Future Institute of Engineering and Management

PROJECT REPORT ON

"VULNERABILITYSCANNER ON CROSS SITE SCRIPTING USING PYTHON"

SUBMITTED BY–

SAYANTAN DUTTA

MAKAUT ROLLNO–14800118054

CLASS ROLLNO–18/CSE/66

TEAM MEMBERS-

SASWATI SAMANTA

AVINNA CHAKROBORTY

DINESH GHORA

SAYAN GHOSH

SUBHOJEET KARMAKAR

UNDER THE GUIDANCE OF

Mr.Ranit Mishra

and

Mr.Ritaban Das

National Institute for Industrial Training

CERTIFICATE

This is to certify that the project entitled "**VULNERABILITY SCANNER ONCROSS SITE SCRIPTING USING PYTHON**" submitted by **SAYANTAN DUTTA** in partial fulfillments for the requirements for the award of Bachelor of Technology Degree in Computer Science & Engineering at Future Institute of Engineering & Management, Kolkata is an authentic work carried out by her under our supervision and guidance.

Mr. Ranit Mishra & Mr. Ritaban Das

Department of Cyber security

National Institute for Industrial Training

Date: 05-05-2021

Kolkata, West Bengal

ACKNOWLEDGEMENT

The successful completion of this project marks the beginning of an ever – going learning experience of converting ideas and concepts into real life, practical system. At thesame time, it has given me confidence to work in professional setup. I feel the experience gained during the project will lead me to gain the bright prospect in the future. First of all, I would like to make my deepest appreciation and gratitude to **Mr. Ritaban Das and Mr. Ranit Mishra** for their invaluable guidance, constructive criticism and encouragement during the course of this project. Without their active support and continuous guidance, it would have been difficult for me to complete this project.

I am also thankful to **Mr. Tapas Roy**, **Head of the Department**, **Future Institute of Engineering & Management** for helping me whenever I needed and he gave right direction towards completion of project.

SAYANTAN DUTTA
MAKAUTRollNo**14800118054**
Class Roll No–**18/CSE/66**
Department of Computer Science
& Engineering
Date: 05-05-2021
FUTURE INSTITUTE OF
ENGINEERING &
MANAGEMENT
KOLKATA–700150, WEST BENGAL

Future Institute of Engineering and Management

CONTENTS

No.	TOPICS	PAGENO
1	Vulnerability Scanner 1.1 Definition 1.2 How Vulnerability Scanners Secure Our Network? 1.3 What Are the Types of Vulnerability Scanners? 1.4 Key Features of The Best Vulnerability Scanning tools	5-6
2.	Cross Site Scripting (XSS) 2.1 Howdoes XSS work? 2.2 Types of XSS Attacks 2.3 What can XSS be used for? 2.4 Impact of XSS Vulnerabilities 2.5 How to prevent XSS attacks?	7-9
3.	Development of Vulnerability Scanner 3.1 Language used to develop Vulnerability Scanner 3.2 Tools used to develop Vulnerability Scanner 3.3 Source code 3.4 Output	10-15
4.	Benefits of Using the Best Vulnerability Scanning Tools	16
5.	Conclusion	16

VULNERABILITY SCANNER

DEFINITION :

Vulnerability scanning, also commonly known as 'vuln scan,' is an automated process of proactively identifying network, application, and security vulnerabilities. Vulnerability scanning is typically performed by the IT department of an organization or a third-party security service provider. This scan is also performed by attackers who try to find points of entry into your network.

Vulnerability scanners are automated tools that allow organizations to check if their networks, systems and applications have security weaknesses that could expose them to attacks. Vulnerability scanning is a common practice across enterprise networks and is often mandated by industry standards and government regulations to improve the organization's security posture.

There are many tools and products in the vulnerability scanning space that cover different types of assets and offer additional features that help companies implement a complete vulnerability management program — the combined processes related to identifying, classifying and mitigating vulnerabilities.

HOW VULNERABILITY SCANNERS SECURE OUR NETWORK?

The foundation of a vulnerability scanner rests upon the vendor's vulnerability database, which contains details on every known security vulnerability. The security research team of vendors frequently updates the database with information on new vulnerabilities. The vulnerability scanners typically begin by executing a network asset inventory, which gathers information from an existing asset management system and identifies each system running on the network. Then it performs a baseline scan to identify applications and the operating system running on that host to detect possible vulnerabilities.

Once the vulnerability scan completes, the results may overwhelm the security professionals with thousands of configuration flaws. The real power of the top vulnerability scanning tools resides in its capability to support security teams in sorting through the chaos of information and prioritize the actions, which have the highest impact on the company's security posture. They do this by combining the knowledge about the severity and impact of the discovered security weaknesses, the system's priority as well as compliance issues if any exist in the given environment.

This vulnerability scanning best practice is what transforms a simple scanner into the best vulnerability management platform. The scanner helps you prioritize the process as per the risk level and suggest remediation to alleviate the risks.

WHAT ARE THE TYPES OF VULNERABILITY SCANNERS?

