Smart Bridge
Project Report

Course: Cyber Security and Ethical Hacking

Project Title: WEB APPLICATION PENETRATION TESTING

Submitted By

Devendra Dev - 20BCE2191
Archisman Das - 20BCE2229
Prakhar Goyal - 20BCE2211
Adwyait Pawar - 20BCE2088

DECLARATION

We hereby declare that the project entitled “WEB APPLICATION PENETRATION TESTING” submitted by us to the Smart Bridge, fulfillment of the requirements for the Cyber Security and Ethical Hacking course, is a record of bonafide work carried out by us under the supervision of Smart Bridge faculties. 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 course or degree or diploma of this institute or of any other institute or university.

INDEX

1. INTRODUCTION

1.1 OVERVIEW

Our project focuses on conducting a comprehensive web application penetration testing to assess the security posture of a target web application. Web application penetration testing, also known as ethical hacking or white-box testing, involves simulating real-world attacks to identify vulnerabilities and potential security weaknesses within a web application.

The primary objective of this project is to evaluate the resilience of the web application against malicious activities and uncover any vulnerabilities that could potentially be exploited by attackers. By proactively identifying and addressing these security flaws, we aim to enhance the overall security and protect sensitive information from unauthorized access, data breaches, and potential financial losses.

The project will involve a systematic and methodical approach to the penetration testing process. It will include the following key steps:

- **Scoping:** Identifying the scope and boundaries of the web application to be tested, including its functionality, interfaces, and underlying technologies.

- **Reconnaissance:** Gathering information about the web application and its environment through various techniques, such as web crawling, port scanning, and analyzing publicly available data.

- **Vulnerability Assessment:** Conducting a thorough examination of the web application to identify potential vulnerabilities, including common security misconfigurations, input validation flaws, authentication weaknesses, and other application-specific vulnerabilities.

- **Exploitation:** Attempting to exploit the identified vulnerabilities to determine the extent of their impact and potential risks associated with them. This step involves various techniques like SQL injection, cross-site scripting (XSS), and session hijacking.

- **Post-Exploitation:** Assessing the consequences of successful exploitation, such as gaining unauthorized access, escalating privileges, or exfiltrating sensitive data. This step helps measure the potential impact of a successful attack.

- **Reporting:** Documenting all identified vulnerabilities, their potential risks, and providing actionable recommendations to mitigate the discovered security issues. The report will include detailed findings, supporting evidence, and a prioritized list of recommendations to improve the web application's security.

Throughout the project, we will adhere to ethical guidelines, ensuring that all penetration testing activities are carried out with proper permissions and legal considerations. Our aim is to provide valuable insights and actionable recommendations that empower organizations to proactively secure their web applications, safeguard user data, and maintain trust with their stakeholders.

By performing a thorough web application penetration testing, we will contribute to enhancing the security posture of the tested application, thus reducing the likelihood of successful attacks and potential financial and reputational damage.

1.2 PURPOSE

The purpose of our project on web application penetration testing is to identify and address potential security vulnerabilities within a target web application. By conducting a thorough assessment of the application's security posture, we aim to achieve the following objectives:

- **Identify Vulnerabilities:** The primary purpose of the project is to systematically uncover security vulnerabilities and weaknesses within the web application. By simulating real-world attacks, we can identify vulnerabilities such as code flaws, misconfigurations, and implementation errors that could potentially be exploited by malicious actors.

- **Assess Security Risks:** Through the penetration testing process, we aim to assess the potential risks associated with the identified vulnerabilities. By exploiting these vulnerabilities, we can gauge the extent to which an attacker could compromise the application, gain unauthorized access, or compromise sensitive data. This helps in understanding the potential impact of successful attacks and prioritizing remediation efforts.

- **Enhance Security Measures:** The project serves as a proactive measure to improve the overall security of the web application. By uncovering vulnerabilities and providing actionable recommendations, we enable the development team or organization to address the identified weaknesses, mitigate risks, and strengthen the application's defenses. This helps protect the application and its users from potential attacks and data breaches.

- **Maintain Compliance:** Web application penetration testing plays a crucial role in meeting compliance requirements. Many regulatory frameworks and industry standards, such as Payment Card Industry Data Security Standard (PCI DSS) and General Data Protection Regulation (GDPR), mandate regular security assessments. By conducting penetration testing, organizations can demonstrate their commitment to compliance and ensure that appropriate security measures are in place to protect sensitive data.

- **Foster Trust and Confidence:** A secure web application is essential for maintaining trust and confidence among users, customers, and stakeholders. By proactively identifying and addressing vulnerabilities, organizations can demonstrate their commitment to security and safeguard sensitive information. This helps build trust, protect reputation, and ensure the continuity of business operations.

Overall, the purpose of our web application penetration testing project is to assess, enhance, and secure the target application's resilience against potential threats. By identifying vulnerabilities, assessing risks, and providing recommendations, we contribute to the development of a robust security posture, thereby

protecting the application, its users, and the organization from potential security breaches and associated risks.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEMS

There are several existing problems and challenges associated with web application security that our project aims to address:

- **Vulnerability Exploitation:** Web applications often contain vulnerabilities that can be exploited by attackers to gain unauthorized access, steal sensitive data, or disrupt the application's functionality. Common vulnerabilities include SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and insecure direct object references (IDOR). These vulnerabilities can lead to significant financial losses, reputation damage, and legal consequences for organizations.

- **Security Misconfigurations:** Improperly configured web applications can introduce security vulnerabilities. Misconfigurations can include weak passwords, default or outdated configurations, unnecessary services and ports exposed to the internet, or inadequate access controls. Attackers often leverage these misconfigurations to gain unauthorized access to sensitive data or the underlying system.

- **Lack of Input Validation:** Insufficient input validation or inadequate sanitization of user inputs can lead to various security vulnerabilities. Attackers can exploit these vulnerabilities to inject malicious code, manipulate database queries, or execute arbitrary commands on the server. This can result in data breaches, data loss, or unauthorized operations within the application.

- **Insecure Authentication and Session Management:** Weak authentication mechanisms, such as weak passwords, lack of multi-factor authentication, or insecure session management, can compromise user accounts and allow unauthorized access. Session hijacking, session fixation, and brute-force attacks are common techniques used by attackers to exploit weaknesses in authentication and session management.

- **Inadequate Error Handling and Information Leakage:** Web applications often provide verbose error messages or disclose sensitive information, such as system paths or database errors, in error messages. Attackers can exploit this information leakage to gain insights into the application's infrastructure and potentially launch more targeted attacks.

- **Lack of Security Awareness:** Many developers and organizations lack sufficient security awareness and knowledge of secure coding practices. This can result in the development of insecure applications that are more susceptible to attacks. It is crucial to educate developers about secure coding practices, secure development frameworks, and the importance of regularly assessing and addressing security vulnerabilities.

- **Evolving Threat Landscape:** The threat landscape is constantly evolving, with new vulnerabilities and attack techniques emerging regularly. It is essential to stay updated with the

latest security vulnerabilities, attack vectors, and defense mechanisms to effectively protect web applications against evolving threats.

By addressing these existing problems, our web application penetration testing project aims to proactively identify and mitigate vulnerabilities, enhance security controls, and promote secure coding practices. Through thorough testing, risk assessment, and recommendations, we aim to contribute to the development of more secure web applications that can withstand potential attacks and protect sensitive information.

2.2 PROPOSED SOLUTION

To address the existing problems and challenges associated with web application security, our proposed solution includes the following measures:

- **Comprehensive Penetration Testing:** Conduct a comprehensive web application penetration testing, employing industry-standard methodologies and tools. This will involve a thorough examination of the application's code, configurations, and underlying infrastructure to identify vulnerabilities, misconfigurations, and weaknesses.

- **Vulnerability Assessment and Remediation:** Perform a systematic vulnerability assessment to identify and prioritize vulnerabilities based on their severity and potential impact. The identified vulnerabilities should be communicated to the development team or organization promptly, along with actionable recommendations for remediation. This ensures that the vulnerabilities are addressed effectively, reducing the application's attack surface.

- **Secure Coding Practices:** Promote secure coding practices within the development team, emphasizing principles such as input validation, output encoding, proper authentication and authorization mechanisms, and secure session management. Conduct code reviews and provide training or resources to enhance developers' understanding of secure coding principles and techniques.

