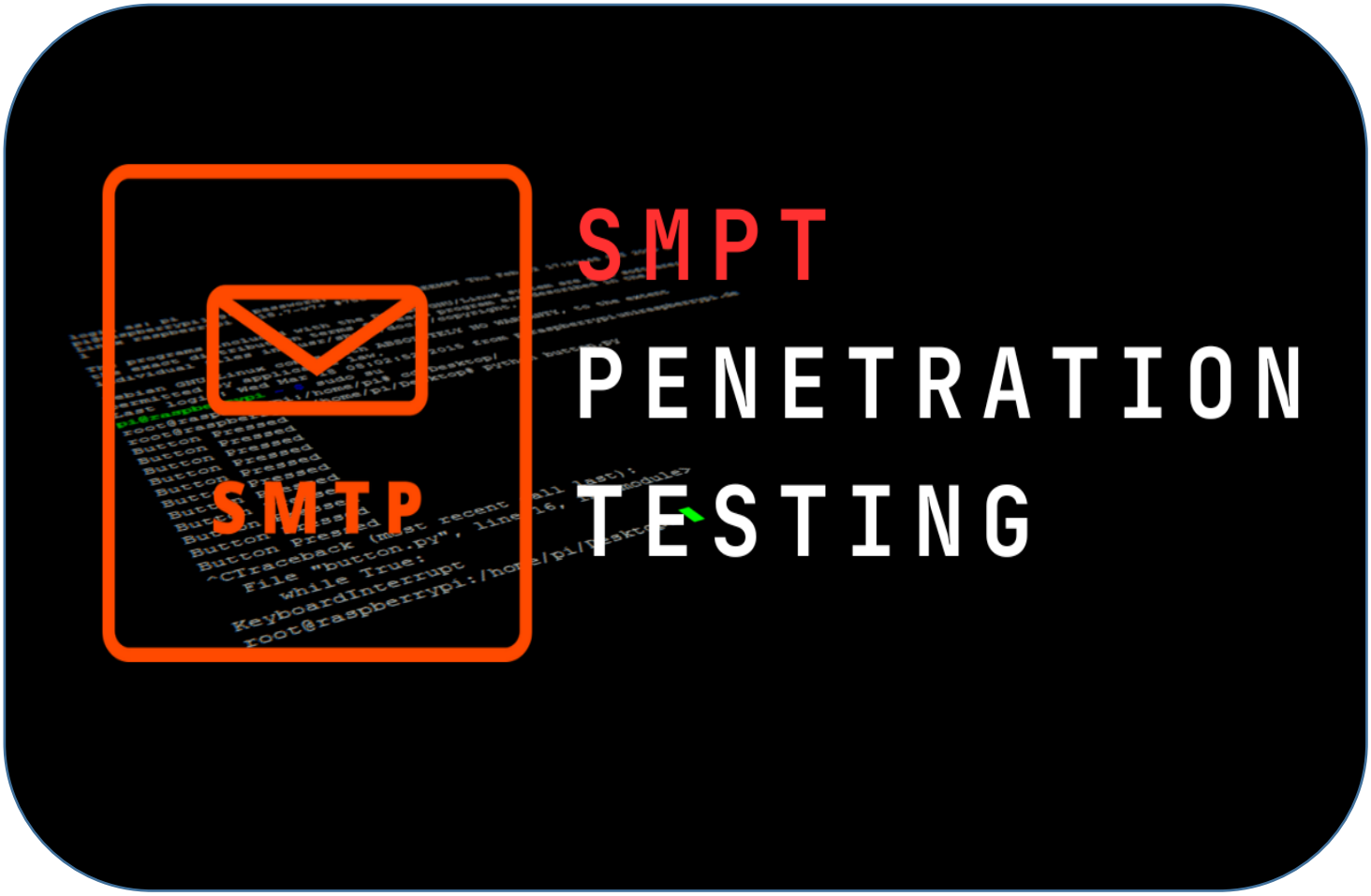

SMTP
SMPT
PENETRATION
TESTING

SMTP Penetration Testing Research Report

Table of Contents

1. Executive Summary

This report explores the process of penetration testing on Simple Mail Transfer Protocol (SMTP) servers. SMTP is a fundamental protocol in internet communications, enabling email exchanges between servers. However, poorly configured or outdated SMTP servers can be vulnerable to several security risks, including unauthorized access, user enumeration, and spam relay attacks.