There are various types of vulnerability scanners and cover off a wide range of attack scenarios with their powerful features. A hacker could enter your network by exploiting web server vulnerabilities or through unpatched software. As such, different attack vectors could be addressed by different vulnerability scanner features and use-cases. It is worth considering the potential risks to your businesses and choose the best vulnerability scanning tools, which are suitable. The three main types of vulnerability scanners are:

- ***Network-Based Vulnerability Scanners*** *– As its name implies, network-based vulnerability scanners scan the system across the network, by sending inquiries searching for all open services and ports, and then examine each service further to identify known vulnerabilities and configuration weakness.*
- ***Agent-Based Vulnerability Scanners*** *– This type of vulnerability scanning tool involves installing a lightweight scanner on each machine, runs vulnerability scan locally on the device and reports the results back to the server.*
- ***Web Application Vulnerability Scanners*** *– This specialized type of vulnerability scanners focuses on finding security gaps in websites and web applications.*
- ***Database Scanners -****Database vulnerability scanners identify the weak points in a database so as to prevent malicious attacks*

WE CAN ALSO CLASSIFY THE VULNERABILITY SCANNERS INTO THESE TYPES:

External vulnerability scanner vs internal vulnerability scanners *– External scanners perform the vulnerability scan from outside the network whereas internal scanners aid you to strengthen the security scan from inside of your network.*

Unauthenticated vulnerability scans vs authenticated vulnerability scan *– Unauthenticated vulnerability scans offer visibility into what a hacker could gain access to without obtaining login credentials whereas authenticated scans perform vulnerability scanning with privileged credentials and dig deeper into your network to detect threats around malware, weak passwords, and configuration is*

: KEY FEATURES OF THE BEST VULNERABILITY SCANNING TOOLS :

- *Asset Discovery*
- ***Scanning*** *Capabilities*
- *Policy Compliance*
- *Action Plans &* ***Vulnerability*** *Management*
- *Overall Risk Score &* ***Vulnerability*** *Reporting*

OUR PROJECT MAINLY FOCUSES ON

"VULNERABILITY SCANNER ON CROSS SITE SCRIPTING".

: CROSS-SITE SCRIPTING (XSS):

Cross-site Scripting (XSS) is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. The actual attack occurs when the victim visits the web page or web application that executes the malicious code. The web page or web application becomes a vehicle to deliver the malicious script to the user's browser. Vulnerable vehicles that are commonly used for Cross-site Scripting attacks are forums, message boards, and web pages that allow comments.

A web page or web application is vulnerable to XSS if it uses unsensitized user input in the output that it generates. This user input must then be parsed by the victim's browser. XSS attacks are possible in VBScript, ActiveX, Flash, python and even CSS. However, they are most common in JavaScript.

HOW DOES XSS WORK?

Cross Site Scripting attack means sending and injecting malicious code or script. Malicious code is usually written with client-side programming languages such as JavaScript, HTML, VBScript, python,Flash, etc.
This attack can be performed in different ways. Depending upon the type of XSS attack, the malicious script may be reflected on the victim's browser or stored in the database and executed every time, when the user calls the appropriate function.

The main reason for this attack is inappropriate user's input validation, where malicious input can get into the output. A malicious user can enter a script, which will be injected into the website's code. Then the browser is not able to know if the executed code is malicious or not.

: TYPES OF XSS ATTACKS:

The attack can be classified into the following major categories –

- *Persistent or stored XSS*
- *Non-persistent or reflected XSS*

Persistent or stored XSS

In this kind of XSS attack, an attacker injects a script, referred to as the payload, that is permanently stored on the target web application, for example within a database. This is the reason; it is called persistent XSS attack. It is actually the most damaging type of XSS attack. For example, a malicious code is inserted by an attacker in the comment field on a blog or in the forum post.

Blind Cross-site Scripting:

Blind Cross-site Scripting is a form of persistent XSS. It generally occurs when the attacker's payload saved on the server and reflected back to the victim from the backend application. For example, in feedback forms, an attacker can submit the malicious payload using the form, and once the backend user/admin of the application will open the attacker's submitted form via the backend application, the attacker's payload will get executed. Blind Cross-site Scripting is hard to confirm in the real-world scenario but one of the best tools for this is XSS Hunter.

Non-persistent or reflected XSS

It is the most common type of XSS attack in which the attacker's payload has to be the part of the request, which is sent to the web server and reflected, back in such a way that the HTTP response includes the payload from the HTTP request. It is a non-persistent attack because the attacker needs to deliver the payload to each victim. The most common example of such kinds of XSS attacks are the phishing emails with the help of which attacker attracts the victim to make a request to the server which contains the XSS payloads and ends-up executing the script that gets reflected and executed inside the browser.

Other Types of XSS Vulnerabilities

In addition to Stored and Reflected XSS, another type of XSS, ***DOM Based XSS*** *was identified by Amit Klein in 2005.*

DOM XSS

This type of attack occurs when the DOM environment is being changed, but the client-side code does not change. When the DOM environment is being modified in the victim's browser, then the client-side code executes differently.