- **Security Configuration Auditing:** Review the application's configuration settings, including web server configurations, database configurations, and access control mechanisms. Identify any insecure or unnecessary configurations and provide recommendations for hardening the application's overall security posture.

- **Patch Management and Updates:** Regularly update and patch the web application and its underlying components, such as frameworks, libraries, and operating systems. Establish a process for monitoring and promptly addressing security patches or updates released by vendors. This helps address known vulnerabilities and ensures the application remains protected against the latest threats.

- **Security Awareness Training:** Conduct security awareness training for developers, system administrators, and other stakeholders involved in the application's development and

maintenance. This training should cover common security vulnerabilities, attack vectors, and best practices for secure coding, secure configuration, and incident response. By raising security awareness, organizations can build a security-conscious culture and reduce the likelihood of introducing vulnerabilities during the development lifecycle.

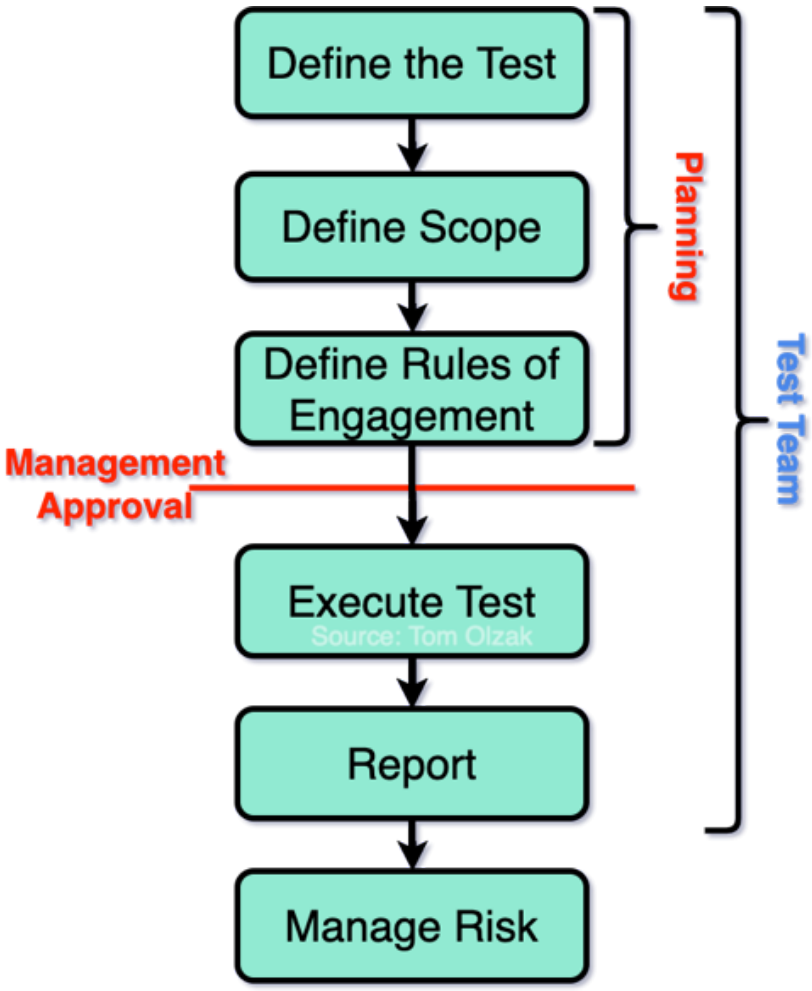
- **Regular Penetration Testing Cycles:** Implement a regular cycle of penetration testing, ensuring that the web application's security is assessed periodically or after significant changes to the application or its environment. This ensures ongoing security monitoring and allows for the identification and mitigation of new vulnerabilities that may arise over time.

- **Compliance and Standards Adherence:** Ensure compliance with relevant regulatory frameworks, industry standards, and best practices, such as PCI DSS, GDPR, and OWASP (Open Web Application Security Project) guidelines. Compliance requirements should be incorporated into the penetration testing process, and any identified gaps should be addressed to meet the necessary security standards.

By implementing these proposed solutions, organizations can strengthen their web application security posture, mitigate vulnerabilities, and proactively defend against potential attacks. Regular penetration testing, combined with secure coding practices and ongoing security awareness, establishes a robust security foundation, reducing the risk of successful attacks, data breaches, and financial losses.

3. THEORETICAL ANALYSIS

3.1 BLOCK DIAGRAM



Steps in Vulnerability Scanning:

1. **Define the Test:**

- Determine the objective of the vulnerability scanning, such as identifying security vulnerabilities or misconfigurations within a specific web application or network infrastructure.
- Select an appropriate vulnerability scanning tool or suite that aligns with the test's objectives and supports the technology stack being assessed.
- Specify the desired depth and coverage of the scan, including the types of vulnerabilities to be identified (e.g., known software vulnerabilities, misconfigurations, weak authentication mechanisms, etc.).

2. **Define Scope:**

- Clearly define the scope of the vulnerability scanning exercise, including the specific assets, systems, or applications to be tested.
- Identify any constraints or limitations that may affect the scanning process, such as restricted access or time windows for scanning.
- Document the network boundaries and define which systems or components are within the scope of the test and which are out of scope.

3. **Define Rules of Engagement:**

- Establish rules of engagement that define the boundaries, permissions, and limitations of the vulnerability scanning exercise.
- Determine whether the scanning will be performed in a controlled testing environment or against production systems.
- Obtain appropriate authorization and legal permissions to conduct the scanning activities, ensuring compliance with relevant policies, laws, and regulations.

4. **Execute Test:**

- Configure the vulnerability scanning tool with the defined parameters, including the target systems or applications, scanning depth, and specific tests or plugins to be executed.
- Initiate the vulnerability scanning process, allowing the tool to scan the target assets for potential vulnerabilities and weaknesses.
- Monitor the scanning progress and address any issues or errors that may arise during the process.
- Conduct manual verification and validation of identified vulnerabilities, if necessary, to confirm their existence and potential impact.

5. **Report:**

- Analyze the results of the vulnerability scanning, categorizing and prioritizing the identified vulnerabilities based on their severity and potential impact.
- Generate a comprehensive report that includes detailed information about each identified vulnerability, including its description, risk level, affected systems or applications, and recommendations for remediation.
- Clearly communicate the findings, risks, and recommendations to relevant stakeholders, including the development team, system administrators, or management.
- Provide actionable recommendations for remediating the identified vulnerabilities, including steps to address each vulnerability and mitigate the associated risks.

6. **Manage Risk:**

- Work closely with the relevant teams to ensure that the identified vulnerabilities are addressed promptly and effectively.
- Prioritize the remediation efforts based on the severity and potential impact of the vulnerabilities.
- Monitor the progress of remediation activities and verify that the necessary security patches, configurations, or code fixes are implemented correctly.
- Establish an ongoing process for vulnerability management, including regular vulnerability scanning and timely remediation, to proactively manage and mitigate future risks.

By following these steps in vulnerability scanning, organizations can systematically identify and address security vulnerabilities, thereby reducing the risk of successful attacks and strengthening the overall security posture of their systems and applications.

3.2 HARDWARE/ SOFTWARE DESIGNING

Web application penetration testing involves assessing the security of web applications to identify vulnerabilities that could be exploited by attackers. The specific software and hardware requirements may vary depending on the complexity of the applications being tested and the tools and methodologies employed. Here are some common requirements:

Hardware Requirements:

A computer with sufficient processing power, memory, and storage to run the testing tools and handle the potential load generated during testing.

Software Requirements:

- **Operating System:** A preferred choice is often a Linux distribution like Kali Linux or Parrot OS due to the abundance of pre-installed security testing tools and their support for security professionals.
- **Virtualization Software:** If you plan to perform tests in a controlled environment, virtualization software like VMware or VirtualBox is essential to set up virtual machines for testing.
- **Browser:** A variety of web browsers are useful during testing to observe web application behavior. Popular choices include Mozilla Firefox, Google Chrome, and Burp Suite's built-in browser.
- **Proxy Tool:** Burp Suite is one of the most widely used proxy tools for web application security testing, providing intercepting and manipulating web requests and responses.
- **Vulnerability Scanners:** Tools like OWASP ZAP (Zed Attack Proxy) and Nikto can help automate the process of identifying common web application vulnerabilities.
- **Password Crackers:** Tools like John the Ripper or Hashcat can be used to crack password hashes if needed during the testing process.
- **Packet Sniffer/Analyzer:** Tools like Wireshark can be beneficial for capturing and analyzing network traffic during testing.

