WAVESTONE

SANS DFIR SUMMIT 2022
Hunting for Active Directory persistence with FarsightAD

Thomas DIOT (@_Qazeer) | CERT-W at Wavestone

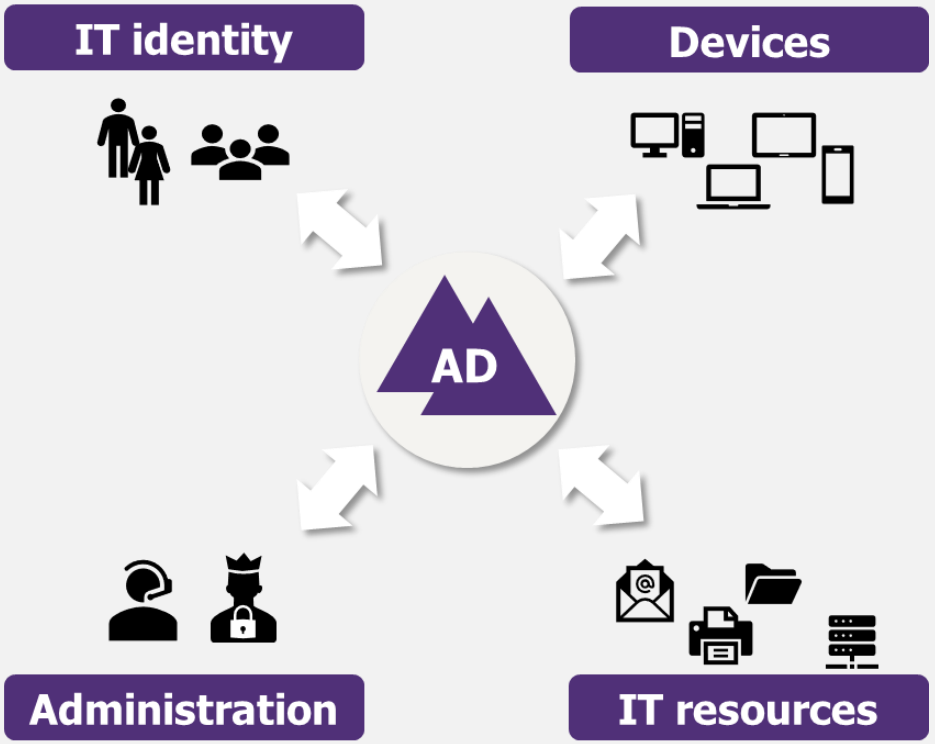
Whoami

- **Lead incident responder** at CERT-W / Wavestone
- **Pentester** / **Red Teamer**, with a focus on Windows & Active Directory
- **Open-source tools & notes author**
  EDRSandblast w/ @themaks, OffensivePythonPipeline, and InfoSec-Notes

Active Directory, a core and critical infrastructure

"The keys of the kingdom"

Active Directory is **the heart IT systems** of most major companies…



…And will most likely **continue to be.**

Created in 2000

- LDAP directory.
- For user authentication, computer and resources management.

Maintained in years

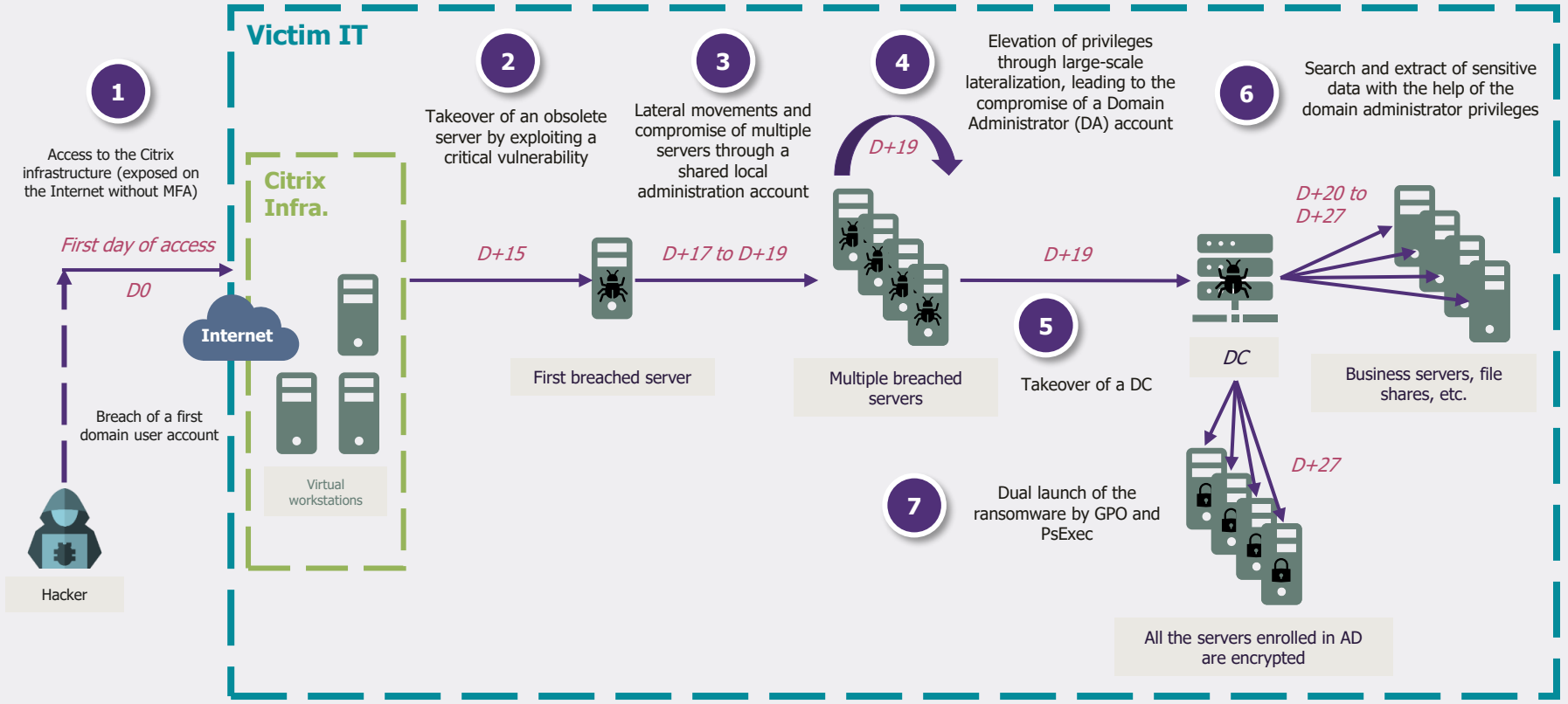
- **Windows Server 2003, 2008 (R2)**: optimization (eg. RODC, DFS for SYSVOL replication).
- **Windows Server 2012 (R2)**: security improvements (protected users, Authentication Policies and Authentication Policy Silos, …).
- **Windows Server 2016, 2019**: few evolutions.

Challenged by AAD

- But with **interconnections between AD** and **AAD** (Password Hash Sync and more recently groups writeback).

Why care about Active Directory security?

WHAT GETTING AN ACTIVE DIRECTORY COMPROMISED CAN LOOK LIKE, BASED ON A REAL-LIFE IR ENGAGEMENT.



Why care about Active Directory security?

WHAT GETTING AN ACTIVE DIRECTORY COMPROMISED CAN LOOK LIKE, BASED ON A REAL-LIFE IR ENGAG

TLP:WHITE

FBI FLASH

FEDERAL BUREAU OF INVESTIGATION • CYBER DIVISION

19 April 2022

The following information is being provided by the FBI, with no guarantees or warranties, for potential

FLASH Number

CU-000167-MW

BlackCat/ALPHV ransc
the victim system. On
administrator account
Objects (GPOs) to dep
in conjunction with Cc
BlackCat/ALPHV ransc
tools during comprom

BlackCat/ALPHV steals
providers where comp

Hacker

New data-wiping malware used in destructive attacks on Ukraine

By Lawrence Abrams — February 23, 2022 — 05:31 PM — 0

Cybersecurity firms have found a new data wiper used in destructive attacks today against Ukrainian networks just as Russia moves troops into regions of Ukraine.

ESET warned that in at least one of these attacks, it was not targeted at individual computers and was deployed directly from the Windows domain controller.

This indicates that the threat actors had access to these networks for some time.

"In one of the targeted organizations, the wiper was dropped via the default (domain policy) GPO meaning that attackers had likely taken control of the Active Directory server," explains ESET.

Ohio hospital diverting ambulances, canceling appointments amid cyberattack

Jessica Davis November 12, 2021

3

al movement
promise of multiple
ugh a
cal
account

D+19

D+19

D+19

D+20 to
D+27

5

ansomware now encrypts Windows domains using group

rams — July 27, 2021 — 05:10 PM — 1

of the LockBit 2.0 ransomware has been found that automates the encryption of a
ain using Active Directory group policies.

All the servers enrolled in AD are encrypted

The domain is compromised, what's next?

Building a new secure domain controller and replicating existing objects (for each domain)*

1. **Create a new hardened infrastructure**
   Hardened Windows server and privileged access workstation(s) in an isolated network.
2. **Move the existing primary domain controller into the isolated network**
   Remove all other domain controllers.
3. **Clean Active Directory objects**
   Reset accounts passwords, identify and remove attacker persistence.
4. **Promote the new harden server to DC**
5. **Promote as PDC the new DC and shutdown the former PDC**
   Eventually do a second "pivot", by doing a second round of replication from the new PDC to a second fresh DC.
6. **Harden the Active Directory forest / domain**
   Audit the current configuration and implement Tier 0.
7. **…**

Benefits

- Faster than rebuilding "from scratch".
- Shorter service interruption.

Drawbacks

- Attacker persistence may still be present in the active directory.
- Even if backups are restored, forensic analysis are required to ensure the backups are in a safe state.

** If backups have been compromised.*
Otherwise (nonauthoritative) restore of AD DS and an (authoritative) restore of SYSVOL of the first writeable DC in the domain can be performed from backups.

The lack of historical data is often the first challenge to overcome in commercial IR to identify AD persistence

Quick rotation of Security event logs on Domain Controllers

Historical data often limited to a few hours, with new events continuously overwriting previous ones.

```
PS > $LastEvent = Get-WinEvent -Path "Security.evtx" -MaxEvents 1
PS > $FirstEvent = Get-WinEvent -Path "Security.evtx" -MaxEvents 1 -Oldest
PS > $LastEvent

   ProviderName: Microsoft-Windows-Security-Auditing

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
12/04/2022 15:00:00           4768 Information      A Kerberos authentication ticket (TGT) was requested.…

PS > $FirstEvent

   ProviderName: Microsoft-Windows-Security-Auditing

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
12/04/2022 12:42:44           4624 Information      An account was successfully logged on.…
```

Lack or partial centralization of Domain Controllers logs

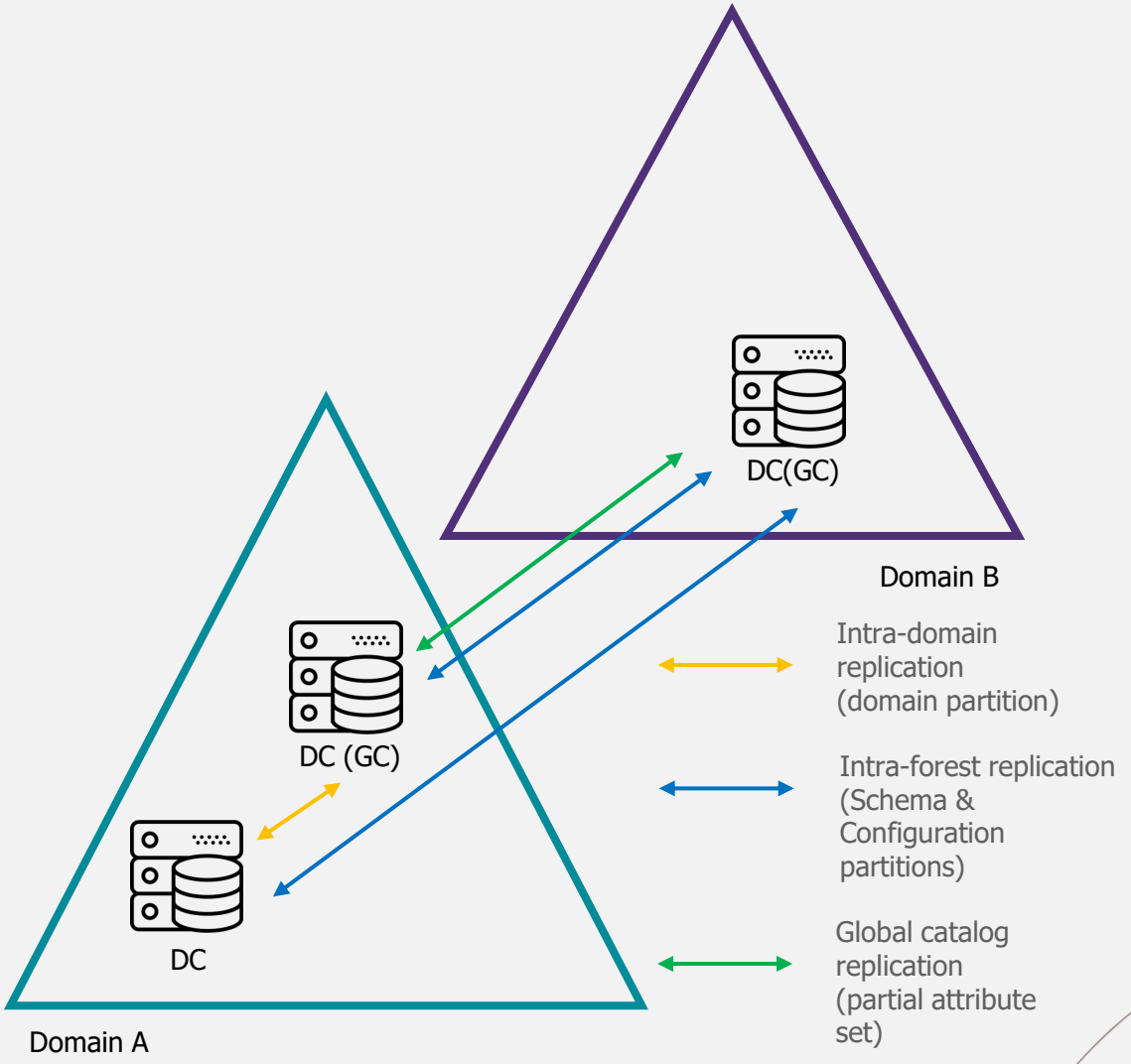
In the absence of Windows Event Forwarding (or other logs forwarder) to centralize logs in a SIEM or with a limited number of event types being centralized (logon events only for example).