WHAT CAN XSS BE USED FOR?

An attacker who exploits a cross-site scripting vulnerability is typically able to:

- *Impersonate or masquerade as the victim user.*
- *Carry out any action that the user is able to perform.*
- *Read any data that the user is able to access.*
- *Capture the user's login credentials.*
- *Perform virtual defacement of the web site.*
- *Inject trojan functionality into the web site.*

: IMPACT OF XSS VULNERABILITIES:

The actual impact of an XSS attack generally depends on the nature of the application, its functionality and data, and the status of the compromised user. For example:

- *In a brochureware application, where all users are anonymous and all information is public, the impact will often be minimal.*
- *In an application holding sensitive data, such as banking transactions, emails, or healthcare records, the impact will usually be serious.*
- *If the compromised user has elevated privileges within the application, then the impact will generally be critical, allowing the attacker to take full control of the vulnerable application and compromise all users and their data.*

: HOW TO PREVENT XSS ATTACKS:

Preventing cross-site scripting is trivial in some cases but can be much harder depending on the complexity of the application and the ways it handles user-controllable data.

In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures:

- ***Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.*
- ***Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.*
- ***Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.*
- ***Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.*

DEVELOPMENT OF VULNERABILITY SCANNER

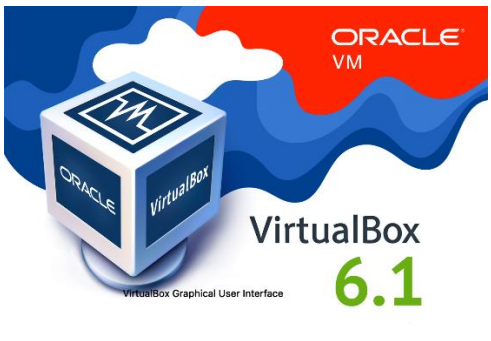
Language Used in Development ofVulnerabilityScanner

PYTHON – Python is an interpreted, object oriented high level and general-purpose programming language with dynamic semantics. It's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.
In our project development we have developed Vulnerability Scanner with the help of python3.

Python

Tools Used in Development of Vulnerability Scanner

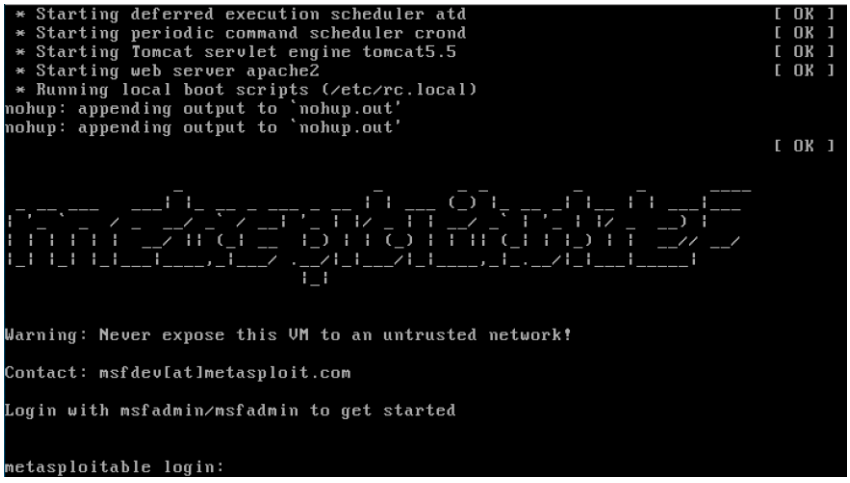
- **Oracle VM VirtualBox** – Oracle VM VirtualBox is a free and open-source hosted hypervisor for x86 virtualization, developed by Oracle Corporation. Created by Innotek, it was acquired by Sun Microsystems in 2008, which was in turn acquired by Oracle in 2010.It is designed to run virtual machines on our physical machine without reinstalling our OS that is running on a physical machine. An OS and applications installed inside a VM "think" that they are running on a regular physical machine since emulated hardware is used for running VMs on VirtualBox. Virtual machines are isolated from each other and from the host operating system. Thus, we can perform our tests in isolated virtual machines without any concerns of damaging our host operating system or other virtual machines.



- **Kali Linux**- Kali Linux *(formerly known as Backtrack Linux)* is an open-source, Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing. Kali Linux contains several hundred tools targeted towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering. Kali Linux is a multiplatform solution, accessible and freely available to information security professionals and hobbyists.