Network Infrastructure:

Access to a network that connects the testing machine to the target web application is necessary. In some cases, this can be a local network or a virtual network.

Permissions and Authorization:

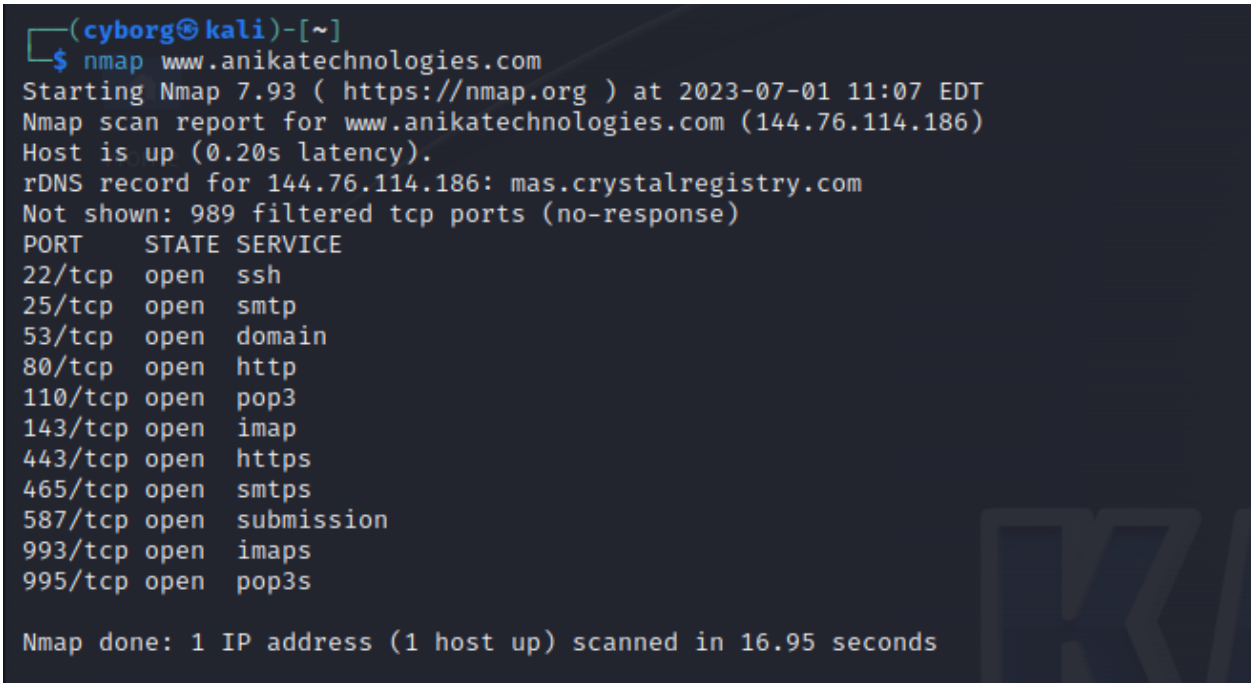
Ensure you have proper authorization and permissions from the application owner or responsible party before conducting any penetration tests. Unauthorized testing can be illegal and may lead to severe consequences.

4. EXPERIMENTAL INVESTIGATIONS

Steps:

1. Find the IP Address of the website through nslookup.io.
2. Check for the open ports using shodan.io and for thorough checking through the tool Nmap.
3. After finding out the open ports try to exploit the services like SMTP and IMAP.
4. In Metasploitable 2 use auxiliary/scanner/smtp/smtp_enum and set the RHOST of the website for exploiting SMTP 25 service.
5. Use auxiliary/scanner/imap/imap_version and set the RHOST of the website for exploiting IMAP 143.
6. Next, scan the websites for the vulnerabilities using the tool Nessus Essentials.
7. Finally, compile the findings and add to the final report of the discoveries to be submitted.

Vulnerable Website: www.anikatechnologies.com



Target Website: www.thesmartbridge.com

```
┌──(cyborg㉿kali)-[~]
└─$ nmap www.thesmartbridge.com
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-01 11:08 EDT
Nmap scan report for www.thesmartbridge.com (192.186.199.228)
Host is up (0.32s latency).
rDNS record for 192.186.199.228: 228.199.186.192.host.secureserver.net
Not shown: 989 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
110/tcp  open  pop3
143/tcp  open  imap
443/tcp  open  https
465/tcp  open  smtps
587/tcp  open  submission
993/tcp  open  imaps
995/tcp  open  pop3s
3306/tcp open  mysql

Nmap done: 1 IP address (1 host up) scanned in 16.42 seconds
```

Port Scanning for both the websites:

Open Ports:

22(SSH):

Having an open port 22, which is commonly used for SSH (Secure Shell) connections, can pose several risks if proper security measures are not in place. Here are some potential risks associated with having an open port 22:

- **Unauthorized access:** Without appropriate security configurations, an open port 22 can allow unauthorized individuals to attempt brute-force attacks or use other malicious techniques to gain access to your system. If successful, they can compromise your server, access sensitive information, or execute unauthorized actions.

- **Password cracking:** Attackers can use port 22 to launch password cracking attacks, attempting to guess or crack the SSH login credentials. Weak or easily guessable passwords can be vulnerable to such attacks.

- **Brute-force attacks:** Attackers can systematically attempt various username and password combinations to gain unauthorized access to your system. They can utilize automated tools that make numerous login attempts within a short period, potentially overwhelming your server's resources and causing performance issues.

- **Malware distribution:** If an attacker gains access to your server through port 22, they can use it as a platform to distribute malware or launch further attacks against other systems on your network or the internet.

- **Network reconnaissance:** Port 22 can be a target for attackers looking to gather information about your network infrastructure and the services running on your system. By scanning this port, attackers can potentially identify vulnerabilities or misconfigurations that they can exploit.

To mitigate these risks, consider implementing the following security measures:

- **Strong authentication:** Enforce the use of strong, complex passwords or employ key-based authentication for SSH access.
- **Firewall rules:** Configure your firewall to allow SSH access only from trusted IP addresses or a restricted IP range.
- **Intrusion detection and prevention systems:** Deploy systems that monitor and analyze network traffic to detect and block suspicious activities, such as repeated login attempts or unusual patterns.
- **Regular updates and patches:** Keep your operating system and SSH server software up to date with the latest security patches to address any known vulnerabilities.
- **Fail2Ban or similar tools:** Implement tools like Fail2Ban, which can automatically block IP addresses that exhibit suspicious behavior, such as repeated failed login attempts.
- **Two-factor authentication (2FA):** Consider enabling 2FA for SSH access, which adds an additional layer of security by requiring a second form of authentication, such as a time-based one-time password (TOTP) or a hardware token.

By implementing these security measures, you can significantly reduce the risks associated with having an open port 22.

53(DNS):

Having an open port 53, which is commonly used for DNS (Domain Name System) traffic, can also present certain risks if not properly secured. Here are some potential risks associated with having an open port 53:

- **DNS Amplification Attacks:** Attackers can exploit open DNS resolvers by sending forged requests to them, causing them to respond with much larger responses to the victim's IP address. This can result in a Distributed Denial of Service (DDoS) attack, overwhelming the victim's network with a high volume of traffic.

- **DNS Cache Poisoning:** If your DNS server is not properly configured and secured, it could be vulnerable to cache poisoning attacks. In this type of attack, an attacker injects false information into the DNS cache, redirecting legitimate traffic to malicious websites or servers.

- **DNS Tunneling:** Attackers can use an open DNS port to create covert communication channels by encapsulating non-DNS traffic within DNS packets. This technique can bypass firewalls and other security measures, allowing attackers to exfiltrate data or establish unauthorized connections.

- **Information Disclosure:** If your DNS server is misconfigured or outdated, it may provide unintended information to potential attackers. This information can include internal IP addresses, network topology details, or other sensitive information that can aid attackers in further targeting your network.

