Bad Meets Evil → Why do you think that I chose this title?

GDI Foundation

Who am I?

- **<u>Huy Kha</u>**
- Information Security @ International Law Firm
- Advisory Board Member @ GDI.Foundation
- **Background**: System Administrator & Service Desk Analyst
- **Likes**: Management, Governance, Risk, Compliance, IT Auditing
- **Dislikes**:

Inspiration

- **<u>What is my goal?</u>**
- Inspiring **RED Teamers & Pentesters** to improve their attacking techniques and let them look to things that they might have overlooked in the past.
- Making sure that both **RED** / **BLUE** Teamers understand that attackers don't need to be Domain Admin to own you. There are several other ways to do it.
- Helping **BLUE Teamers** to improve their defensive capabilities.
- Helping System Administrators to understand the different risks and how they can recognize it or even avoiding it.
- Let's help each other to improve :)

Hackers
don't give a shit:

- About your project's scope
- It's managed by a third party
- It's a legacy system
- It's "too critical to patch"
- About your outage windows
- About your budget
- You've always done it that way
- About your Go-Live Date
- It's only a pilot/proof of concept
- About Non-Disclosure Agreements
- It wasn't a requirement in the contract
- It's an internal system
- It's really hard to change
- It's due for replacement
- You're not sure how to fx it
- It's handled in the Cloud
- About your Risk Register entry
- The vendor doesn't support that conf guration
- It's an interim solution
- It's [insert standard here] compliant
- It's encrypted on disk
- The cost benef t doesn't stack up
- "Nobody else could f gure that out"
- You can't explain the risk to "The Business"
- You've got other priorities
- About your faith in the competence of your internal users
- You don't have a business justif cation
- You can't show Return on Investment
- You contracted out that risk

Attackers Path

- **Insecure ACL Configurations**
- GenericAll
- GenericWrite
- WriteDacl
- WriteOwner
- AllExtendedRights

- **Kerberos Attacks**
- Kerberoasting
- AS-REP Roasting
- Unconstrained Kerberos Delegation
- SeEnableDelegation Privilege

- **Attacking Group Policy Objects**

Unauthorized user with Full control on GPO's

- Compromising users of Group Policy Creator Owners

- **Credential Harvesting**
- LLMNR/NBT-NS Poisoning and Relay
- SMB Relay Attack

- **Lateral Movement**
- Pass-The-Hash
- Golden Ticket
- Silver Ticket

- **Defense Evasion**
- DCSync
- DCShadow
- Applocker / Windows Firewall

Insecure ACL Configurations

- **GenericAll**
- The right to create or delete children, delete a subtree, read and write properties, examine children and the object itself, add and remove the object from the directory, and read or write with an extended right**.**

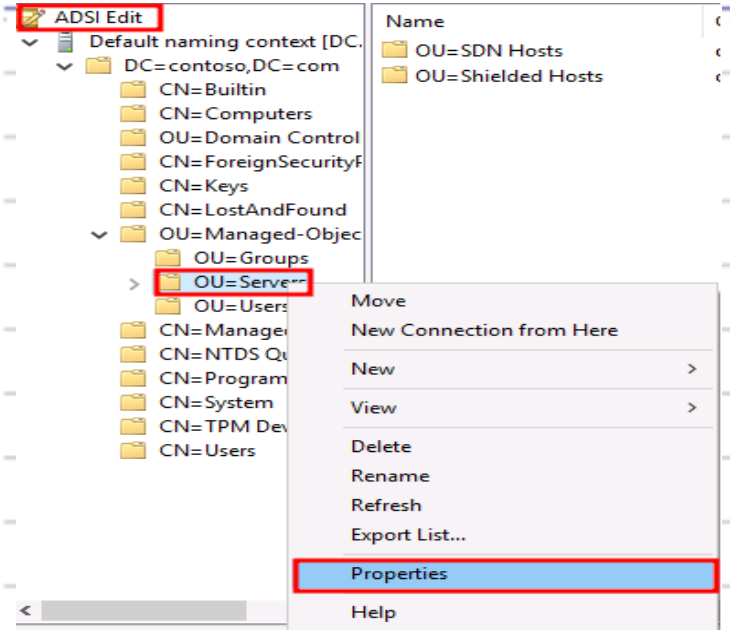
- **GenericWrite**
- The right to read permissions on this object, write all the properties on this object, and perform all validated writes to this object.

- **WriteDacl**
- The right to modify the DACL in the object security descriptor.

- **AllExtendedRights**
- The ability to perform any action associated with extended Active Directory rights against the object.

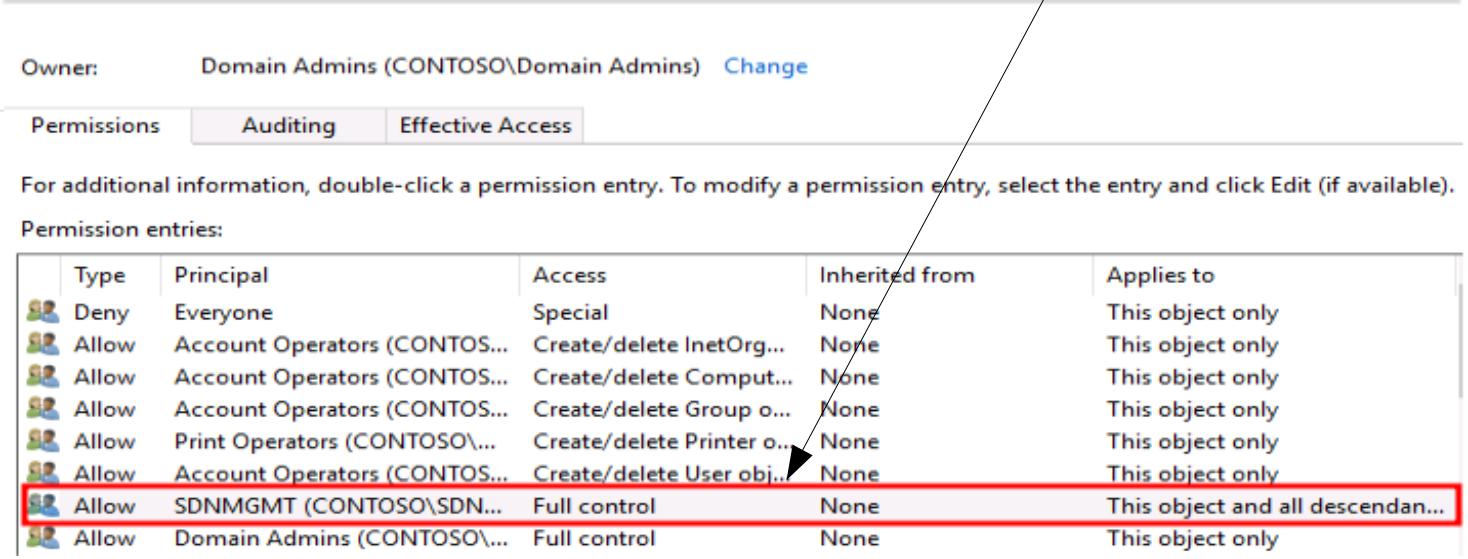
- **WriteOwner**
- The right to assume ownership of the object. The user must be an object trustee. The user cannot transfer the ownership to other users.

GenericAll = Full control



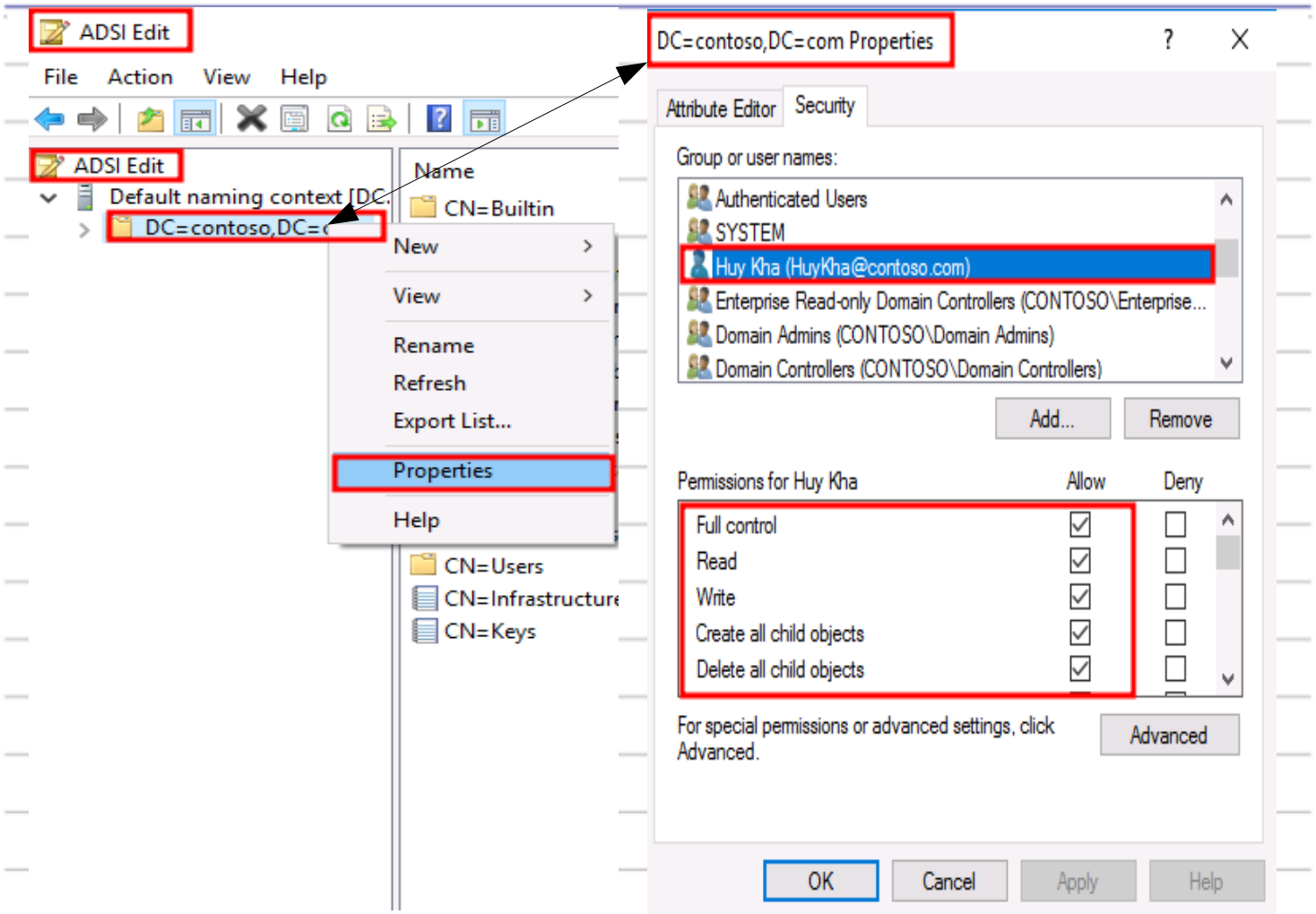
- The rights of **SDNMGMT** has been delegated to the **OU=Servers** with **Full control**, which allows everyone in that group being able to modify every object in that (specific) OU.

- This is ~~often~~ not needed, so be aware who has the **GenericAll** rights, because it can lead to serious breaches or damage.



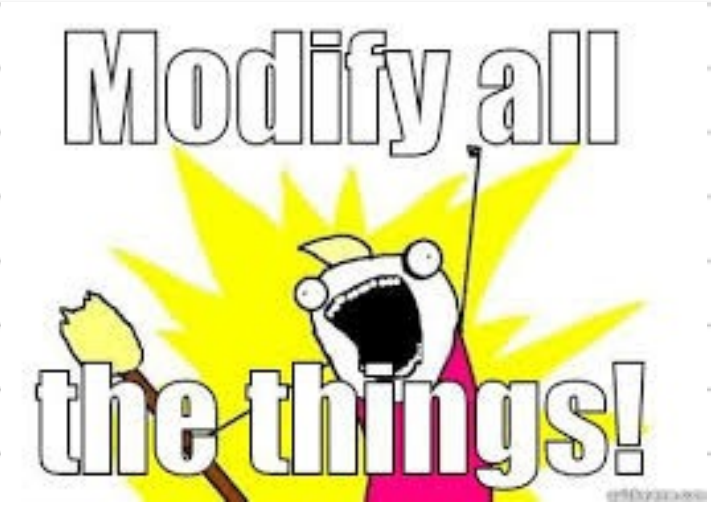
Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Special	None	This object only
Allow	Account Operators (CONTOS...	Create/delete InetOrg...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Comput...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Group o...	None	This object only
Allow	Print Operators (CONTOSO\...	Create/delete Printer o...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete User obj...	None	This object only
Allow	SDNMGMT (CONTOSO\SDN...	Full control	None	This object and all descendan...
Allow	Domain Admins (CONTOSO\...	Full control	None	This object only

GenericAll →(on) Domain Root



Impact & Risk → GenericAll

- **Impact:** GenericAll
- What if the attacker compromised a user with GenericAll privilege?
- Attacker would be able to add himself to different groups and granting himself access to different resources. Also it's possible to modify user accounts & OU's



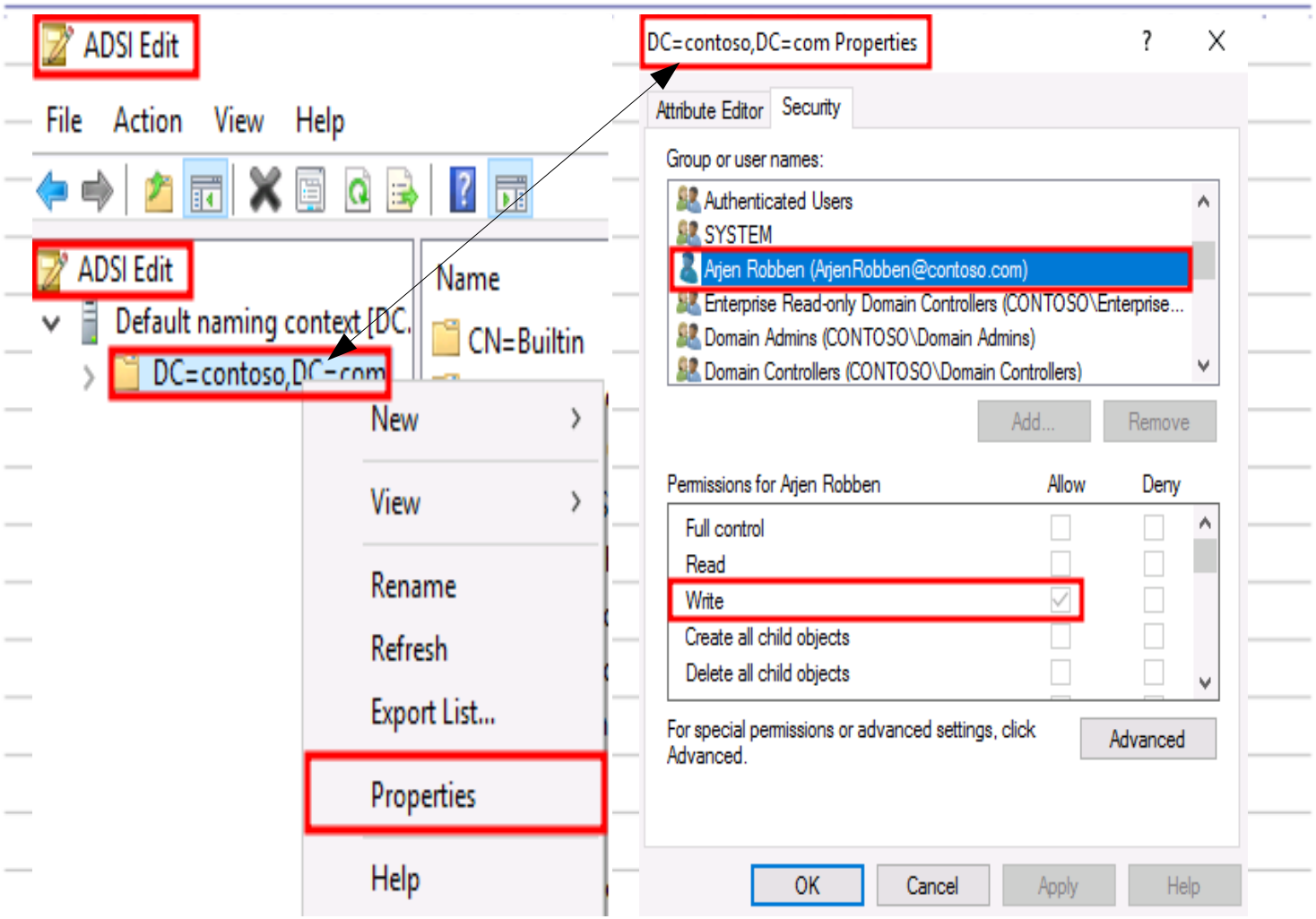
- **Risk:** GenericAll
- GenericAll (Full control) is ~~often~~ not needed in an environment.
- Risk depends if the GenericAll is set on the domain root or a particular OU.
- GenericAll on Domain Root = DCSync
- **Unauthorized users with GenericAll could do the following:**
- Add themselves to different groups and OU's
- Create / Delete / Add / Reset – Accounts
- Modify permissions of users

GenericWrite → Information

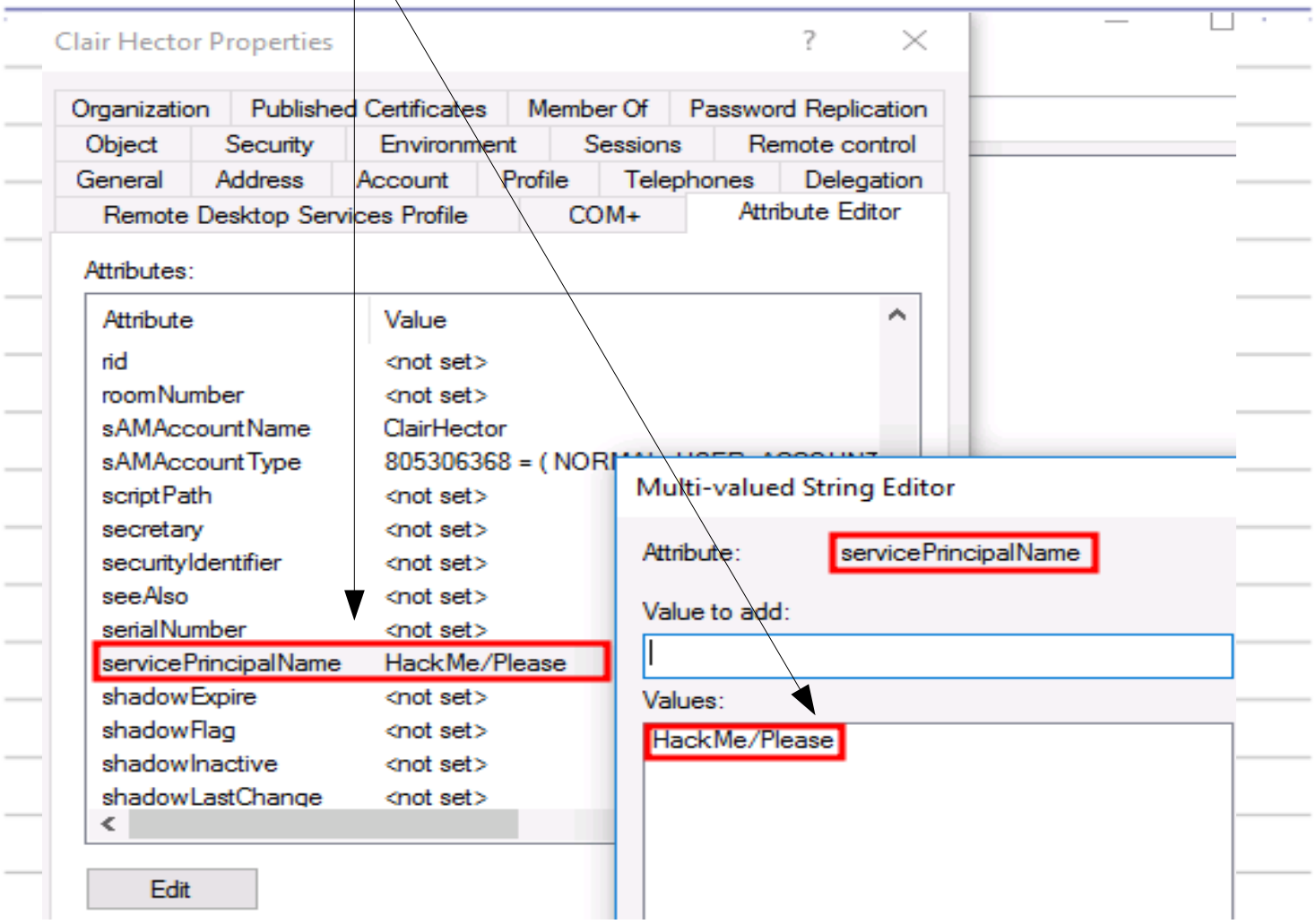
- **GenericWrite** gives the ability to users to write any non-protected property of an object.
- Every object in AD has the ''Attribute Editor'' tab, which can be abused by attackers, if they are able to compromise users with GenericWrite permissions.
- Attackers are able to modify the attributes from user accounts.
- The risk depends on if the permission is delegated on the **domain root** or to an **specific OU**.
- SID-History can be modified as well to escalate privileges.