Penetration testing helps identify these vulnerabilities and enables organizations to secure their SMTP infrastructure. This report covers essential SMTP vulnerabilities, techniques such as banner grabbing and user enumeration, as well as more advanced attack vectors like brute force attacks and SMTP relay exploitation. Practical exercises are provided to demonstrate these techniques in a controlled environment.

2. Introduction to SMTP

The Simple Mail Transfer Protocol (SMTP) is an application-layer protocol that facilitates the sending and receiving of emails between mail servers. Developed in the early 1980s, SMTP remains the backbone of email delivery systems. The protocol uses a straightforward, client-server architecture where a mail client sends a message to an SMTP server, which in turn relays that message to the recipient's mail server.

SMTP commonly operates on port 25, though it may also use ports 465 (for SMTP over SSL) or 587 (for secure, authenticated transmission). The protocol works by establishing a connection between mail servers, issuing commands to initiate and manage email transfers, and reporting on the status of the delivery process.

Key Features of SMTP:

- Simple and reliable
- Supports plain text transmission by default, but can be secured using TLS/SSL
- Widely supported across the internet

- Typically relies on other protocols, such as POP3 or IMAP, for message retrieval by end-users

SMTP Security Concerns: While SMTP is a necessary part of the modern internet, it was not designed with security in mind. SMTP is vulnerable to numerous threats, such as man-in-the-middle (MITM) attacks, email spoofing, and open relay abuses. These risks make it essential for organizations to adopt security practices that protect their mail servers from attack.

3. SMTP Architecture and Workflow

SMTP works by establishing a connection between two mail servers using a client-server model. When an email is sent, the client connects to the SMTP server of the sender's domain, which then routes the message to the recipient's mail server. The recipient's server accepts the message and forwards it to the user, usually via protocols like POP3 or IMAP.

SMTP's architecture can be broken down into three core components:

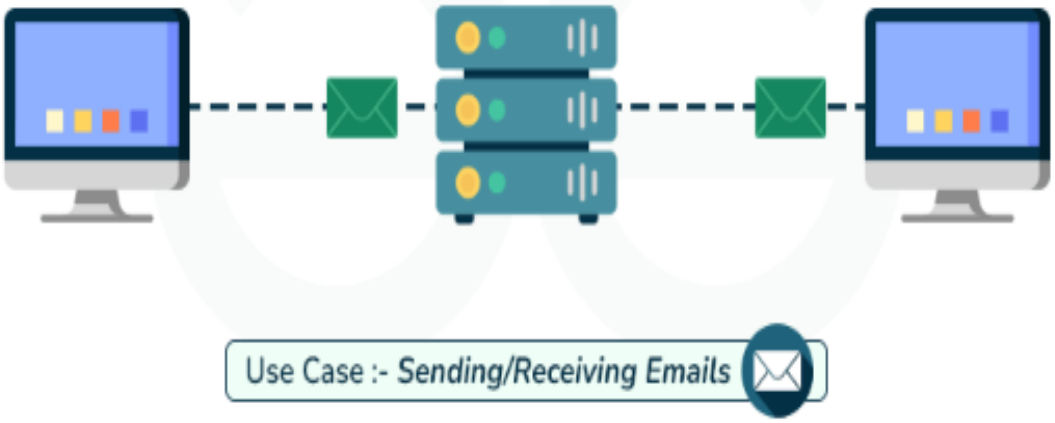
- **Mail Transfer Agent (MTA):** Responsible for transferring email between servers.
- **Mail Delivery Agent (MDA):** Handles the actual delivery of the email to the recipient's mailbox.
- **Mail User Agent (MUA):** The email client software used by the end user.