Absence of Advanced Audit Policy

The default audit policy offers a good baseline (for example using "4738: A user account was changed" events to detect users attribute changes) but is limited on certain aspects (security descriptor updates, certificate requests and issuances, …).

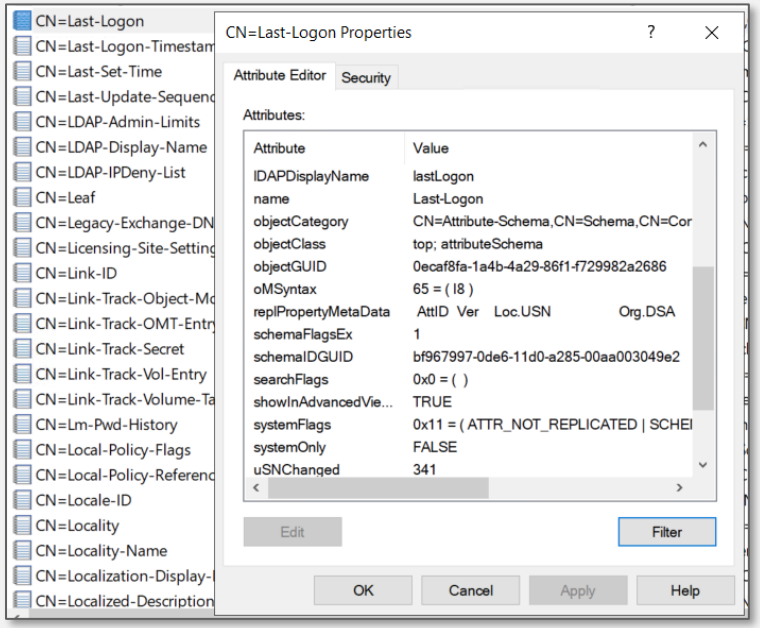
AD replication metadata to the rescue

- Used by AD DS to **replicate objects across the domain and forest**, through the Directory Replication Service (DRS) RPC protocol.
- Huge value to **identify** and **timestamp object's attributes modifications**, for instance for persistence, **in the absence of logs**.
- Nothing new, the use of replication metadata for DFIR has been known for a few years:
  - "Metadata, what is it and why do we care?" by Pierre Audonnet (piaudonn)
  - "Metadata de réplication et analyse Forensic Active Directory (fr-FR)" by Grégory LUCAND (@Greg_Lucand)
  - "Hunting With Active Directory Replication Metadata" by Will Schroeder (@harmj0y)
- **ADTimeline** by ANSSI to **generate a timeline of Active Directory changes using replication metadata**.
  Presented at SANS DFIR in 2021.



AD replication metadata - *msDS-Repl**Attribute**MetaData*

- Constructed attribute (i.e., attribute constructed by the DC at the time of request) that specifies a **list of metadata for each "regular" replicated attribute of an object**.
- Notably contains the **timestamp of modification and DC from which the modification originated**.
- **Not all attributes are replicated**. Attributes with their systemFlags (as defined in the Schema partition) having:
  - The flag "FLAG_ATTR_NOT_REPLICATED" (0x1) set are not replicated
  - The flag "FLAG_ATTR_REQ_PARTIAL_SET_MEMBER" (0x2) set are only replicated on Global Catalog




```
Get-ADObject "CN=Administrator,CN=Users,DC=adhunting,DC=lab" -Properties "msDS-ReplAttributeMetaData" `
>> | Select-Object -ExpandProperty "msDS-ReplAttributeMetaData"
<DS_REPL_ATTR_META_DATA>
        <pszAttributeName>lastLogonTimestamp</pszAttributeName>
        <dwVersion>1</dwVersion>
        <ftimeLastOriginatingChange>2022-06-13T22:42:23Z</ftimeLastOriginatingChange>
        <uuidLastOriginatingDsaInvocationID>ab025830-75ef-4c59-ba71-5ef71c33f859</uuidLastOriginatingDsaInvocationID>
        <usnOriginatingChange>20507</usnOriginatingChange>
        <usnLocalChange>20507</usnLocalChange>
        <pszLastOriginatingDsaDN>CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,
DC=adhunting,DC=lab</pszLastOriginatingDsaDN>
</DS_REPL_ATTR_META_DATA>
```

AD replication metadata - *msDS-Repl**Value**MetaData*

- Similar to msDS-ReplAttributeMetaData but for **replication metadata of linked value attributes**.
- **Linked attributes are an associated pair of attributes**, with **one forward link attribute** and **one back link attribute**. The value of the forward link attribute is stored while the value of the back link attribute is constructed (from the value of the forward link attribute).
- For example, the **member** and **memberOf attribute are linked attributes**, with member being the forward link attribute.

name	ldapdisplayname	linkid
Member	member	2
Is-Member-Of-DL	memberOf	3

- Same data as msDS-ReplAttributeMetaData with **metadata for each value of an attribute**.

  For the member attribute, the DistinguishedName of each current and past members of the group will be stored.

```
PS C:\> Get-ADReplicationAttributeMetadata -ShowAllLinkedValues -Server DC1.ADHunting.lab "CN=Domain Admins,CN=Users,DC=adhunting,DC=lab" -Properties member


AttributeName                           : member
AttributeValue                          : CN=user_da_removed,CN=Users,DC=adhunting,DC=lab
FirstOriginatingCreateTime              : 6/15/2022 12:05:09 PM
IsLinkValue                             : True
LastOriginatingChangeDirectoryServerIdentity     : CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=adhunting,DC=lab
LastOriginatingChangeDirectoryServerInvocationId : ab025830-75ef-4c59-ba71-5ef71c33f859
LastOriginatingChangeTime               : 6/15/2022 12:05:30 PM
LastOriginatingChangeUsn                : 24657
LastOriginatingDeleteTime               : 6/15/2022 12:05:30 PM
LocalChangeUsn                          : 24657
Object                                  : CN=Domain Admins,CN=Users,DC=adhunting,DC=lab
Server                                  : DC1.adhunting.lab
Version                                 : 2

AttributeName                           : member
AttributeValue                          : CN=user_da2,CN=Users,DC=adhunting,DC=lab
FirstOriginatingCreateTime              : 6/15/2022 11:58:27 AM
IsLinkValue                             : True
LastOriginatingChangeDirectoryServerIdentity     : CN=NTDS Settings,CN=DC1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=adhunting,DC=lab
LastOriginatingChangeDirectoryServerInvocationId : ab025830-75ef-4c59-ba71-5ef71c33f859
LastOriginatingChangeTime               : 6/15/2022 11:58:27 AM
LastOriginatingChangeUsn                : 24637
LastOriginatingDeleteTime               : 12/31/1600 4:00:00 PM
LocalChangeUsn                          : 24637
Object                                  : CN=Domain Admins,CN=Users,DC=adhunting,DC=lab
Server                                  : DC1.adhunting.lab
Version                                 : 1
```

Enabled by default, User Access Login can be used to identify compromised accounts 1/2

- Consolidates data on **client activity**. On Domain Controllers, yield information on **sessions opening on domain-joined computers** (if the particular DC was reached for the logon).
- Introduced, and **enabled by default**, in Windows Server 2012.
- Great value to **map lateral movements** but also to **identify compromised accounts**.
- **Historical data** going **back to 2 years** (2020 as of 2022).
- Can be parsed on live systems with **Get-Ual* PowerShell cmdlets** or, in CSV / JSON output, with **Eric Zimmerman's SumECmd.exe** utility.

Files in the "**%SystemRoot%\Windows\System32\Logfiles\SUM**" folder:

- Current.mdb which contains data for the last 24-hour.
- Systemidentity.mdb which contains metadata.
- Up to three <GUID>.mdb files, which contain data for an entire year.

```
C:\>dir /t:c %SystemRoot%\System32\Logfiles\SUM\*.mdb
 Volume in drive C is System
 Volume Serial Number is ECE5-F4BE

 Directory of C:\Windows\System32\Logfiles\SUM

08/17/2020  12:42 PM         1,048,576 Current.mdb
08/17/2020  12:42 PM         1,048,576 SystemIdentity.mdb
08/24/2020  10:09 PM         1,048,576 {944065E5-FD8B-4220-8788-DABD3C55C063}.mdb
01/21/2021  04:54 PM         1,048,576 {ACF391D5-5C63-4254-8365-68212F8B006A}.mdb
01/08/2022  06:14 PM         1,048,576 {F8B9D1CB-17FF-4B35-B8B3-E95B06C24D8A}.mdb
               5 File(s)      5,242,880 bytes
```

Enabled by default, User Access Login can be used to identify compromised accounts 2/2

Information of interest:

- **Role accessed**.
  For example: Active Directory Domain Services => GUID ad495fc3-0eaa-413d-ba7d-8b13fa7ec598.
- **Client domain** and **username**.
- **Number of access** per day and in total.
- **First**, **last**, and **daily access timestamps**.
- **Client IPv4** or **IPv6 address**.

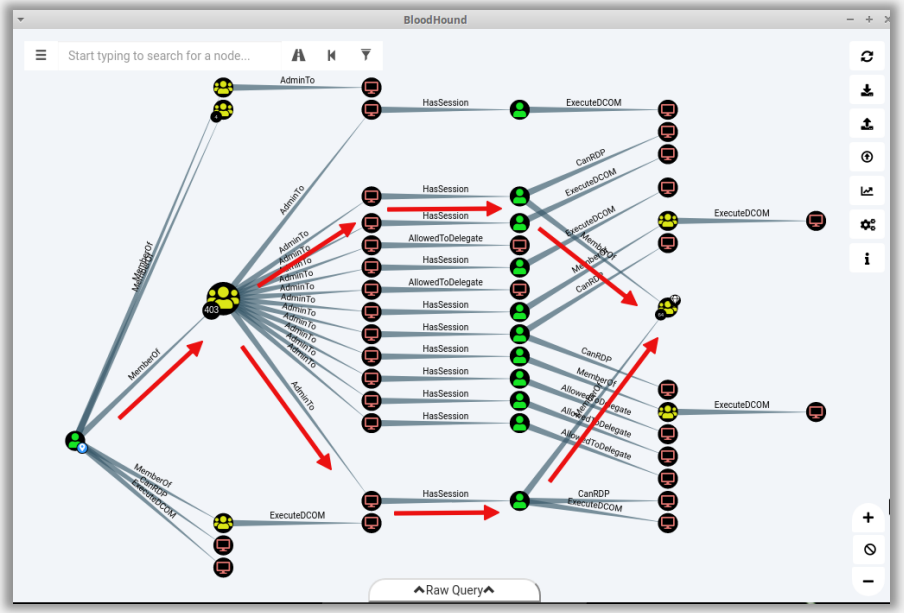
```
Get-UalDailyAccess | % {
    [PsCustomObject]@{
        AccessDate = $_.AccessDate.ToString("yyyy-MM-dd HH:mm:ss")
        AccessCount = $_.AccessCount
        UserName = $_.UserName
        IPAddress = $_.IPAddress
        RoleGuid = $_.RoleGuid
        RoleName = $_.RoleName
    }
} | Sort-Object -Property AccessDate

[...]

AccessDate  : 2021-11-28 21:42:20
AccessCount : 8
UserName    : forest1\user_ea
IPAddress   : 192.168.15.66
RoleGuid    : 10a9226f-50ee-49d8-a393-9a501d47ce04
RoleName    : File Server

AccessDate  : 2021-11-28 21:42:21
AccessCount : 114
UserName    : forest1\user_ea
IPAddress   : 192.168.15.66
RoleGuid    : ad495fc3-0eaa-413d-ba7d-8b13fa7ec598
RoleName    : Active Directory Domain Services

AccessDate  : 2021-11-28 21:42:37
AccessCount : 16
UserName    : forest1\labad-srvtools$
IPAddress   : 192.168.15.66
RoleGuid    : 10a9226f-50ee-49d8-a393-9a501d47ce04
RoleName    : File Server
```