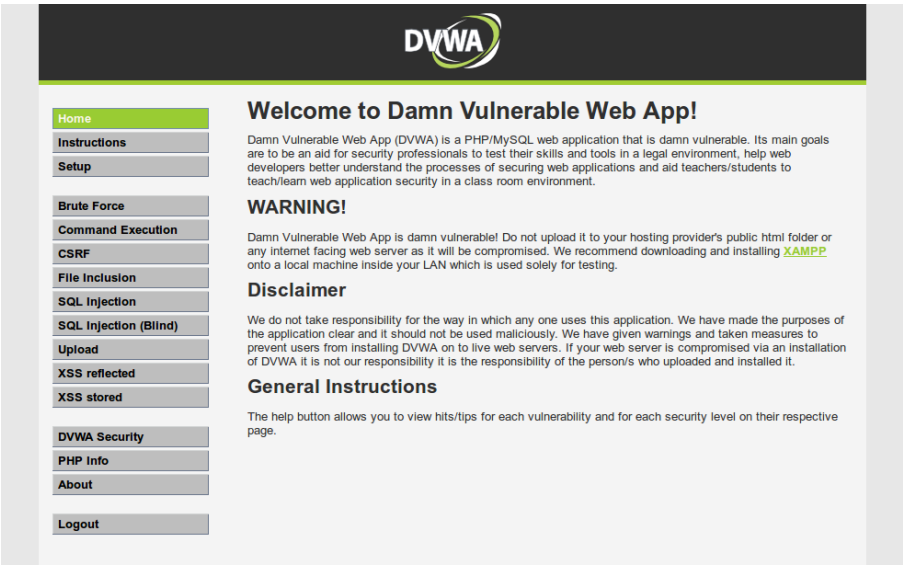
  Kali Linux was released on the 13th March 2013 as a complete, top-to-bottom rebuild of BackTrack Linux, adhering completely to Debian development standards.

- **Metasploitable 2**- The Metasploitable virtual machine is an intentionally vulnerable version of Ubuntu Linux designed for testing security tools and demonstrating common vulnerabilities. Version 2 of this virtual machine is available for download and ships with even more vulnerabilities than the original image. This virtual machine is compatible with VMWare, VirtualBox, and other common virtualization platforms. By default, Metasploitable's network interfaces are bound to the NAT and Host-only network adapters, and the image should never be exposed to a hostile network.



- **DVWA**- DVWA is a DAMM VULNERABLE WEB APP coded in *PHP/MYSQL*. In this app security professionals, ethical hackers test their skills and run these tools in a legal environment. It also helps web developer better understand the processes of securing web applications and teacher/students to teach/learn web application security in a safe environment.

  The aim of DVWA is to practice some of the *most common web vulnerability*, with various *difficulties levels*. Every vulnerability has four different security levels, low, medium, high and impossible. The security levels give a challenge to the 'attacker' and also shows how each vulnerability can be counter measured by secure coding.



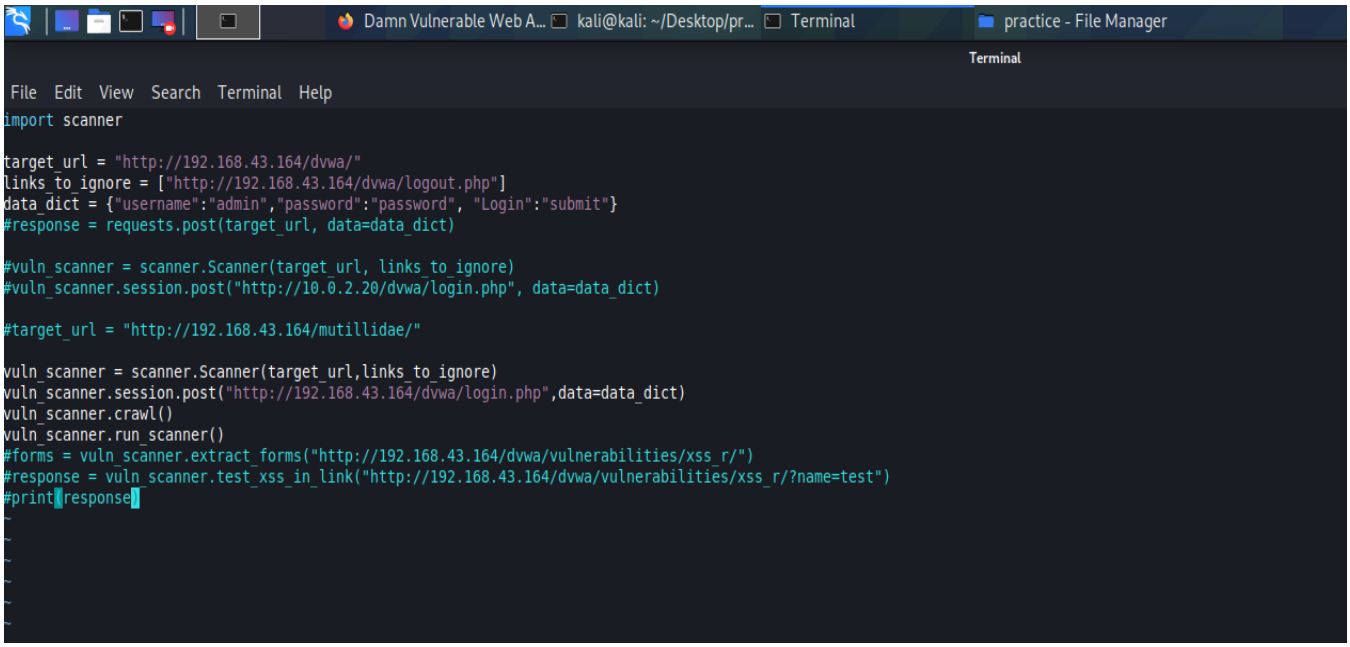
- **Mutillidae**-Mutillidae is a free, open-source web application provided to allow security enthusiest to pen-test and hack a web application. Mutillidae contains dozens of vulnerabilities and hints to help the user exploit them; providing an easy-to-use web hacking environment deliberately designed to be used as a hack-lab for security enthusiast, classroom labs, and vulnerability assessment tool targets. Mutillidae has been tested/attacked with Cenzic Hailstorm ARC, W3AF, SQLMAP, Samurai WTF, Backtrack, HP Web Inspect, Burp-Suite, NetSparker Community Edition, and other tools. If you would like to practice pen-testing/hacking a web application by exploiting cross-site scripting, sql injection, response-splitting, html injection, javascript injection, clickjacking, cross frame scripting, forms-caching, authentication bypass, or many other vulnerabilities, then Mutillidae is for you. The current version of Mutillidae, code named "NOWASP Mutillidae 2.x", wasdeveloped by Jeremy Druin aka webpwnized.