SMTP Process Workflow:

1. **Sender Initialization:** The user sends an email from their MUA (e.g., Gmail or Outlook).
2. **Outgoing Mail:** The MUA forwards the email to the MTA of the sender's domain.
3. **Routing:** The sender's MTA determines the recipient's mail server using DNS (Domain Name System) and MX (Mail Exchange) records.
4. **Receiving Mail:** The recipient's MTA accepts the message and hands it off to the MDA.
5. **Email Retrieval:** The MUA on the recipient's side retrieves the email via IMAP or POP3 for the end user to read.

The use of DNS for determining MX records is a critical element of how SMTP operates. Misconfigured MX records can result in emails being routed incorrectly, causing service interruptions or delivering messages to the wrong recipient.

SMTP



Use Case :- *Sending/Receiving Emails*

4. Common SMTP Vulnerabilities

Despite its importance, many SMTP servers are poorly configured or outdated, making them a prime target for attackers. Below are some of the most common vulnerabilities found in SMTP servers:

1. Open Relays:

SMTP servers that are configured as open relays allow anyone to send emails through them without authentication. This enables attackers to abuse the server to send spam or malicious emails.

2. Banner Disclosure:

Many SMTP servers disclose too much information in their initial greeting banner. This information often includes the server's software version, making it easier for attackers to identify vulnerabilities.

3. Weak or No Authentication:

SMTP servers may allow users to send mail without authentication or may use weak authentication mechanisms. This can lead to unauthorized users accessing the server or sending emails as legitimate users.

4. Lack of Encryption:

When SMTP communication is not encrypted, the contents of the email can be intercepted by attackers. This vulnerability allows for man-in-the-middle attacks, where the attacker can read or alter the message before it reaches the recipient.

5. User Enumeration:

Improperly configured SMTP servers may allow attackers to verify whether a specific email address exists on the system by using commands such as VRFY or EXPN. These commands provide a mechanism for attackers to gather valid email addresses for future attacks.

Mitigation Strategies:

- Disable open relays by requiring proper authentication.
- Limit the information disclosed in SMTP banners.
- Use secure authentication mechanisms (e.g., SASL).
- Enforce TLS/SSL for secure communication.
- Disable the VRFY and EXPN commands to prevent user enumeration.

5. Techniques for Banner Grabbing in SMTP

Banner grabbing is a technique used to gather information about the services running on a network, specifically their versions and configurations. In the context of SMTP, banner grabbing refers to obtaining the initial greeting banner from the server, which often contains the server's software version and other useful information for attackers.

Common Tools for Banner Grabbing:

1. **Telnet**:

   Telnet can be used to manually connect to the SMTP server and retrieve the banner.

   Command: `telnet <Target_IP> 25`

2. **Netcat (nc):**

   Another tool for banner grabbing is Netcat, which works similarly to Telnet but is more versatile.

   Command: `nc <target_IP> 25`

3. **Nmap**:

   Nmap can automate the process of banner grabbing by using its version detection feature.

   Command:

   ```
   nmap –sV  -p 25 <target_IP>
   ```

Each of these methods can expose critical information about the server software, including the type and version, which an attacker can use to identify known vulnerabilities in that software.

6. Methods for User Enumeration in SMTP

User enumeration is a crucial step in the SMTP penetration testing process. This technique involves using SMTP commands to determine whether certain users exist on the mail server. Attackers can use this information to build a list of valid email addresses for further attacks such as phishing, spam, or brute force.

Key SMTP Commands Used for User Enumeration:

1. **VRFY (Verify)**:

   The VRFY command asks the server to confirm if a specific user exists. If the server responds with "250 OK," it means the user is valid.

   Example: `VRFY admin@example.com`

2. **EXPN (Expand)**:

   The EXPN command requests the server to reveal the members of a mailing list. If a mailing list is specified, the server will return the addresses of all users subscribed to it.

   Example: `EXPN staff@example.com`

3. **RCPT TO (Recipient To):**

   While primarily used to specify a recipient during email delivery, the RCPT TO command can sometimes be used for user enumeration if the server responds differently when a valid and an invalid user are provided.