Date	Insert Date	Count	Total Accesses	Role Description	Authenticated User Name	Ip Address	Last Access
= 2021-11-29 00:00:00	=	=	=	abc	abc	= 192.168.15.66	=
2021-11-29 00:00:00	2021-09-09 20:42:20	3	16	File Server	forest1\user_ea	192.168.15.66	2021-11-29 01:24:25
2021-11-29 00:00:00	2021-09-09 20:42:21	8	146	Active Directory Domain Services	forest1\user_ea	192.168.15.66	2021-12-09 06:10:58
2021-11-29 00:00:00	2021-09-09 20:42:37	4	179	File Server	forest1\labad-srvtools$	192.168.15.66	2021-12-23 13:45:11
2021-11-29 00:00:00	2021-09-09 20:42:38	5	164	Active Directory Domain Services	forest1\labad-srvtools$	192.168.15.66	2021-12-23 13:30:11

Evaluating the objects in the domain can help uncover pre-existing compromise paths and new persistence

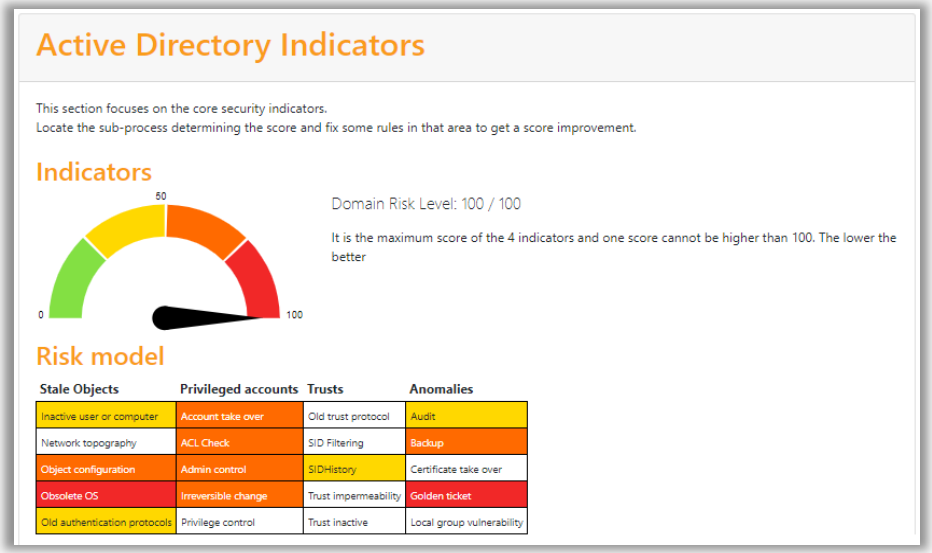
Identify **attack paths** to **privileged principals and assets (Tier 0)** from lower privilege users.

*Tool example: **BloodHound** (free and open-source version available)*



Assess the **current security level** and identify **possible "quick wins" fixes**.

*Tool example: **PingCastle** (free for usage on your own AD, subject to license for commercial use as a contractor)*

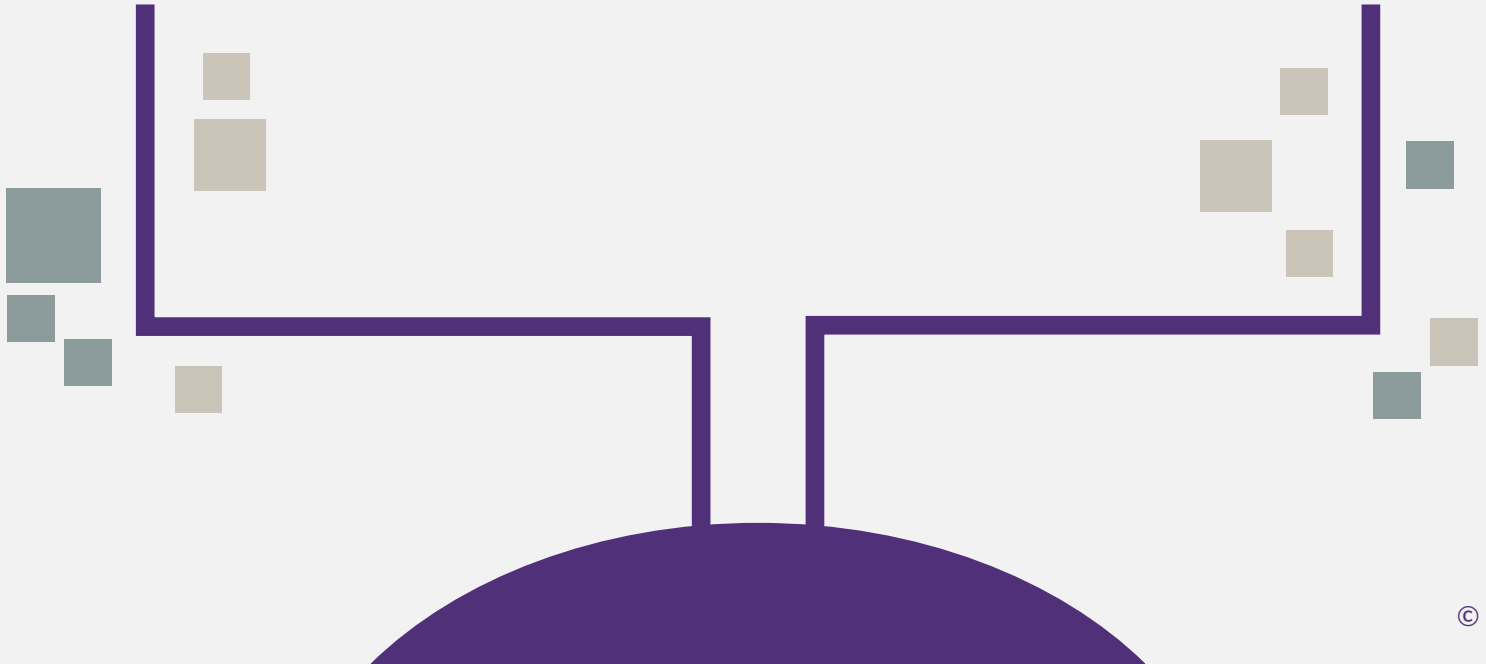


Introducing the FarsightAD PowerShell script

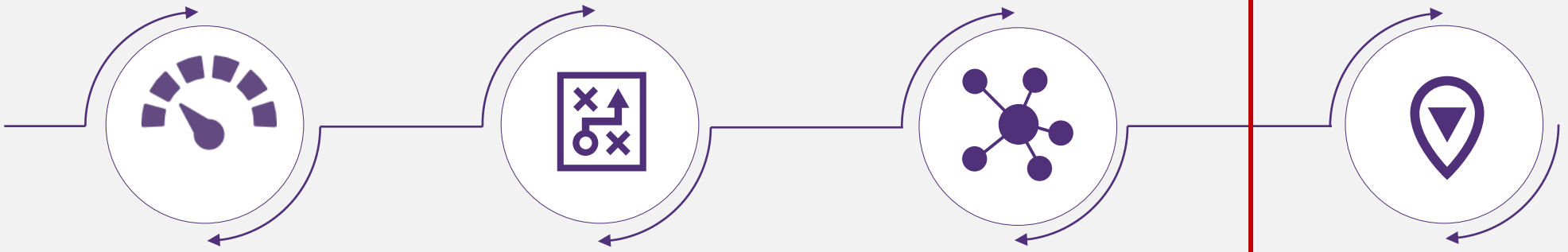
A mix of **reviewing the current configuration for persistence** and **getting historical information / timestamps** whenever possible. Aimed to **help uncovering** (eventual) **persistence mechanisms deployed by a threat actor** following an Active Directory domain compromise.

The script produces CSV / JSON file exports of AD data related to various persistence mechanisms.

https://github.com/Qazeer/FarsightAD



Active Directory kill chain overview



First user / asset compromise

- LLMNR / NBT-NS **poisoning**
- **Unauthenticated RPC users enumeration** and **password spraying**
- **System** or **applicative vulnerability** / **faulty configuration** exploit
- **Social engineering**
- **Physical access** to **unencrypted drives**
- ...

Privilege escalation

- Horizontal **brute force** or **password spraying**
- **Cleartext credentials** in network shares or objects' public attributes
- **Kerberoasting** / **ASP-REP roasting**
- **ACL-based exploits**
- **Domain Controller authentication coercion** and **relay**
- **AD Certificate Services exploits** (vulnerable template, ...)
- **Workstation mastering process** configuration flaw exploit
- ...

Lateral movement

- Domain accounts **credentials theft** and **shuffling**
- Shared **local account compromise** and **reuse**
- **Pass the Hash** : NTLM authentication
- **Pass the Ticket** : Kerberos authentication
- **GPO alteration**
- Use of **legitimate administration tools**
- ...

Domain domination and persistence

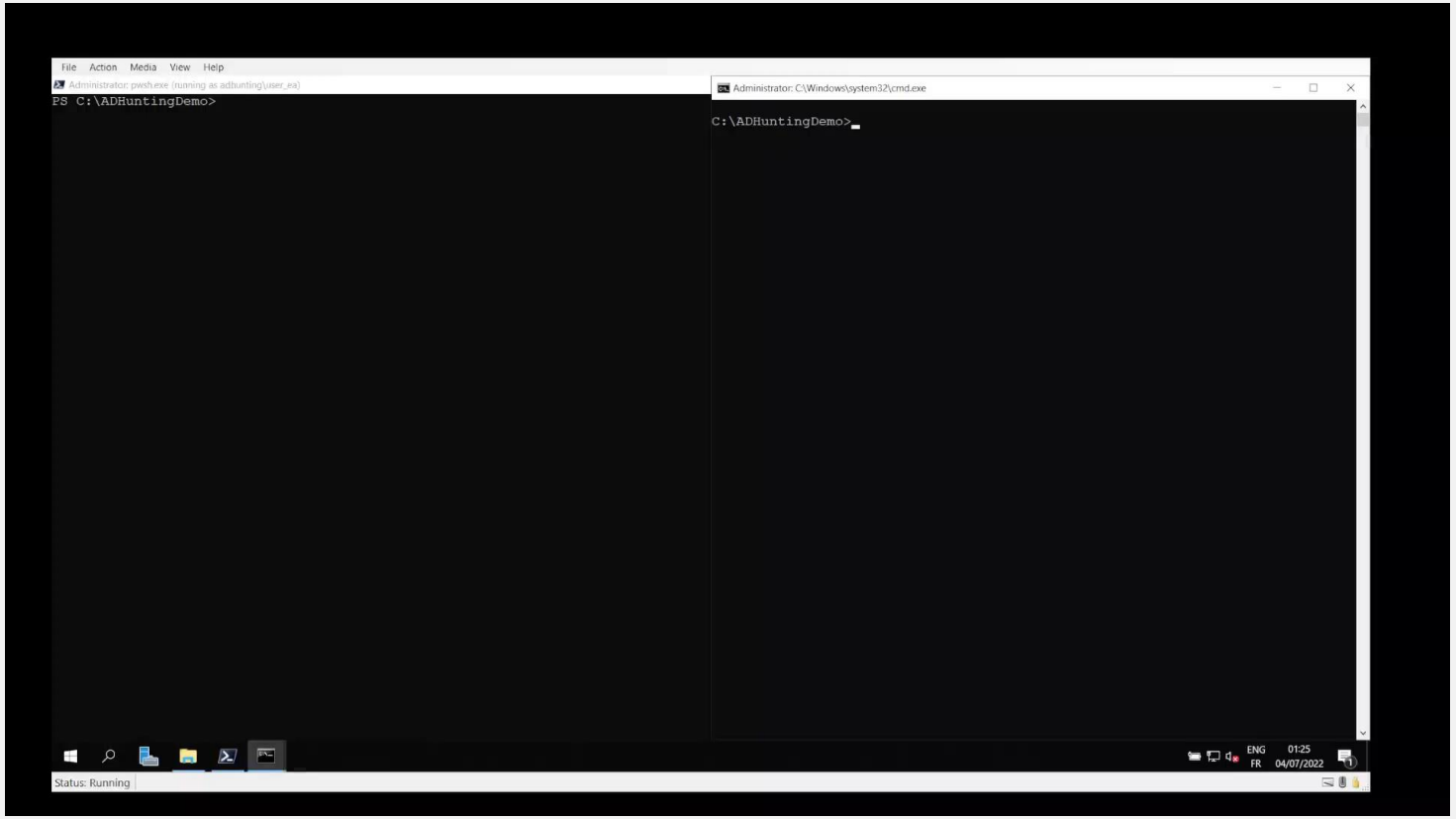
Our focus for this talk

Before analyzing objects, we must first make sure that we got the right data 1/3

Objects leveraged for persistence by a threat actor can be **fully hidden** by:

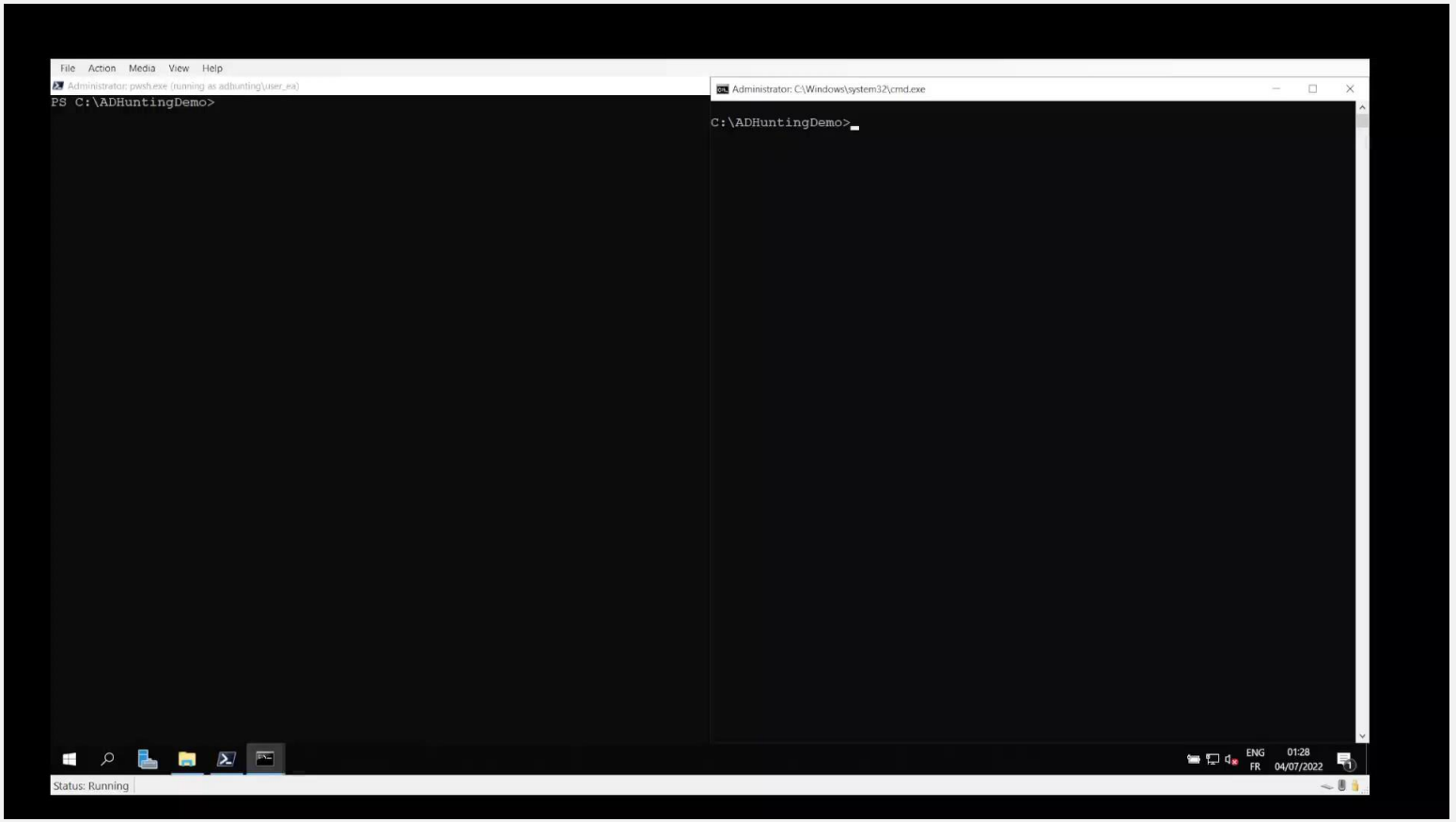
/ Setting **ReadProperty deny** or by **removing all allow ReadProperty access rights** on the object.

/ And **preventing children listing** on the parent **Organizational Unit** (through explicit deny or removal of ListChildren access right).



Before analyzing objects, we must first make sure that we got the right data 2/3

Stealthier, **a subset of an object's attribute(s)** can be **hidden**, either by setting **ReadProperty deny** or by **removing allow ReadProperty access rights** on the **specific attributes** to hide.



Before analyzing objects, we must first make sure that we got the right data 3/3

Relying only on ACL to identify hidden objects is prone to false negatives, as multiple edge cases must be considered:

- Fully hidden objects.
- Explicit ReadProperty Deny but also absence of ReadProperty Allow access right.
- Non-accessible security descriptor.
- Improper Allow / Deny ACE order
- Etc.

The **Directory Replication Service (DRS) Remote Protocol** RPC protocol can be leveraged instead, as the **objects attributes replication data** can be **retrieved independently** of the **object DACL**.

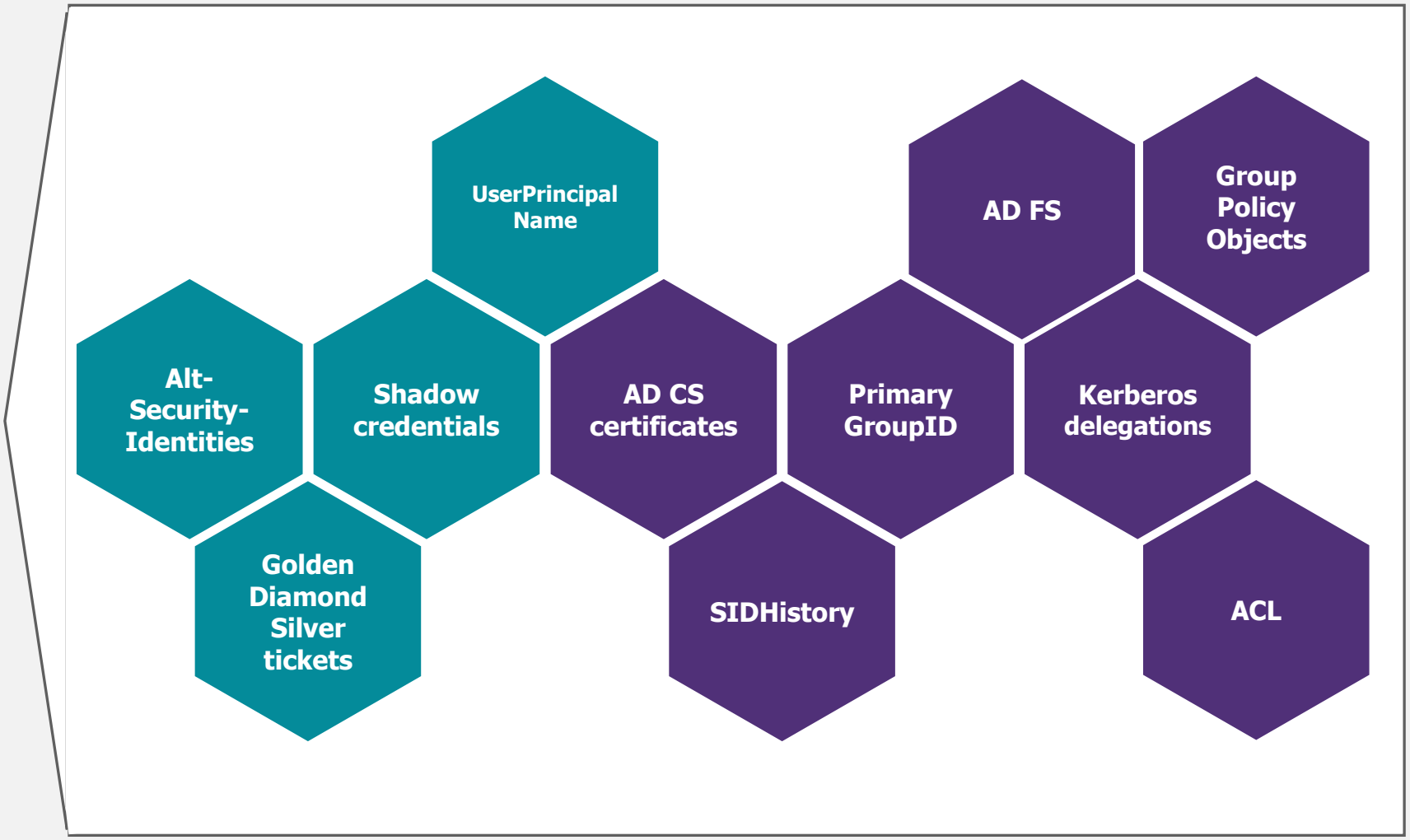
Export-ADHuntingHiddenObjectsWithDRSRepData will:

1. Retrieve replication data on a partial set of sensitive attributes for all objects.
   Implemented using code from MakeMeEnterpriseAdmin (by @vletoux), mimikatz (by @gentilkiwi & @vletoux), and SharpKatz (by @b4rtik).
2. Compare the attribute replication data with the data accessible through direct queries to identify non-accessible attributes.

Before analyzing objects, we must first make sure that we got the right data 3/3

Numerous Active Directory persistence techniques can be leveraged by a threat actor

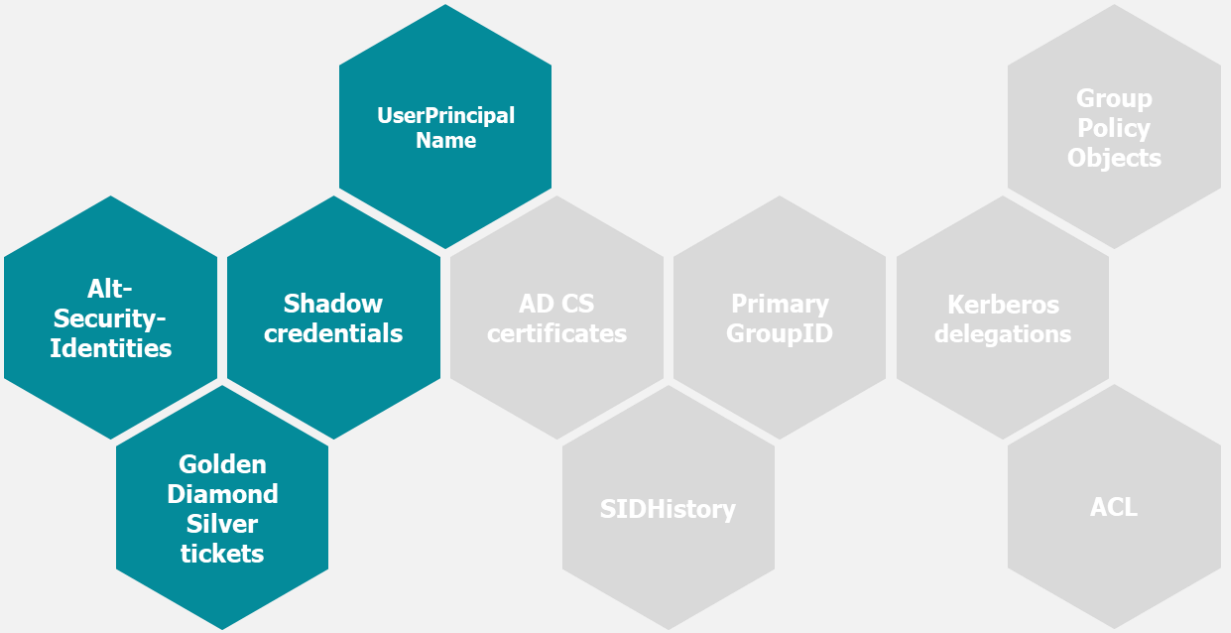
Domain persistence



01

Techniques to maintain access following passwords reset

(x2 resets for krbtgt / trust accounts)