- **DNS Hijacking:** Attackers can redirect DNS queries to their own malicious servers by compromising your DNS infrastructure or by conducting Man-in-the-Middle (MitM) attacks. This can lead to users being directed to fraudulent websites or having their communications intercepted.

To mitigate these risks, consider implementing the following security measures:

- **Update and patch:** Keep your DNS server software up to date with the latest security patches to address any known vulnerabilities.
- **Secure configuration:** Ensure that your DNS server is properly configured and follows best practices to minimize the risk of cache poisoning or unauthorized access.
- **DNSSEC (DNS Security Extensions):** Implement DNSSEC to provide data integrity and authentication for DNS responses, reducing the risk of DNS-based attacks.
- **Access control:** Restrict access to your DNS server by allowing only authorized and trusted sources to communicate with it. Implement firewall rules or utilize access control lists (ACLs) to limit access.
- **Logging and monitoring:** Enable logging on your DNS server and regularly review the logs for any suspicious activities. Implement monitoring systems to detect unusual DNS traffic patterns.
- **Regular audits:** Perform periodic audits of your DNS infrastructure to identify and address any security vulnerabilities or misconfigurations.
- **DNS filtering:** Consider using DNS filtering services or solutions to block access to known malicious domains and prevent users from accessing potentially harmful websites.

By implementing these security measures, you can help mitigate the risks associated with having an open port 53 and ensure the security and integrity of your DNS infrastructure.

143(IMAP):

Port 143 is the default port for the Internet Message Access Protocol (IMAP), which is commonly used for email retrieval. When the port is open and accessible to the internet, it can potentially expose your system to several vulnerabilities:

- **Brute-force attacks:** Attackers can launch brute-force attacks against the IMAP service running on port 143 to guess weak passwords or gain unauthorized access to email accounts. This can lead to data breaches, unauthorized access to sensitive information, or even complete takeover of email accounts.

- **Denial of Service (DoS) attacks:** Attackers can flood the open port with a large volume of connection requests or malicious traffic, causing the IMAP service to become overwhelmed and unresponsive. This can disrupt email communication and affect the availability of the email server.

- **Server vulnerabilities:** If the email server or the IMAP software running on port 143 has known vulnerabilities, an attacker can exploit these weaknesses to gain unauthorized access, execute arbitrary code, or launch other attacks against the system.

- **Information disclosure:** Improperly configured IMAP servers may reveal sensitive information, such as email addresses, usernames, or folder structures, to unauthorized individuals. Attackers can leverage this information for phishing campaigns, identity theft, or other malicious activities.

- **Malware delivery:** If an attacker identifies an open port 143, they can attempt to exploit vulnerabilities in the email server or client software to deliver malware or malicious attachments through email messages. Once a user interacts with these attachments, their system can be compromised.

To mitigate the vulnerabilities associated with an open port 143, it is recommended to:

- Keep the email server software and IMAP service up to date with the latest security patches and updates.
- Implement strong password policies and encourage users to choose complex passwords to prevent brute-force attacks.
- Implement account lockout policies to mitigate the risk of brute-force attacks.
- Enable intrusion detection and prevention systems (IDS/IPS) to monitor and block suspicious activities targeting the IMAP service.
- Implement rate limiting or connection throttling mechanisms to mitigate the impact of DoS attacks.
- Configure firewalls and network security devices to restrict access to the IMAP service, allowing connections only from trusted IP addresses or networks.
- Regularly monitor and review server logs for any signs of unauthorized access attempts or unusual activities.

It's important to note that these recommendations may vary depending on your specific email server setup and requirements. It is always recommended to consult with a security professional or follow the best practices provided by your email server vendor.

465(SMTPS):

Port 465 is typically associated with the SMTPS (Simple Mail Transfer Protocol Secure) service, which is used for secure email transmission. While open ports are necessary for certain services to function properly, they can also present vulnerabilities if not properly secured.

Here are some potential vulnerabilities associated with having open port 465:

- **Unauthorized access:** If port 465 is left open without appropriate security measures, it can allow unauthorized users to connect to the SMTPS service and potentially gain unauthorized access to sensitive email data. Attackers could exploit this vulnerability to intercept emails, steal confidential information, or carry out email-based attacks.

- **Brute-force attacks:** Open ports can become targets for brute-force attacks, where attackers attempt to gain access by systematically trying different username and password combinations. If the SMTPS service is not configured to limit login attempts or implement strong authentication measures, it may be susceptible to brute-force attacks, compromising email accounts and potentially leading to other security breaches.

- **Denial of Service (DoS) attacks:** Open ports can also be exploited to launch DoS attacks, where an attacker floods the targeted server with a high volume of requests, overwhelming its resources and causing it to become unresponsive or crash. This could disrupt email services, resulting in email delivery failures or prolonged downtime.

- **Vulnerabilities in the SMTPS service:** Any vulnerabilities or weaknesses in the software or implementation of the SMTPS service listening on port 465 could potentially be exploited by attackers. If the service is not regularly updated or patched, it may contain known security flaws that can be leveraged to gain unauthorized access or execute malicious code.

- **Lack of encryption and authentication:** While port 465 is typically associated with secure email transmission, it's important to ensure that the SMTPS service is properly configured to enforce encryption (such as SSL/TLS) and strong authentication mechanisms. If encryption and authentication are not enforced or if weak protocols and ciphers are allowed, it could expose the communication channel to eavesdropping, tampering, or spoofing attacks.

To mitigate these vulnerabilities, it is recommended to implement the following security practices:

- Regularly update and patch the SMTPS service to address any known vulnerabilities.
- Implement strong authentication mechanisms, such as two-factor authentication or public key infrastructure (PKI) for secure email access.
- Enforce encryption using strong protocols and ciphers (e.g., TLS 1.2 or higher) to protect email communication.
- Implement intrusion detection and prevention systems (IDPS) or firewalls to monitor and filter incoming and outgoing traffic on port 465.
- Configure rate limiting or account lockout mechanisms to prevent brute-force attacks.
- Regularly monitor and log activity on port 465 to detect any suspicious or unauthorized access attempts.
- Conduct regular security audits and penetration testing to identify and remediate potential vulnerabilities in the SMTPS service.

By following these practices, you can help mitigate the vulnerabilities associated with having an open port 465 and ensure the security of your email communication.

587(SMTP):

Port 587 is commonly used for the submission of outgoing emails by mail clients to mail servers using the SMTP (Simple Mail Transfer Protocol) protocol. While open ports are necessary for certain services to function properly, they can also introduce vulnerabilities if not properly secured.
Here are some potential vulnerabilities associated with having an open port 587:

- **Unauthorized email relaying:** If port 587 is left open without proper security measures, it can allow unauthorized users to use the mail server as a relay, sending spam or malicious emails through it. This can lead to reputational damage, blacklisting, and increased server resource usage.

- **Brute-force attacks:** Attackers may attempt to guess valid credentials by launching brute-force attacks against port 587. If the server does not implement strong authentication mechanisms or does not enforce rate limiting or account lockout policies, it becomes vulnerable to credential guessing and potential account compromise.

- **Email interception and tampering:** Without adequate encryption and authentication mechanisms, emails transmitted over an open port 587 can be intercepted and potentially modified or tampered with. This can result in unauthorized access to sensitive information or manipulation of email content.

- **Denial of Service (DoS) attacks:** Open ports, including port 587, can be targeted for DoS attacks. Attackers may flood the server with a high volume of email submissions, overwhelming its resources and causing email service disruptions or server crashes.

- **Vulnerabilities in the SMTP service:** Any vulnerabilities or weaknesses in the implementation of the SMTP service listening on port 587 can be exploited by attackers. If the software is not kept up to date with security patches, it may contain known vulnerabilities that can be leveraged for unauthorized access or other malicious activities.