Type	Principal	Access	Inherited from	Applies to
Allow	ENTERPRISE DOMAIN CONT...	Read only replication s...	None	This object only
Allow	Authenticated Users	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only
Allow	Cloneable Domain Controller...	Allow a DC to create a ...	None	This object only
Allow	Enterprise Read-only Domain...	Replicating Directory ...	None	This object only
Allow	Domain Controllers (CONTO...	Replicating Directory ...	None	This object only
Allow	Arjen Robben (ArjenRobben...	Write	None	This object and all descendan...
Allow	SELF		None	This object and all descendan...
Allow	SELF	Special	None	All descendant objects

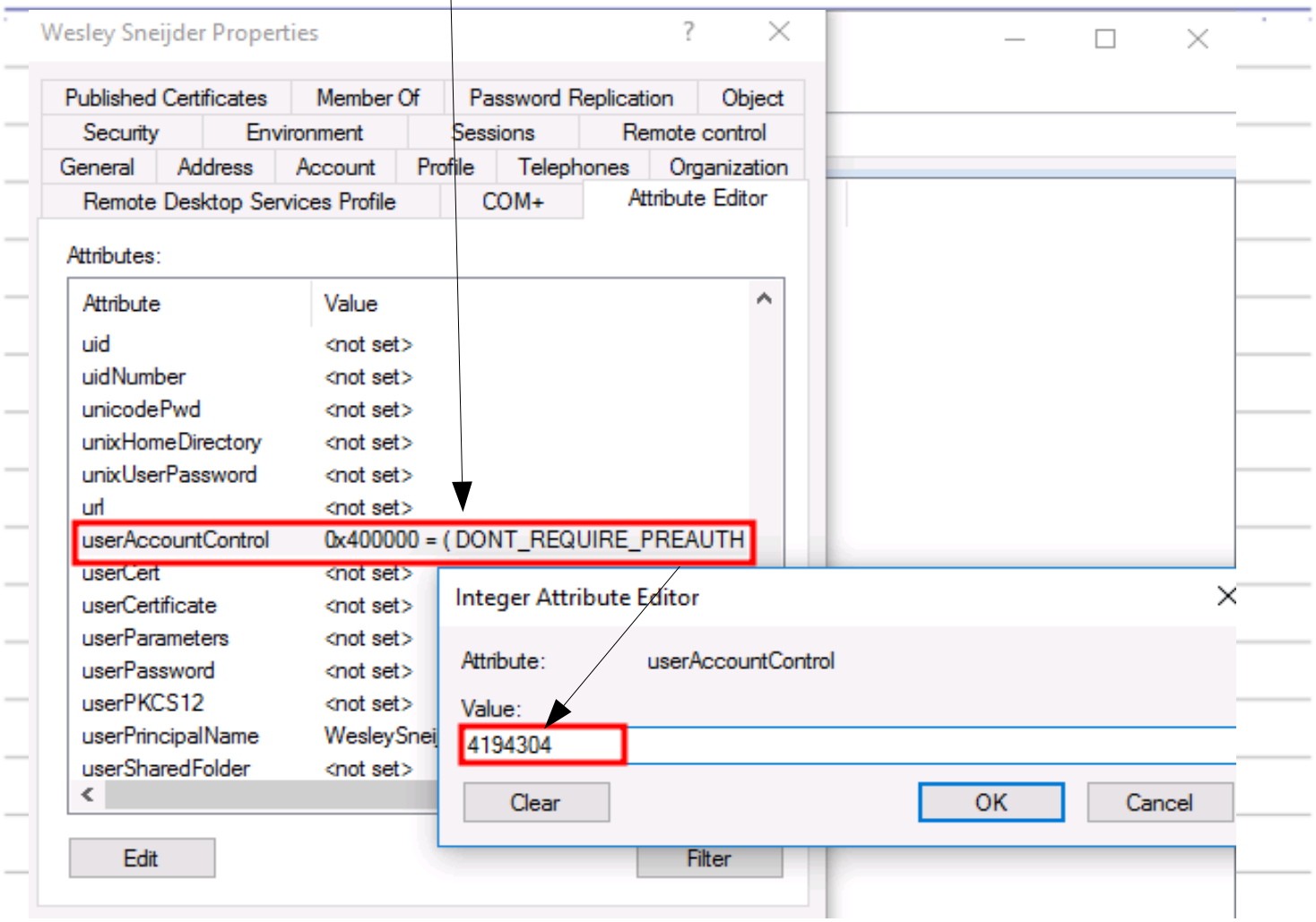
GenericWrite → (on) Domain Root



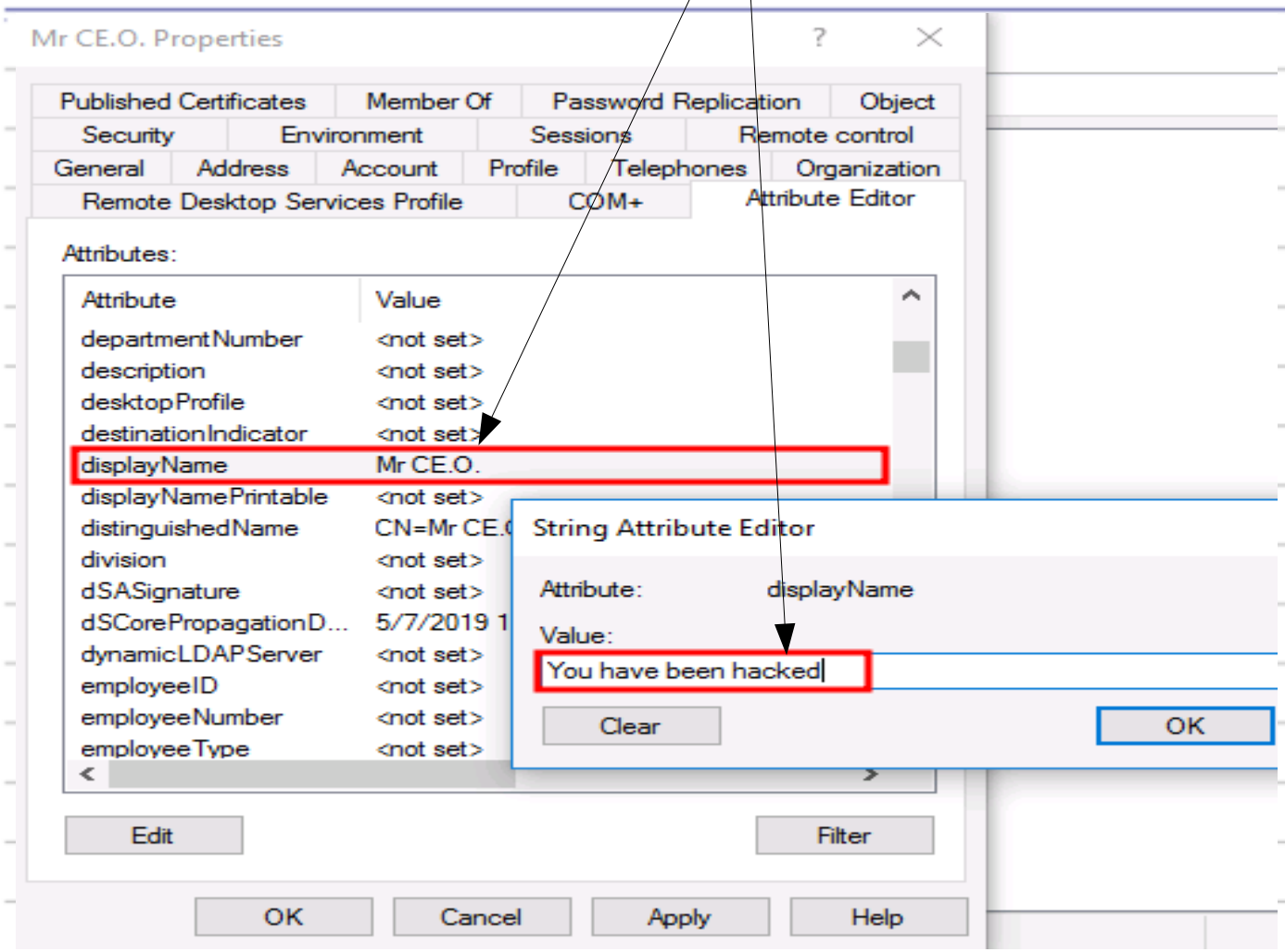
GenericWrite → Modify attribute *(Adding SPN value for Kerberoasting)*



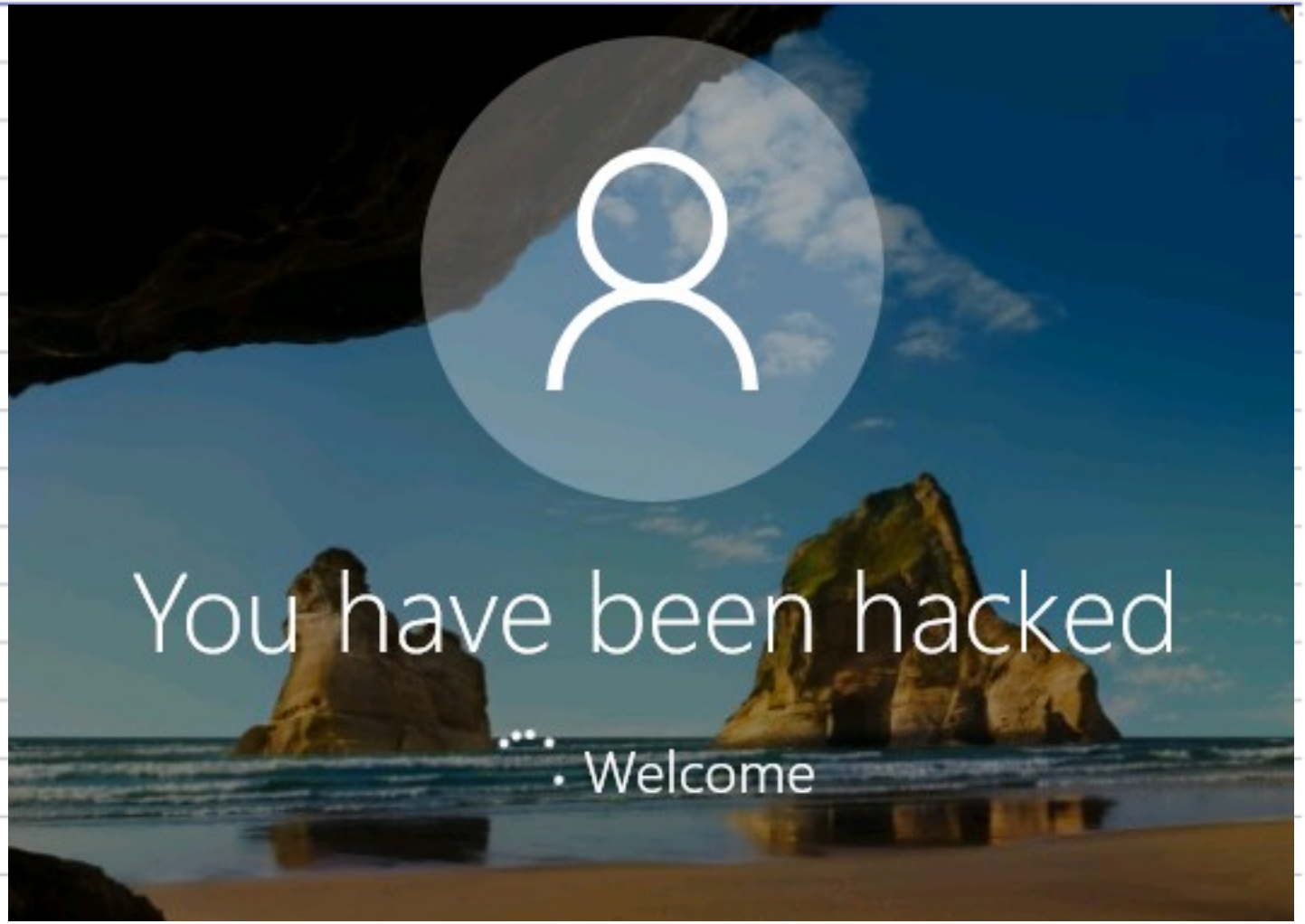
GenericWrite → Enable DONT_REQ_PREAUTH for user account



GenericWrite → Scare your CEO by changing the displayName



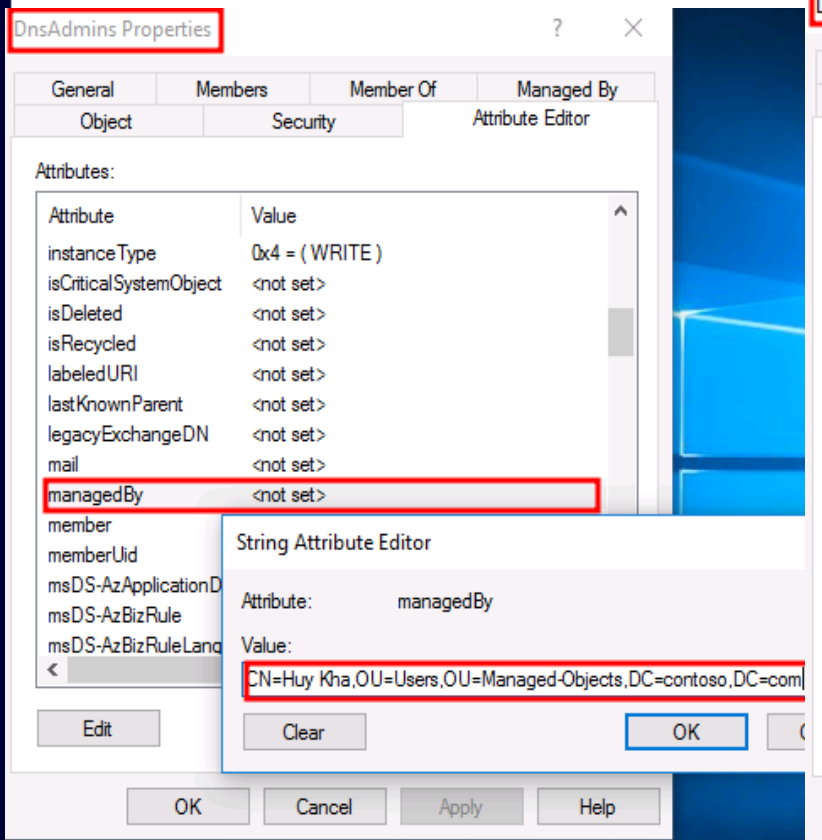
He is scared… now

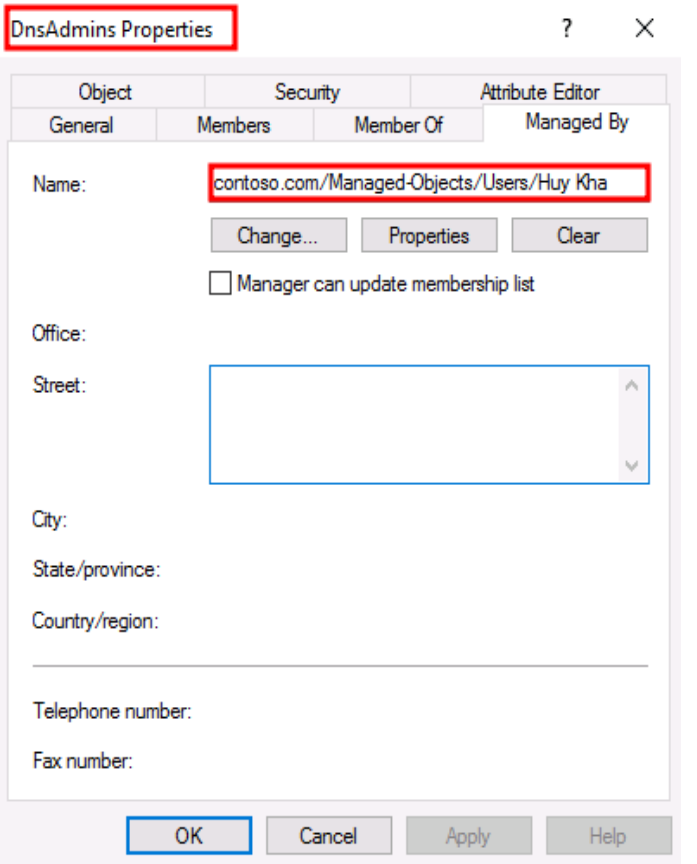


GenericWrite → Descedant Group Objects → Modify "Managed by" attribute

Example: Huy has GenericWrite on the Domain Root for Group Objects

Type	Principal	Access	Inherited from	Applies to
Allow	Enterprise Admins (CONTOS...	Full control	None	This object and all descendan...
Allow	Administrators (CONTOSO\A...	Special	None	This object and all descendan...
Allow	Pre-Windows 2000 Compatib...	List contents	None	This object and all descendan...
Allow	Pre-Windows 2000 Compatib...	Special	None	This object only
Allow	ENTERPRISE DOMAIN CONT...	Special	None	This object only
Allow	Huy Kha (Huy@contoso.com)	Special	None	Descendant Group objects
Allow	CREATOR OWNER	Validated write to com...	None	Descendant Computer objects
Allow	SELF	Special	None	All descendant objects



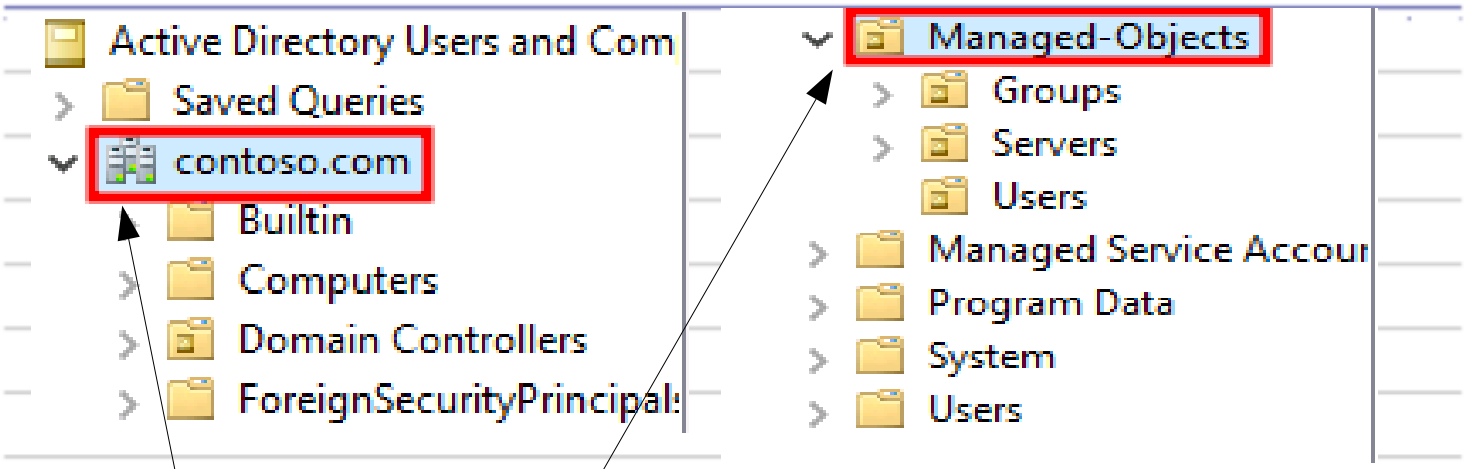


GenericWrite → Interesting <u>userAccountControl</u> values

- **PASSWD_NOTREQD** - an account can be configured to have a null value for a password. With this set anyone could login with the account and access authorised resources.
- **PASSWD_CANT_CHANGE** - the user cannot change the account password. Worth flagging for its rarity.
- **ENCRYPTED_TEXT_PWD_ALLOWED** - the password is stored with reversible encryption. The password hash can be converted to plain text.
- **DONT_EXPIRE_PASSWORD** - the password never expires, leaving the account susceptible to brute force attacks.
- **DONT_REQ_PREAUTH** - the account doesn't require Kerberos pre-authentication. Opens up the possibility of offline brute-forcing of encrypted TGT.

- **PASSWD_NOTREQD =** 32
- **PASSWD_CANT_CHANGE =** 64
- **ENCRYPTED_TEXT_PWD_ALLOWED =** 128
- **DONT_EXPIRE_PASSWORD =** 65536
- **DONT_REQ_PREAUTH =** 4194304
- **TRUSTED_FOR_DELEGATION =** 524288
- **TRUSTED_TO_AUTH_FOR_DELEGATION =** 16777216

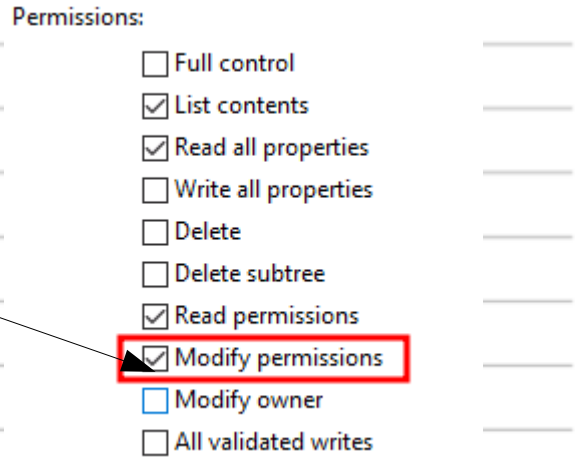
WriteDacl → Delegated rights



“Domain root” → Delegated WriteDacl on root

“Managed Objects” → Delegated WriteDacl on *specific* OU.

“WriteDacl”



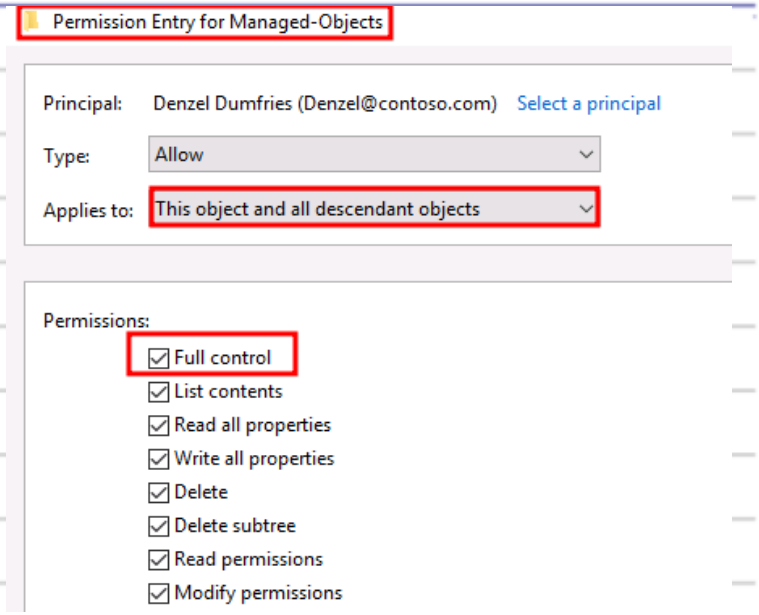
WriteDacl → Delegated OU

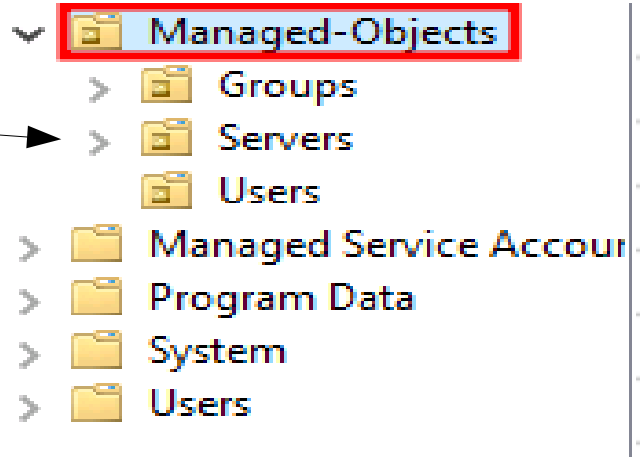
- Assigning the **WriteDacl** to unauthorized users could lead to the following: *(Depending if it's on the domain root or a specific OU)*
- User can modify permissions of user accounts. Including himself, which means that he grant himself the **''Full control''** permission or to someone else.
- Add and remove users to OU.
- Basically having **Full control** and also on Sub-OU's (if exist).

Allow	Domain Admins (CONTOSO\Domain Ad...	Special	None	This object only
Allow	Enterprise Admins (CONTOSO\Enterprise ...	Special	None	This object only
Allow	Pre-Windows 2000 Compatible Access (C...	Special	None	This object only
Allow	Administrators (CONTOSO\Administrators)	Special	None	This object only
Allow	Everyone	Modify permissions	None	This object only
Allow	Authenticated Users	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only

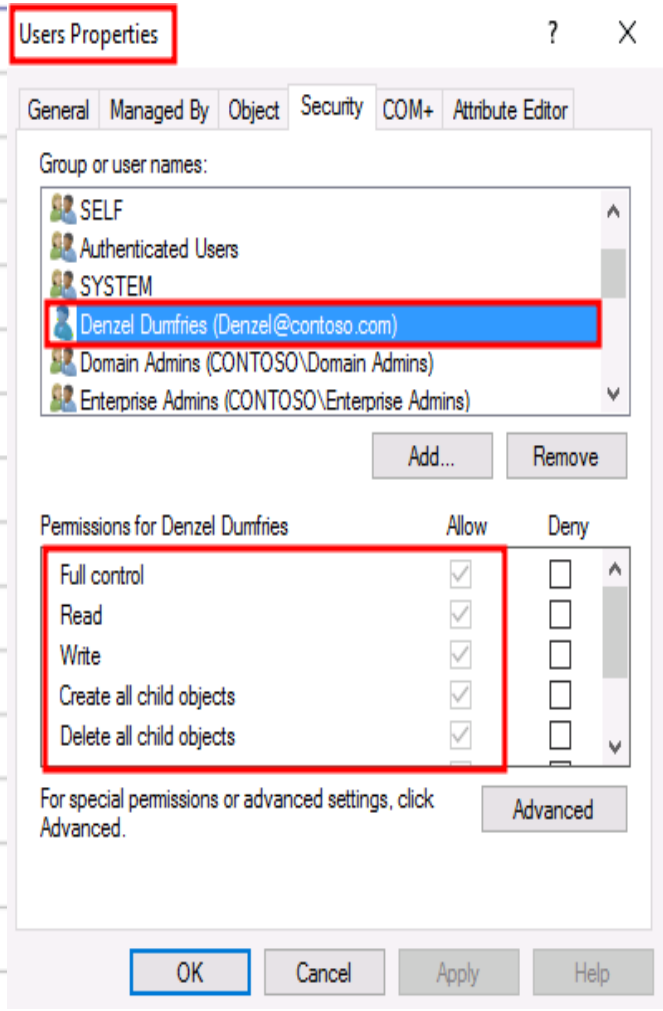
WriteDacl → Assigning Full control

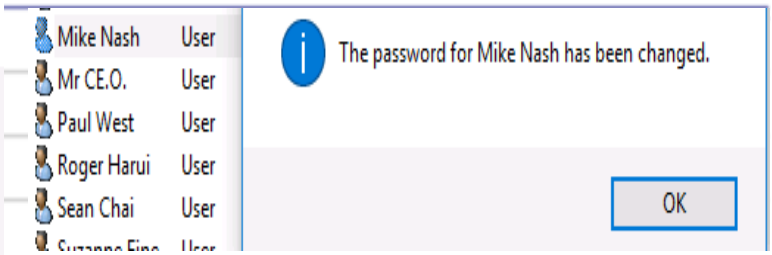
- Modify permissions (**WriteDacl**) allow us to edit permissions of user accounts. Because we're allowed to do this. It is possible to grant **Full control**.
- With **"Full control"** - We can modify the **Managed-Objects** and change the **Applies to** → *"This object and all descendant objects"*
- Now we have **''Full control"** on all the sub OU's of Managed-Objects.
- Since we now have Full control rights on all the Sub OU's. We have the privileges to modify the ''Users'' OU and reset passwords of those accounts for account take-over.