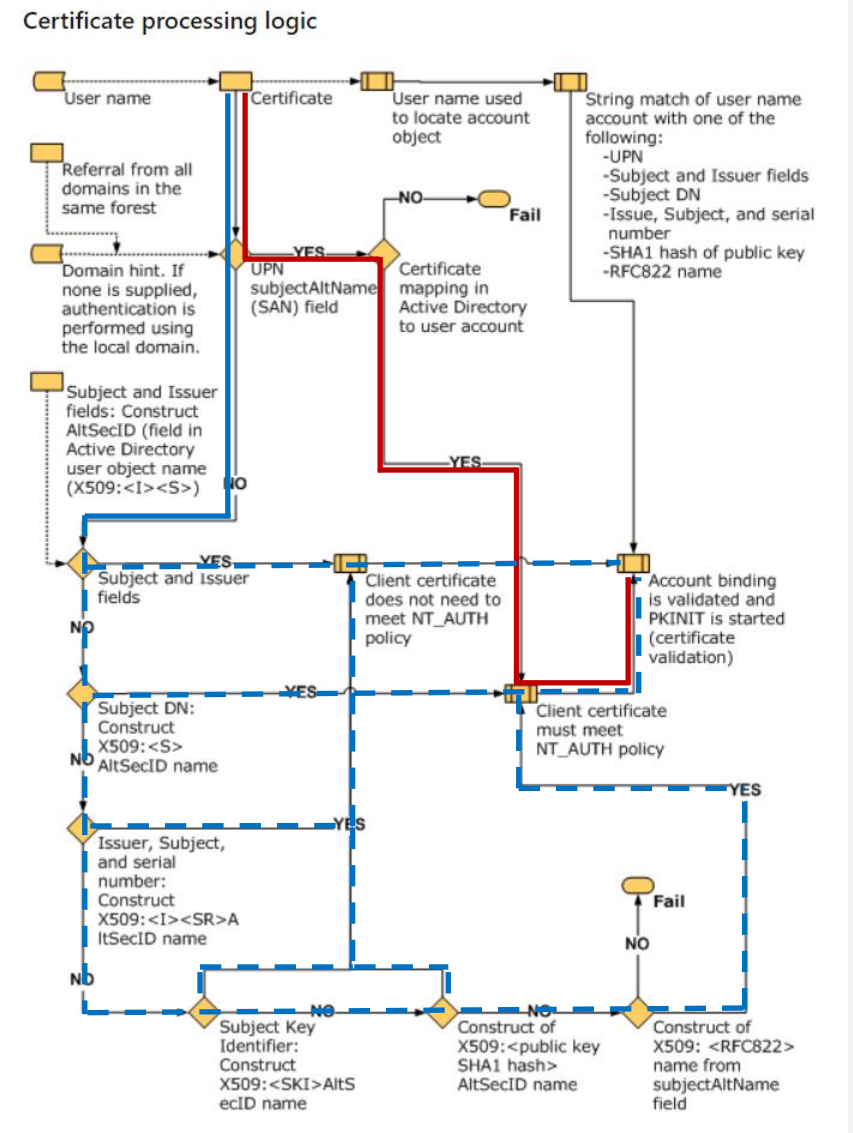


User impersonation with UserPrincipalName or Alt-Security-Identities 1/2

- The **UserPrincipalName (UPN)** or the **Alt-Security-Identities (AltSecID)** attributes of a user are used in **PKINIT authentication** (using public key cryptography as a Kerberos pre-authentication mechanism).

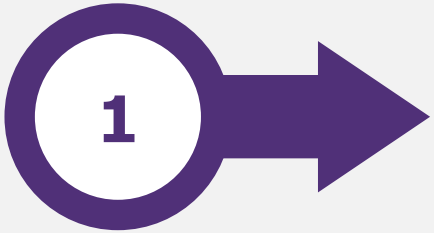
- In **implicit mapping** (no username hint provided for the authentication), the **mapping between a user** and **a certificate** is done:
  - Using the **certificate's SubjectAltName (SAN)** field and the **user's UPN**.
  - If the certificate has no SAN, based on **parameters of the certificate** (subject, issuer, and / or serial number) and the **user's AltSecID**.

- As a result, setting the **UPN** or **AltSecID attributes** of a **privileged user to match a controlled user / certificate** can be **leveraged to impersonate the privileged user** using a certificate from the controlled user / certificate.



https://docs.microsoft.com/en-us/windows/security/identity-protection/smart-cards/smart-card-certificate-requirements-and-enumeration#kdc-certificate

Exploiting UserPrincipalName or Alt-Security-Identities for persistence



UPN / AltSecId modification

An user UPN / AltSecId attribute can be **modified using "standard" AD tools** (RSAT, ADExplorer, …)

PKINIT authentication

A **PKINIT authentication** can be conducted with **tools such as Rubeus or Kekeo** and a **controlled certificate**

unPAC-the-hash (optional)

The **NTHash of the user** can be retrieved with **a subsequent U2U service ticket request** (unPAC-the-hash)

```
Rubeus.exe asktgt /user:<USERNAME> /certificate:<CERT> /password:"<CERT_PWD>" [/domain:<DOMAIN>] [/dc:<DC>]
/getcredentials /show
```

User impersonation with UserPrincipalName or Alt-Security-Identities 2/2

UserPrincipalName

- **Composed as "prefix@suffix"** and should by convention match the user email address:
  - The **UPN should match** the user's **mail attribute** or the **prefix should match** the user's **SamAccountName / mailNickName** attributes.
  - The suffix must match the DNS name of a domain in the forest or a name in the Partitions container's upnSuffixes attribute.
- **Relatively easy to spot anomalies**.

Alt-Security-Identities

- **Supported format**:
  - X509:<I><S>
  - X509:<S>*
  - X509:<I><SR>
  - X509:<SKI>
  - X509:<SHA*-PUKEY>
  - X509:<RFC822>*
- The format **may not be linked to data from the user**, thus making **spotting anomalies much harder**.

Hunting for UserPrincipalName or Alt-Security-Identities persistence as implemented in FarsightAD

Export-ADHuntingPrincipalsUPNandAltSecID

Enumerate UPN and AltSecId to **help investigation** but **doesn't replace manual analysis** (especially for AltSecId)

Highlight **UPN that do not match** the user **SamAccountName**, **mail** or **mailNickName** attribute

UPN and AltSecId timestamp of last modification of the attribute through **replication data**

Note: when in doubt, **unknown / suspicious AltSecID entries should be deleted** (especially for privileged principals).

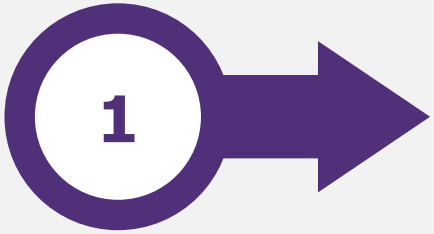
User impersonation and persistence with Shadow Credential

- A **user/computer object's msDS-KeyCredentialLink** attribute **holds Key Credentials information** for the given account.
- Introduced to **support Kerberos PKINIT authentication** in environments **without a Public Key Infrastructure** (PKI) trusted by Active Directory (such as AD CS).
- Can be used to **authenticate as a privileged account** and **retrieve its NTHash** (similarly to UPN or AltSecId based persistence).
- Presented by Michael Grafnetter (@MGrafnetter)* and later popularized by Elad Shamir (@elad_shamir)** notably to **take over account in ACL-based attacks** (as an alternative to the Kerberos RBCD exploit primitive – more on that later).

** https://www.dsinternals.com/wp-content/uploads/eu-19-Grafnetter-Exploiting-Windows-Hello-for-Business.pdf*

*** https://posts.specterops.io/shadow-credentials-abusing-key-trust-account-mapping-for-takeover-8ee1a53566ab*

Exploiting Shadow Credential for persistence



msDS-KeyCredentialLink modification

The msDS-KeyCredentialLink attribute of a user / computer object can be **modified using DSInternals** / **Whisker** / **pywhisker**

PKINIT authentication

unPAC-the-hash (optional)

Similar to UPN or AltSecId primitives

```
Whisker.exe add /target:<USERNAME> [/domain:<DOMAIN>] [/dc:<DC>] [/path:<CERT_FILE>] [/password:<CERT_PWD>]

Rubeus.exe asktgt /user:<USERNAME> /certificate:<CERT> /password:"<CERT_PWD>" /domain:<DOMAIN_FQDN> /dc:<DC>
/getcredentials /show
```

Hunting for Shadow Credential persistence as implemented in FarsightAD

/ **Multiple Key Credentials** can be added in the **msDS-KeyCredentialLink attribute** making **replication data timestamp less relevant**.

But two timestamps are among **each Key Credential**: the **key created** and **approximate last use timestamps**.

Export-ADHuntingPrincipalsShadowCredentials

Enumerate and **parse each Key Credentials**, using code from the ADComputerKeys PowerShell module*, to identify each key:

- **Source** (AD / AAD)
- **Type** (NextGenCredentials being used for user object)
- **Key created timestamps**
- **Approximate last use timestamps**

**https://www.powershellgallery.com/packages/ADComputerKeys/1.0.0/Content/ADComputerKeys.psm1*

Exploiting (AD CS) users / computers certificates for persistence

/ **Certificates** valid for client authentication can be **leveraged for persistence**, even if the **associated account password is reset**.

Extended / Enhanced Key Usage (EKU) extensions for client authentication:

- anyExtendedKeyUsage (OID 2.5.29.37.0)
- clientAuth (OID 1.3.6.1.5.5.7.3.2)
- keyPurposeClientAuth (OID 1.3.6.1.5.2.3.4)
- Smartcard logon (OID 1.3.6.1.4.1.311.20.2.2)

/ **Certificate templates published by default** allow **any authenticated users** or **computers** ("User" / "Computer" templates) to request a **certificate for client authentication valid for one year**.

Hunting for certificates persistence as implemented in FarsightAD

Export-ADHuntingPrincipalsCertificates

Enumerate and **parse users / computers certificates**, identifying certificates **valid for client authentication**

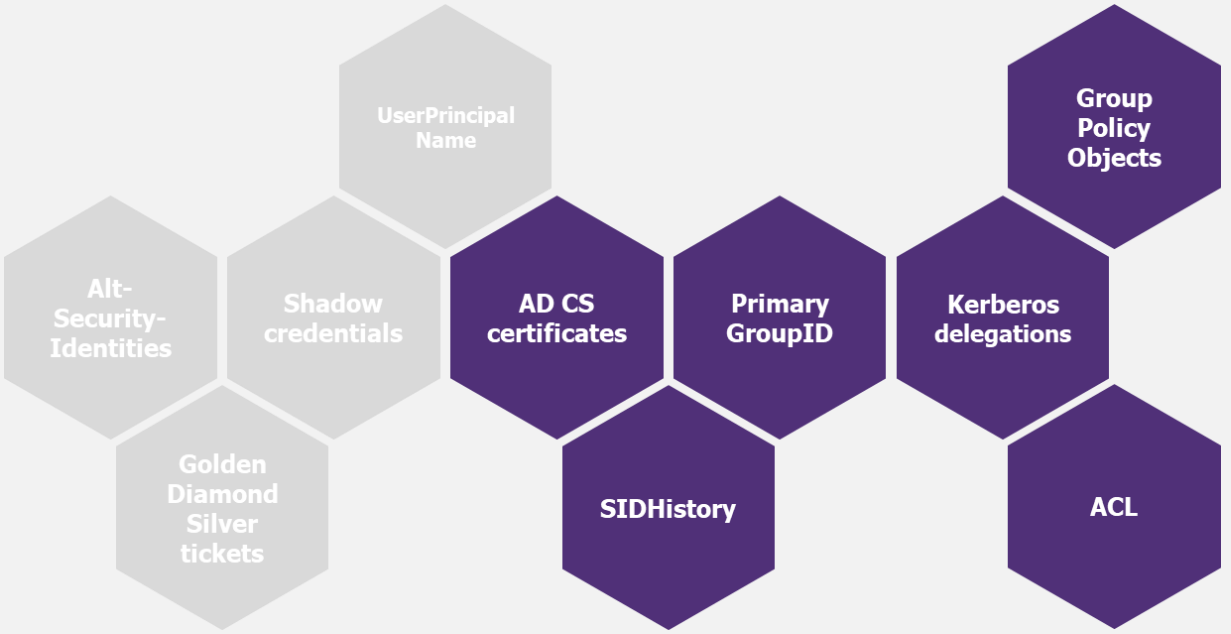
Extract **each certificate eventual SubjectAltName(s)**, determining if **any UPN do not match the current account UPN** and if the **UPN is linked to a privileged account**

Retrieve certificate validity timestamps (not before / not after) and **timestamp of last modification** of the account certificate attribute through **replication data**

Note: while new certificates can be requested by a threat actor for persistence, **already existing certificates may also have been retrieved following endpoints compromise** or the CA certificate and private key stolen to forge arbitrary certificate.
It is thus recommended to **renew the CA certificate in a forest / domain recovery procedure**.

02

Techniques to maintain a path to high privileges



What does "high privileges" mean?

/ **Privileges granted** by the **Default Domain Controllers Policy GPO** offer **escalation path** to **Domain Admins** / **Enterprise Admins** to **several built-in groups**.