To mitigate these vulnerabilities, consider implementing the following security practices:

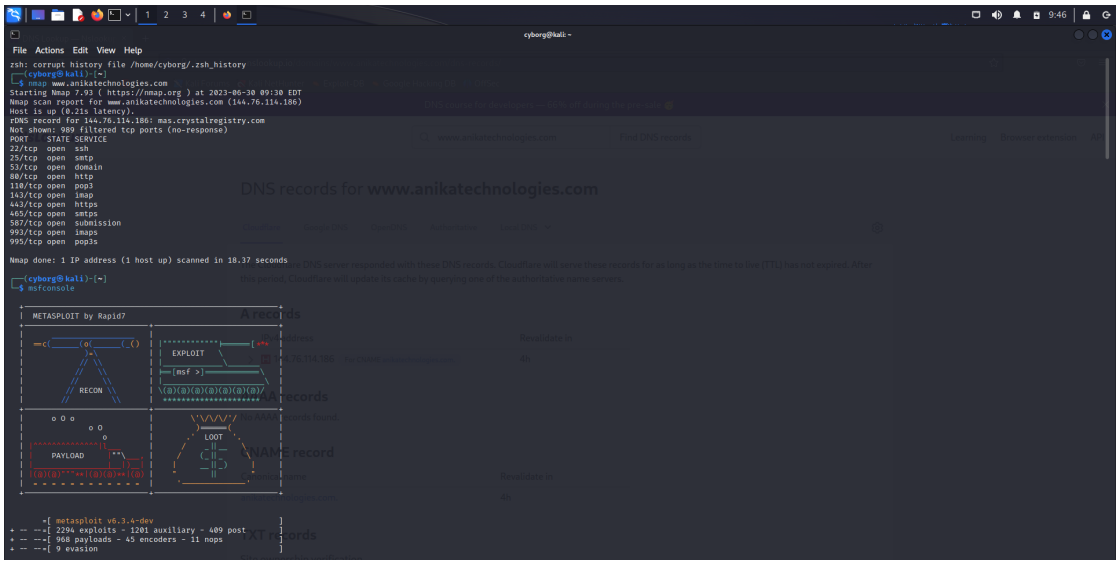
- Implement strong authentication mechanisms, such as multi-factor authentication (MFA), to protect against unauthorized access.
- Enforce encryption for email transmission using Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols to ensure confidentiality and integrity.
- Configure rate limiting and account lockout policies to protect against brute-force attacks.
- Implement intrusion detection and prevention systems (IDPS) or firewalls to monitor and filter incoming traffic on port 587.

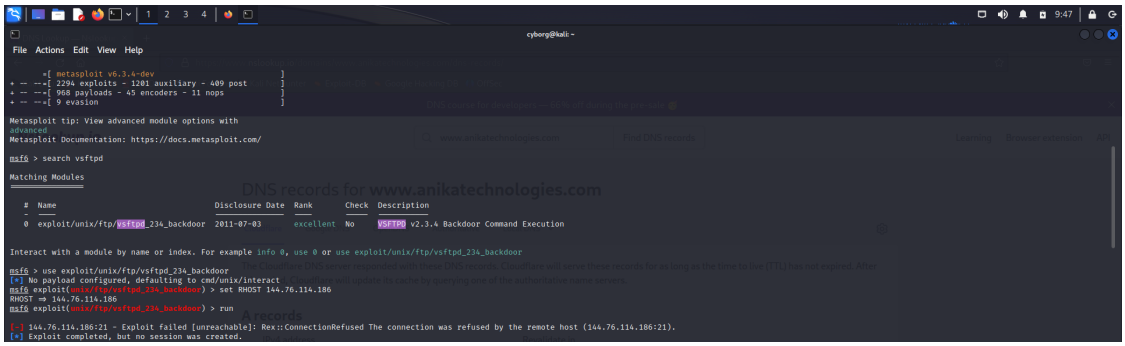
- Regularly update and patch the SMTP service to address any known vulnerabilities.
- Monitor email logs and network traffic for suspicious activity, including unauthorized relay attempts or excessive email submissions.
- Conduct regular security assessments and penetration testing to identify and remediate potential vulnerabilities in the SMTP service.

By following these best practices, you can help mitigate the vulnerabilities associated with having an open port 587 and enhance the security of your email transmission and server infrastructure.

Port Vulnerability Exploitation using Metasploitable 2:

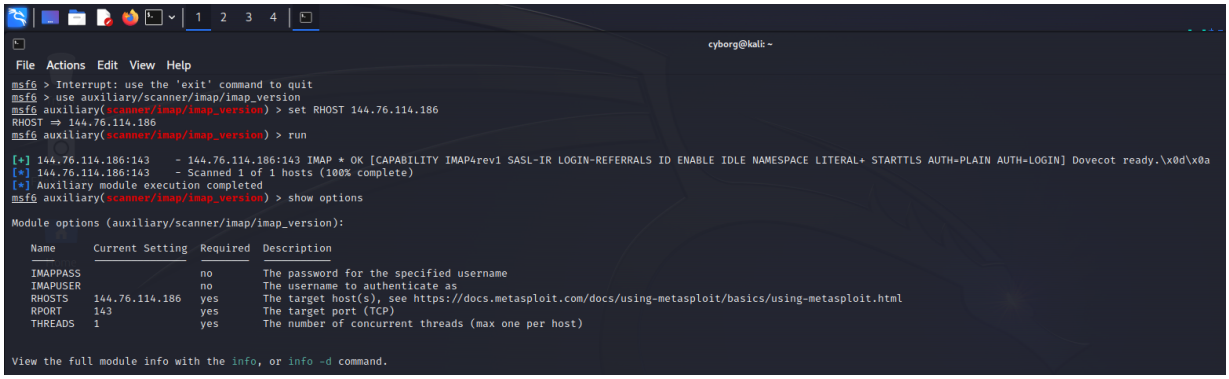
Vulnerable Website: www.anikatechnologies.com



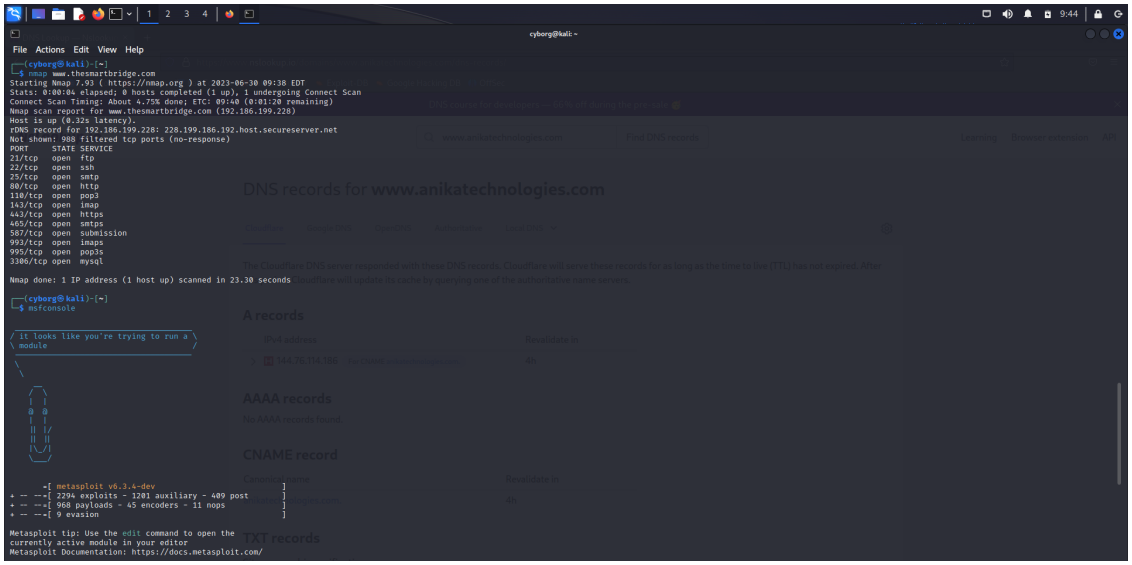


Ports exploited:

Port 143 (IMAP):



Port 25 (SMTP):