   Example:

```
RCPT TO:<invalid_user@example.com>
```

Tools for Automated User Enumeration:

- **Metasploit**:

  The auxiliary/scanner/smtp/smtp_enum module in Metasploit can be used to automate user enumeration by sending VRFY or EXPN commands to the target server.

  Command:

  ```
  use auxiliary/scanner/smtp/smtp_enum
  set RHOSTS <target_ip>
  set USER_FILE /path/to/usernames.txt
  run
  ```

- **Nmap SMTP Enumeration Script**:

  Nmap includes an SMTP enumeration script (smtp-enum-users) that can automate the process of verifying users.

  Command:

  ```
  nmap --script smtp-enum-users -p <target_ip>
  ```

- **SMTP User Enum Tool**:

  This tool specifically targets SMTP servers for user enumeration using VRFY and EXPN commands.

  Command:

  ```
  smtp-user-enum -M VRFY -U /path/to/userlist.txt -t <target_ip>
  ```

Mitigating User Enumeration Risks:

- Disable VRFY and EXPN commands on the SMTP server.
- Use proper access controls and rate limiting to prevent automated attacks.
- Employ sender verification techniques such as SPF, DKIM, and DMARC to mitigate the risk of email-based attacks following user enumeration.

7. Advanced Enumeration Techniques and Tools

Attackers often use more advanced enumeration techniques when the basic VRFY and EXPN commands are disabled. These techniques can include:

- **Timing-based enumeration**:
  The attacker sends an email using RCPT TO commands for different users and measures the time taken for the server to respond. A slower response for invalid users can indicate successful user enumeration.

  Example:

  `RCPT TO:valid_user@example.com`

  Response time: 150ms

  RCPT TO:invalid_user@example.com

  Response time: 100ms

- **SMTP Response Code Analysis**:

  Even when VRFY and EXPN are disabled, variations in the SMTP server's response codes can indicate whether a user exists. For example:

  - `250 OK`: Valid user.
  - `550 No such user`: Invalid user.

- **Email Headers**:

  Analyzing email headers can sometimes reveal internal information about the mail server, including valid email addresses or internal forwarding addresses.

Tools for Advanced Enumeration:

- **SMAP**:
  An enumeration tool that focuses on identifying open relays and valid users, even in secured environments.
- **Burp Suite**:
  Although primarily used for web application testing, Burp Suite can be configured to perform SMTP enumeration through its Intruder module, allowing for advanced manipulation of SMTP requests.

Mitigation Strategies:

- Implement uniform response times and messages for both valid and invalid users to avoid timing attacks.
- Restrict access to SMTP servers and disable unnecessary commands.

8. SMTP Relay Attacks

An SMTP relay attack occurs when an attacker takes advantage of an improperly configured SMTP server that allows unauthorized third parties to send emails through it (known as an open relay). These attacks often result in the server being used to send spam or malicious emails.

How Open Relays Work:

An open relay SMTP server does not enforce proper authentication and allows any user to send emails to external addresses. Attackers can abuse this by sending bulk emails through the server, thereby obscuring their identity.

SMTP Relay Attack Process:

1. The attacker identifies an SMTP server with open relay functionality.
2. They craft an email with a spoofed sender address and send it via the open relay server.
3. The SMTP s
4. erver forwards the email to the recipient as though it came from the spoofed address, potentially bypassing spam filters.

Risks of Open Relays:

- **Spam Propagation**: Attackers use open relays to send large volumes of spam.
- **IP Blacklisting**: SMTP servers that are used for spam will likely be blacklisted, preventing legitimate email delivery.
- **Phishing and Malware Delivery**: Attackers can use open relays to distribute phishing emails or malware while hiding their true identity.

Tools to Detect Open Relays:

- **Open Relay Test Tools**: Several online tools are available to test whether an SMTP server is configured as an open relay.
- **Nmap SMTP Relay Scanner**:
  Nmap can be used to scan for open relay functionality on SMTP servers.