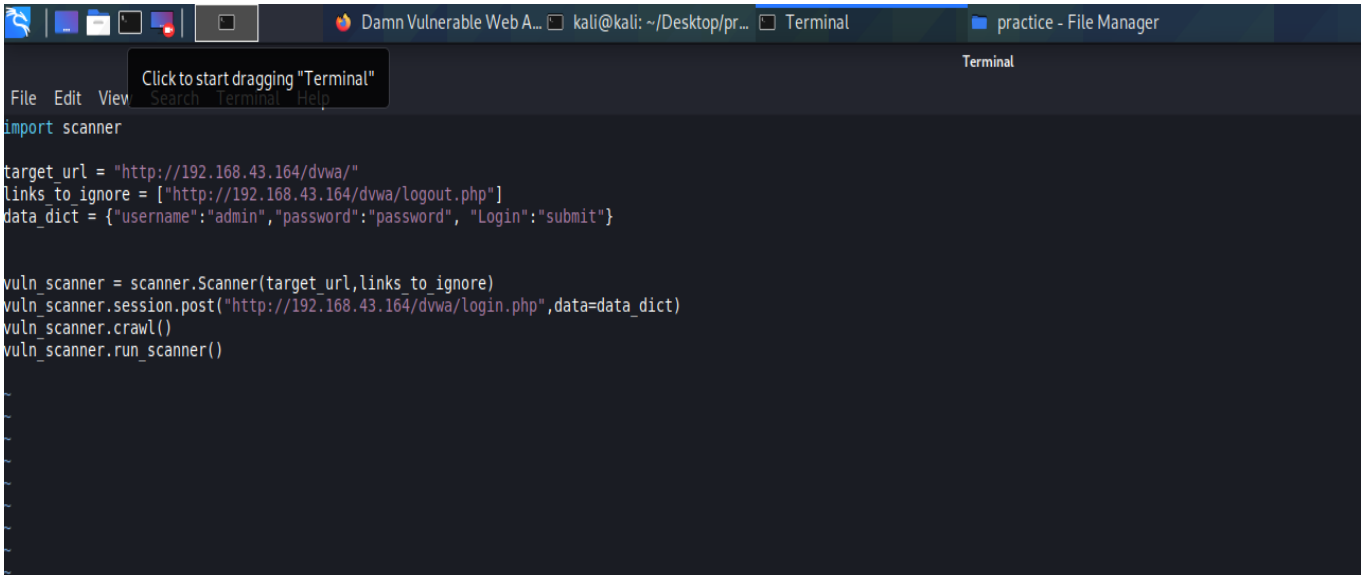
Source code used

The code has been divided into 2 main parts: -

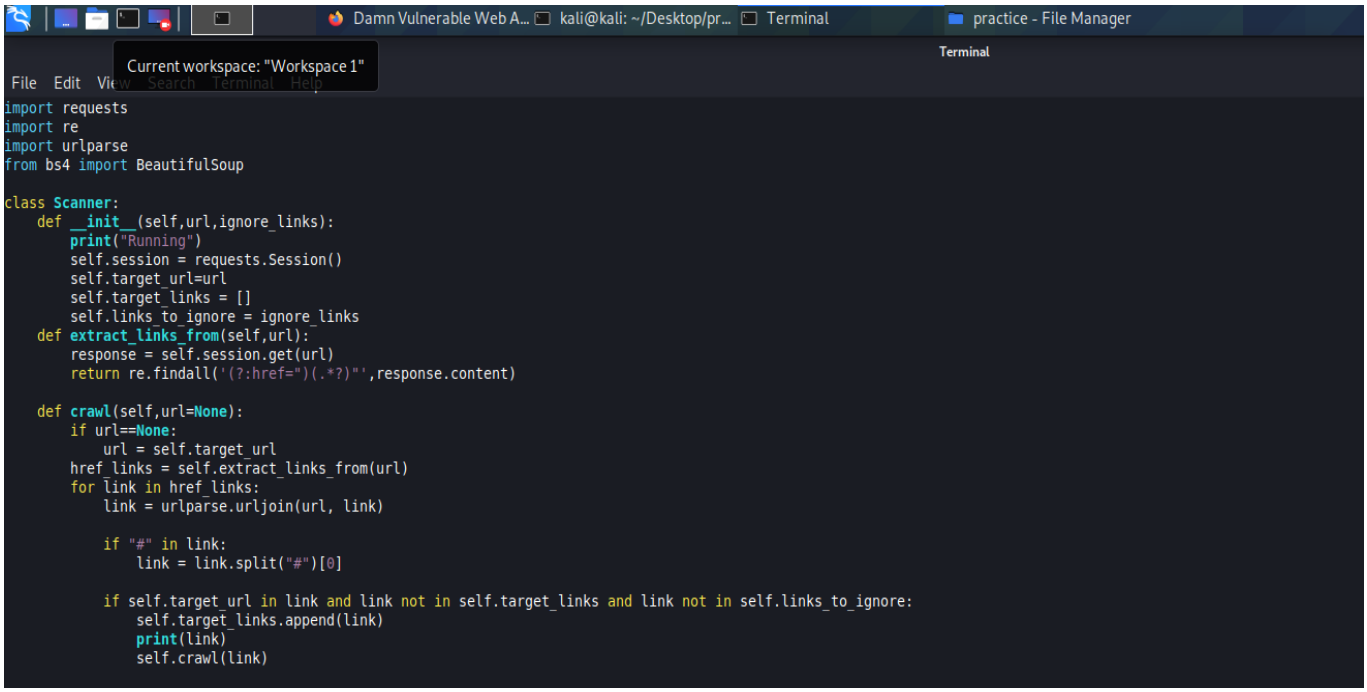
- ❖ vulnerability scanner
- ❖ scanner

1) vulnerability scanner





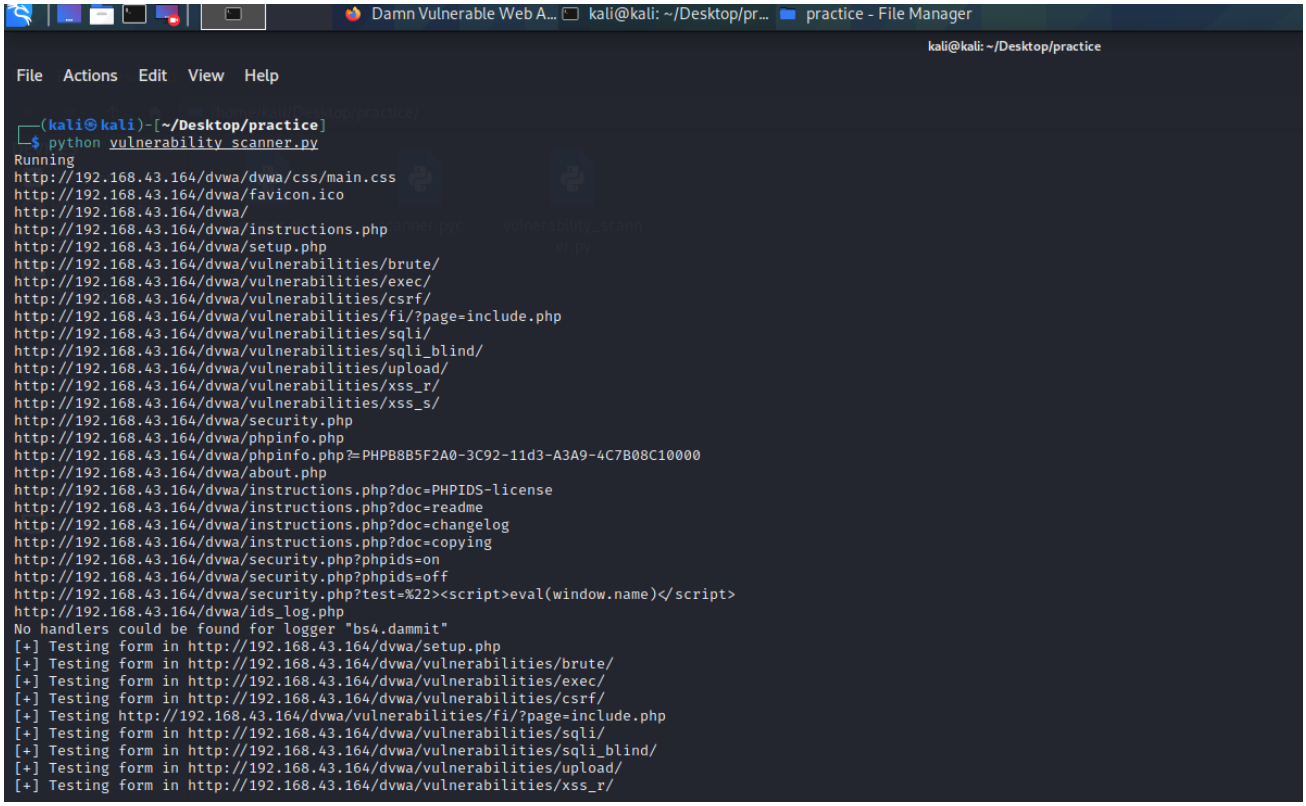
2) Scanner

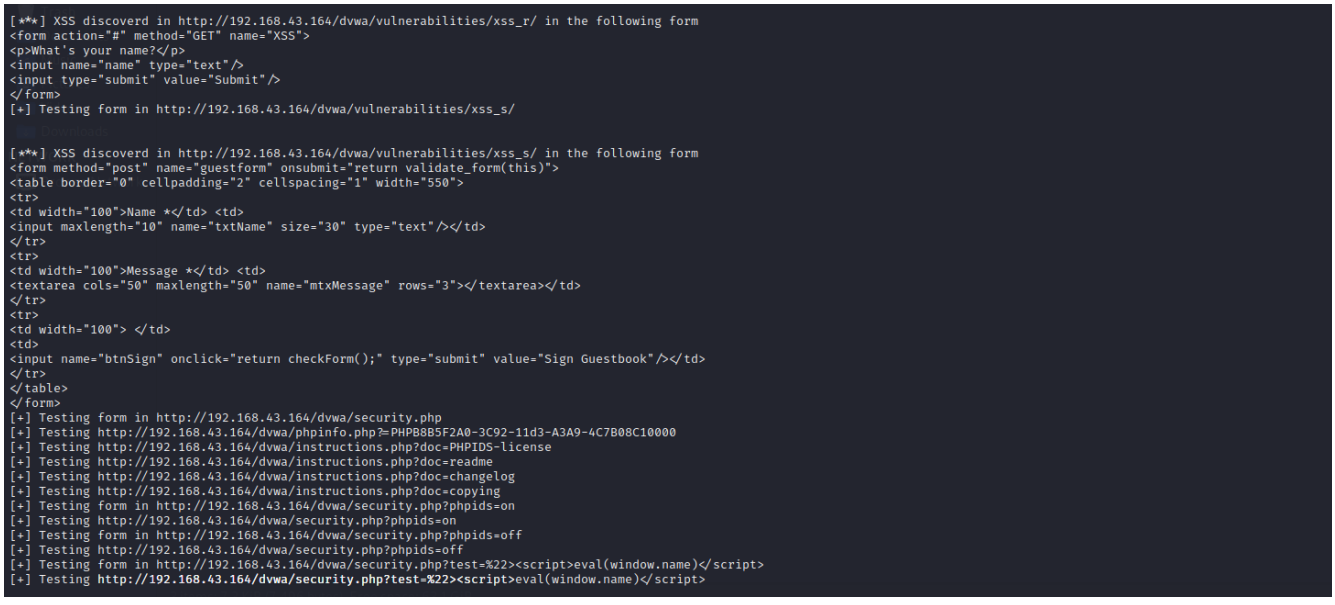



```
    def test_xss_in_link(self, url):
        xss_test_script = "<sCript>alert('test')</scriPt>"
        url = url.replace("=","=" + xss_test_script)
        response = self.session.get(url)
        if xss_test_script in response.content:
            return True

    def test_xss_in_form(self,form,url):
        xss_test_script = " <sCript>alert('test')</scriPt>"
        response = self.submit_form(form,xss_test_script,url)
```

Output


Damn Vulnerable Web A...
kali@kali: ~/Desktop/pr...
practice - File Manager
kali@kali: ~/Desktop/practice
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/practice]
$ python vulnerability_scanner.py
Running