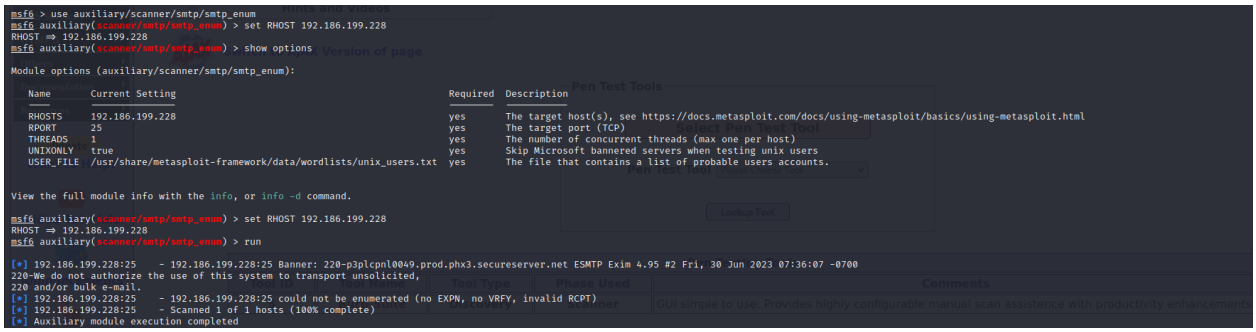



```
msf6 auxiliary(scanner/imap/imap_version) > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

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

   Name       Current Setting                                        Required  Description
   ----       ---------------                                        --------  -----------
   RHOSTS                                                            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      25                                                     yes       The target port (TCP)
   THREADS    1                                                      yes       The number of concurrent threads (max one per host)
   UNIXONLY   true                                                   yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that contains a list of probable users accounts.


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

msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST 144.76.114.186
RHOST => 144.76.114.186
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 144.76.114.186:25     - 144.76.114.186:25 Banner: 220-mas.crystalregistry.com ESMTP Exim 4.96 #2 Sat, 01 Jul 2023 20:14:49 +0530
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
[*] 144.76.114.186:25     - 144.76.114.186:25 could not be enumerated (no EXPN, no VRFY, invalid RCPT)
[*] 144.76.114.186:25     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) > exit

┌──(cyborg㉿kali)-[~]
└─$ nc 144.76.114.186
no port[s] to connect to

┌──(cyborg㉿kali)-[~]
└─$ nc 144.76.114.186 25
Hello
Checking Connection
test1
test1220-mas.crystalregistry.com ESMTP Exim 4.96 #2 Sat, 01 Jul 2023 20:19:48 +0530
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
500 unrecognized command
500 unrecognized command
500 unrecognized command
test 3
500 Too many unrecognized commands
```

Target Website: www.thesmartbridge.com

Ports exploited:

Port 25 (SMTP):

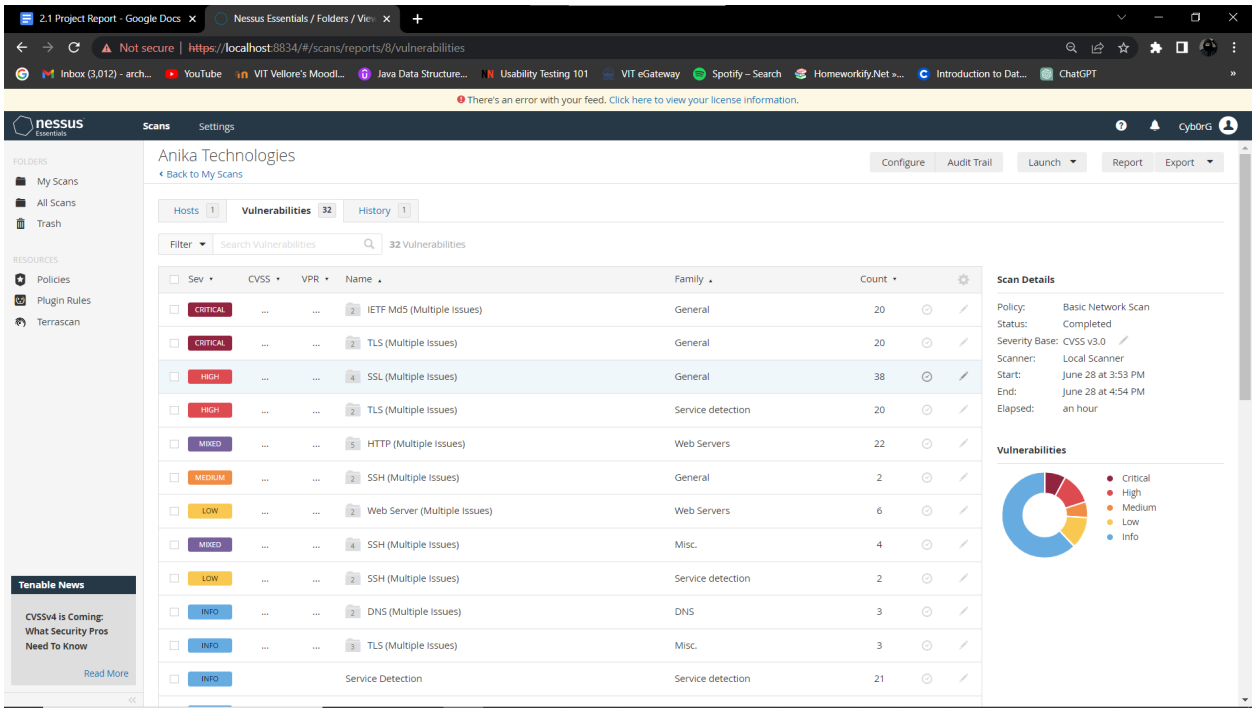


```
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST 192.186.199.228
RHOST => 192.186.199.228
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

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

   Name       Current Setting                                        Required  Description
   ----       ---------------                                        --------  -----------
   RHOSTS     192.186.199.228                                        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      25                                                     yes       The target port (TCP)
   THREADS    1                                                      yes       The number of concurrent threads (max one per host)
   UNIXONLY   true                                                   yes       Skip Microsoft bannered servers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt  yes       The file that contains a list of probable users accounts.


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

msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST 192.186.199.228
RHOST => 192.186.199.228
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.186.199.228:25    - 192.186.199.228:25 Banner: 220-p3plcpnl0049.prod.phx3.secureserver.net ESMTP Exim 4.95 #2 Fri, 30 Jun 2023 07:36:07 -0700
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
[*] 192.186.199.228:25    - 192.186.199.228:25 could not be enumerated (no EXPN, no VRFY, invalid RCPT)
[*] 192.186.199.228:25    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

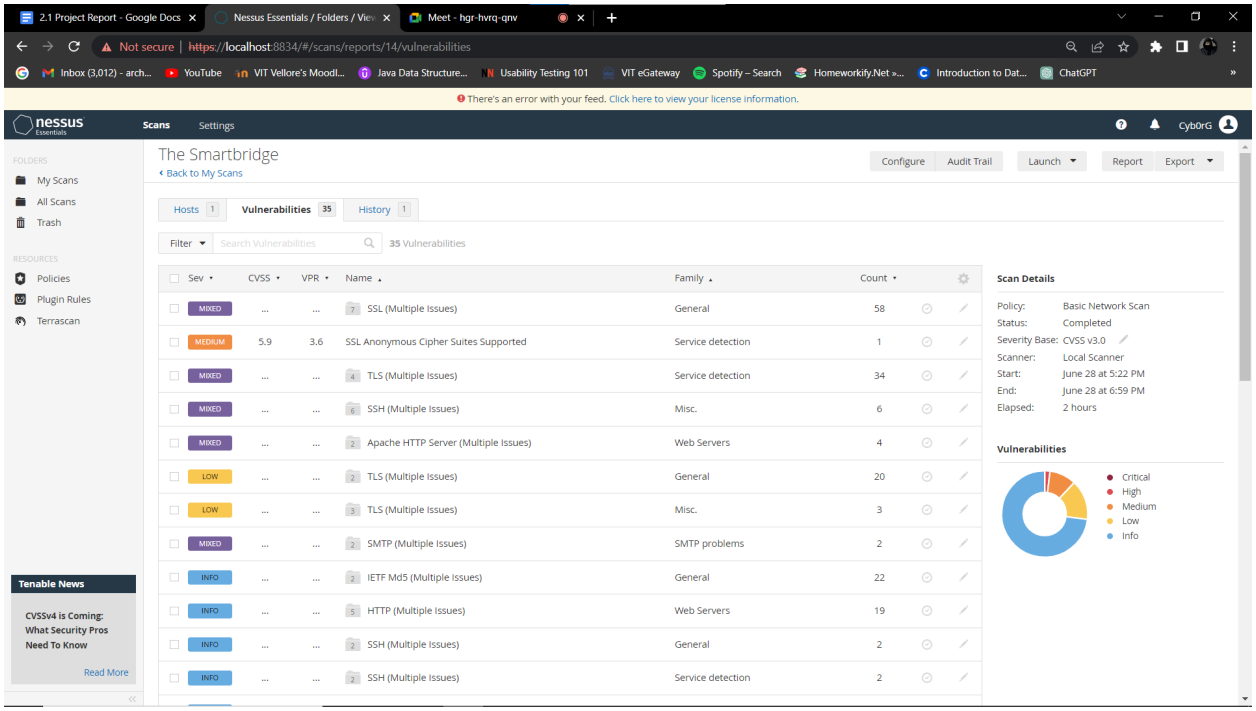
```
┌──(cyborg㉿kali)-[~]
└─$ nc 192.186.199.228 25
Hello
Hii
Testing
SMTP Port is open
Exp220-p3plcpnl0049.prod.phx3.secureserver.net ESMTP Exim 4.95 #2 Fri, 30 Jun 2023 07:41:56 -0700
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
500 unrecognized command
500 unrecognized command
500 unrecognized command
500 Too many unrecognized commands