  Command:

  ```
  nmap -p 25 --script smtp-open-relay 192.168.1.100
  ```

Mitigating SMTP Relay Attacks:

- Disable open relay functionality by configuring the SMTP server to require proper authentication.
- Use Sender Policy Framework (SPF), DomainKeys Identified Mail (DKIM), and Domain-based Message Authentication, Reporting, and Conformance (DMARC) to validate sender identities and prevent email spoofing.

9. Brute Force Attacks in SMTP

Brute force attacks on SMTP servers involve systematically guessing usernames and passwords to gain unauthorized access to the server. Attackers typically use automated tools to try multiple combinations until they find valid credentials.

Common Tools for Brute Force Attacks:

- **Hydra**:

  Hydra is a widely used tool for conducting brute force attacks on various protocols, including SMTP.

  Command:

  ```
  hydra -l user -P /path/to/passwords.txt smtp://<target_ip> -V
  ```

- **Medusa**:

  Medusa is another tool that supports SMTP brute forcing.

  Command:

  ```
  medusa -h <target_ip> -u user -P /path/to/passwords.txt -M smtp
  ```

- **Metasploit**:

  Metasploit also provides modules for conducting brute force attacks on SMTP servers.

  Command:

```
use auxiliary/scanner/smtp/smtp_login
set RHOSTS <target_ip>
set USER_FILE /path/to/usernames.txt
set PASS_FILE /path/to/passwords.txt
run
```

Preventing Brute Force Attacks:

- Implement account lockout mechanisms to prevent continuous login attempts after a certain number of failed attempts.
- Use strong, complex passwords for email accounts.
- Enforce multi-factor authentication (MFA) to add an additional layer of security.

10. Best Practices for Securing SMTP Servers

Securing SMTP servers involves a combination of proper configuration, strong authentication, and encryption. Some best practices include:

1. **Authentication**:

   Ensure that only authenticated users can send emails through the SMTP server. Use SASL (Simple Authentication and Security Layer) for this purpose.

2. **Encryption**:

   Enforce TLS/SSL to encrypt email communications and prevent data interception during transmission.

3. **Disable Unnecessary SMTP Commands**:

   Turn off commands like VRFY and EXPN to prevent user enumeration.

4. **Implement Access Controls**:

   Restrict access to the SMTP server by IP address or domain to prevent unauthorized use.

5. **Use Anti-spam Measures**:

   Deploy anti-spam tools and enable email filtering to prevent the server from being used to send spam or phishing emails.

6. **Monitor Logs**:

   Regularly review SMTP server logs for suspicious activity, such as failed login attempts or large volumes of outgoing emails.

11. Advanced Configurations for Securing SMTP Servers

To further enhance the security of an SMTP server, administrators should implement advanced configurations that go beyond basic setup. These configurations can help to reduce vulnerabilities and ensure that the server remains protected from various attack vectors.

11.1 Implementing SPF, DKIM, and DMARC

SPF (Sender Policy Framework):

SPF allows domain owners to specify which mail servers are permitted to send emails on their behalf. This helps to prevent email spoofing.

Example of an SPF record in DNS:

```
example.com.  IN TXT  "v=spf1 mx ip4:<192.168.1.100> -all"
```

DKIM (DomainKeys Identified Mail):

DKIM adds a digital signature to emails to verify that the message has not been altered in transit. It also ensures the sender's authenticity.

Example of a DKIM signature in email headers:

```
DKIM-Signature: v=1; a=rsa-sha256; d=example.com; s=key;
c=relaxed/relaxed;
h=from:to:subject:date;
```

DMARC (Domain-based Message Authentication, Reporting, and Conformance):

DMARC builds on SPF and DKIM to ensure that both are aligned and provides a way for domain owners to receive reports about fraudulent emails.