Privilege	Default Domain Controllers Policy
SeDebugPrivilege	Administrators
SeBackupPrivilege	Administrators Server Operators Backup Operators
SeInteractiveLogonRight	Administrators Account Operators Server Operators Backup Operators Print Operators
SeLoadDriverPrivilege	Administrators Print Operators
SeRemoteShutdownPrivilege	Administrators Server Operators
SeRestorePrivilege	Administrators Server Operators Backup Operators
SeTakeOwnershipPrivilege	Administrators

/ **Administrators** (S-1-5-32-544) can **remotely connect to DCs** and **dump the LSASS process** / **ntds.dit database** (and takeover most AD objects including the AdminSDHolder container anyway).

/ **Backup operators** (S-1-5-32-551) and **Server Operators** (S-1-5-32-549) can **remotely connect to DCs** and **access** (backup) or **modify** (restore) **any file**, effectively bypassing access controls.

/ **Print Operators** (S-1-5-32-550) can **remotely connect to DCs** and **load drivers**. Loading a **vulnerable driver** can be leveraged to **achieve code execution in the kernel space** (to dump the ntds.dit database or takeover AD objects).

/ **Account Operators** (SID S-1-5-32-548) have **full control over user** and **machine accounts** that are **not protected** by the **AdminSDHolder mechanism** (T1 / T2 compromise).

/ **Schema admins** (RID 518) can modify the forest schema and have lasting impact or **update the objects default ACL** (impacting new objects).

/ DnsAdmins can / could remotely execute code on DCs as SYSTEM by making the DNS service execute an arbitrary DLL* but it got patched**.

*https://medium.com/@esnesenon/feature-not-bug-dnsadmin-to-dc-compromise-in-one-line-a0f779b8dc83 **https://mobile.twitter.com/cnotin/status/1467791440176726022

Using the SIDHistory / primaryGroupID attributes as shadow user impersonation or group membership 1/2

Access token

Following a login, the LSA process generates an **access token** that represent the **user security context**.

For instance, following a **Kerberos authentication to a service**, the user access token is constructed from the **PAC** of the **service ticket (ST)**, itself copied from the PAC of the TGT (used to request the ST).

The PAC contains the **user PrimaryGroupId** and **SIDs** from the **user SIDHistory***.

```
typedef struct _KERB_VALIDATION_INFO** {
  FILETIME LogonTime;
  […]
  ULONG UserId;
  ULONG PrimaryGroupId;
  ULONG GroupCount;
  [size(GroupCount)] PGROUP_MEMBERSHIP
    GroupIds;
  ULONG SidCount;
  [size(SidCount)] PKERB_SID_AND_ATTRIBUTES
    ExtraSids;
  […]
} KERB_VALIDATION_INFO;
```

**Not all SIDs are included, with filtering of SIDs for quarantined domain or across trusts with SID filtering. And only Domain Local Group SIDs from the domain of the resource are included. https://docs.microsoft.com/en-US/troubleshoot/windows-server/windows-security/logging-on-user-account-fails*

*** https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-pac/69e86ccc-85e3-41b9-b514-7d969cd0ed73*

Using the SIDHistory / primaryGroupID attributes as shadow user impersonation or group membership 2/2

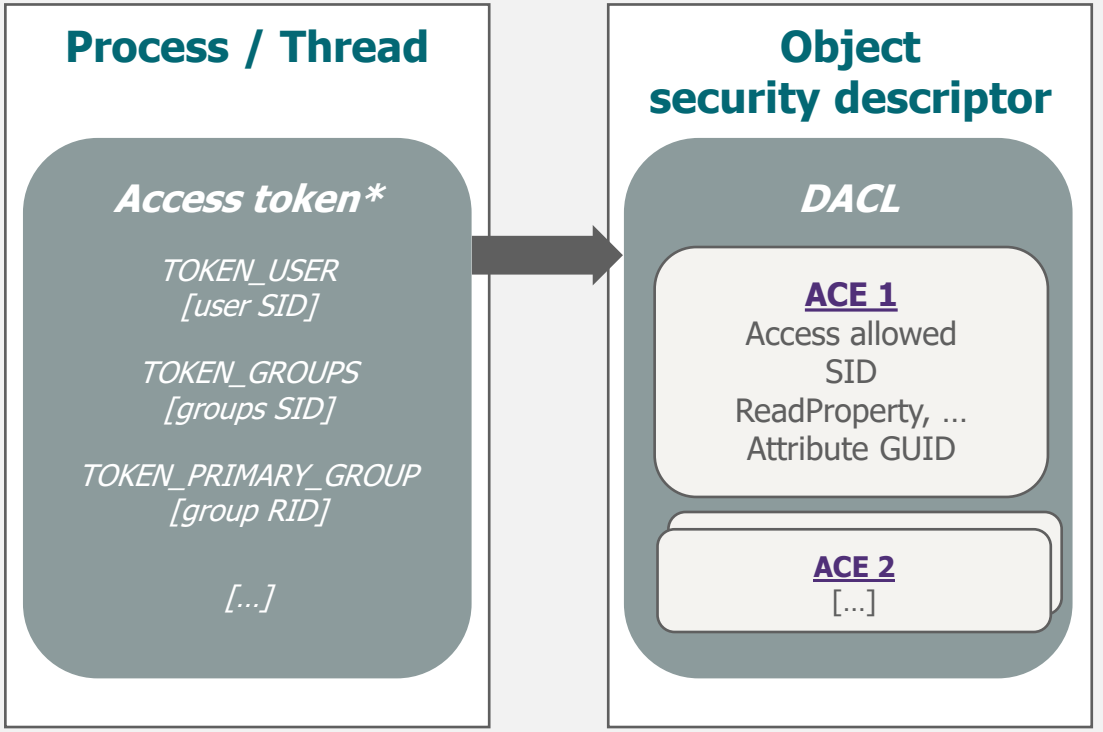
2

Securable resources

Access to securable resources are based on the **accessing user access token** and **object security descriptor**.

The **SIDs** in the user access token are compared to the **Access Control Entries (ACEs)** of the accessed object's **Discretionary Access Control List (DACL)**.

Access is thus also conditioned by the **user PrimaryGroupId** and **SIDs** in **SIDHistory**.



** https://docs.microsoft.com/en-us/windows/win32/api/winnt/ne-winnt-token_information_class*

Exploiting SIDHistory and primaryGroupID for persistence

- **SIDs can be injected in history** (T1134.005) and the **PrimaryGroupID attribute modified freely** through a **DCShadow attack** (T1207) using **mimikatz***.

  The injected **SID / RID** will ultimately end up in the **user access token**.

- The **group membership** granted by the **PrimaryGroupID attribute does not appear** in the **account's MemberOf** nor in the **group's Members LDAP attributes**.

  It **does however appear** in **group membership attributes "constructed"** through Microsoft APIs.

```
# First interpreter (executed as "NT AUTHORITY\SYSTEM").
mimikatz # lsadump::dcshadow /object:<USERNAME> /attribute:<ATTRIBUTE_NAME> /value:<ATTRIBUTE_VALUE>

# Second interpreter (executed as the privileged domain account).
mimikatz # lsadump::dcshadow /push
```

**SIDs can also be injected in history using DSInternals' Add-ADDBSidHistory PowerShell cmdlet through a local execution on a Domain Controller.*

Hunting for SIDHistory & primaryGroupID persistence as implemented in FarsightAD

Export-ADHuntingPrincipalsSIDHistory

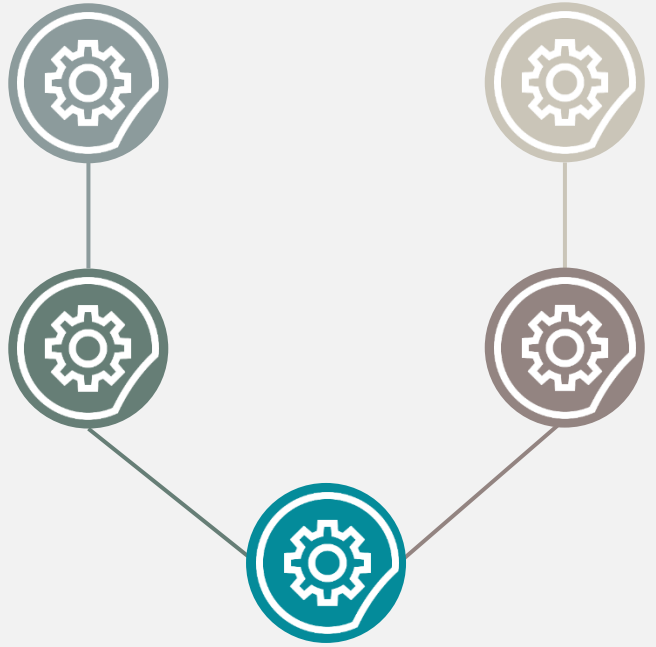
Enumerate SIDs in SIDHistory and **highlight SIDs from the current domain**

Identify SIDs linked to **privileged groups** in history

Export-ADHuntingPrincipalsPrimaryGroupID

Enumerate non-default RID given the object type: 513 for users & 515 for computers (non-DC)

Identify RID of a **privileged group** (built-in or not)

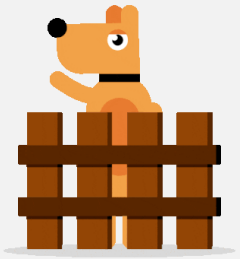


Timestamp of last modification of the attribute through **replication data**

3 kind of Kerberos delegations with different security implications

Kerberos delegations were introduced to **allow an application** / **server to act on the behalf of another user** through the Kerberos protocol.

Unconstrained delegations



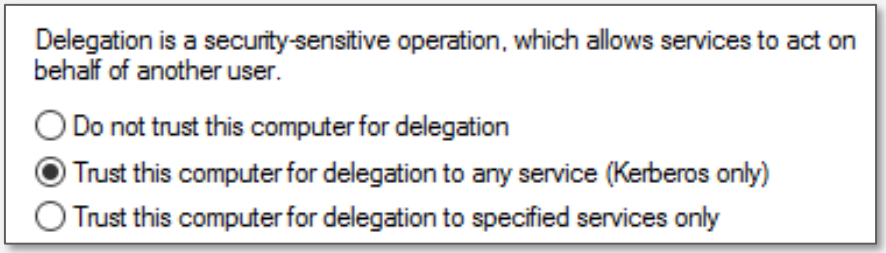
Constrained delegations

Resource-based constrained delegation (RBCD)

Kerberos unconstrained delegations expose users or computers credentials (TGT) 1/2

Service accounts that are **trusted for Kerberos unconstrained delegation**, ADS_UF_TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION flag set in their User-Account-Control, **can fully act on behalf of other domain accounts**.

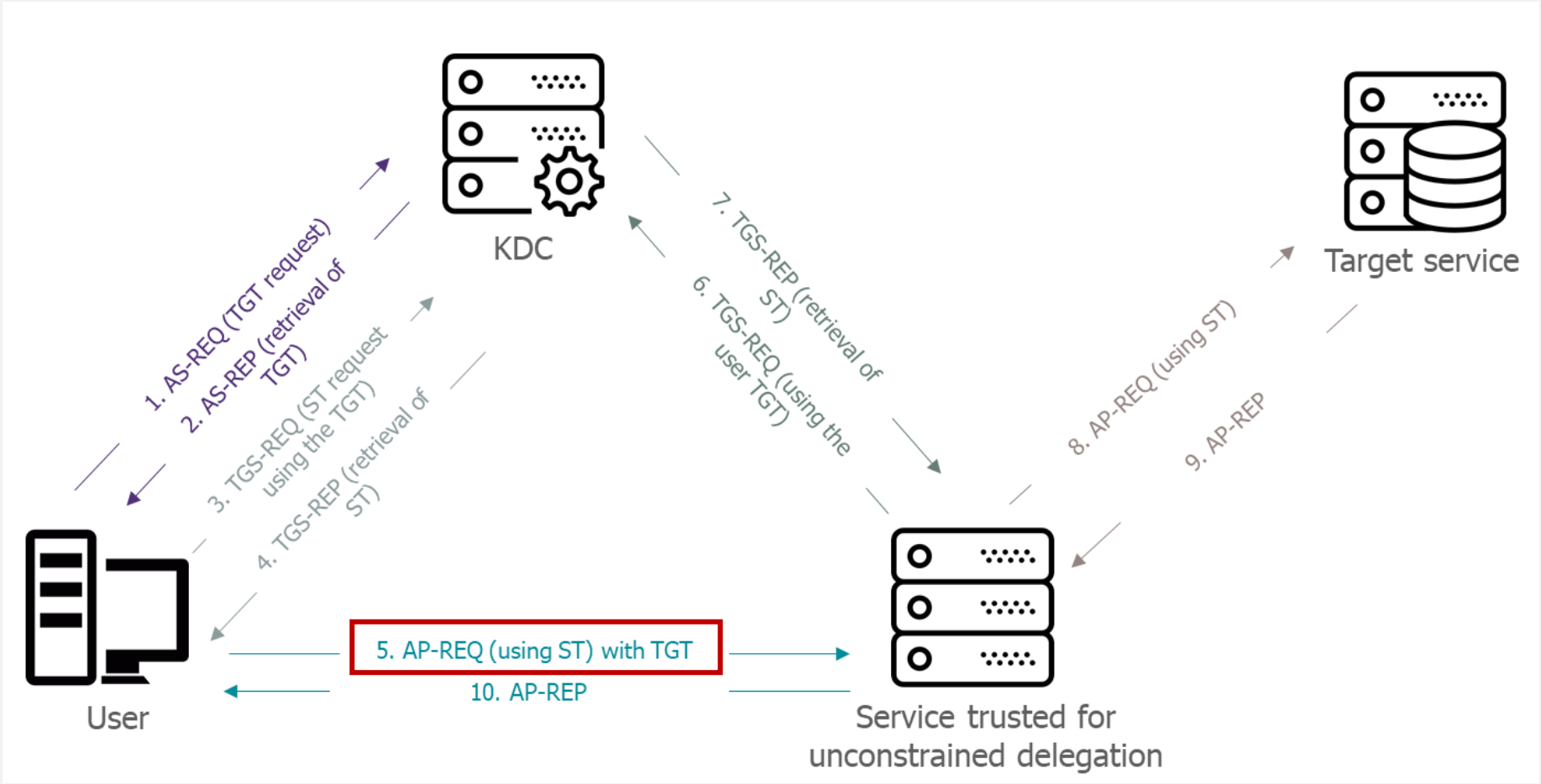
Service tickets received by such services will indeed **contain a copy of the TGT** of the **account accessing the service**.