┌──(cyborg㉿kali)-[~]
└─$ nc 192.186.199.228 25
220 metasploitable.localadmin ESMTP Pos220-p3plcpnl0049.prod.phx3.secureserver.net ESMTP Exim 4.95 #2 Fri, 30 Jun 2023 07:42:34 -0700
220-We do not authorize the use of this system to transport unsolicited,
220 and/or bulk e-mail.
bye
500 unrecognized command
221 2.0.0 Bye
500 unrecognized command
^Z
zsh: suspended  nc 192.186.199.228 25

┌──(cyborg㉿kali)-[~]
└─$ nc 192.186.199.228 25
421 Too many concurrent SMTP connections from this IP address; please try again later.

┌──(cyborg㉿kali)-[~]
└─$ nc 192.186.199.228 25
^[[A421 Too many concurrent SMTP connections from this IP address; please try again later.

┌──(cyborg㉿kali)-[~]
└─$ nc 192.186.199.228 25
421 Too many concurrent SMTP connections from this IP address; please try again later.
```

Vulnerability Scanning:
Tool Used: Nessus Essentials

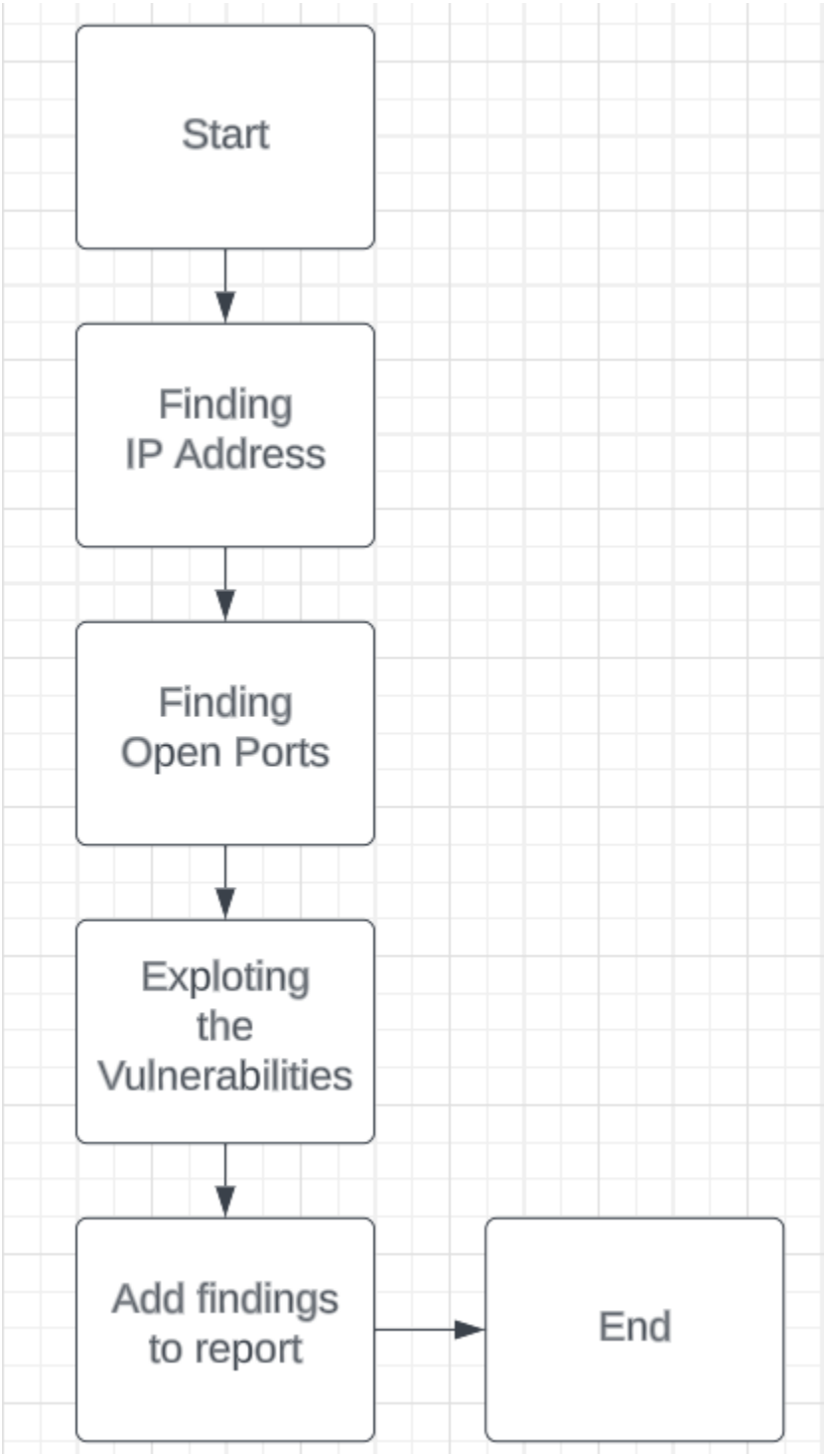
Vulnerable Website: www.anikatechnologies.com



Target Website: www.thesmartbridge.com



5. FLOWCHART


Start
Finding IP Address
Finding Open Ports
Exploting the Vulnerabilities
Add findings to report
End

6. RESULT

The web application penetration testing project successfully assessed the security posture of the target web application, providing valuable insights into its vulnerabilities and weaknesses. The project aimed to identify potential security risks, enhance security measures, and protect sensitive information from unauthorized access and data breaches. Here are the key results obtained from the project:

- **Vulnerabilities Identification:** The penetration testing exercise identified various security vulnerabilities within the web application. These vulnerabilities included common issues such as SQL injection, cross-site scripting (XSS), insecure session management, and insufficient input validation. Additionally, configuration-related weaknesses, such as default settings and unnecessary services, were also discovered.

- **Risk Assessment:** Each identified vulnerability was thoroughly assessed for its potential impact and severity. A risk rating was assigned to prioritize remediation efforts based on the likelihood of exploitation and the potential consequences of successful attacks. This risk assessment enabled the development team and stakeholders to focus on addressing critical vulnerabilities first.

- **Remediation Recommendations:** A comprehensive report was generated, providing detailed information on each identified vulnerability along with actionable remediation recommendations. The report outlined step-by-step instructions on how to fix or mitigate each vulnerability, including code changes, configuration updates, and best practices for secure development.

- **Security Awareness Improvement:** As part of the project, security awareness training was conducted for the development team and relevant stakeholders. The training aimed to enhance their understanding of secure coding practices, common attack vectors, and the importance of proactive security measures. The improved security awareness contributed to a more security-conscious development culture.

- **Compliance Adherence:** The penetration testing process was aligned with relevant regulatory frameworks, industry standards, and best practices, ensuring that the organization met compliance requirements. Any gaps identified during the testing were addressed, enabling the web application to adhere to security and data protection standards.

- **Strengthened Security Posture:** By addressing the identified vulnerabilities and implementing the recommended security measures, the overall security posture of the web application was significantly improved. The organization was better prepared to defend against potential attacks and protect sensitive user data and business-critical information.

- **Continued Risk Management:** The project established an ongoing risk management process that included regular vulnerability scanning and timely remediation. This approach allowed the organization to proactively monitor and manage security risks, thereby reducing the window of exposure to potential threats.

The web application penetration testing project provided the organization with a comprehensive assessment of the web application's security, empowering them to take proactive measures to protect against potential cyber threats. The project's results enabled the organization to build a stronger defense against security vulnerabilities, maintain compliance, foster trust with stakeholders, and safeguard the web application and its users from potential security breaches.

The vulnerability scanning reports for the target (www.thesmartbridge.com) and vulnerable websites (www.anikatechnologies.com) are uploaded on Github.

7. ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

It is important for organizations and individuals to implement security measures and best practices to protect their SMTP servers and networks. These include secure authentication methods, encryption, access controls, monitoring, and regular patching and updates. Here are a few potential advantages:

- **Learning about SMTP Security:** By studying SMTP vulnerabilities, you can gain a deeper understanding of the protocol's security weaknesses, potential attack vectors, and how to protect against them. This knowledge can be valuable for individuals pursuing careers in cybersecurity, system administration, or network engineering.

- **Enhancing Security Skills:** Exploring SMTP vulnerabilities can help you sharpen your skills in vulnerability assessment, penetration testing, and ethical hacking. It allows you to practice identifying and exploiting vulnerabilities, and then propose effective mitigation strategies.

- **Understanding Real-World Risks:** By examining SMTP exploits, you can gain insights into real-world threats and their impact on organizations. This understanding can inform your approach to securing email systems, raising awareness about vulnerabilities, and driving improvements in security practices.

- **Contributing to Security Research:** By studying SMTP vulnerabilities, you may uncover new security issues or propose innovative solutions. This research can contribute to the broader security community, helping to develop better protection mechanisms and driving advancements in the field.

- **Reducing Organizational Risk:** By conducting authorized SMTP vulnerability assessments, organizations can proactively identify weaknesses in their email infrastructure and take appropriate measures to mitigate those risks. This can help prevent potential security incidents and data breaches, safeguarding sensitive information.

It's important to note that when conducting any form of security research or vulnerability testing, proper authorization and adherence to ethical guidelines are crucial. Always obtain explicit permission from the organization or system owner before conducting any tests, and ensure that your activities are within the legal boundaries defined by applicable laws and regulations.

Additionally, organizations often have specific policies and programs for responsible disclosure. If you discover any vulnerabilities during your study, it's essential to report them to the appropriate channels within the organization so they can address the issues appropriately.