Example of a DMARC record:

```
_dmarc.example.com.  IN TXT  "v=DMARC1; p=reject;
rua=mailto:dmarc-reports@example.com"
```

11.2 TLS Encryption for SMTP

To secure email transmissions, it is crucial to enforce the use of Transport Layer Security (TLS). Without TLS, emails are transmitted in plain text and can be intercepted. Configuring an SMTP server to use STARTTLS ensures that communications between mail servers are encrypted.

STARTTLS:

STARTTLS is an extension to the SMTP protocol that allows the server to upgrade an existing insecure connection to a secure, encrypted one using SSL/TLS.

Example configuration for Postfix (an SMTP server):

```
smtpd_tls_security_level = may

smtpd_tls_auth_only = yes
```

11.3 IP-Based Access Control

Restricting which IP addresses are allowed to connect to the SMTP server can significantly reduce the risk of unauthorized access. This can be configured by using firewall rules or configuring the SMTP server directly.

Example of IP restriction in Postfix:

```
smtpd_client_restrictions = permit_mynetworks,
reject_unauth_destination
```

12. Case Study: Securing an SMTP Server in a Real-World Scenario

This case study demonstrates the process of securing an SMTP server for a medium-sized organization. The organization had been experiencing issues with spam being sent from its mail server, leading to IP blacklisting and delivery issues with legitimate emails.

12.1 Problem Overview

- The organization's SMTP server was an open relay, allowing unauthorized users to send emails through it.
- There were no SPF, DKIM, or DMARC records set up, making the organization's domain vulnerable to spoofing attacks.
- The server did not enforce TLS encryption for email transmission, leaving emails vulnerable to interception.
-

12.2 Solution Implementation

1. **Disabling Open Relay**:
   The first step was to disable open relay functionality by ensuring that only authenticated users could send emails.
   Postfix configuration:
   ```
   smtpd_recipient_restrictions = permit_sasl_authenticated, reject_unauth_destination
   ```

2. **Setting Up SPF, DKIM, and DMARC**:
   DNS records were updated to include SPF, DKIM, and DMARC, which helped to prevent spoofing and provided feedback on email deliverability.

3. **Enforcing TLS Encryption**:
   `STARTTLS` was implemented to ensure that emails were encrypted in transit.

4. **Monitoring and Logging**:
   The server was configured to log all email activity and attempts to authenticate. This provided insights into any potential brute force attempts.

12.3 Results

- Spam activity dropped significantly, and the server was removed from several blacklists.
- Legitimate emails were delivered reliably, and phishing attacks targeting the organization's domain decreased due to the SPF, DKIM, and DMARC setup.

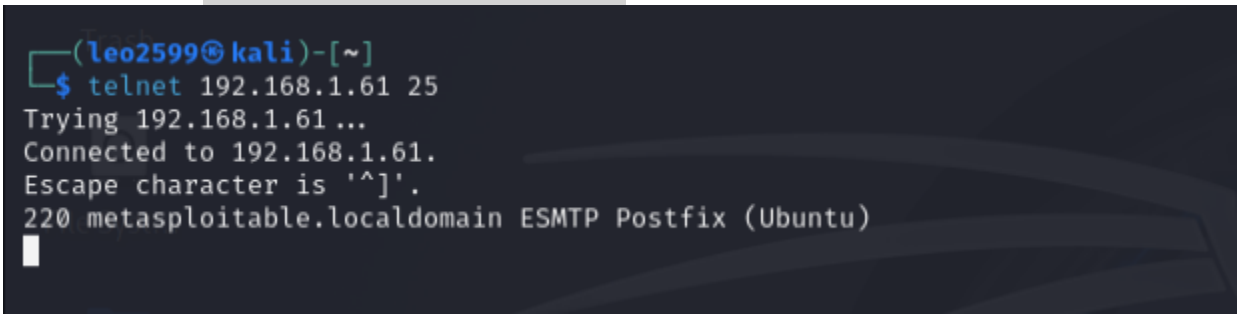
13. Practical Exercises with Metasploitable 2 VM: SMTP Testing

Target: Metasploitable2 VM [192.168.1.61] (SMTP Service)