Coercing a DC machine account Kerberos authentication, for example through RPC calls to the MS-RPRN (printer bug)[1], MS-EFSRPC (PetitPotam)[2], MS-FSRVP (ShadowCoerce)[3], or MS-DFSNM (DFSCoerce)[4], interfaces can lead to the **compromise of a TGT for the DC machine account**.

1 @tifkin_ and @elad_shamir - 2 @topotam77 - 3 @topotam77 and @_nwodtuhs - 4 @filip_dragovic

Kerberos unconstrained delegations expose users or computers credentials (TGT) 2/2



The Service-for-User (S4U) extension was introduced in Kerberos to support constrained delegations and RBCD

S4U2Proxy

Allows service to **request service tickets** (ST) to the KDC (TGS-REQ) **on behalf of other users** by:

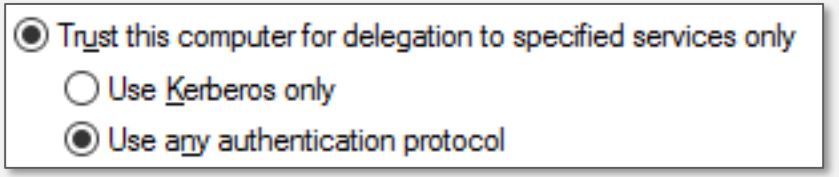
- Arbitrarily specifying the user the service ticket should be emitted for.
- **Joining**, in the *req-body.additional-tickets* field of the TGS-REQ request, **a ST marked as forwardable from the specified user except for S4U2Proxy request in RBC flow***.

** https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html*

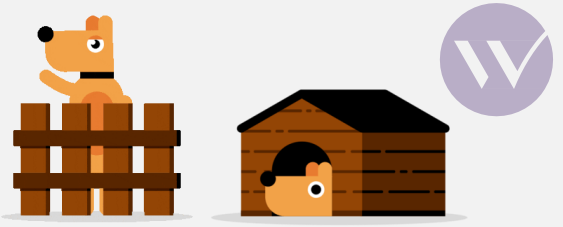
S4U2Self

Provide protocol transition from NTLM to Kerberos, by allowing a service to get a ST for itself of an arbitrary user.

Only service accounts that can **"use any authentication protocol"** (*ADS_UF_TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION* flag set in the service accounts' UAC attribute) **will receive ST marked as forwardable** from S4U2Self requests.



Kerberos constrained delegations or RBCD can be leveraged to persist to specific services 1/2

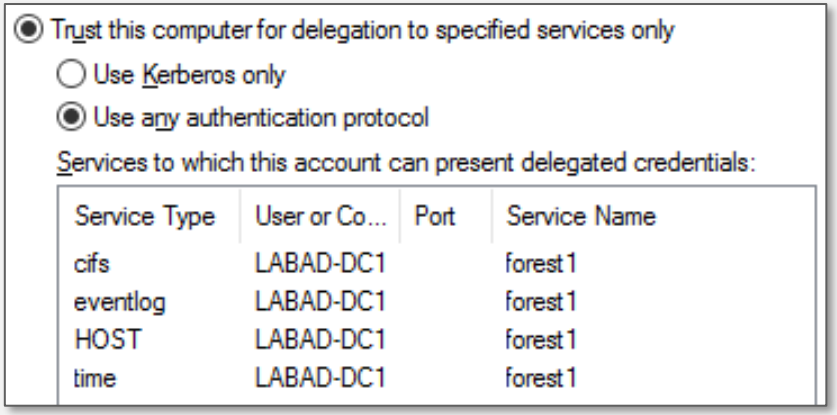


Constrained delegations

Services configured for constrained delegations can **assume the identity of users** to **target services defined in their msDS-AllowedToDelegateTo attribute**.

1. The **service must have a forwardable ST** from the **user to impersonate** or **obtain one using a S4U2Self request** (if the service can "use any authentication protocol").

2. Then the **service can make a S4U2Proxy request** to **obtain a ST on behalf of the user to impersonate**. The ST must be for a target service the service configured for constrained delegation is allowed to delegate to.

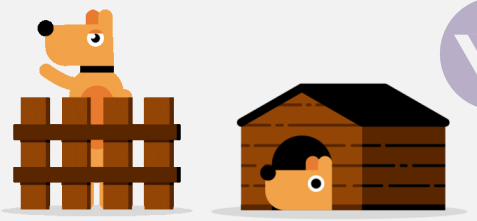


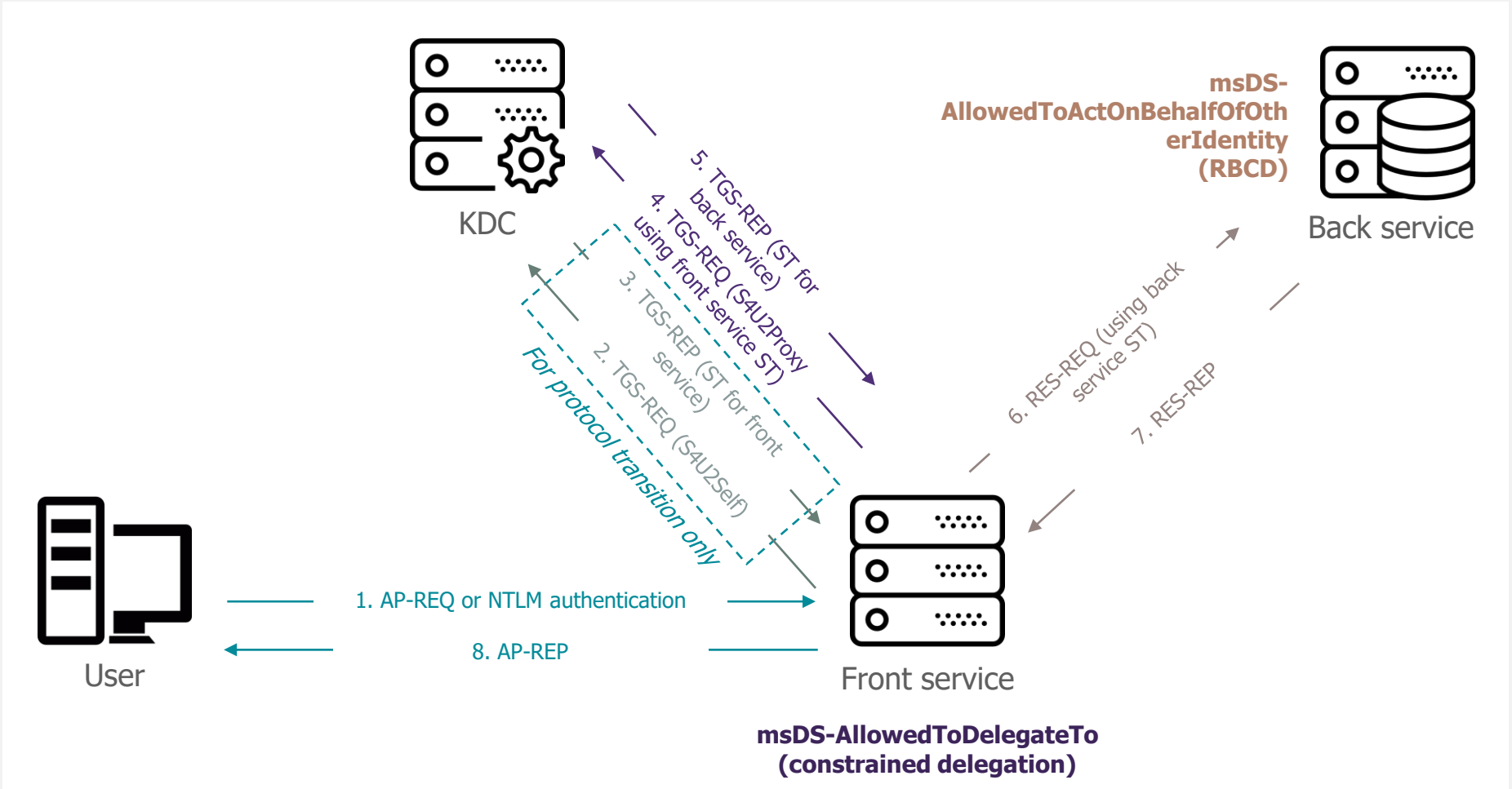
Resource-based constrained delegations

Introduced in Windows Server 2012 and **work similarly to constrained delegations**, **except the trust is shifted to the target / final service**. A **service account** can indeed **allow other services to delegate to it**, by setting the services they accept delegated authentication from in **its own msDS-AllowedToActOnBehalfOfOtherIdentity attribute**.

Unlike constrained delegations, the **intermediary service does not need to be in possession of a forwardable ST** in order to make "S4U2proxy" requests.

Kerberos constrained delegations or RBCD can be leveraged to persist to specific services 2/2





Hunting for Kerberos delegations persistence as implemented in FarsightAD

Export-ADHuntingKerberosDelegations

Enumerate all Kerberos delegations (unconstrained delegations, constrained delegations, and RBCD)

Parse services SPN in **constrained delegations and RBCD**, determining the target user or computer service SID

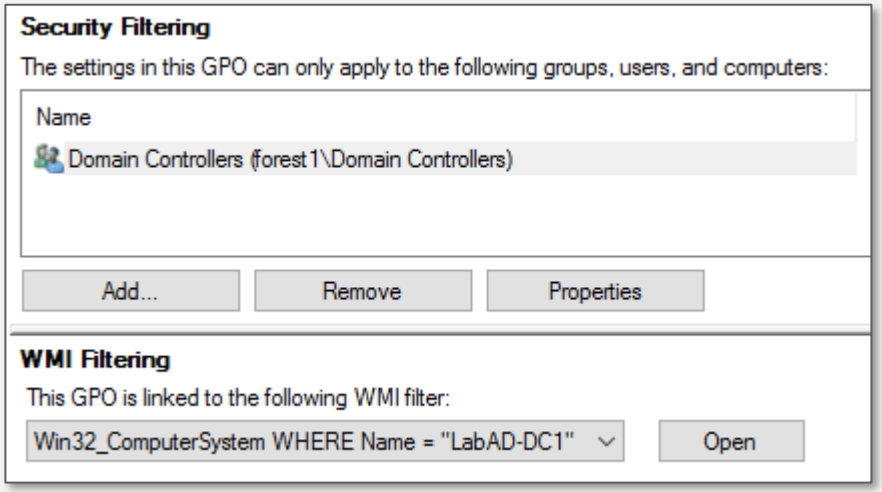
Determine dangerous Kerberos delegations, that is:

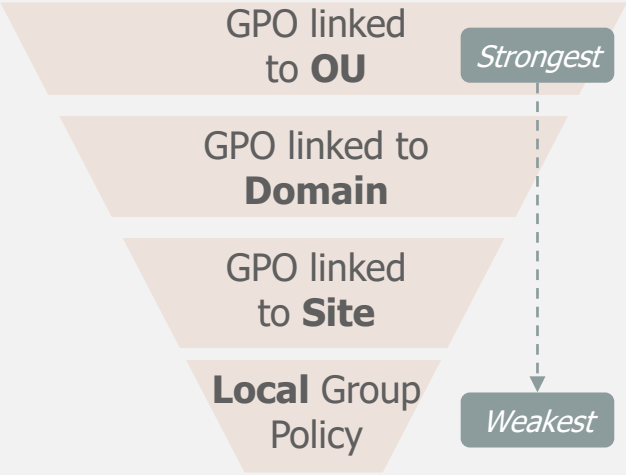
- Unconstrained delegations (except for DCs)
- Constrained delegations where the target service is privileged
- RBCD where the source service account is privileged

msDS-AllowedToDelegateTo and *msDS-AllowedToActOnBehalfOfOtherIdentity* attributes **timestamp of last modification** through **replication data**

Group Policy Object are designed to centrally manage users or computers

- **Per-machine** or **per-user settings** to configure **numerous parameters** (Windows system and security settings, logon and logoff scripts, …).
- **Can be linked** to **Organizational Unit (OU)**, **Site**, or **Domain**, with a **precedence order for conflicting settings**: the GPO closest to the client location in the directory will overwrite any conflicting settings applied.
- A **GPO may not necessarily apply** as an OU can blocks inheritance (on a not enforced GPO), filtering can restrict the GPO application (on specific users or through a WMI query), the link can be disabled (but still present), or only the GPO user / computer settings may apply.