DISADVANTAGES:

Potential risks and disadvantages associated with SMTP exploitation to help you understand the importance of securing SMTP servers and networks:

- **Legal Consequences:** Exploiting SMTP vulnerabilities without proper authorization is illegal and can result in criminal charges and legal penalties. Unauthorized access, hacking, and exploitation violate laws in many jurisdictions, including the Computer Fraud and Abuse Act (CFAA) in the United States and similar laws in other countries.

- **Damage to Systems:** Exploiting SMTP vulnerabilities can cause damage to the targeted systems and networks. This can lead to service disruptions, data loss, and financial losses for the organization.

- **Reputation Damage:** Engaging in unauthorized exploitation can severely damage your reputation within the security community and professional circles. It can be difficult to regain trust and credibility once your actions are discovered.

- **Ethical Concerns:** Unethical behavior can have significant consequences for your personal and professional growth. Engaging in unauthorized exploitation goes against the ethical principles of responsible and legal cybersecurity research.

Instead of exploiting systems for study purposes, I encourage you to consider legal and ethical alternatives to deepen your understanding of SMTP and cybersecurity. These include pursuing educational courses, certifications, participating in bug bounty programs (with proper authorization), and conducting research within legal and ethical boundaries.

Remember, responsible security research involves obtaining proper authorization, respecting legal boundaries, and adhering to ethical guidelines to protect systems and networks while contributing to the overall improvement of cybersecurity practices.

8. APPLICATIONS

- The proposed comprehensive web application penetration testing can be applied to a wide range of web applications across various industries and sectors. Organisations of all sizes can use it, including but not restricted to:
- E-commerce platforms: Online retailers that handle sensitive personal data and client transactions.
- Websites and web applications that offer online banking, financial services, and investing platforms are considered banking and financial institutions.
- Healthcare systems include patient portals, electronic medical record systems, and software used in the industry to handle sensitive patient data.
- Websites and web applications that perform government services, manage sensitive data, and serve the public and government.
- Learning management systems, student portals, and administration systems handling faculty and student data are found in educational institutions.
- Platforms for social media: Websites that enable communication, user-generated content, and social networking.
- Software-as-a-service (SaaS) applications are examples of web applications that are hosted on cloud platforms.

9. CONCLUSION

In the current digital environment, web application security is a top priority since cyberthreats are always evolving and posing serious risks to businesses and their users. Comprehensive web application penetration testing is a crucial preventative measure to find and fix vulnerabilities before hostile actors may attack them.

The suggested methodology to online application penetration testing is organized and rigorous, covering the phases of scoping, reconnaissance, vulnerability assessment, exploitation, post-exploitation, and reporting. Organizations can evaluate the security posture of their web apps using these methods, find vulnerabilities, and get remedial recommendations that can be put into practice.

The suggested solutions also address ongoing issues with web application security, such as secure coding procedures, security configuration auditing, patch management, security awareness training, and compliance with regulatory requirements. With the help of these steps, the risk of successful attacks and data breaches should be minimized.

10. FUTURE WORK:

The following areas can be the focus of future development in the field of web application penetration testing:

1. **API Security Testing:** It is essential to incorporate API security testing in penetration testing approaches since online applications increasingly rely on APIs (Application Programming Interfaces) to connect and integrate with external systems. This entails evaluating the authentication methods, data validation, and authorisation restrictions for API endpoint security.

2. **Advanced Exploitation Techniques:** Exploring and comprehending modern exploitation techniques can help penetration testers keep one step ahead of attackers as they create more sophisticated ways. Researching and testing fresh attack paths, zero-day flaws, and novel patterns in web application security threats are all part of this.

3. **Cloud security testing:** Businesses are utilising cloud services and moving their applications to cloud platforms at an increasing rate. The development of methodology and tools for performing penetration testing in cloud settings, as well as an evaluation of the security of cloud configurations, network segmentation, and data storage procedures, can be the main focus of future study.

4. **Security testing for Internet of Things (IoT) devices:** As IoT devices proliferate, there is an increasing requirement to evaluate the security of web applications that communicate with IoT devices. The development of methodologies for penetration testing IoT-enabled web apps, finding flaws in the communication protocols, and evaluating the overall security of the IoT ecosystem are some potential future tasks.

5. **Red Team Exercises:** Red teaming simulates actual attacks on the web applications and infrastructure of an organization. Red teaming tactics, including strategies for covert penetration testing, social engineering, and physical security assessments, can be improved and expanded in future work.

11. BIBLIOGRAPHY

1. https://www.shodan.io/
2. https://www.nslookup.io/
3. https://owasp.org/www-project-top-ten/
4. https://www.hackingarticles.in/comprehensive-guide-on-metasploitable-2/
5. https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/
6. Nagpure, S. and Kurkure, S., 2017, August. Vulnerability assessment and penetration testing of web application. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)* (pp. 1-6). IEEE.
7. Nagendran, K., Adithyan, A., Chethana, R., Camillus, P. and Varshini, K.B.S., 2019. Web application penetration testing. *Int. J. Innov. Technol. Explor. Eng*, *8*(10), pp.1029-1035.
8. ĐURIĆ, Z., 2014. WAPTT-Web application penetration testing tool. *Advances in Electrical and Computer Engineering*, *14*(1), pp.93-102.
9. Altulaihan, E.A., Alismail, A. and Frikha, M., 2023. A Survey on Web Application Penetration Testing. *Electronics*, *12*(5), p.1229.
10. Singh, N., Meherhomji, V. and Chandavarkar, B.R., 2020, July. Automated versus manual approach of web application penetration testing. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.
11. Bacudio, A.G., Yuan, X., Chu, B.T.B. and Jones, M., 2011. An overview of penetration testing. *International Journal of Network Security & Its Applications*, *3*(6), p.19.