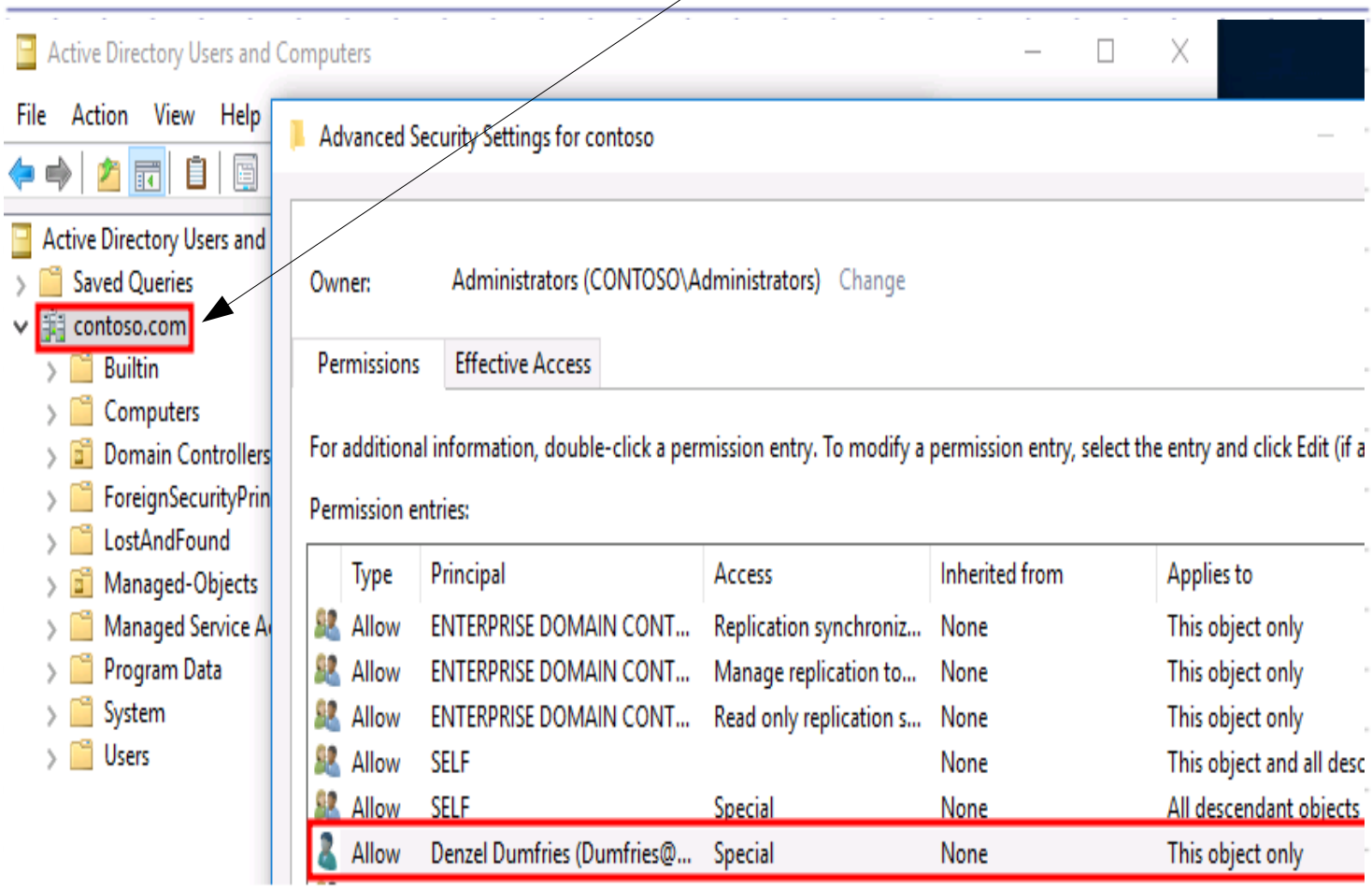
WriteDacl → Leverage with GenericAll permissions



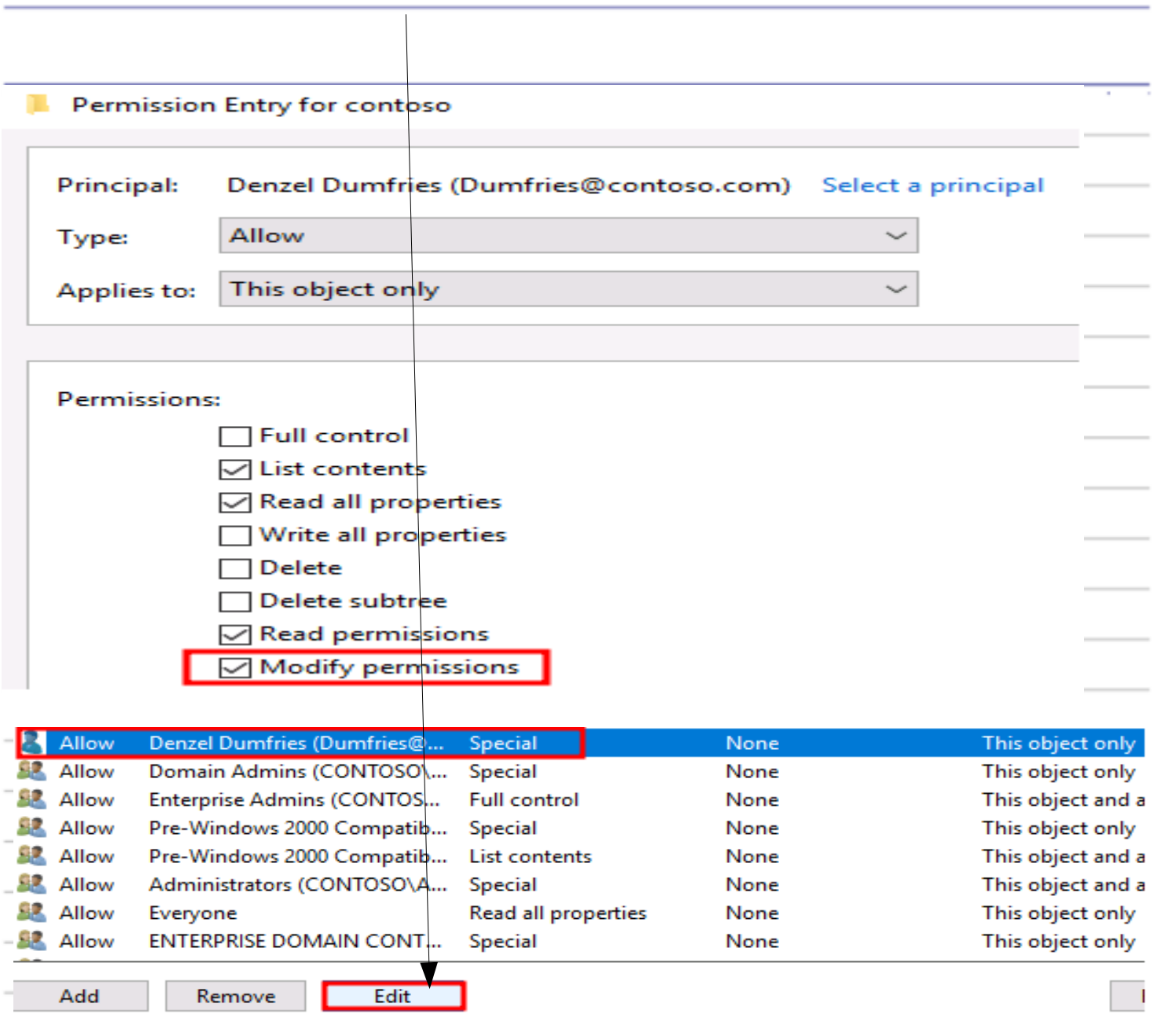


WriteDacl → Full control →OU=**Managed-Objects** → Change Applies to *"This object and all descendant object"* → Full control on all sub OU's → Modifying (Sub)OU=**Users** → <u>User-Force-Change-Password</u> → Account takeover completed!

WriteDacl → on Domain Root



WriteDacl →"Edit" Modify permission to Full control on Domain Root.

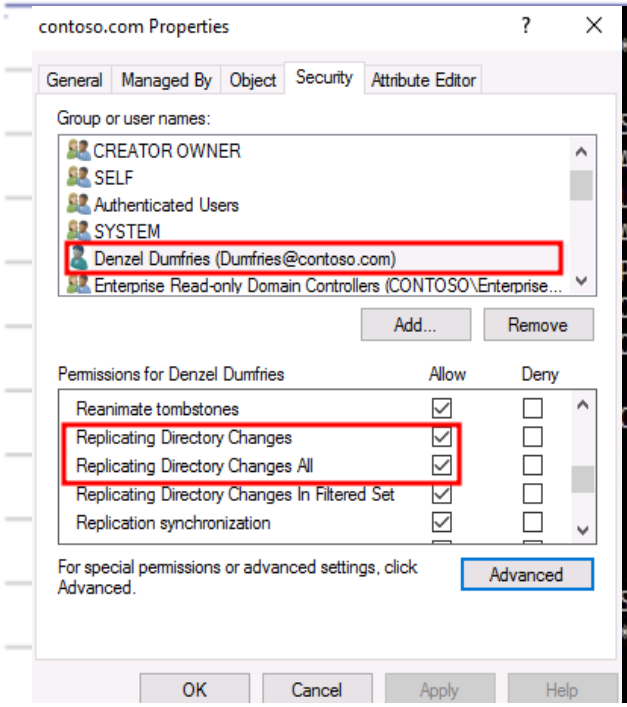


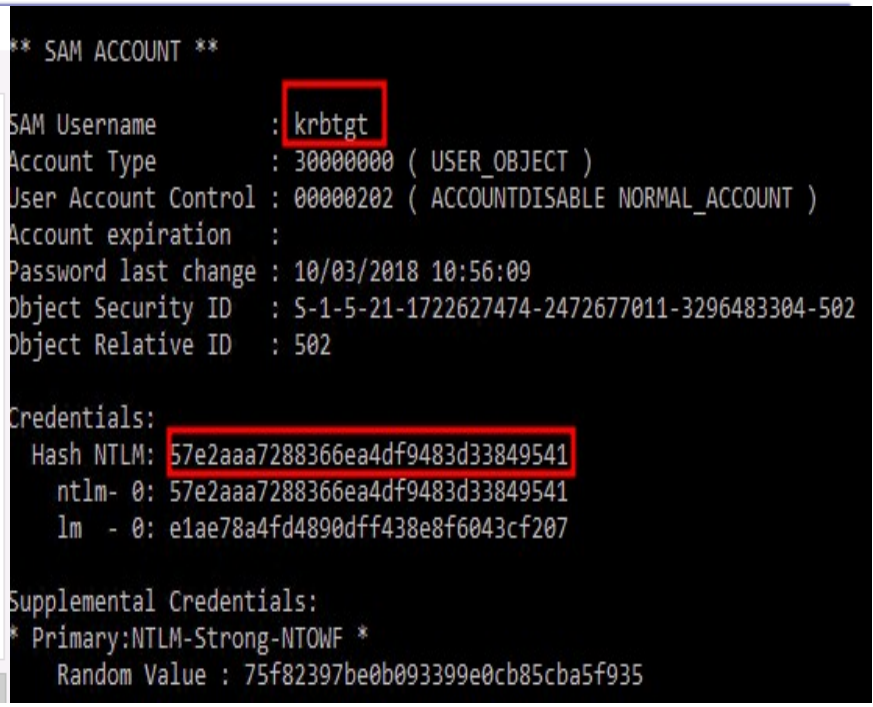
WriteDacl – Full control on Domain Root

Type	Principal	Access
Allow	Denzel Dumfries (Dumfries@...	Full control
Allow	Domain Admins (CONTOSO\...	Special
Allow	Enterprise Admins (CONTOS...	Full control
Allow	Pre-Windows 2000 Compatib...	Special
Allow	Pre-Windows 2000 Compatib...	List contents
Allow	Administrators (CONTOSO\A...	Special
Allow	Everyone	Read all properties
Allow	ENTERPRISE DOMAIN CONT...	Special
Allow	Authenticated Users	Special
Allow	SYSTEM	Full control
Allow	Cloneable Domain Controller...	Allow a DC to create a ...

Add Remove Edit

WriteDacl – DCSync without Domain Admin





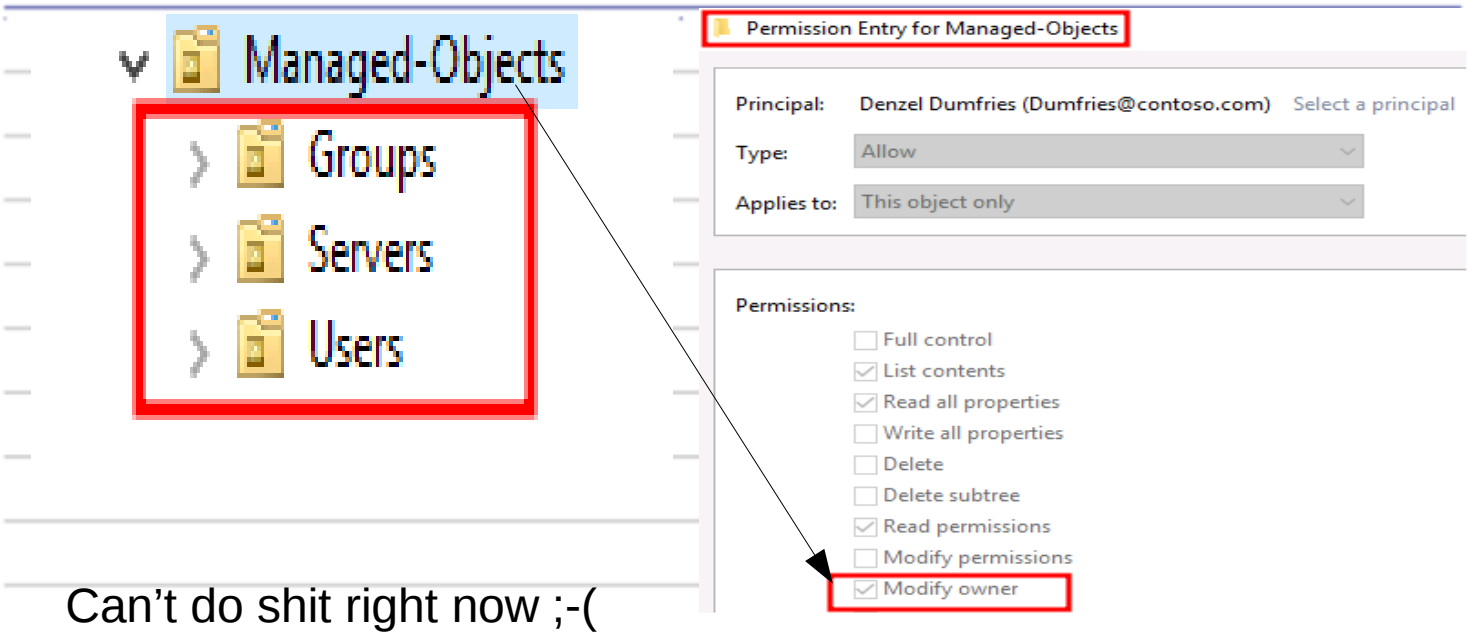
lsadump::dcsync /domain:example.com /user:krbtgt

"DCSync impersonates the behavior of Domain Controller and requests account password data from the targeted Domain Controller."

WriteOwner → Information

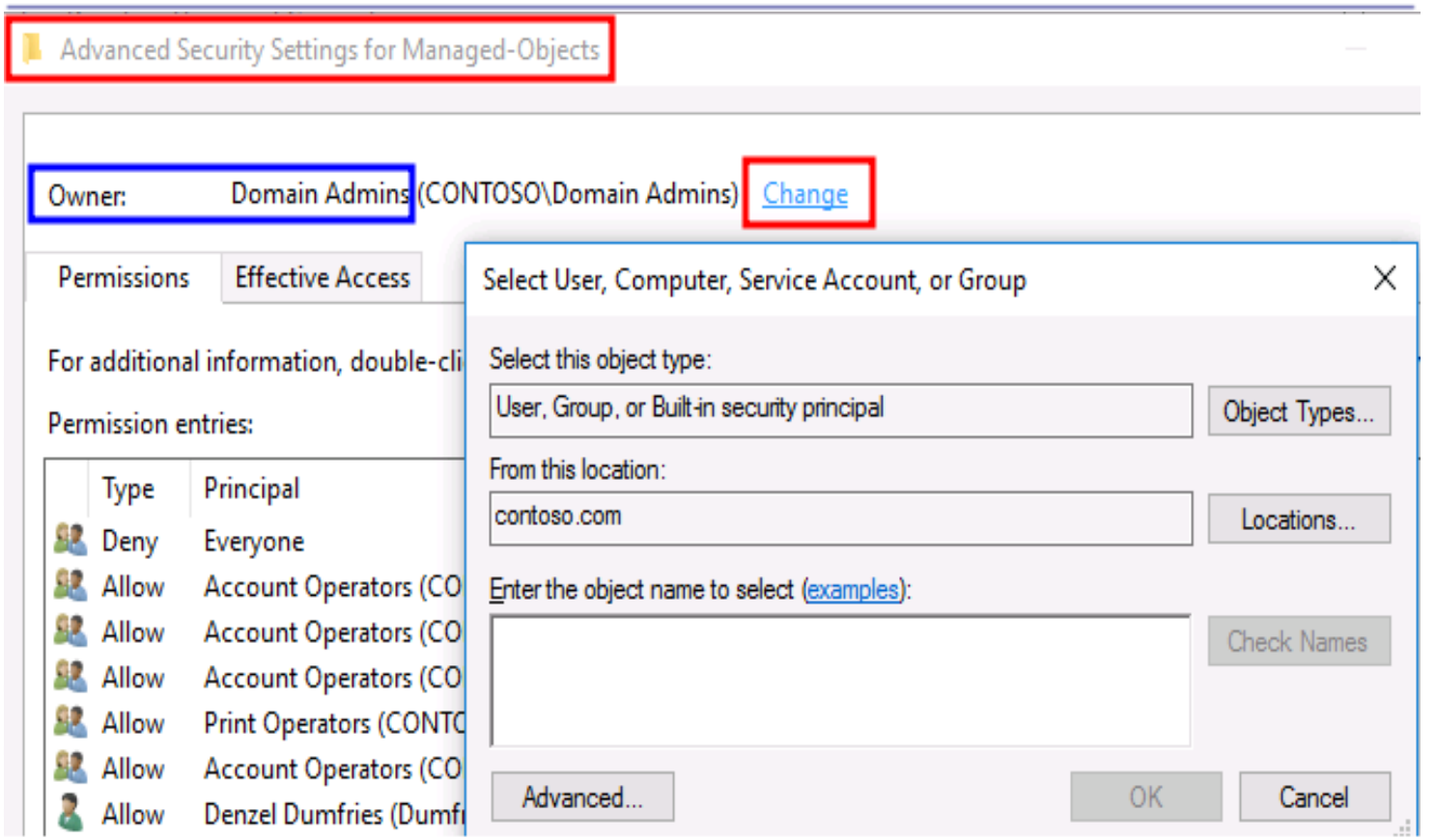
- **WriteOwner**
- The right to assume ownership of the object. The user must be an object trustee. The user cannot transfer the ownership to other users.
- Compromising a user or group with the **WriteOwner** permission depends on if it's delegated on the domain root or an specific OU.
- **WriteOwner** allows an attacker to leverage to **GenericAll** (Full control) by removing the owner of an OU and no further explanation is needed anymore.

WriteOwner → Delegated rights on OU



Can't do shit right now ;-(

WriteOwner → Domain Admin as **Owner**? I don't think so!



WriteOwner → Meme time :D

WriteOwner → I'm now the Owner and can grant Full control to Everyone ;-)

Advanced Security Settings for Managed-Objects

Owner: Denzel Dumfries (Dumfries@contoso.com) Change

Permissions | Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Special	None	This object only
Allow	Account Operators (CONTOS...	Create/delete InetOrg...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Comput...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Group o...	None	This object only
Allow	Print Operators (CONTOSO\...	Create/delete Printer o...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete User obj...	None	This object only
Allow	Denzel Dumfries (Dumfries@...	Special	None	This object only
Allow	Domain Admins (CONTOSO\...	Full control	None	This object only
Allow	ENTERPRISE DOMAIN CONT...	Special	None	This object only
Allow	Authenticated Users	Special	None	This object only

Add | Remove | Edit | Restore defaults

Disable inheritance

Advanced Security Settings for Managed-Objects

Owner: Denzel Dumfries (Dumfries@contoso.com) Change

Permissions | Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

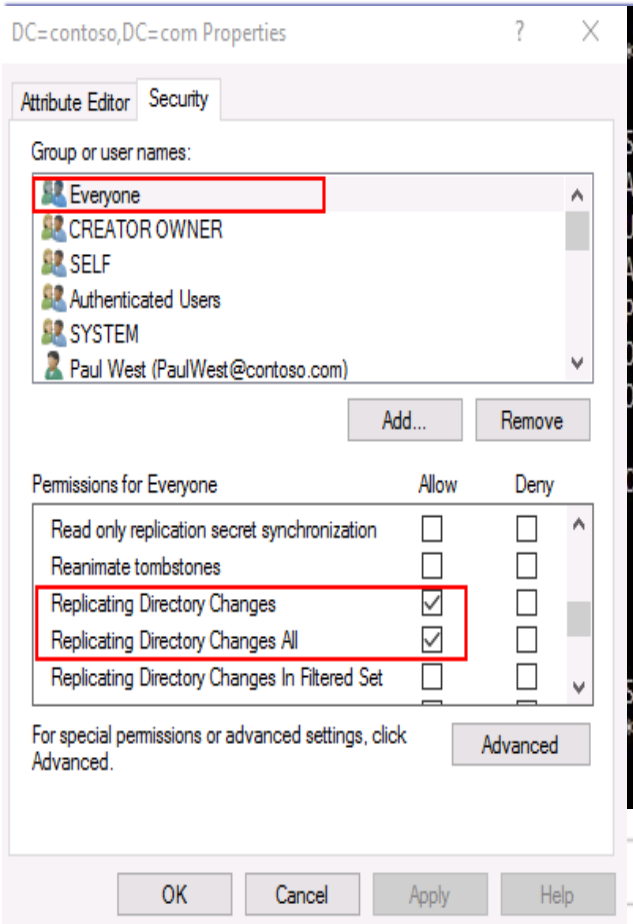
Permission entries:

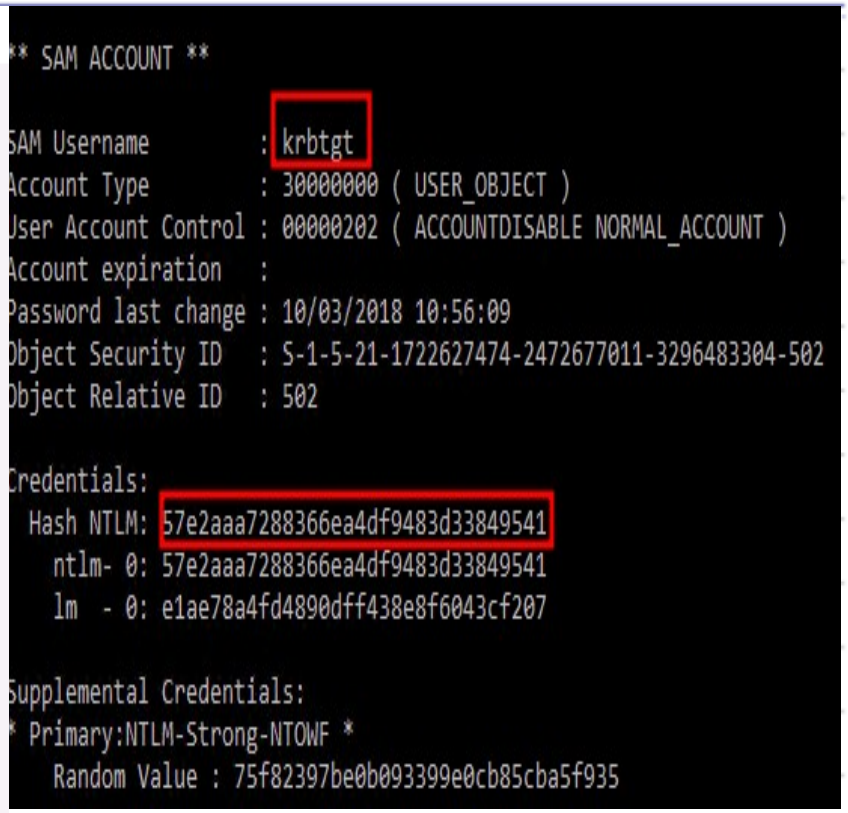
Type	Principal	Access	Inherited from	Applies to
Allow	Authenticated Users	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only
Allow	Everyone	Full control	None	This object and all descendan...
Allow	Pre-Windows 2000 Compatib...	Special	DC=contoso,DC=com	Descendant InetOrgPerson o...
Allow	Pre-Windows 2000 Compatib...	Special	DC=contoso,DC=com	Descendant Group objects

WriteOwner → Impact

- **WriteOwner** can leverage to Full Control on an OU and will be allowed to have full permissions on all the Sub-OU's
- Since we have leveraged to Full control, we're allowed to have full control on all user accounts in the sub-OU's.
- **Impact:**
- Reset account password of your CFO?
- Set SPN on accounts to kerberoast them
- Enable ''Do not require Kerberos preauthentication'' for users
- Add/remove them in different groups that exist in the (Sub) OU's
- Delete accounts
- SID-History modification?
- **Critical**
- If you compromise a user with **WriteOwner** on the **Domain Root**, you can perform a DCSync attack then without Domain Admin.