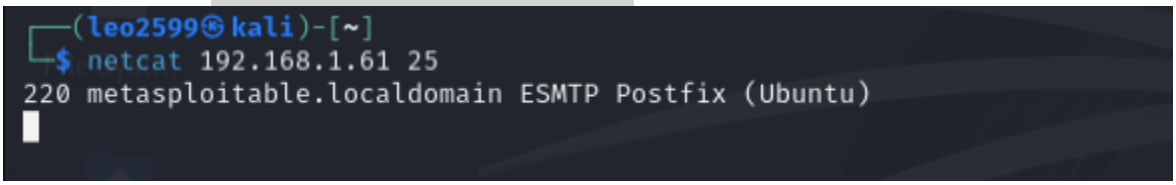
In this practical exercise, we will perform banner grabbing and user enumeration on an SMTP service running on a Metasploitable2 VM.

Banner Grabbing on the SMTP Server

1. **Using Telnet**:
   Command: `telnet 192.168.1.61 25`



2. **Using Netcat**:
   Command: `netcat 192.168.1.61 25`



3. **Using Nmap**:
   Command: `nmap -sV -p 25 192.168.1.61`

```
┌──(leo2599㉿kali)-[~]
└─$ nmap -sV -p 25 192.168.1.61
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 15:08 IST
Nmap scan report for 192.168.1.61 (192.168.1.61)
Host is up (0.0049s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
Service Info: Host:  metasploitable.localdomain

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.39 seconds

┌──(leo2599㉿kali)-[~]
└─$
```

NMAP Script Scanning

Command: `nmap 192.168.1.61 -p 25 -sV -sC`

```
┌──(leo2599㉿kali)-[~]
└─$ nmap 192.168.1.61 -p 25 -sV -sC
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 15:19 IST
Nmap scan report for 192.168.1.61 (192.168.1.61)
Host is up (0.011s latency).

PORT   STATE SERVICE VERSION
25/tcp open  smtp    Postfix smtpd
|_ssl-date: 2024-09-18T09:49:40+00:00; +3s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_    SSL2_RC4_128_WITH_MD5
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
Service Info: Host:  metasploitable.localdomain

Host script results:
|_clock-skew: 2s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.14 seconds

┌──(leo2599㉿kali)-[~]
```

User Enumeration on the SMTP Server

1. **Using Telnet (VRFY Command)**:

   Command: `telnet 192.168.1.61 25`

   Once connected, use the following command:
   VRFY msfadmin

   Output

```
┌──(leo2599㉿kali)-[~]
└─$ telnet 192.168.1.61 25
Trying 192.168.1.61 ...
Connected to 192.168.1.61.
Escape character is '^]'.
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
VRFY msfadmin
252 2.0.0 msfadmin
```