http://192.168.43.164/dvwa/dvwa/css/main.css
http://192.168.43.164/dvwa/favicon.ico
http://192.168.43.164/dvwa/
http://192.168.43.164/dvwa/instructions.php
http://192.168.43.164/dvwa/setup.php
http://192.168.43.164/dvwa/vulnerabilities/brute/
http://192.168.43.164/dvwa/vulnerabilities/exec/
http://192.168.43.164/dvwa/vulnerabilities/csrf/
http://192.168.43.164/dvwa/vulnerabilities/fi/?page=include.php
http://192.168.43.164/dvwa/vulnerabilities/sqli/
http://192.168.43.164/dvwa/vulnerabilities/sqli_blind/
http://192.168.43.164/dvwa/vulnerabilities/upload/
http://192.168.43.164/dvwa/vulnerabilities/xss_r/
http://192.168.43.164/dvwa/vulnerabilities/xss_s/
http://192.168.43.164/dvwa/security.php
http://192.168.43.164/dvwa/phpinfo.php
http://192.168.43.164/dvwa/phpinfo.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000
http://192.168.43.164/dvwa/about.php
http://192.168.43.164/dvwa/instructions.php?doc=PHPIDS-license
http://192.168.43.164/dvwa/instructions.php?doc=readme
http://192.168.43.164/dvwa/instructions.php?doc=changelog
http://192.168.43.164/dvwa/instructions.php?doc=copying
http://192.168.43.164/dvwa/security.php?phpids=on
http://192.168.43.164/dvwa/security.php?phpids=off
http://192.168.43.164/dvwa/security.php?test=%22><script>eval(window.name)</script>
http://192.168.43.164/dvwa/ids_log.php
No handlers could be found for logger "bs4.dammit"
[+] Testing form in http://192.168.43.164/dvwa/setup.php
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/brute/
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/exec/
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/csrf/
[+] Testing http://192.168.43.164/dvwa/vulnerabilities/fi/?page=include.php
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/sqli/
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/sqli_blind/
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/upload/
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/xss_r/