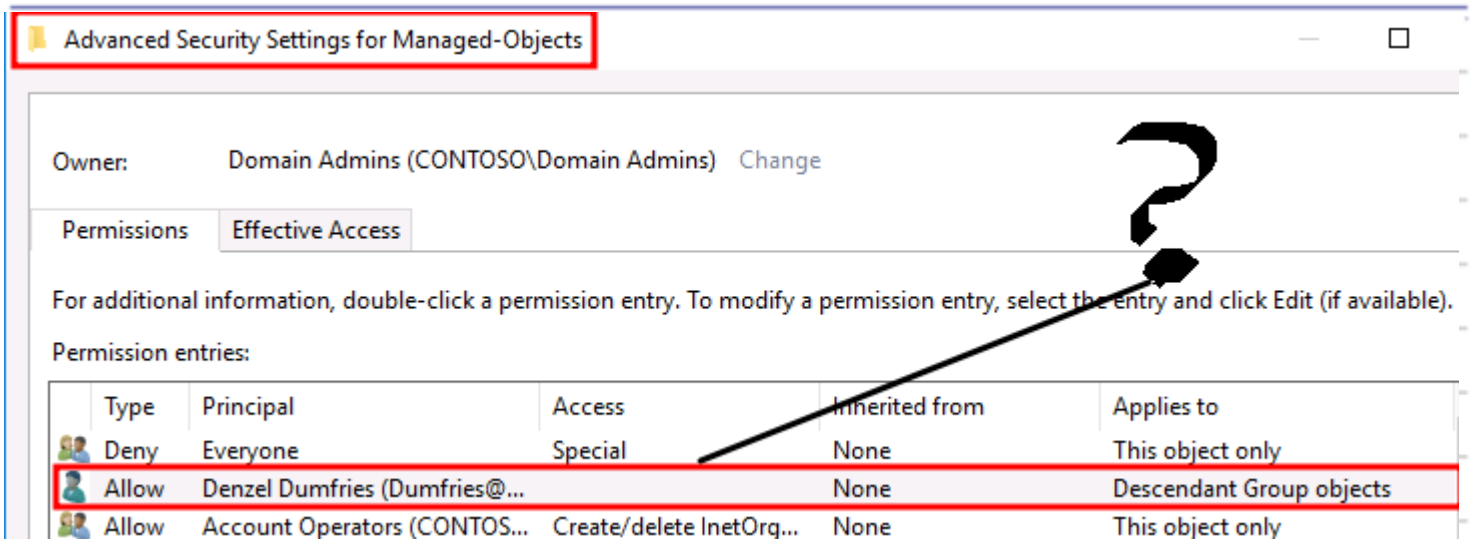
WriteOwner → Change Owner (on Domain Root) → Become Owner → Full Control = DcSync

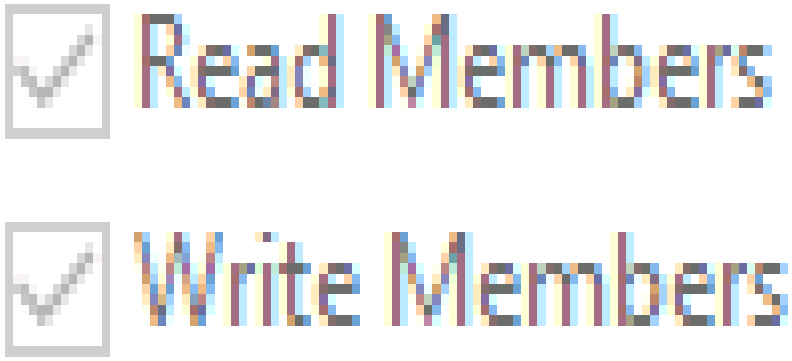




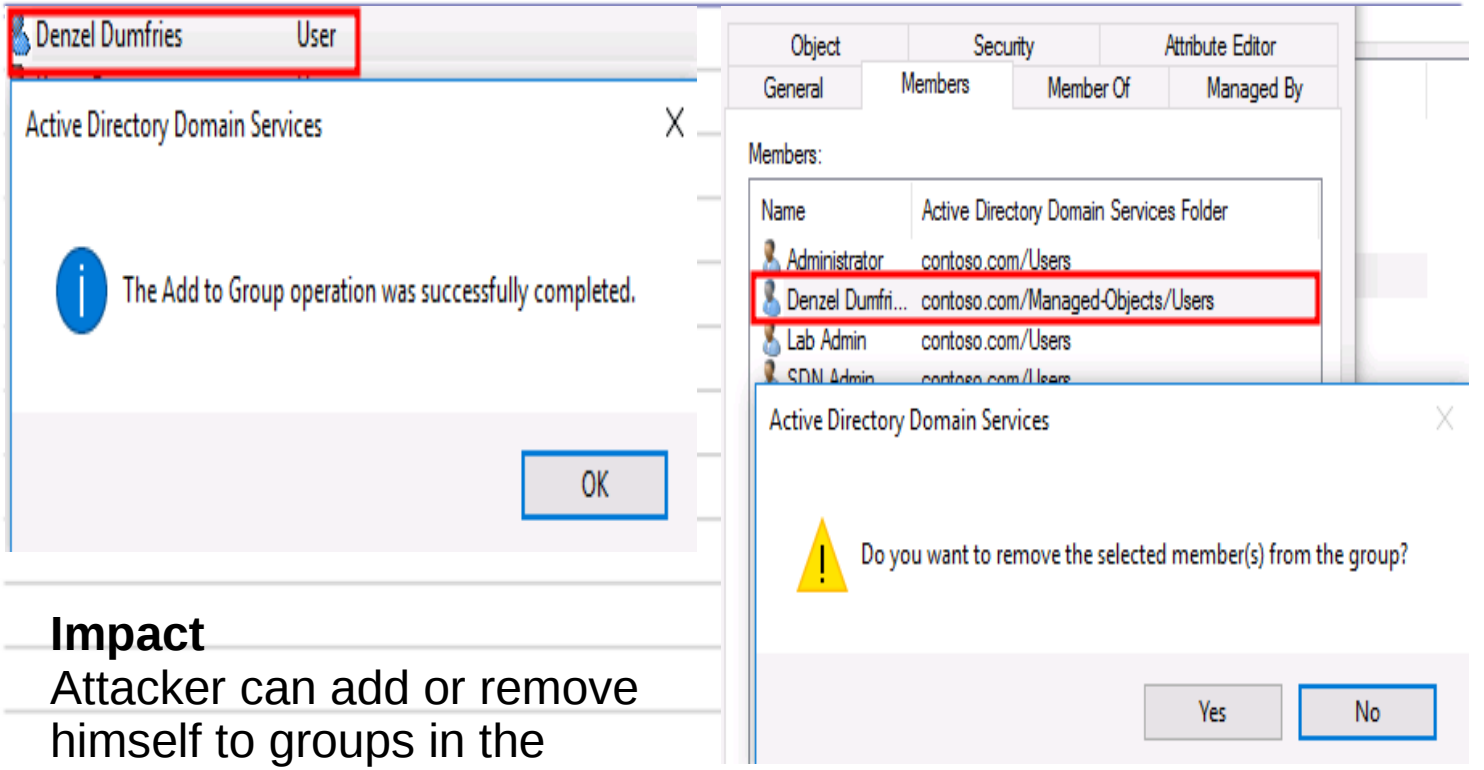
It's not always common, but if you find a user with something as WriteOwner on the Domain Root (Applies to all objects). It's enough to compromise that user to own the entire domain.

WriteMember → Add or remove any user to OU





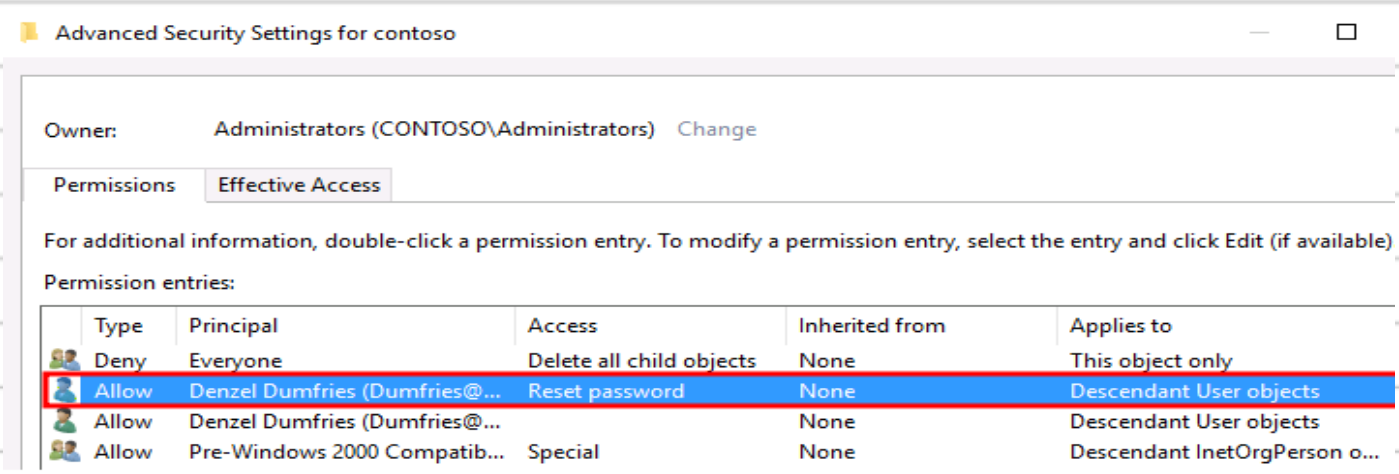
WriteMember → Impact



Impact
Attacker can add or remove himself to groups in the specific OU, and those groups might have access to resources that are sensitive.

User-Force-Change-Password

- Permits resetting a password on a user account.
- Look for all OU's and look if there are (unauthorized) users who can reset your CEO's password
- Compromising a user who can reset the CEO's password is sometimes enough as well for the attacker. Can be detected easy if logging is in place.



Keep in mind → ***''Applies to''*** who? ;-)

What is the difference?

Allow	ENTERPRISE DOMAIN CONTROLLERS		None	Descendant Computer objects
Allow	ENTERPRISE DOMAIN CONTROLLERS		None	Descendant Group objects
Allow	ENTERPRISE DOMAIN CONTROLLERS		None	Descendant User objects
Allow	Jeroen Zoet (Zoet@contoso.com)	Special	None	Descendant Group objects
Allow	SELF		None	Descendant Computer objects

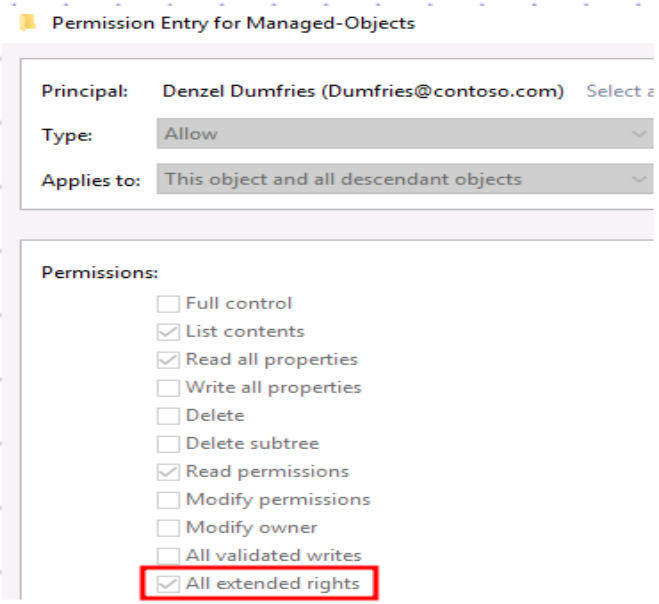
Allow	SELF		None	This object and all descendant objects
Allow	SELF	Special	None	All descendant objects
Allow	Denzel Dumfries (Dumfries@contoso.com)	Special	None	This object only
Allow	Domain Admins (CONTOSO\Domain Ad...	Special	None	This object only

Stupid meme.

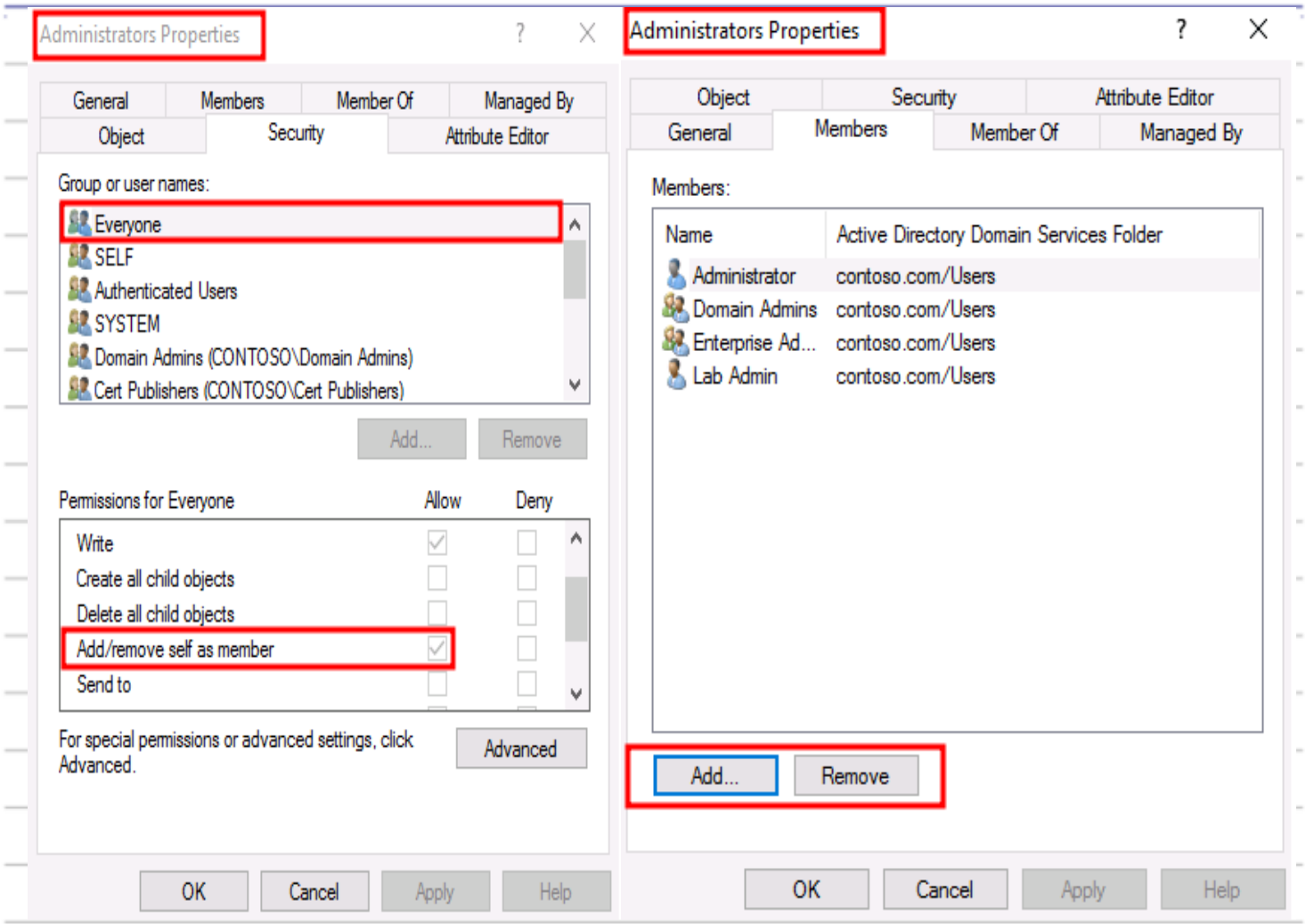

I CHANGED ALL MY PASSWORDS TO "INCORRECT"
SO WHENEVER I FORGET, IT WILL TELL ME "YOUR PASSWORD IS INCORRECT."

AllExtendedRights → Ability to change passwords

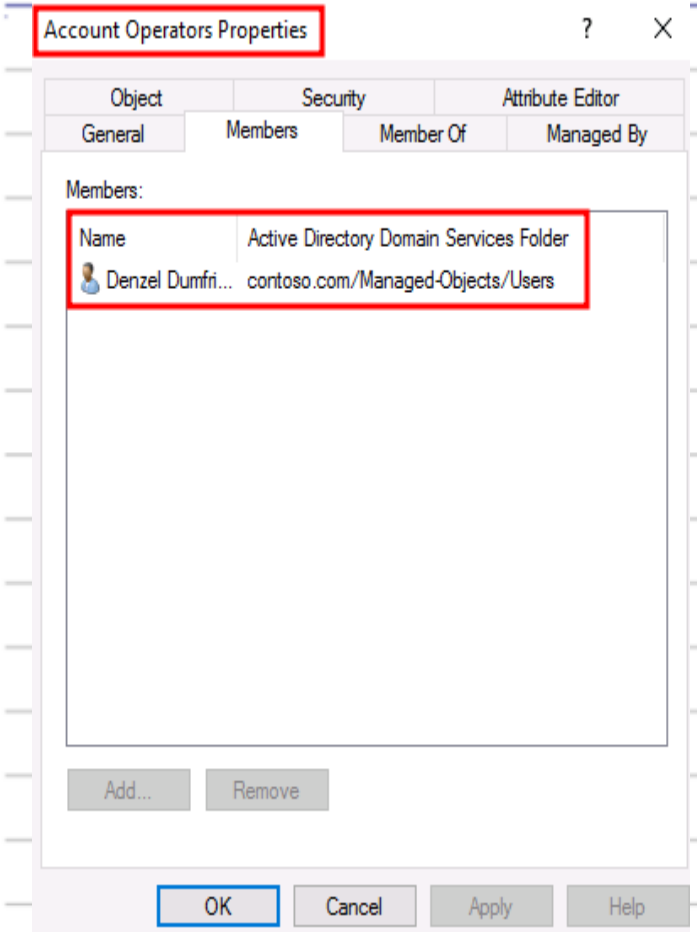
- All Extended Rights is needed in organizations that has deployed Microsoft LAPS solutions.
- Users with the ‘’All Extended Rights’’ are able to see the password(s) in **ExtendedRightholders** of LAPS
- Compromising a user with the ‘’All Extended Rights’’ could allow an attacker to **reset the password of users** in OU or see the password of users their built-in local administrator account if LAPS is deployed.

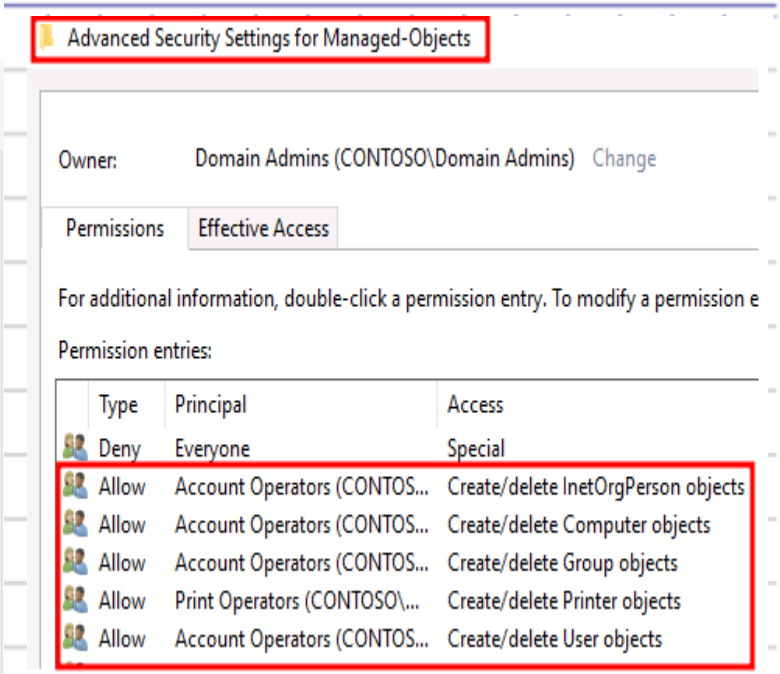


You only live once, they said? :D



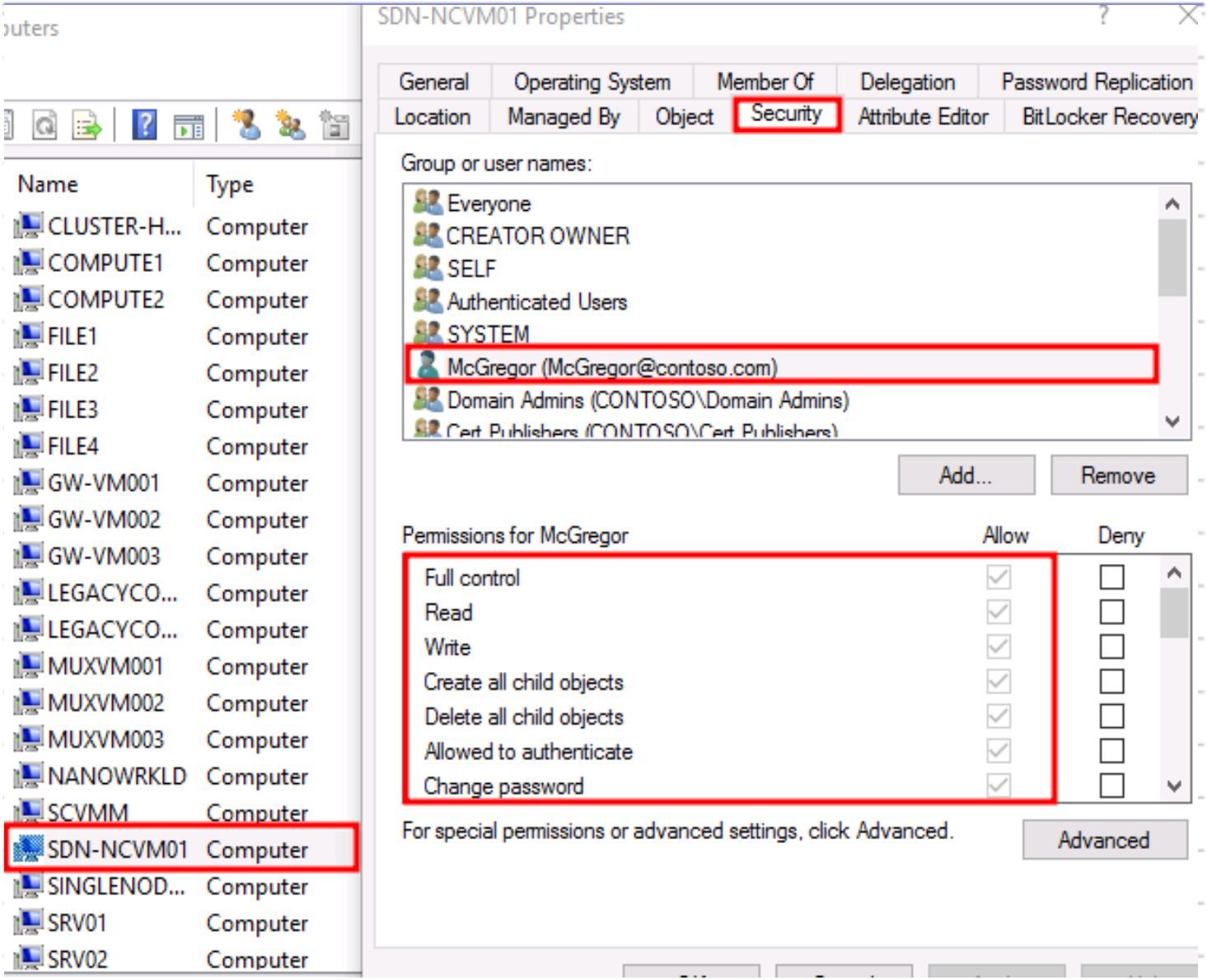
Account Operators → Too many permissions than needed!