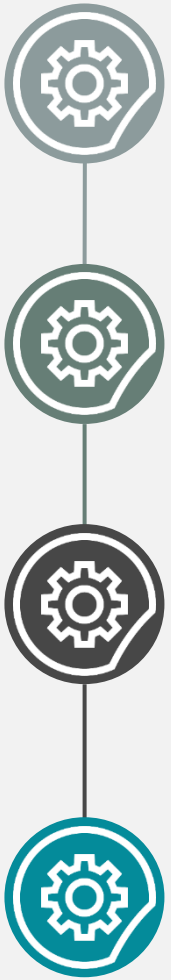
** Precedence order for non-enforced GPOs. With enforcement, the parent GPO link always has precedence. https://serverfault.com/questions/510624/gpo-enforced-precedence*

Group Policy Object offer (many) many persistence opportunities

- **Logon rights** and **Windows privileges** to remotely logon and maintain local Administrator privileges on a system.
- **AD-level privileges** such as the right to enable services for Kerberos unconstrained delegation.
- **Restricted Groups** to add members to the computer local groups.
- **Computer** or **user logon / logoff scripts** to execute code at the computer startup / shutdown or user logon / logoff.
- **GPO deployed files** through **MSI installation package**.
- Arbitrary **HKEY_LOCAL_MACHINE** / **HKEY_CURRENT_USER ASEP registry keys**.
- **GPO scheduled** or **immediate tasks** to be executed by the Task Scheduler.
- **Startup shortcut / LNK files** to execute an arbitrary program.
- **GPO files not hosted in the SYSVOL directory** or **access rights allowing modification** of the **GPO AD objects**, **GPO SYSVOL files**, and / or **files executed through GPO**.
- Others likely missing here…

Hunting for Group Policy Object persistence as implemented in FarsightAD 1/2

Export-ADHuntingGPOObjectsAndFilesACL



Determine if the **GPOs are applied on privileged users or computers** (at OU, Domain or Site level, and by processing OU inheritance block / GPO enforcement)

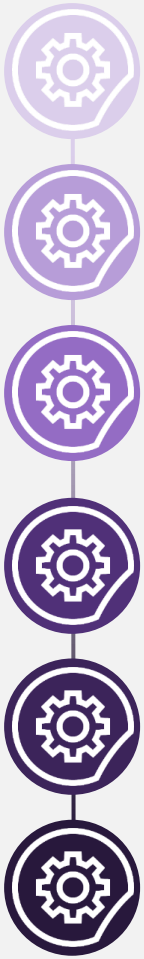
Check GPO objects and **GPO files ownership** and **access rights**, highlighting takeover / modifications **rights granted to non-privileged principals** or **everyone**

Check if the GPO files are hosted on DCs by parsing the gPCFileSysPath attribute

Retrieve multiple timestamps: **GPOs creation** and **last modification**, GPOs **security descriptor** and **gPCFileSysPath** attributes **last modification** through replication data

Hunting for Group Policy Object persistence as implemented in FarsightAD 2/2

Export-ADHuntingGPOSettings



Determine if the **GPOs are applied on privileged users or computers** (at OU, Domain or Site level, and by processing OU inheritance block / GPO enforcement)

Directly parse the GPO files on the SYSVOL to **retrieve (some) settings** deployed by the GPOs and **track GPO that couldn't be evaluated**, notably to identify **access denied errors**

Retrieve the privileges and logon rights, determining if dangerous privileges* / logon rights are **granted to non-privileged principals or everyone**

Retrieve the restricted groups membership, highlighting privileged groups and unprivileged members

Retrieve the scheduled and immediate tasks configured, by (somewhat) parsing the XML tasks definition

Retrieve the machine and user logon / logoff scripts, checking if the target scripts are hosted on the DC, if **takeover / modification rights** are granted to non-privileged or everyone, and the **scripts MACB timestamps**

** Based on https://github.com/gtworek/Priv2Admin and privileges dangerous at the domain level*

(D)ACL also offer (many) many persistence possibilities

Using numerous takeover/dangerous ACE

- "Allow" access and apply to the object (i.e not InheritOnly).
- GenericAll, WriteDacl, or WriteOwner.
- GenericWrite or WriteProperty on all properties or on sensitive attributes:
  - `UPN, AltSecId, Public-Information`
  - `Member`
  - `msDS-AllowedToActOnBehalfOfOtherIdentity, msDS-ManagedPassword`
  - `gPLink, gPCFileSysPath`
  - …
- Self, for all validates write or to add one-self to a group.
- All extended rights or sensitive extended rights:
  - `User-Force-Change-Password`
  - `[DCSync] DS-Replication-Get-Changes + DS-Replication-Get-Changes-All*`
  - `[DCShadow] DS-Install-Replica + DS-Replication-Manage-Topology + DS-Replication-Synchronize`
  - …

On various privileged objects

- AdminSDHolder container, whose security descriptor is replicated by the SDProp mechanism (every 60 minutes by default) on a number of privileged accounts and groups (DA, EA, …).
- On the Domain Root object.
- On the Domain Controllers group and machine accounts.
- The OU under which privileged principals reside.
- The GPOs applied to privileged users and computers (at OU, Domain, and Site level).
- The DPAPI backup and Key Distribution Service (KDS) root keys.
- …

For more information:

- An ACE up the Sleeve by Will Schroeder (@harmj0y), Andy Robbins (@_wald0), and Lee Christensen (@tifkin_): https://www.specterops.io/assets/resources/an_ace_up_the_sleeve.pdf
- ACTIVE DIRECTORY SECURITY ASSESSMENT CHECKLIST by ANSSI: https://www.cert.ssi.gouv.fr/uploads/guide-ad.html
- https://notes.qazeer.io/active-directory/exploitation-acl_exploiting

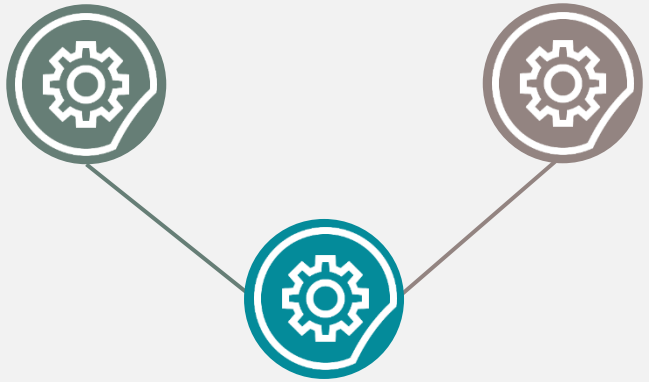
* Computer account with the userAccountControl SERVER_TRUST_ACCOUNT flag can also conduct replication operations

Hunting for ACL persistence as implemented in FarsightAD

Export-ADHuntingACLPrivilegedObjects

Export-ADHuntingACLDangerousAccessRights

Enumerate ACL on **privileged objects**



Enumerate ACL on **all (securable) objects**

Filter / **highlight dangerous ownership** and **ACE**

Note: for forest / domain recovery, **it can be easier to restore ACL from their default values** (as stored in the Schema) on privileged objects and **disable ACL inheritance on Tier 0 OUs**.

Export-ADHuntingACLDefaultFromSchema

Compare the domain default ACL, from defaultSecurityDescriptor attribute of Schema classes, to their expected **values from Microsoft documentation** / **fresh AD install**.
Additionally check if any **dangerous rights are positioned** and retrieve the **defaultSecurityDescriptor's last modification timestamps** through replication metadata.

Exploiting AD CS for persistence

- **Various misconfigurations can allow privilege escalation / persistence through AD CS**, notably if user-supplied data from the certificate request is used for the certificate SAN or if the access rights of published certificate templates are too permissive (or illegitimately modified).
- A **rogue CA certificate** may also be added to the **NTAuthCertificates trusted certificates** to **forge arbitrary certificates** (permitting client authentication).

Export-ADHuntingADCSCertificateTemplates

Enumerate and **review certificate templates** (SAN construction, manager approval, ACL, …)

Export-ADHuntingADCSPKSObjects

Enumerate and **review Public Key Services objects** (NTAuthCertificates, certificationAuthority, and pKIEnrollmentService), parsing certificates and attempting to find rogue CA certificates

- For more information on AD CS attacks:

Certified Pre-Owned by Will Schroeder (@harmj0y) and Lee Christensen (@tifkin_)
https://posts.specterops.io/certified-pre-owned-d95910965cd2

Microsoft ADCS – Abusing PKI in Active Directory Environment by Jean Marsault (@iansus)
https://www.riskinsight-wavestone.com/en/2021/06/microsoft-adcs-abusing-pki-in-active-directory-environment/

Introduced FarsightAD cmdlets, and more… 1/2

Cmdlet	Synopsis
Invoke-ADHunting	Execute all the FarsightAD AD hunting cmdlets.
Export-ADHuntingACLDangerousAccessRights	Export dangerous ACEs, i.e ACE that allow takeover of the underlying object, on all the domain's objects.
Export-ADHuntingACLDefaultFromSchema	Export the ACL configured in the defaultSecurityDescriptor attribute of Schema classes.
Export-ADHuntingACLPrivilegedObjects	Export the ACL configured on the privileged objects in the domain and highlight potentially dangerous access rights.
Export-ADHuntingADCSCertificateTemplates	Export information and access rights on certificate templates.
Export-ADHuntingADCSPKSObjects	Export information and access rights on sensitive PKS objects.
Export-ADHuntingGPOObjectsAndFilesACL	Export ACL access rights information on GPO objects and files.
Export-ADHuntingGPOSettings	Export information on various settings configured by GPOs that could be leveraged for persistence.
Export-ADHuntingHiddenObjectsWithDRSRepData	Export the objects' attributes that are accessible through replication but not by direct query.
Export-ADHuntingKerberosDelegations	Export the Kerberos delegations that are considered dangerous.
Export-ADHuntingPrincipalsAddedViaMachineAccountQuota	Export the computers that were added to the domain by non-privileged principals.
Export-ADHuntingPrincipalsCertificates	Export parsed accounts' certificate(s).
Export-ADHuntingPrincipalsDontRequirePreAuth	Export the accounts that do not require Kerberos pre-authentication.

Introduced FarsightAD cmdlets, and more… 2/2

Cmdlet	Synopsis
Export-ADHuntingPrincipalsOncePrivileged	Export the accounts that were once member of privileged groups.
Export-ADHuntingPrincipalsPrimaryGroupID	Export the accounts that have a non default primaryGroupID attribute, highlighting RID linked to privileged groups.
Export-ADHuntingPrincipalsPrivilegedAccounts	Export detailed information about members of privileged groups.
Export-ADHuntingPrincipalsPrivilegedGroupsMembership	Export privileged groups' current and past members, retrieved using replication metadata.
Export-ADHuntingPrincipalsSIDHistory	Export the accounts that have a non-empty SID History attribute, with resolution of the associated domain and highlighting of privileged SIDs.
Export-ADHuntingPrincipalsShadowCredentials	Export parsed Key Credentials information (of accounts having a non-empty msDS-KeyCredentialLink attribute).
Export-ADHuntingPrincipalsTechnicalPrivileged	Export the technical privileged accounts (SERVER_TRUST_ACCOUNT and INTERDOMAIN_TRUST_ACCOUNT).
Export-ADHuntingPrincipalsUPNandAltSecID	Export the accounts that define a UserPrincipalName or AltSecurityIdentities attribute, highlighting potential anomalies.
Export-ADHuntingTrusts	Export the trusts of all the domains in the forest.

Main take away points

Identifying AD persistence is hard and **automation has its limit.** But it is often necessary as rebuilding the forest from scratch may not be an option (due to resources, business and / or time constraint).

Following a forest / domain compromise, **attack paths**, **especially to Tier 0**, **should be identified and addressed** (or at least closely monitored) before returning to production.

Security mechanisms such as authentication silos can greatly help secure the Tier 0.

AD backups isolation and **testing is a must**, in order to not make a difficult situation even worse.