[**] XSS discoverd in http://192.168.43.164/dvwa/vulnerabilities/xss_r/ in the following form
<form action="#" method="GET" name="XSS">
What's your name?
<input name="name" type="text"/>
<input type="submit" value="Submit"/>
</form>
[+] Testing form in http://192.168.43.164/dvwa/vulnerabilities/xss_s/
[**] XSS discoverd in http://192.168.43.164/dvwa/vulnerabilities/xss_s/ in the following form
<form method="post" name="guestform" onsubmit="return validate_form(this)">
<table border="0" cellpadding="2" cellspacing="1" width="550">
<tr>
<td width="100">Name *</td> <td>
<input maxlength="10" name="txtName" size="30" type="text"/></td>
</tr>
<tr>
<td width="100">Message *</td> <td>
<textarea cols="50" maxlength="50" name="mtxMessage" rows="3"></textarea></td>
</tr>
<tr>
<td width="100"> </td>
<td>
<input name="btnSign" onclick="return checkForm();" type="submit" value="Sign Guestbook"/></td>
</tr>
</table>
</form>
[+] Testing form in http://192.168.43.164/dvwa/security.php
[+] Testing http://192.168.43.164/dvwa/phpinfo.php?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000
[+] Testing http://192.168.43.164/dvwa/instructions.php?doc=PHPIDS-license
[+] Testing http://192.168.43.164/dvwa/instructions.php?doc=readme
[+] Testing http://192.168.43.164/dvwa/instructions.php?doc=changelog
[+] Testing http://192.168.43.164/dvwa/instructions.php?doc=copying
[+] Testing form in http://192.168.43.164/dvwa/security.php?phpids=on
[+] Testing http://192.168.43.164/dvwa/security.php?phpids=on
[+] Testing form in http://192.168.43.164/dvwa/security.php?phpids=off
[+] Testing http://192.168.43.164/dvwa/security.php?phpids=off
[+] Testing form in http://192.168.43.164/dvwa/security.php?test=%22><script>eval(window.name)</script>
[+] Testing http://192.168.43.164/dvwa/security.php?test=%22><script>eval(window.name)</script>

BENEFITS OF USING THE BEST VULNERABILITY SCANNING TOOLS

It is significant to find the best vendor who can give you abundant services as well as cover a wide range of security protection. An organization can get the below benefits through choosing the best vulnerability scanners:

- Identify possible security weaknesses before attackers exploit them
- Define the level of security risks, which exist on the network
- Make an inventory of all the systems in the organization which is mandatory for the planning and future upgrades
- Perform a better assessment on what requires to be fixed
- Constant monitoring saves the business time – When comparing the time company spend on addressing huge risk, which is detected too late, little attention every day can be a great time saver
- As it saves time, it should save business money as well. Yes, it does – the best vulnerability scanner saves you from costly security breaches.

CONCLUSION

PURPOSE OR IMPORTANCE OF VULNERABILITY SCANS

Every time a computer connects to the Internet, there is a risk of a hacker taking advantage of some new vulnerability. This needle in the cyber-haystack can wreak havoc on networks and computers. Most disconcerting, these vulnerabilities can cause more than annoying pop-ups. They can worm their way into a network and steal proprietary information and other data critical to the profitability of a business.

Vulnerability scanning is an organized approach to the testing, identification, analysis and reporting of potential security issues on a network. An external scan will mimic how hackers on the Internet can attempt to gain access to a network. An internal scan is run from inside the network. The results can show the path a hacker can take once they have gained access to the network and exactly how much data they could collect.

Vulnerability scanning is a non-destructive form of testing that provides immediate feedback on the health and security of a network. Based on the information provided, the IT team can take direct action to better protect a network and the information housed within it.

References

- ✓ wikipedia.org
- ✓ geeksforgeeks.org
- ✓ javapoint.com
- ✓ irongeek.com
- ✓ kali.org
- ✓ allcovered.com