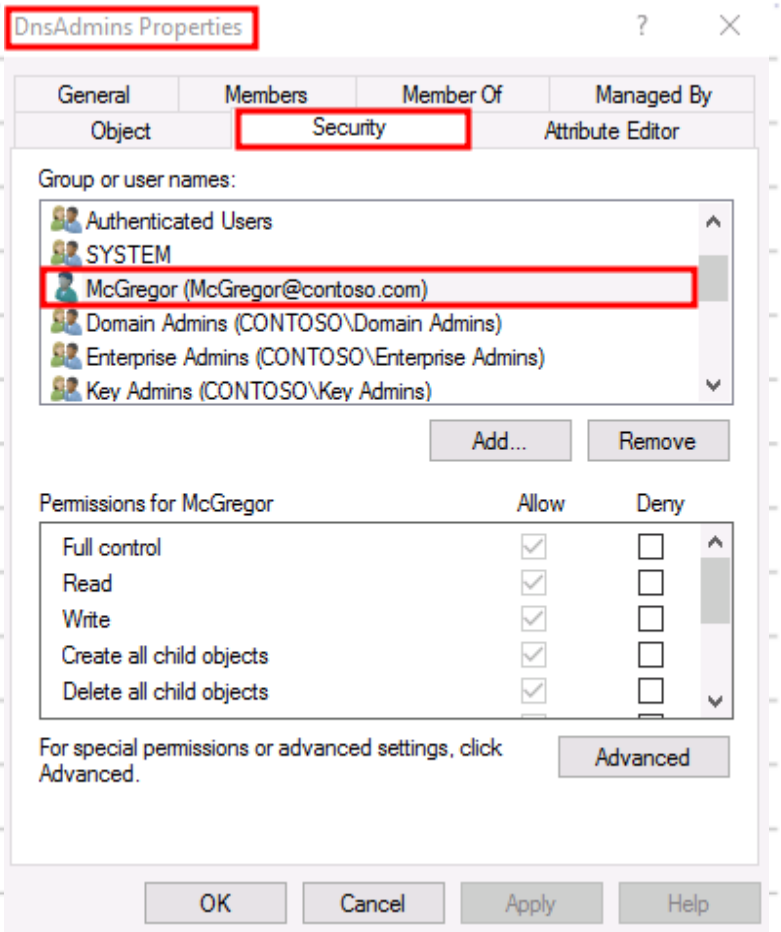
Compromising users from the Account Operators group allows the attacker to reset passwords of accounts, set SPN's, add/remove users in group or enable *"Do not require Kerberos authentication" etc.*

Overlooked things → Who can control which Computer Objects?

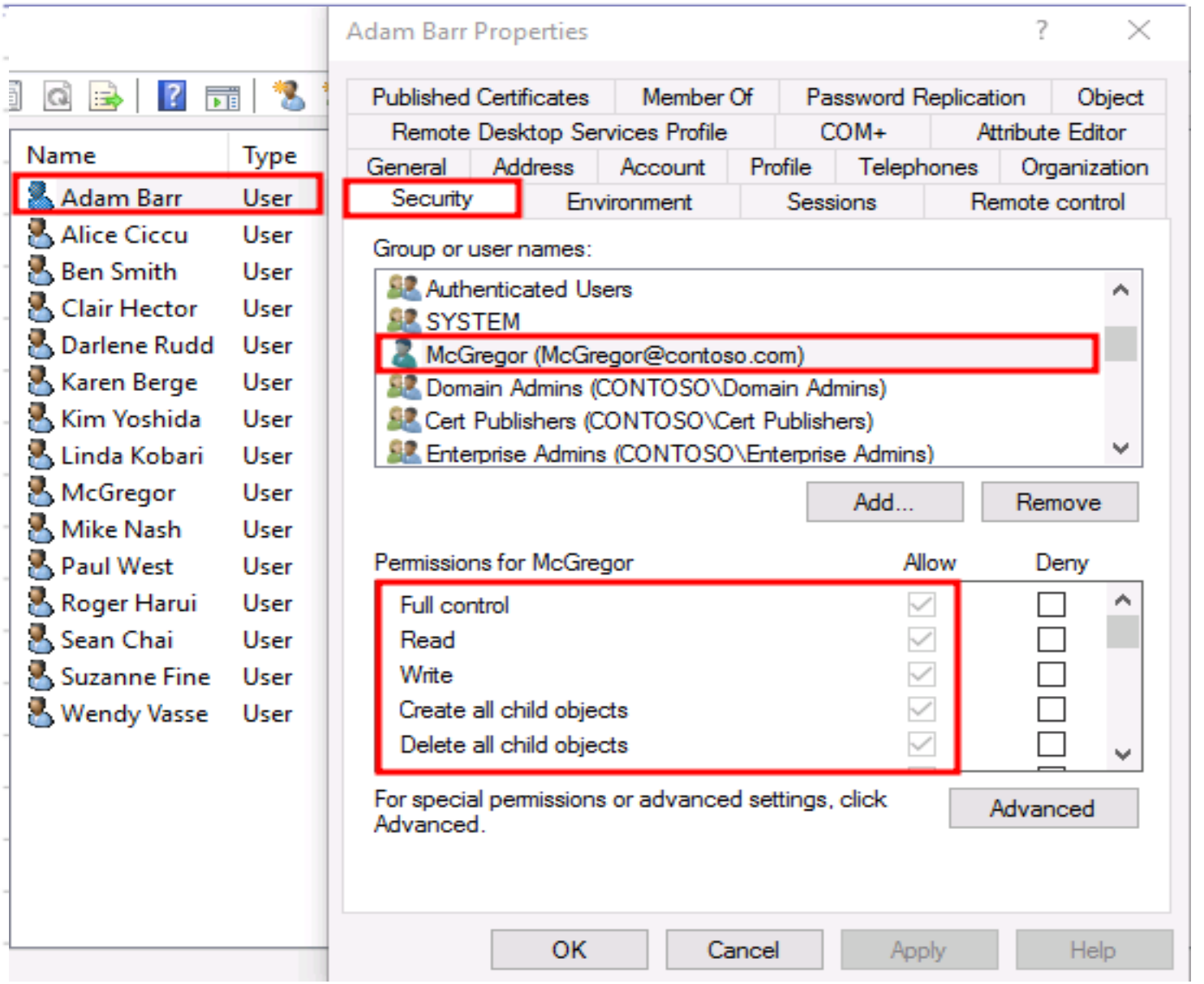


Overlooked things → Who can control AD groups?

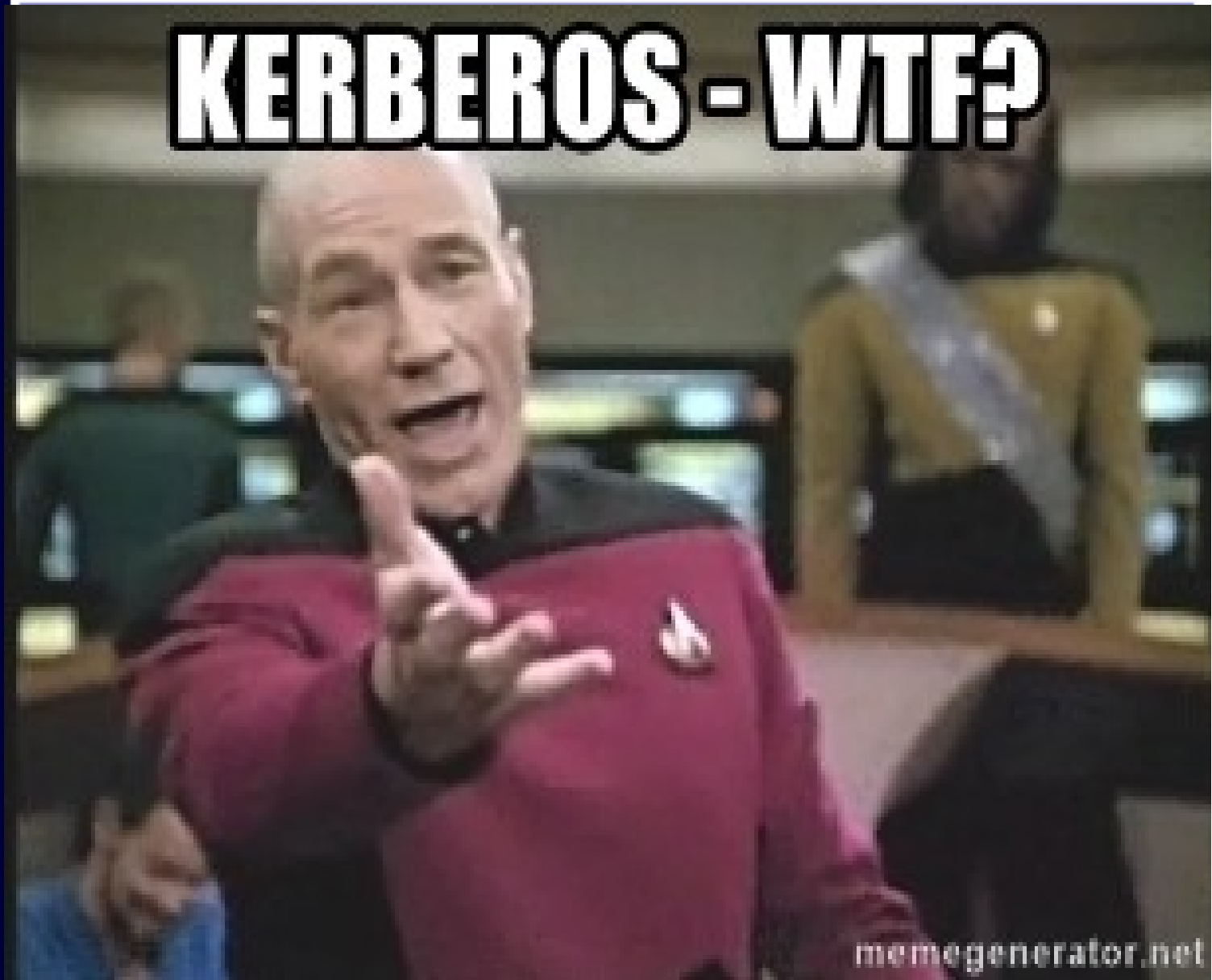
- McGregor has Full control on the DnsAdmins group for example. Allowing him to add or remove any users in that group.
- Look to which groups have the Add/remove self as member.



Overlooked things → Who can control User accounts?

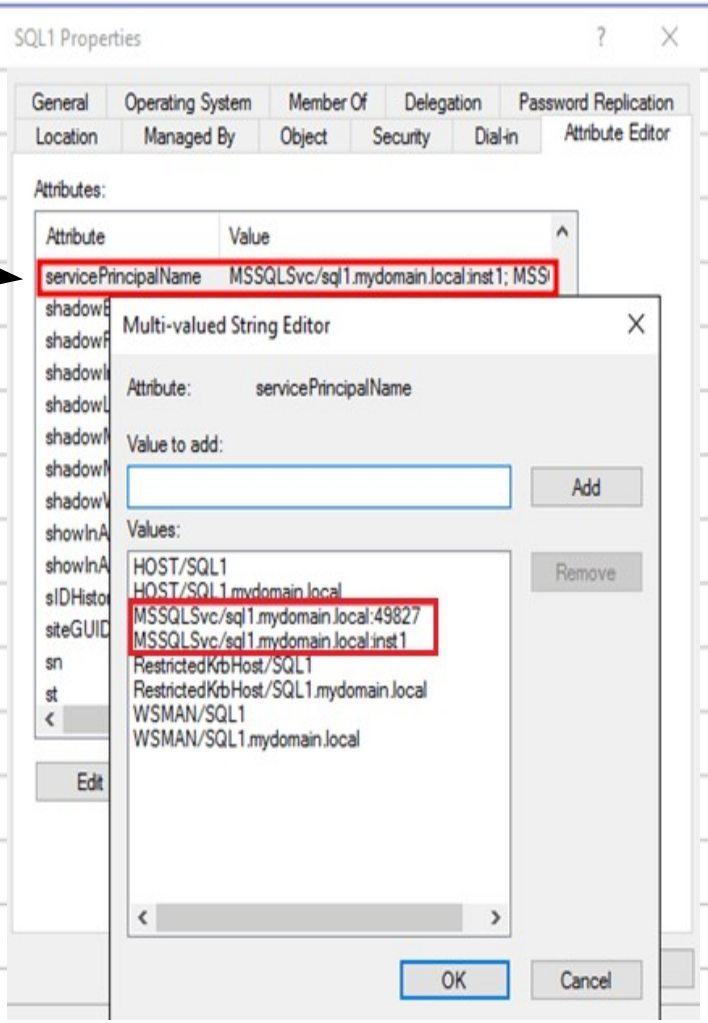


Kerberos attacks → WTF?



Kerberoasting → Accounts with SPN values

- Kerberoasting is basically brute force on Service accounts with a SPN value (**servicePrincipalName**) – It takes advantage of accounts using weak passwords.
- A service account is a special user account that an application or service uses to interact with the operating system. Like SQL Server, Microsoft-IIS Web server, SharePoint etc.
- This attack is usually done, when an authenticated user hasn't much elevated privileges in the domain.
- **Attack phase**
- Discover SPN accounts
- Request service tickets with open-source tools like Mimikatz or Kerberom
- Extract the tickets
- Brute force the tickets with Hashcat



Kerberoasting → Request service tickets

```
C:\>klist

Current LogonId is 0:0x4b0b7

Cached Tickets: (4)

#0>     Client: jasonAdmin @ 2008R2.LOCAL
        Server: krbtgt/2008R2.LOCAL @ 2008R2.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 7/3/2013 10:54:18 (local)
        End Time:   7/3/2013 20:54:18 (local)
        Renew Time: 7/10/2013 10:54:18 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96

#1>     Client: jasonAdmin @ 2008R2.LOCAL
        Server: HTTP/teams.2008r2.local @ 2008R2.LOCAL
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
        Start Time: 7/3/2013 11:20:05 (local)
        End Time:   7/3/2013 20:54:18 (local)
        Renew Time: 7/10/2013 10:54:18 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
```

Attacker will request the <u>service ticket</u> of the SPN account and start cracking it without worrying for lockouts.

Indicator of Compromise

Event ID: 4769 / 4768
A Kerberos service ticket was requested.
Ticket Options: 0x40810000
Ticket Encryption Type: 0x17

Kerberoasting → Used in the wild

BRONZE BUTLER has created forged Kerberos Ticket Granting Ticket (TGT) and Ticket Granting Service (TGS) tickets to maintain administrative access.[2]

Fun Fact:

Users with "**GenericWrite**" can set SPN values on accounts for Kerberoasting.

Users in **Account Operators** can do this as well.

- Kerberoasting has been used in real life attacks by an APT group called "**Bronze Butler**"
- *"**BRONZE BUTLER** is a cyber espionage group with likely Chinese origins that has been active since at least 2008. The group primarily targets Japanese organizations, particularly those in government, biotechnology, electronics manufacturing, and industrial chemistry."*
- Kerberoasting is often, not always, but often a first step that is required to perform a Silver Ticket to remain persistence for the SQL Server for example.

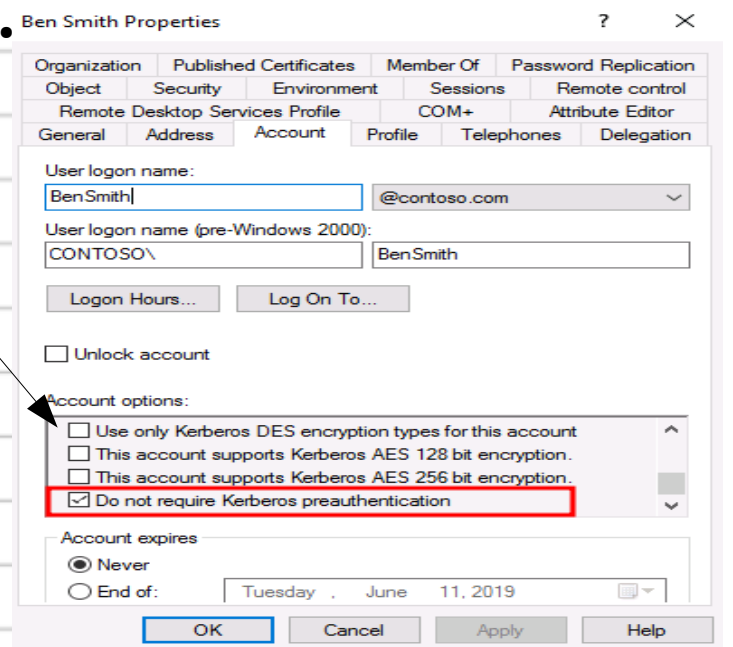
Recommendation

- Since the attack doesn't required any privileges and even can be used to crack the service ticket(s) **offline**, it's required to set at least a **25+ character for all the Service accounts.**
- Using Group Managed Service accounts is good, because it will enforce random, complex passwords that can be automatically rotated and managed centrally within Active Directory.
- **NOTE**
- If you have plans to use this, please test this first in steps to see if you don't break things. More important is just to have a procedure for resetting service accounts their passwords.

AS-REP Roasting → Additional Information

- Kerberos Pre-Authentication is a security feature which offers protection against password-guessing attacks. The AS request identifies the client to the KDC in Plaintext.
- **<u>Without</u> Kerberos Pre-Authentication** a malicious attacker can directly send a dummy request for authentication. The KDC will return an encrypted TGT and the attacker can brute force it offline.
- Checking the KDC logs, nothing will be seen except a single request for a TGT.
- AS-Rep Roasting is basically also brute force, but a bit different from Kerberoasting.

- **Difference:** AS-REP requests a Kerberos Authentication Ticket (TGT) not a service authenitcation ticket (TGS).
- **Do not require Kerberos pre authentication** needs to be enabled to do this.



AS-REP Roasting → Cracking users password

```
$ hashcat -m 18200 -O ./GetNPUsers-impacket-output.txt -a 3 '?u?a?a?a?a?a?l?l'
hashcat (v5.0.0-8-g81a460496) starting...

Session..........: hashcat
Status...........: Running
Hash.Type........: Kerberos 5 AS-REP etype 23
Hash.Target......: $krb5asrep$23$rachel@domain.local:806b2c95090c80775...3049df
Time.Started.....: Wed Oct 31 12:49:49 2018 (3 secs)
Time.Estimated...: Wed Oct 31 22:00:45 2018 (9 hours, 10 mins)
Guess.Mask.......: ?u?a?a?a?a?a?l?l [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   423.3 MH/s (8.27ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#2.........:   422.8 MH/s (8.25ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#3.........:   420.7 MH/s (8.28ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#4.........:   420.8 MH/s (8.28ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#5.........:   423.7 MH/s (8.24ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#6.........:   418.8 MH/s (8.36ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#7.........:   421.6 MH/s (8.29ms) @ Accel:128 Loops:16 Thr:64 Vec:1
Speed.#8.........:   287.2 MH/s (8.72ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Speed.#9.........:   292.7 MH/s (8.57ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Speed.#10........:   295.4 MH/s (8.47ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Speed.#11........:   287.2 MH/s (8.74ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Speed.#12........:   287.2 MH/s (8.74ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Speed.#*.........:  4401.5 MH/s

$krb5asrep$23$rachel@domain.local:806b2c95090c807753166ac2b8b86df1$027caf31e91c3a0382d526d760c5008be89c4862726cd66
7394378882c4ab014801759022de3d73f9f03bb3aa0de8c97c0f36192ccb5c94771ae1eb99a94b3e39f8397daff2287802ad4ab2861afb129d
d50df2879a50d3b1dadc2e250b545da9b779699e422da0e0e0874a4e97c693766f152dd2753364e4dbf10a42ae3b05bc752b7c2fd802ab937e
ce398d406383210964ab59717598f4b136140225dcadc70db1b9c0c75e0059790bbc9d6727c7d2e89560ae3761e71ff9ab30e77454bf9227d9
36ddc3049df:P@ssw0rd
```

AS-Resp Roasting is possible with Hashcat for example since this module has been added to it somewhere in 2018.

Event ID: 4768
A Kerberos authentication ticket (TGT) was requested
Ticket Options: 0x40810000
Ticket Encryption Type: 0x17

AS-REP Roasting → Recommendation

Recommendation

- Scan for all accounts that are vulnerable with the ASREPROAST tool and **uncheck** the *"Do not require preauthentication"* checkbox
- Enforce using strong passwords on accounts, because the attacker doesn't need to worry about getting locket-out.
- Monitor event 4768 on the Domain Controller, where a TGT is requested with 0x17 as additional information.

Rubeus.exe asreproast /outfile:hashes.txt /format:hashcat [/user:USER] [/domain:DOMAIN] [/dc:DOMAIN_CONTROLLER] [/ou:"OU=,..."]