2. **Metasploit:**

   Commands are listed step by step

   i. Enter to Metasploit Framework

   `Msfconsole`

   ii. Search Module

   `search smtp user`

```
   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/smtp/apache_james_exec                  2015-10-01       normal     Yes    Apache James Server 2.3.2 Insecure User Creation Arb
itrary File Write
   1     \_ target: Bash Completion                          .                .          .      .
   2     \_ target: Cron                                     .                .          .      .
   3   auxiliary/scanner/http/gavazzi_em_login_loot          .                normal     No     Carlo Gavazzi Energy Meters - Login Brute Force, Ext
ract Info and Dump Plant Database
   4   exploit/linux/smtp/exim4_dovecot_exec                 2013-05-03       excellent  No     Exim and Dovecot Insecure Configuration Command Inje
ction
   5   exploit/unix/smtp/exim4_string_format                 2010-12-07       excellent  No     Exim4 string_format Function Heap Buffer Overflow
   6   exploit/unix/smtp/opensmtpd_mail_from_rce             2020-01-28       excellent  Yes    OpenSMTPD MAIL FROM Remote Code Execution
   7   exploit/unix/local/opensmtpd_oob_read_lpe             2020-02-24       average    Yes    OpenSMTPD OOB Read Local Privilege Escalation
   8   exploit/unix/smtp/qmail_bash_env_exec                 2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shel
lshock)
   9   auxiliary/scanner/smtp/smtp_enum                      .                normal     No     SMTP User Enumeration Utility
   10  exploit/windows/email/ms07_017_ani_loadimage_chunksize 2007-03-28      great      No     Windows ANI LoadAniIcon() Chunk Size Stack Buffer Ov
erflow (SMTP)
   11    \_ target: Automatic                                .                .          .      .
   12    \_ target: Windows XP SP2 user32.dll 5.1.2600.2622  .                .          .      .
   13    \_ target: Windows XP SP2 userenv.dll English       .                .          .      .
   14    \_ target: Windows XP SP2 userenv.dll French        .                .          .      .
   15    \_ target: Windows XP SP0/SP1 netui2.dll English    .                .          .      .
   16    \_ target: Windows 2000 SP0-SP4 netui2.dll English  .                .          .      .
   17    \_ target: Windows Vista user32.dll 6.0.6000.16386  .                .          .      .
   18    \_ target: Windows XP SP2 user32.dll (5.1.2600.2180) Multi Language .                .          .      .
   19    \_ target: Windows XP SP2 user32.dll (5.1.2600.2180) English .       .          .      .
   20    \_ target: Windows XP SP2 userenv.dll Portuguese (Brazil) .          .          .      .
   21    \_ target: Windows XP SP1a userenv.dll English      .                .          .      .
   22    \_ target: Windows XP SP1a shell32.dll English      .                .          .      .
   23  post/windows/gather/credentials/outlook               .                normal     No     Windows Gather Microsoft Outlook Saved Password Extr
action
```

   iii. User appropriate module

   `use auxiliary/scanner/smtp/smtp_enum`

   Otherwise we can do it by specifying serial number of the module, here it is 9

```
msf6 > use 9
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

   iv. View options

   `show options`

```
msf6 > use 9
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-me
                                                                         tasploit.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/u  yes       The file that contains a list of probable users accounts.
              nix_users.txt


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

msf6 auxiliary(scanner/smtp/smtp_enum) >
```

v. Specify Target host

`Set RHOST 192.168.1.68`

```
msf6 auxiliary(scanner/smtp/smtp_enum) > set RHOST 192.168.1.61
RHOST ⇒ 192.168.1.61
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

vi. Run

`exploit` `run`

```
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.1.61:25       - 192.168.1.61:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
```

```
msf6 auxiliary(scanner/smtp/smtp_enum) > run

[*] 192.168.1.61:25       - 192.168.1.61:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] 192.168.1.61:25       - 192.168.1.61:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc, libuuid, list, lp, mail, man, mysql, news, nobody, p
ostfix, postgres, postmaster, proxy, service, sshd, sync, sys, syslog, user, uucp, www-data
[*] 192.168.1.61:25       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

Now it is showing all usernames…

14. Conclusion

This report has covered the essentials of SMTP penetration testing, from banner grabbing and user enumeration to more advanced topics like brute force attacks and SMTP relay testing. It also provided practical exercises using the Metasploitable 2 VM, demonstrating real-world examples of SMTP vulnerabilities.

Key Takeaways:

- Always disable VRFY and EXPN commands to prevent user enumeration.
- Ensure that open relay functionality is disabled to prevent the server from being used for spam or malicious purposes.
- Implement strong authentication and encryption mechanisms, such as TLS, SPF, DKIM, and DMARC.
- Regularly test SMTP servers for vulnerabilities and apply necessary security patches.

15. References

- *OWASP SMTP Testing Guide*
- *Nmap Scripting Engine Documentation*
- *Hydra Brute Force Attack Documentation*
- *Postfix Security Configuration Guidelines*
- *Metasploit Framework: SMTP Modules*
- *RFC 5321: Simple Mail Transfer Protocol*
- *CVE Database for SMTP Vulnerabilities*
- *Advanced Penetration Testing by* ***Wil Allsop***
- *Penetration Testing Fundamentals: A Hands-On Guide to Reliable Security Audits by* ***Chuck Easttom***