satisfaction
- General appearance

Test for Navigation:
Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

Usability Testing includes the following:

- ✓ The website should be easy to use.
- ✓ The instructions provided should be very clear.
- ✓ Check if the instructions provided are perfect to satisfy its purpose.
- ✓ The main menu should be provided on each page.
- ✓ It should be consistent enough.

Content Checking:
Content should be logical and easy to understand. Check for spelling errors. The usage of dark colors annoys the users and should not be used in the site theme.
You can follow some standard colors that are used for web pages and content building. These are the commonly accepted standards like what I mentioned above about annoying colors, fonts, frames, etc.

Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes.
These are some of the basic important standards that should be followed in web development. Your task is to validate everything for UI testing.

Other user information for user help:
Like the search option, the sitemap also helps with files, etc. The sitemap should be available with all the links on websites with a proper tree view of navigation. Check for all links on the sitemap.
“Search in the site” option will help users to find content pages that they are looking for easily and quickly. These are all optional items and if present they should be validated.

References

[1] Allen, F. E, Cocke, J. "A Program Data Flow Analysis Procedure." Communications of the ACM, 19(3):13147,

March 1976.

[2] Andrews, G. R., Reitman, R. P. "An Axiomatic Approach to Information Flow in Programs." ACM Transactions on

Programming Languages and Systems, 2(1), 56-76, 1980.

[3] Ashcraft, K., Engler, D. "Using Programmer-Written Compiler Extensions to Catch Security Holes." In Proceedings

of the 2002 IEEE Symposium on Security and Privacy, pages 131-147, Oakland, California, 2002.

[4] Augustin, L., Bressler, D., Smith, G. "Accelerating Software Development through Collaboration." In Proceedings

of the 24th International Conference on Software Engineering, pages 559-563, Orlando, Florida, May 19-25, 2002.

[5] Auronen, L. “Tool-Based Approach to Assessing Web Application Security.” Helsinki University of Technology,

Nov 2002.

[6] Ball, T., Rajamani, S. K., "Automatically Validating Temporal Safety Properties of Interfaces." In Proceedings of

the 8th International SPIN Workshop on Model Checking of Software, pages 103-122, volume LNCS 2057,

Toronto, Canada, May 19-21, 2001. Springer-Verlag.

[7] Balzer, R., “Assuring the safety of opening email attachments.” In: DARPA Information Survivability Conference

& Exposition II, 2, 257-262, 2001.

[8] Banatre, J. P., Bryce, C., Le Metayer, D. "Compile-time Detection of Information Flow in Sequential Programs." In

Proceedings of the Third European Symposium on Research in Computer Security, pages 55-73, volume LNCS

875, Brighton, UK, Nov 1994. Springer-Verlag.

[9] Banerjee, A., Naumann, D.A. "Secure Information Flow and Pointer confinement in a Java-Like Language." In:

Proceedings of the 15th Computer Security Foundations Workshop, pages 239-253, Nova Scotia, Canada, 2002.

[10] Barth, J. M. "A Practical Interprocedural Data Flow Analysis Algorithm." Communications of the ACM, 21(9):724-

736, 1978.

[11] Bell, D. E., La Padula, L. J. "Secure Computer System: Unified Exposition and Multics Interpretation." Tech Rep.

ESD-TR-75-306, MITRE Corporation, 1976.

[12] Benedikt M., Freire J., Godefroid P., “VeriWeb: Automatically Testing Dynamic Web Sites.” In: Proceedings of the

11th International Conference on the World Wide Web (Honolulu, Hawaii, May 2002)

****