- **Rubeus** to perform AS-REP Roasting
- Rubeus is a C# tool that has been developed by the authors of Mimikatz

Unconstrained Kerberos Delegation

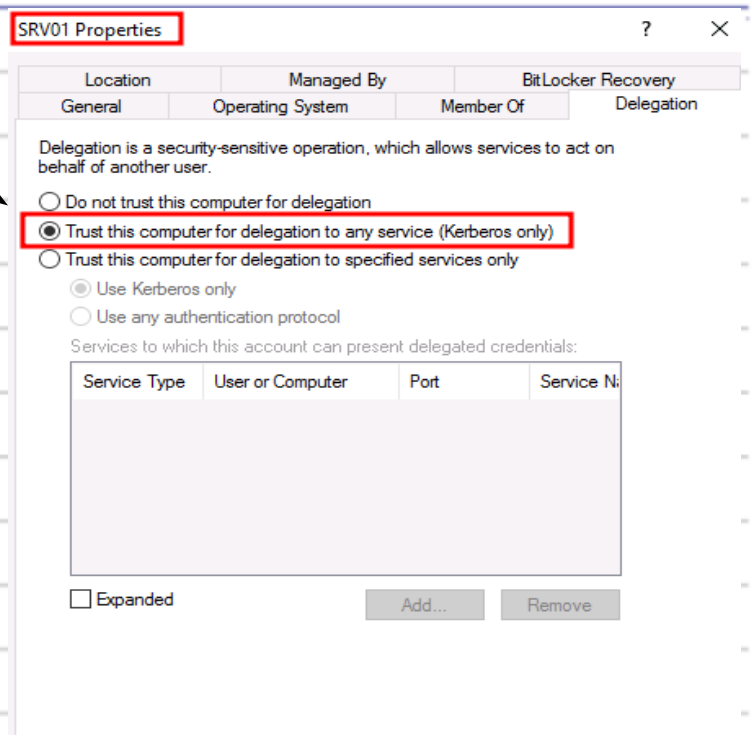
- A server that is trusted for unconstrained delegation is actually allowed to impersonate (almost) any user to any service within the network.
- When a user requests a Service Ticket (ST) from a DC to a service, which is enabled for delegation, the DC will copy the client's Ticket Granting Ticket (TGT) and attach it to the ST, which will later be presented to the service.
- When the user accesses the service with the ST, the user's TGT will be extracted and saved in the server's memory (LSASS) for later use. As a result, the service will be able to impersonate the user to any service within the network.

- **Risk**
- If a server trusted for unconstrained delegation is compromised, the attacker will have access to all of the TGTs of the users that used the service. Using the TGT ticket, an attacker can access all of the resources available in the network with the compromised user's permissions.
-

Unconstrained Kerberos Delegation → **Impact**

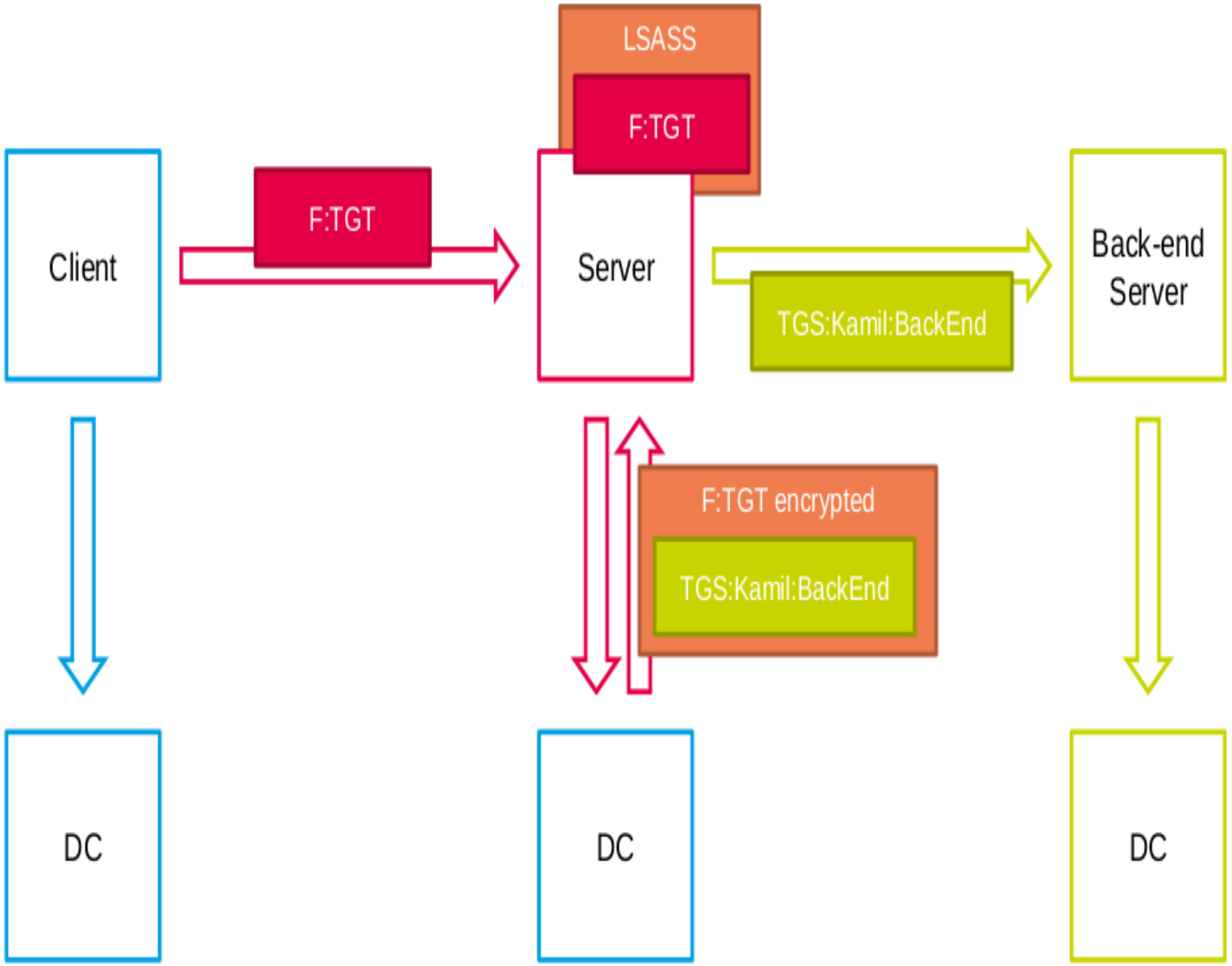
In simple words: This configuration <u>on servers</u> is just bad

- **<u>Impact</u>**
- When the server is compromised, the attacker will gain access to all of the TGT's stored on that server in memory, and consequentially, he might be able to compromise the domain by abusing the SpoolSample for DCSync.
- ''Trust this computer for delegation to any service'' **should only be applied on Domain Controllers**




```
C:\Users\Dumfries>klist

Current LogonId is 0:0xd484f

Cached Tickets: (7)

#0>     Client: Dumfries @ CONTOSO.COM
        Server: krbtgt/CONTOSO.COM @ CONTOSO.COM
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 5/12/2019 11:52:02 (local)
        End Time:   5/12/2019 21:52:02 (local)
        Renew Time: 5/19/2019 11:52:02 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: [redacted]
```

Unconstrained Kerberos Delegation → **Explained**



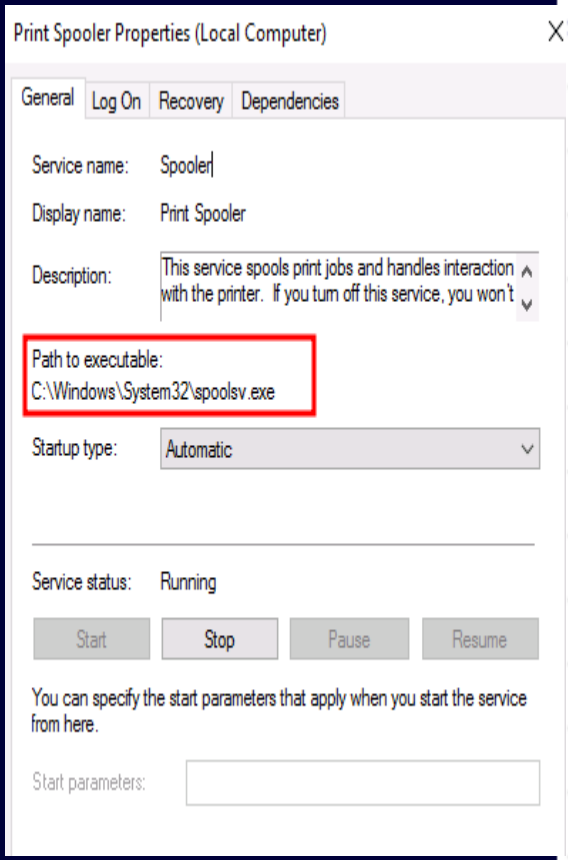
Unconstrained Kerberos Delegation → **DCSync via SpoolSample**

Scenario

- Attacker managed to compromise a server with Unconstrained Kerberos Delegation. Fine, in the two slides before, we've learned that it is possible to impersonate a user that has requested access to the compromised service. Because someone with Local Admin on the compromised server can grab all TGT tickets in the memory.
- **Did you know?!**
- The guys from SpectreOps managed to found a way to compromise just one server that has the unconstrained kerberos delegation enabled to own the entire domain?

- **How the attack works**
- Attacker compromised user with Local Admin privilege on a server that has Unconstrained Kerberos Delegation enabled.
- Attacker looks for the Domain Controller that has the Print Spooler enabled by default.
- Attacker sends the MS-RPRN request to the DC
- DC replies back to the attacker and a service ticket will be created with the DC **Computer account** (e.g. WIN-DC1$) that contains the Kerberos TGT ticket that we've learned.
- Attacker can now impersonate a DC
- DCSync is possible with Unconstrained Kerberos Delegation!!
- Result = **DOMAIN OWNED!**

Unconstrained Kerberos Delegation → **Print Spooler**



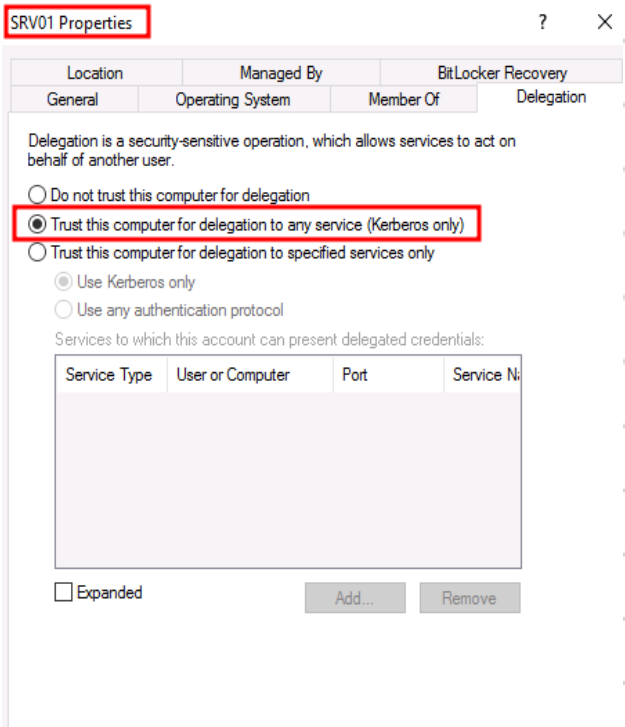
Main Success Scenario:

1. The user initiates a print job at a command prompt by executing a **copy /b FILE \\SERVER\PRINTQ** command where FILE is a local file containing print job data, SERVER is the name of the server, and PRINTQ is the name of a shared print queue.
2. The print client uses the SMB protocol family to submit the file containing the print job data to a printer share on the print server.
3. The print client indicates the start of a new logical page to the print server, repeatedly sends data for the page, and signals the end of a logical page to the print server by using the Print System Remote Protocol [MS-RPRN]. The print client repeats this step for all pages in the document.
4. After sending all the pages of the print job to the print server, the print client ends the print job by using the Print System Remote Protocol.
5. The print client closes the printer handle by using the Print System Remote Protocol.

Do you need to disable it Spoolsv.exe?
No, you don't have to. Because there are more serious problems.
You should avoid using Unconstrained Kerberos Delegation

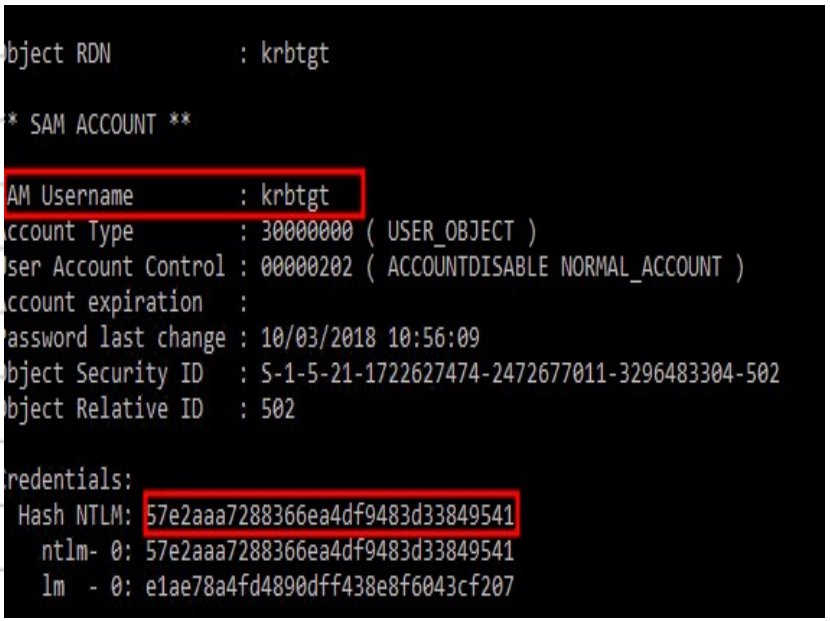
Unconstrained Kerberos Delegation → **DCSync via Spoolsv.exe**

- **Avoid allowing servers having this setting**, but if they really need it for x reasons. Make sure it's Constrained Kerberos Delegation to mitigate the risk.

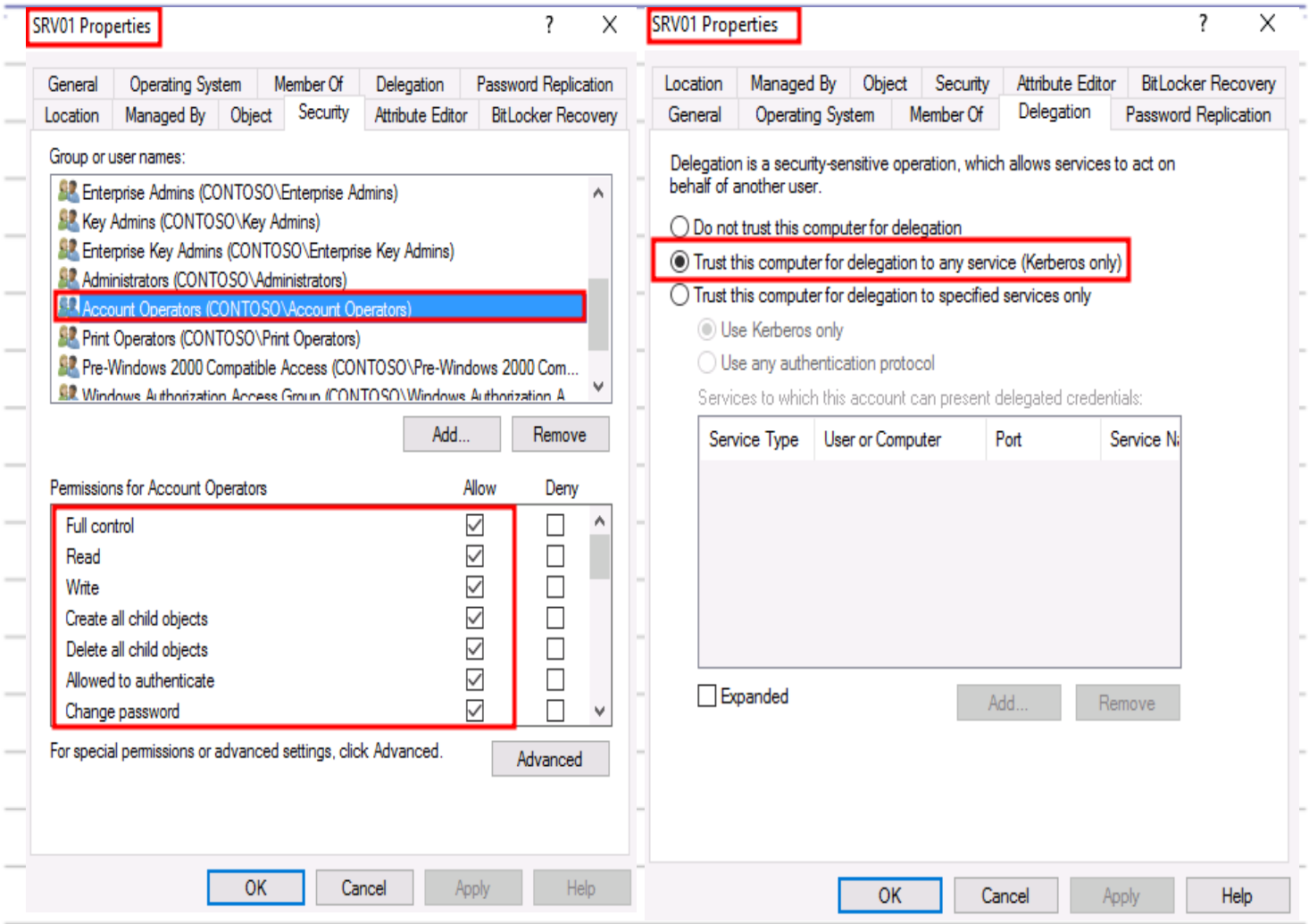


Impact

- Otherwise it's enough for an attacker to compromise just one server with Unconstrained Kerberos Delegation to perform a **DCSync** attack and grab the krbtgt hash to own the domain.



Account Operators → Can enable Unconst. Kerberos Delegation



WriteDacl → Descedant Computer Objects = Unconst. Kerberos Delegation

Advanced Security Settings for Managed-Objects

Owner: Domain Admins (CONTOSO\Domain Admins) Change

Permissions | Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Special	None	This object only
Allow	Denzel Dumfries (Dumfries@...	Special	None	Descendant Computer objects
Allow	Account Operators (CONTOS...	Create/delete InetOrg...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Comput...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete Group o...	None	This object only
Allow	Print Operators (CONTOSO\...	Create/delete Printer o...	None	This object only
Allow	Account Operators (CONTOS...	Create/delete User obj...	None	This object only
Allow	Domain Admins (CONTOSO\...	Full control	None	This object only
Allow	ENTERPRISE DOMAIN CONT...	Special	None	This object only
Allow	Authenticated Users	Special	None	This object only

SRV01 Properties

Location | Managed By | Object | Security | Attribute Editor | BitLocker Recovery
General | Operating System | Member Of | Delegation | Password Replication

Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

- ○ Do not trust this computer for delegation
- ◉ Trust this computer for delegation to any service (Kerberos only)
- ○ Trust this computer for delegation to specified services only
  - ◉ Use Kerberos only
  - ○ Use any authentication protocol

Services to which this account can present delegated credentials:

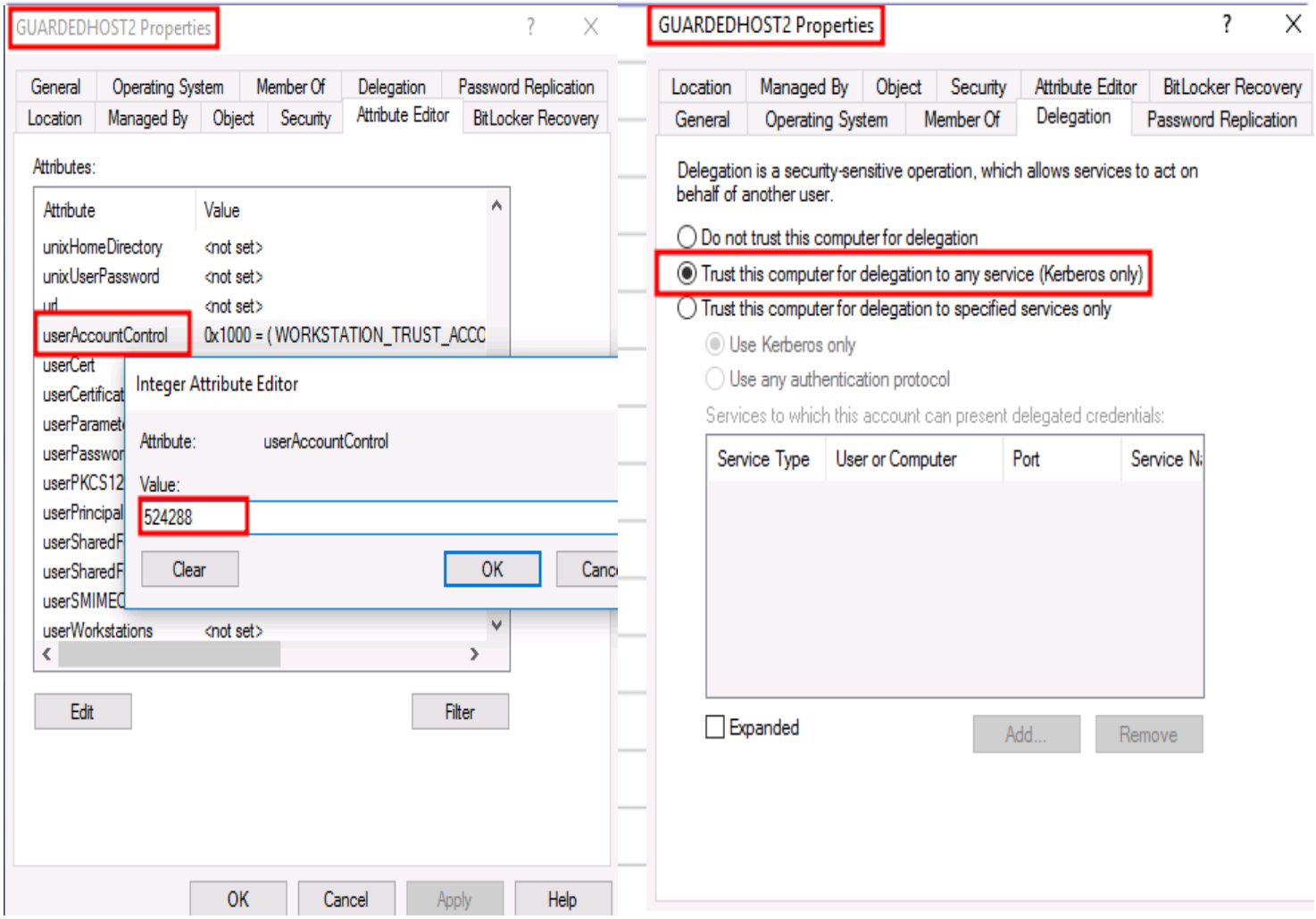
Service Type	User or Computer	Port	Service N

☐ Expanded Add... Remove

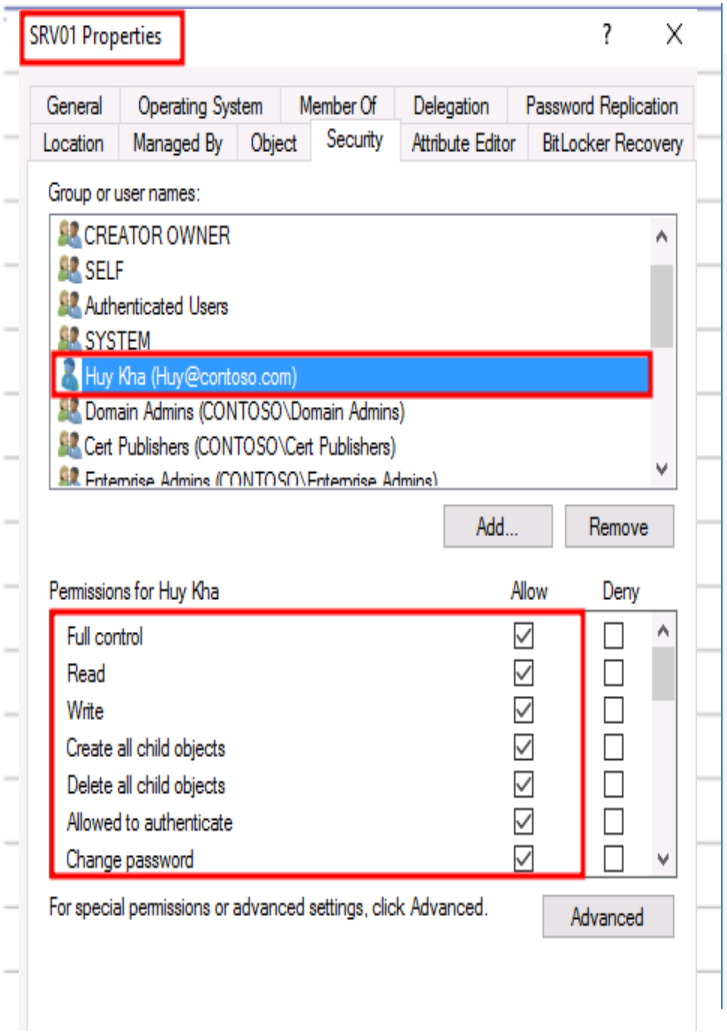
OK | Cancel | Apply | Help

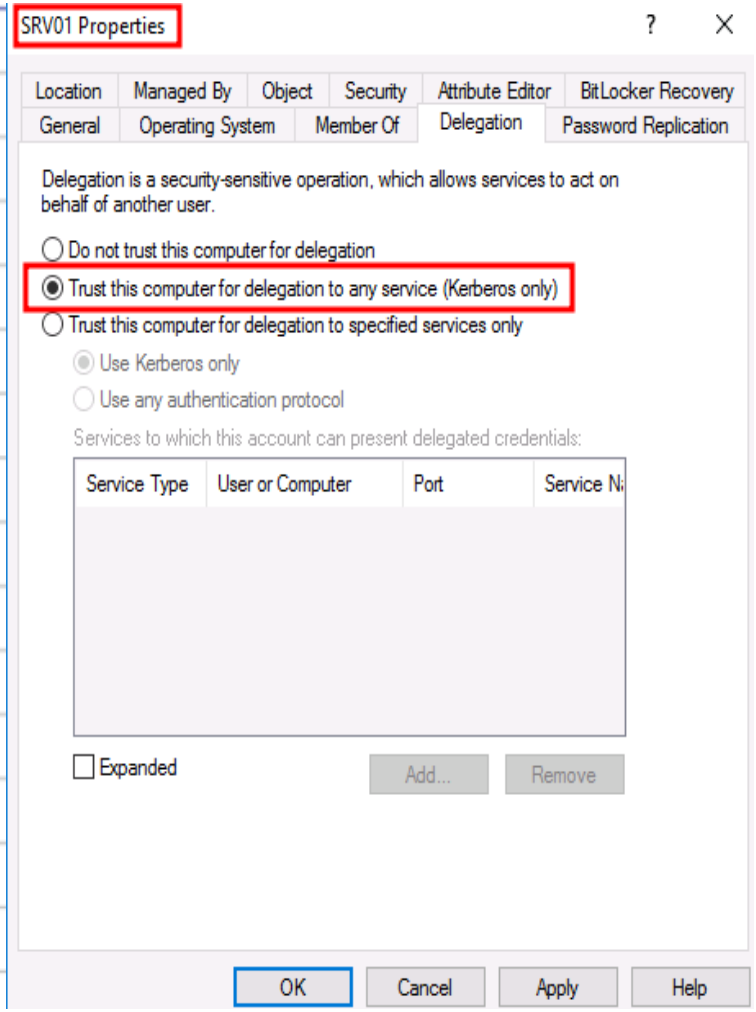
Users with ‘’Modify permission’’ on Descedant Computer Objects can enable Unconstrained Kerberos Delegation.

GenericWrite → Descedant Computer Objects – Can enable Unconst. Kerberos Delegation



(Random) Hiden users with Full control on ‘’critical” servers ¯\(ツ)/¯



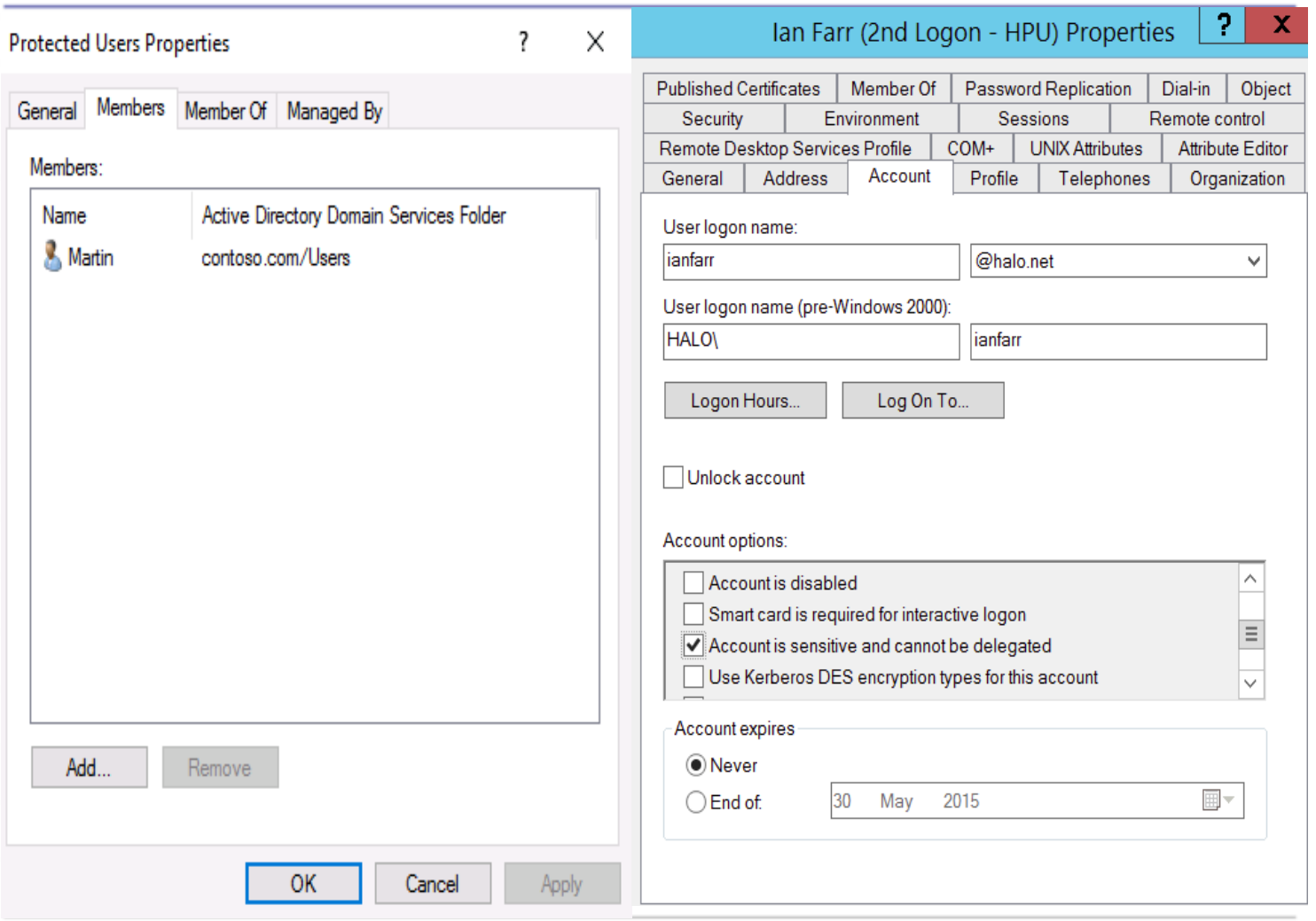


Unconstr. Kerberos Delegation - Conclusion

- Thanks to the guys from SpectreOps. Unconstrained Kerberos Delegation now have much more of an ‘‘high’’ impact.
- **Unconstr can lead to DCSync** :)
- Make sure that your security policy team **prevents** vendors for requiring the ‘‘Unconstrained Kerberos Delegation’’ configuration.
- **Security professional check**
- Who can enable Unconstr. Kerberos Delegation on your servers? Does he/she needs to be able to do that? (Check permission)
- Are there any ‘‘hiden’’ users with Full control on OU=Servers?
- Account Operators can enable it, please avoid this group.

- **What to do about it?**
- CISO & CIO needs to be aware of this insecure configuration. Both needs to agree that this should not (anymore) be supported if a vendor or someone else is requiring this.
- If you really can’t turn it off. Try Constrained Kerberos Delegation. Still not good, but okay.
- All high-priv users, usually DA. Should be in the Protected Users group, and ensure that those accounts are sensitive and cannot be delegated.

Domain Admins → Add them in Protected Users / Account is sensitive and cannot be delegated



Group Policy Objects → Unauthorized users with Edit rights.

Default Domain Controllers Policy

Scope | Details | Settings | Delegation | Status

These groups and users have the specified permission for this GPO

Groups and users:

Name	Allowed Permissions	Inherited
Authenticated Users	Read (from Security Filtering)	No
Domain Admins (CONTOSO\Do...	Custom	No
Enterprise Admins (CONTOSO\E...	Custom	No
ENTERPRISE DOMAIN CONTR...	Read	No
Huy Kha (Huy@contoso.com)	Edit settings, delete, modify security	No
SYSTEM	Edit settings, delete, modify security	No

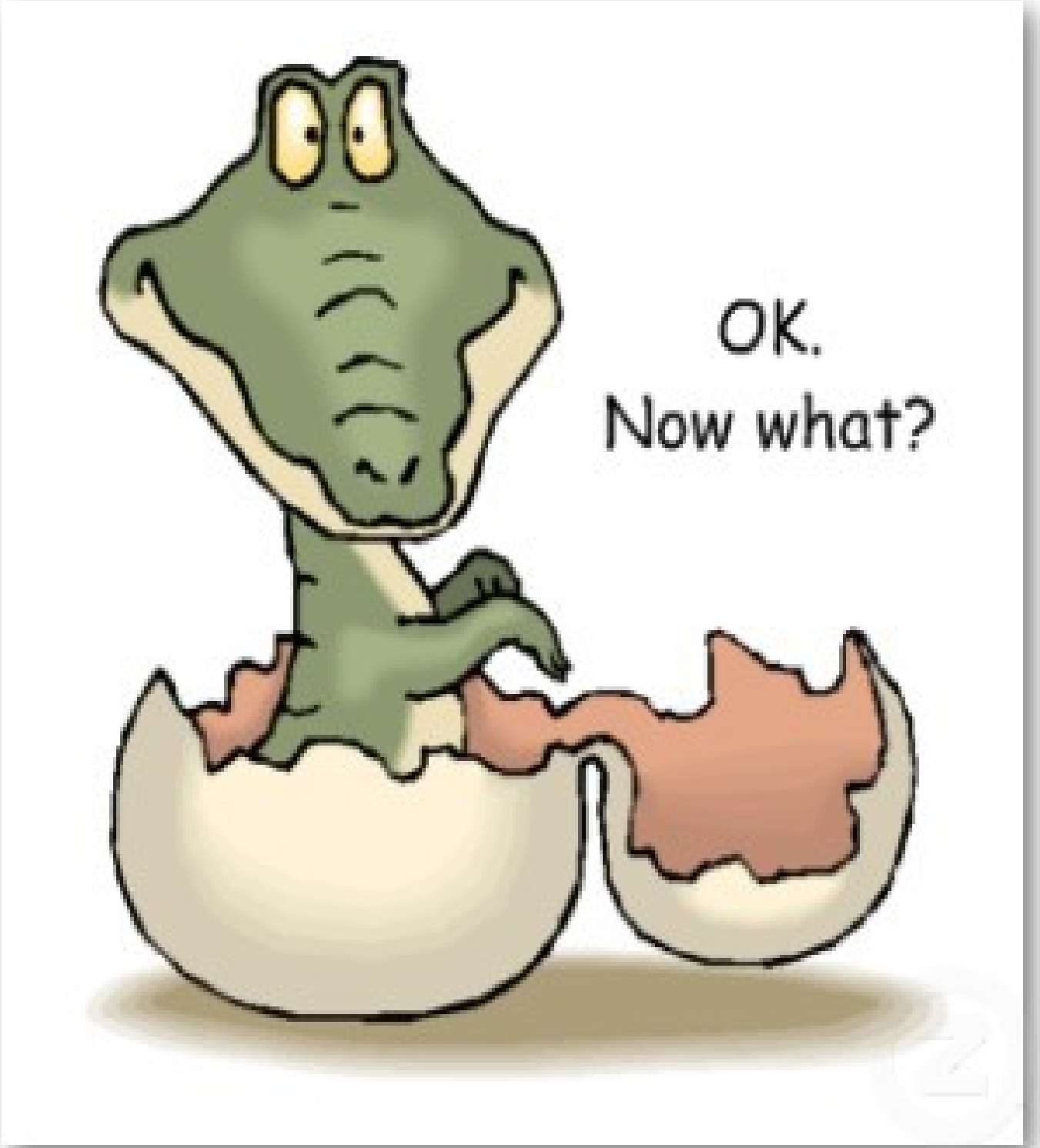
Default Domain Policy

Scope | Details | Settings | Delegation | Status

These groups and users have the specified permission for this GPO

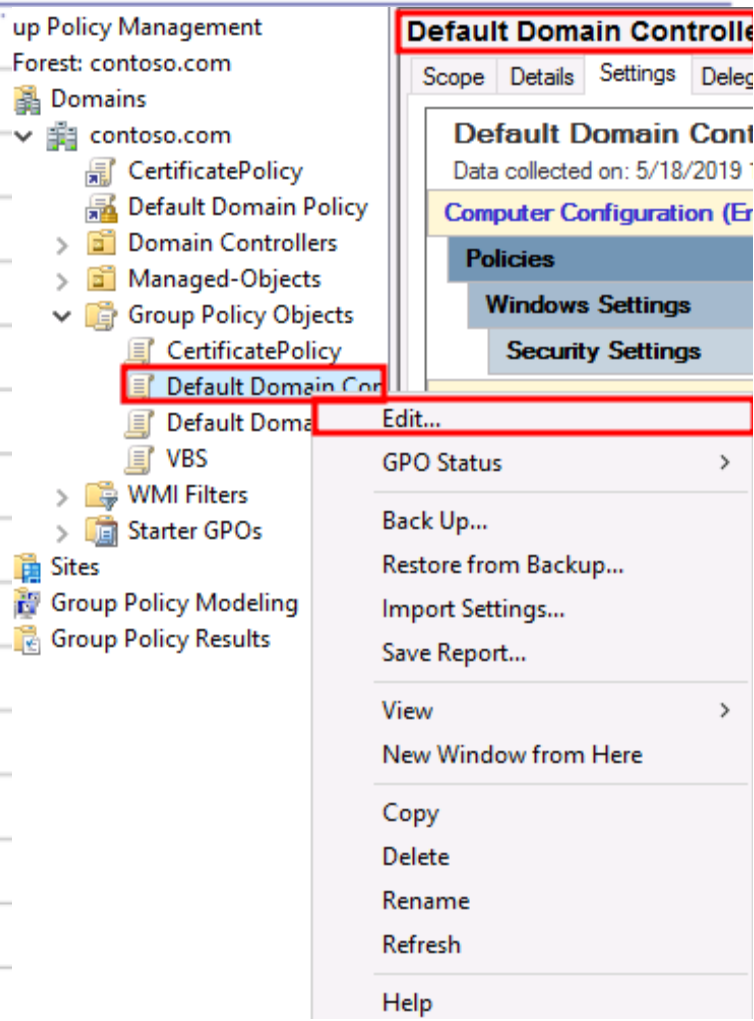
Groups and users:

Name	Allowed Permissions	Inherited
Authenticated Users	Read (from Security Filtering)	No
Domain Admins (CONTOSO\Do...	Custom	No
Domain Users (CONTOSO\Dom...	Edit settings	No
Enterprise Admins (CONTOSO\E...	Custom	No
ENTERPRISE DOMAIN CONTR...	Read	No
SYSTEM	Edit settings, delete, modify security	No


OK.
Now what?

Group Policy Objects → Modify GPO linked to DC :)

Lets start to make it even more insecure (evil laugh) :D



Group Policy Objects → Allow ‘’Everyone’’ to log on the DC

Policy	Policy Setting
Access Credential Manager as a trusted caller	Not Defined
Access this computer from the network	Everyone,ENTERPRISE DOMAIN CONTROLLERS,Authenticated Users,Administrators
Act as part of the operating system	Not Defined
Add workstations to domain	Authenticated Users
Adjust memory quotas for a process	NETWORK SERVICE,LOCAL SERVICE,Administrators
Allow log on locally	Everyone,ENTERPRISE DOMAIN CONTROLLERS,Backup Operators,Administrators
Allow log on through Remote Desktop Services	Everyone
Back up files and directories	Backup Operators,Administrators
Bypass traverse checking	NETWORK SERVICE,LOCAL SERVICE,Everyone,Authenticated Users,Administrators
Change the system time	LOCAL SERVICE,Administrators
Change the time zone	Not Defined
Create a pagefile	Administrators
Create a token object	Not Defined
Create global objects	Not Defined
Create permanent shared objects	Not Defined
Create symbolic links	Not Defined
Debug programs	Administrators
Deny access to this computer from the network	Not Defined
Deny log on as a batch job	Not Defined
Deny log on as a service	Not Defined
Deny log on locally	Not Defined
Deny log on through Remote Desktop Services	Not Defined
Enable computer and user accounts to be trusted for delega...	CONTOSO\Domain Users,Administrators
Force shutdown from a remote system	Administrators
Generate security audits	NETWORK SERVICE,LOCAL SERVICE
Impersonate a client after authentication	Not Defined

Group Policy Objects → Turn of Auditing logs :)

Policy	Policy Setting
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	Success, Failure
Audit logon events	Success
Audit object access	Success, Failure
Audit policy change	Success, Failure
Audit privilege use	Success
Audit process tracking	Success
Audit system events	Success

Subcategory	Audit Events
Audit Credential Validation	Not Configured
Audit Kerberos Authentication Service	Not Configured
Audit Kerberos Service Ticket Operations	Not Configured
Audit Other Account Logon Events	Not Configured

Audit account logon events Properties

Security Policy Setting | Explain

Audit account logon events

☐ Define these policy settings

Audit these attempts:

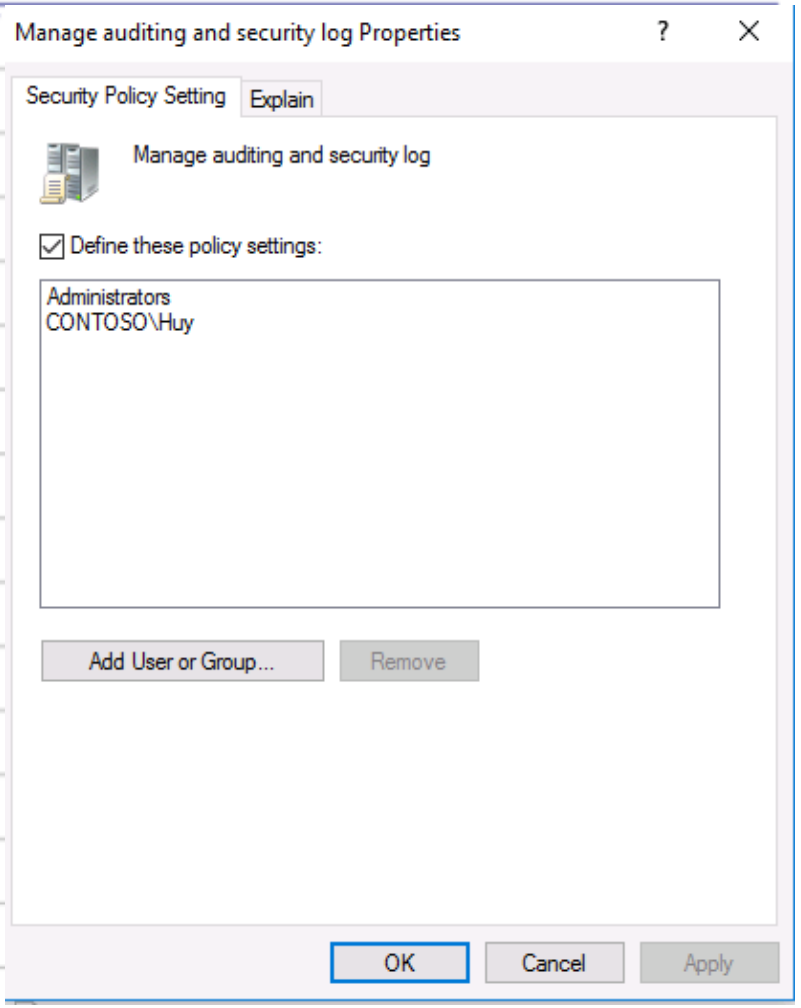
☑ Success

☑ Failure

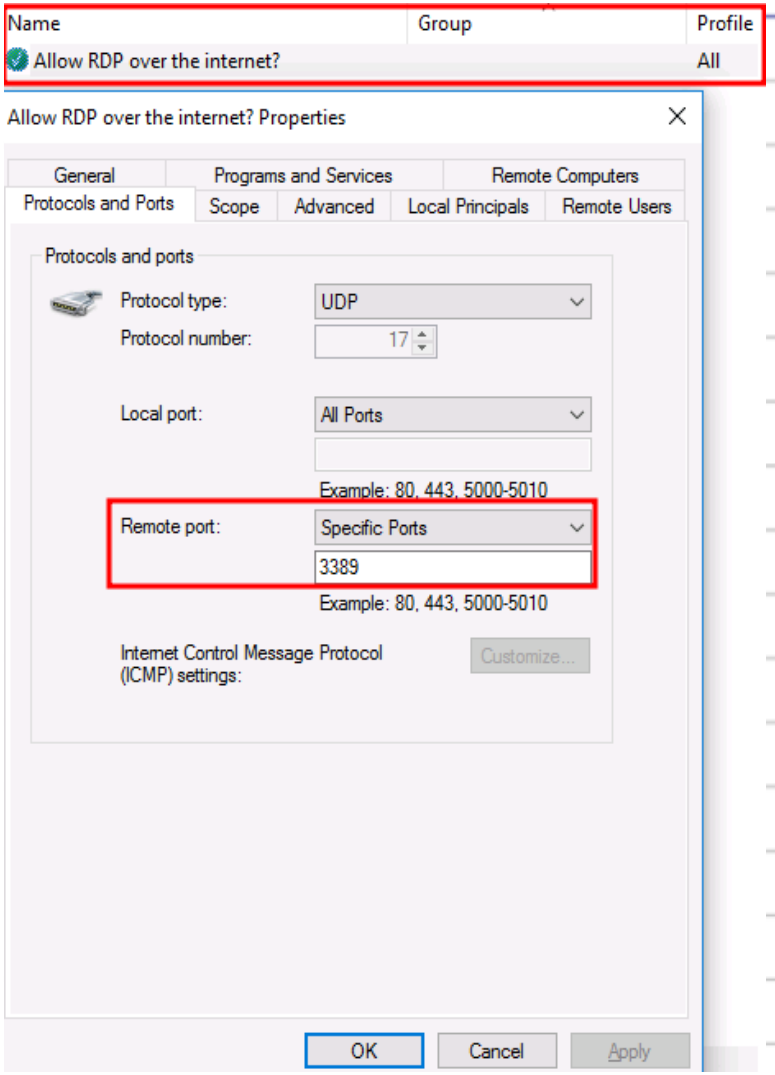
OK | Cancel | Apply

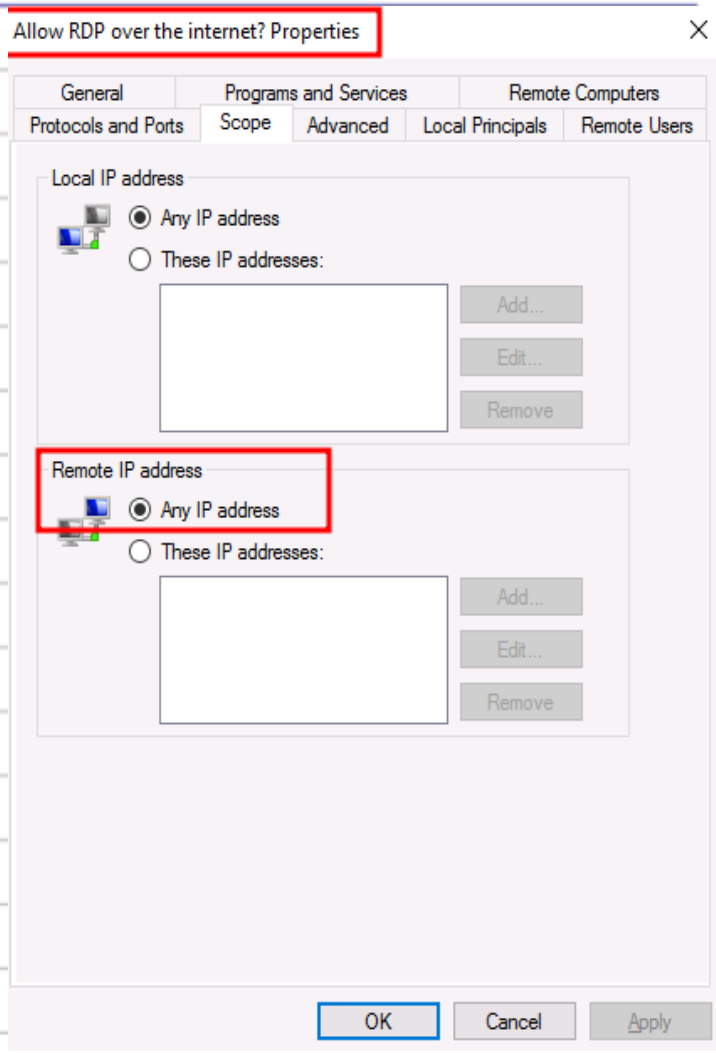
Group Policy Objects → Grant yourself to SE_SECURITY_NAME privilege.

- The Security log is designed for use by the system. However, users can read and clear the Security log if they have been granted the SE_SECURITY_NAME privilege (the "manage auditing and security log" user right).

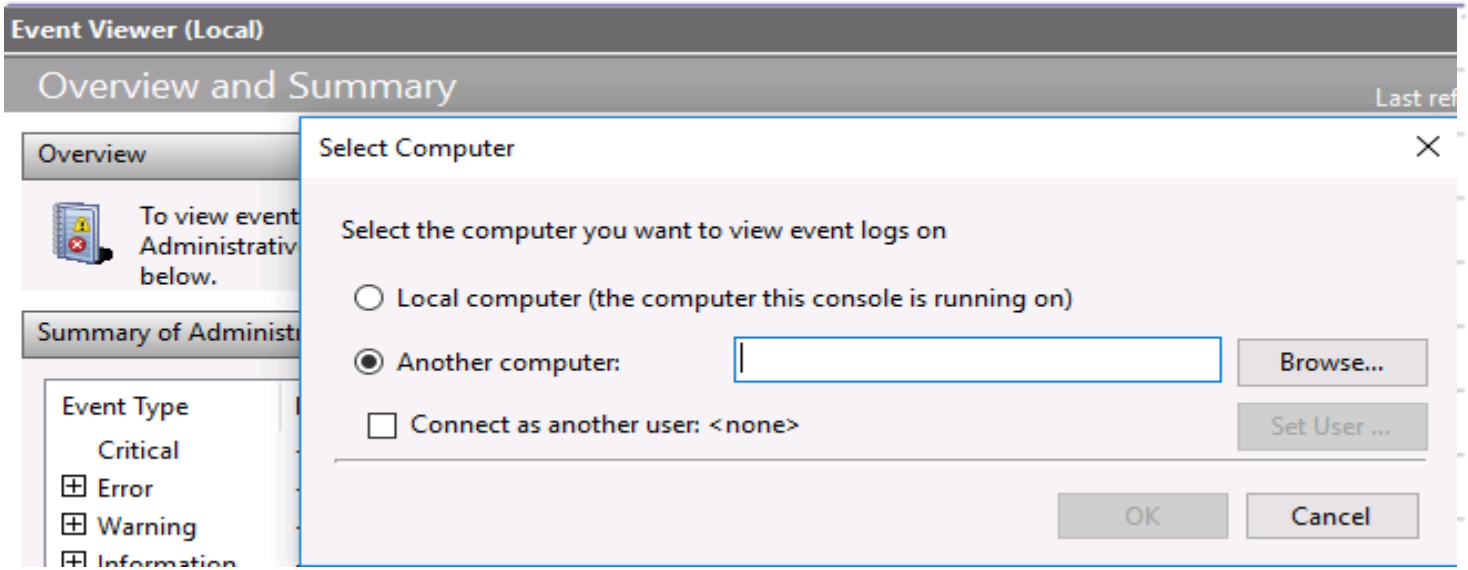


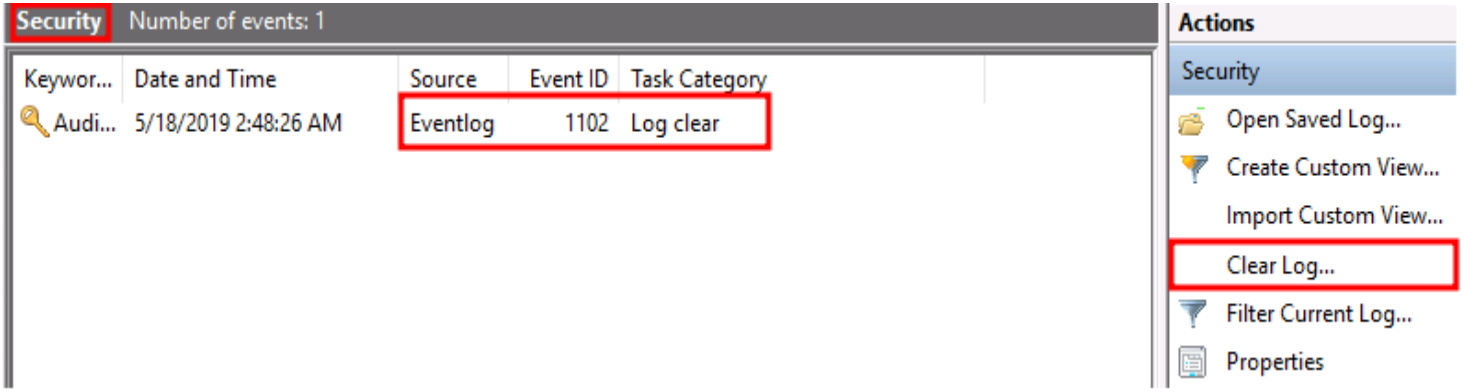
Group Policy Objects → Enable RDP for Remote IP Addresses via WF





Clear logs from DC





Group Policy Objects → Security check

- Check who can modify your GPO’s and kick those out of it, that are not supposed to be able to do that.

LLMNR/NBT-NS Poisoning and Relay

Adversaries can spoof an authoritative source for name resolution on a victim network by responding to LLMNR (UDP 5355)/NBT-NS (UDP 137) traffic as if they know the identity of the requested host, effectively poisoning the service so that the victims will communicate with the adversary controlled system. If the requested host belongs to a resource that requires identification/authentication, the username and NTLMv2 hash will then be sent to the adversary controlled system. The adversary can then collect the hash information sent over the wire through tools that monitor the ports for traffic or through Network Sniffing and crack the hashes offline through Brute Force to obtain the plaintext passwords. In some cases where an adversary has access to a system that is in the authentication path between systems or when automated scans that use credentials attempt to authenticate to an adversary controlled system, the NTLMv2 hashes can be intercepted and relayed to access and execute code against a target system. The relay step can happen in conjunction with poisoning but may also be independent of it. [3][4]

```
[*] [LLMNR]  Poisoned answer sent to 192.168.100.101 for name fielshare
[SMB] NTLMv2-SSP Client    : 192.168.100.101
[SMB] NTLMv2-SSP Username  : AIUK\user2
[SMB] NTLMv2-SSP Hash      : user2::AIUK:1122334455667788:B48729A91322323BD4283DD8C4A4F267:0101000
0000000009F8506E2C4BCD101F5B6DBEBD68390A90000000002000A0073006D006200310032000100140053004500520
560045005200320030003000300038000400160073006D006200310032002E006C006F00630061006C0003002C00530045005
200560045005200320030003000300038002E0073006D006200310032002E006C006F00630061006C00050016007300 6D0062
00310032002E006C006F00630061006C0008003000300000000000000000000000000020000043BD1E3E6C201D3D989B389
B5952622F2CDF272A83DDD0084F2145B78F2119930A0010000000000000000000000000000000000000009001C0063006900
660073002F006600690065006C00730068006100720065000000000000000000
[SMB] Requested Share      : \\FIELSHARE\IPC$
[*] [LLMNR]  Poisoned answer sent to 192.168.100.101 for name fielshare
[*] Skipping previously captured hash for AIUK\user2
[SMB] Requested Share      : \\FIELSHARE\IPC$
```

LLMNR/NBT-NS Poisoning and Relay - **Example**

Download Responder: https://github.com/SpiderLabs/Responder

```
root@kali:~# responder -I eth0

NBT-NS, LLMNR & MDNS Responder 2.3.3.9

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CRTL-C

[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
```

Example;
responder -i 192.168.1.202 -w On -r On -f On

Malicious use of Responder was first publicly documented on August 11, 2017 as being used by APT28, also known as Fancy Bear. The tool was used against hotel visitors to spoof NetBios resources. Victims were coerced into connecting to UDP port 137 and disclosing credentials over SMB to APT28, which the threat actor then used to gain elevated access to the network.

Responder is an LLMNR, NBT-NS, and MDNS poisoner. It will answer to specific NBT-NS (NetBIOS Name Service) queries based on their name suffix.

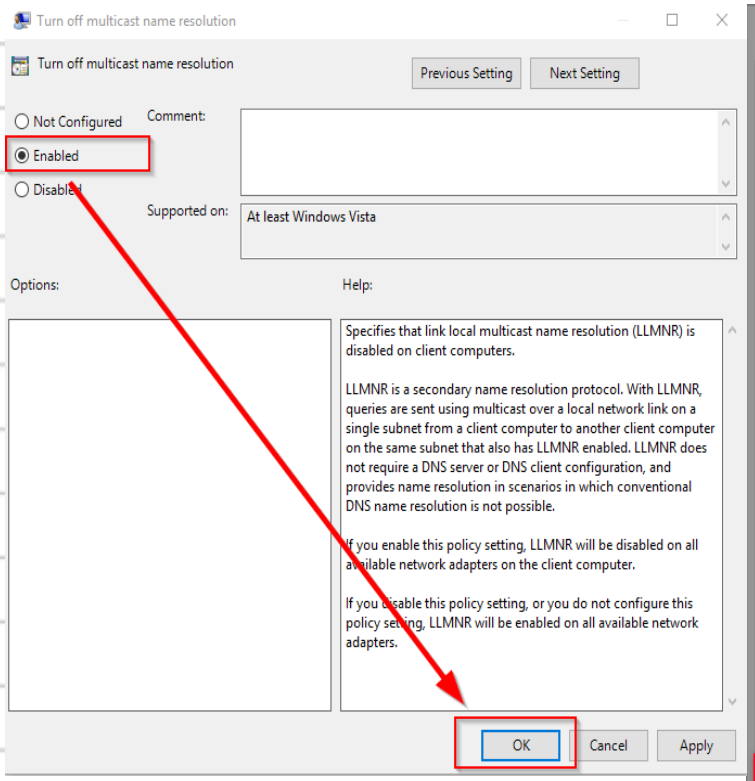
Disclaimer: PtH won't work with NTLMv2 hashes, so it will rely on brute forcing.

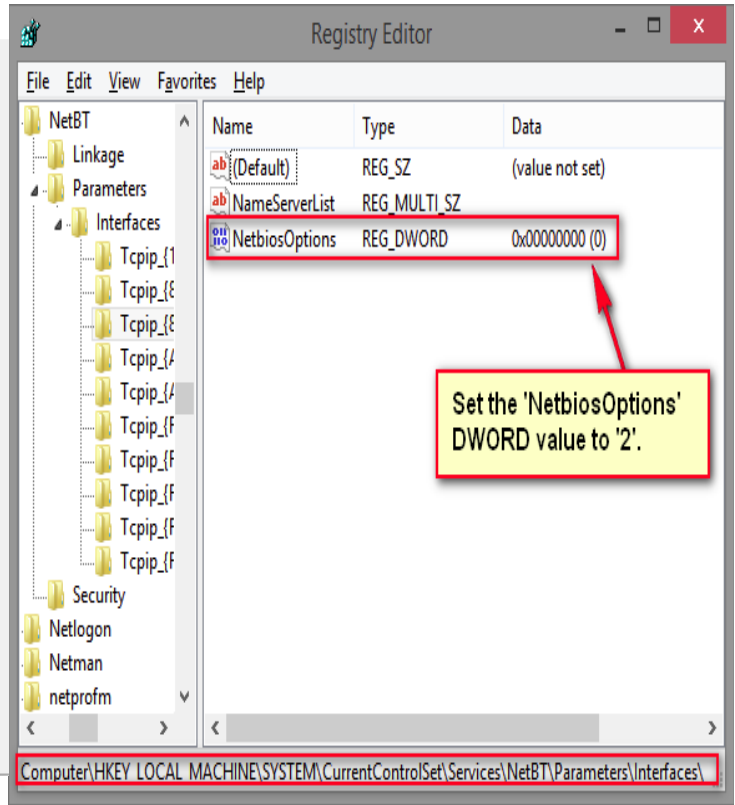
LLMNR/NBT-NS Poisoning and Relay → **How to mitigate**

Click on Enable at ‘’Turn off multicast name resolution’’

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces\

The DWORD value for ‘NetbiosOptions’ will need to be changed to ‘2’



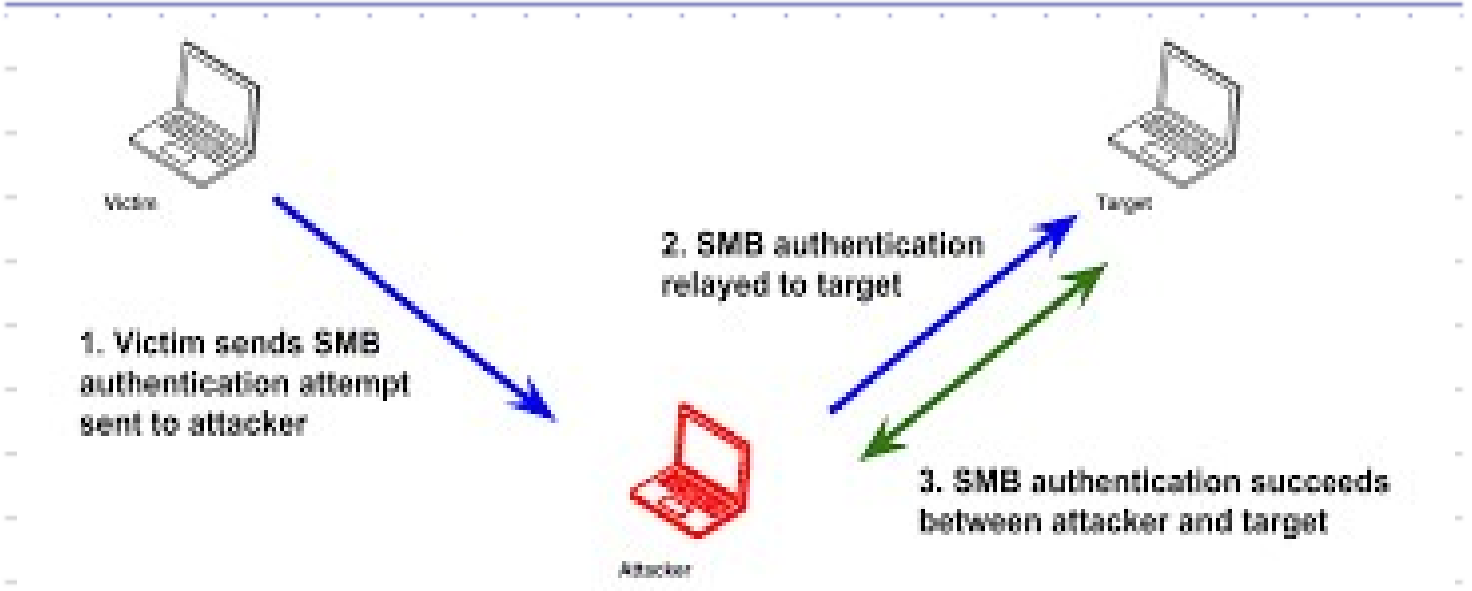


SMB Relay Attack → Info.

- **How it works?**
- SMB Relay attacks allow us to grab authentication attempts and use them to access systems on the network.
- Those hashes can later be used for PtH attacks.
- With this attack you are not bothered with cracking NTLMv2 hashes.
- **Requirements**
- Attacker needs to be on the same local network of the victim

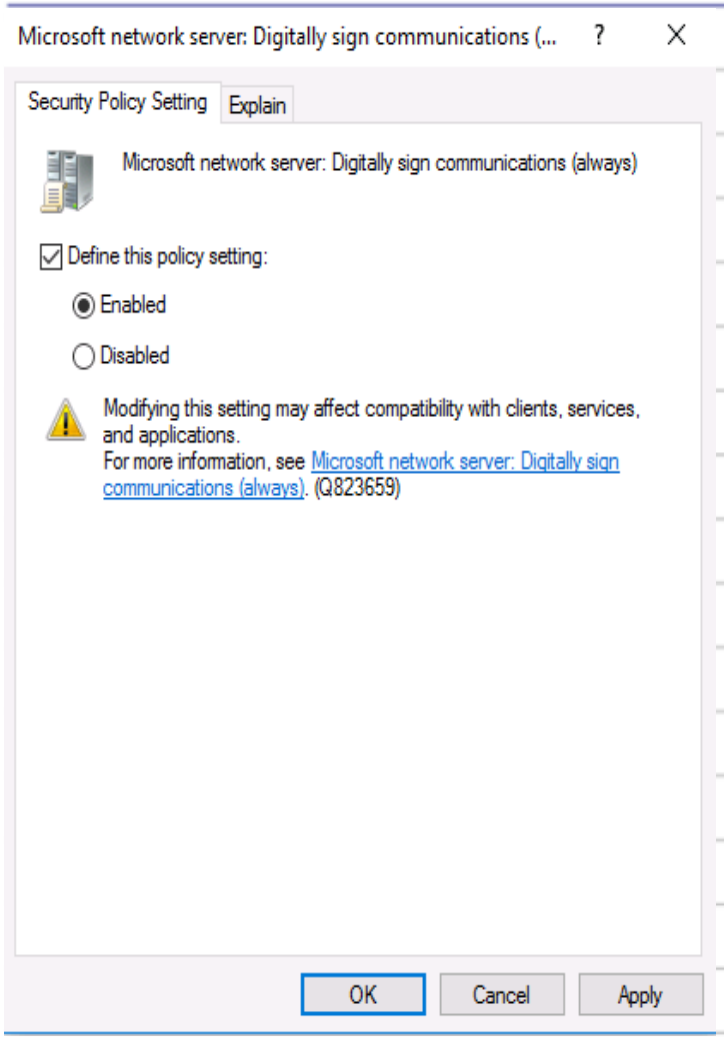
- **Metasploit Module**
- MS08-068 - Microsoft Windows SMB Relay Code Execution
- *"To exploit this, the target system must try to authenticate to this module. The easiest way to force a SMB authentication attempt is by embedding a UNC path (\\SERVER\SHARE) into a web page or email message. When the victim views the web page or email, their system will automatically connect to the server specified in the UNC share (the IP address of the system running this module) and attempt to authenticate" - Rapid7*

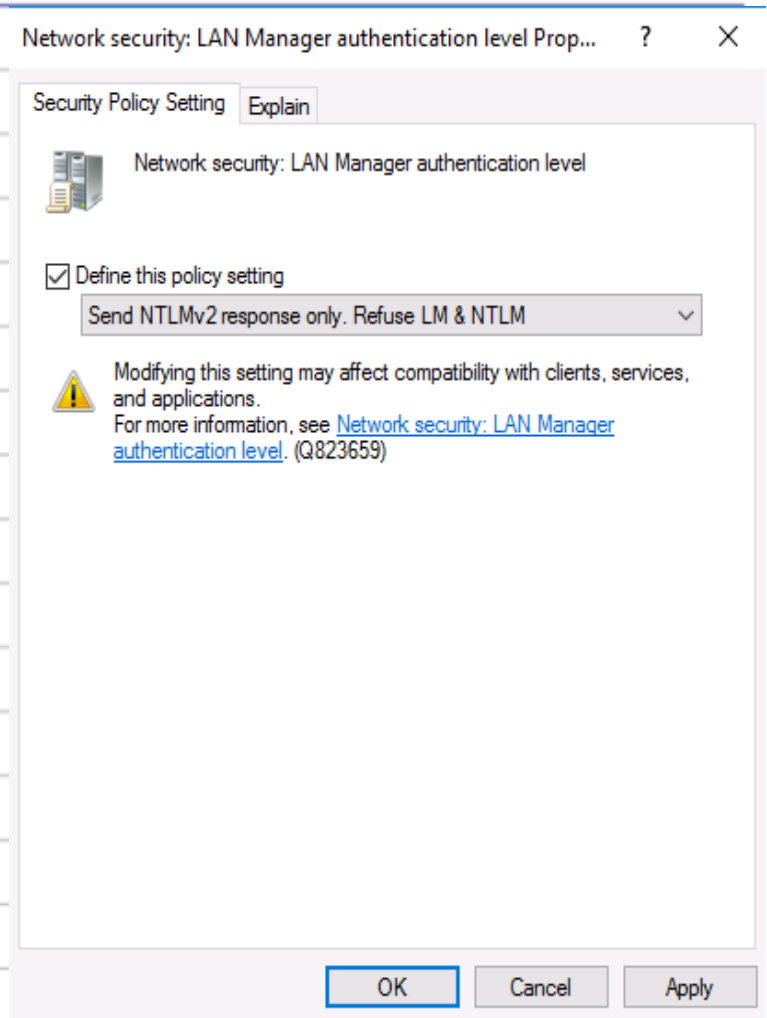
SMB Relay – How it works




```
msf auxiliary(smb) > [*] SMB Captured - 2017-12-11 06:59:05 +0000
NTLMv2 Response Captured from 192.168.1.161:65222 - 192.168.1.161
USER:User DOMAIN:WIN-IH45K7JJ5A7 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:81926ca69b0a3173db24ca5cf165afe8
NT_CLIENT_CHALLENGE:0101000000000000c29361774d72d301603c4c29ea78becd000000000200
0000000000000000000
[*] SMB Captured - 2017-12-11 06:59:05 +0000
NTLMv2 Response Captured from 192.168.1.161:65222 - 192.168.1.161
USER:User DOMAIN:WIN-IH45K7JJ5A7 OS: LM:
LMHASH:Disabled
LM_CLIENT_CHALLENGE:Disabled
NTHASH:0c31043de6e18e5a6ca7c767dec287a5
NT_CLIENT_CHALLENGE:0101000000000000022f563774d72d30181fc68f8c9991f62000000000200
0000000000000000000
```

SMB Relay – How to mitigate?





SMB Relay – Why does it still works?

- Legacy stuff is one of the main reasons why organizations can't enable SMB Signing. So that's why attackers are still able to perform these attacks.
- Enable NTLMv2 and completely refusing NTLM & LM could be possible in most cases, but still. Make sure you test it properly.

Pass-The-Hash → Information

What is PtH?

- Pass-the-Hash (PtH) is a hacking technique that allows an attacker to authenticate to a remote system by using the underlying hash of a user's password rather than having to know the actual password itself.
- Attackers generally use hashes from the current machine to springboard to other machines, grabbing higher privileged credentials as they progress

(Local) SAM Database

```
RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 31d6cfe0d16ae931b73c59d7e0c089c0

RID  : 000001f5 (501)
User : Guest
LM   :
NTLM :

RID  : 000001f7 (503)
User : DefaultAccount
LM   :
NTLM :
```

Attackers can use the stolen credentials of the **built-in local Administrator account** from the compromised workstation to gain access to another workstation with the same password. Which means that they can hop from system to system. workstation with the same password. Which means that they can hop from system to system.

Pass-The-Hash → It still works like always (duh)

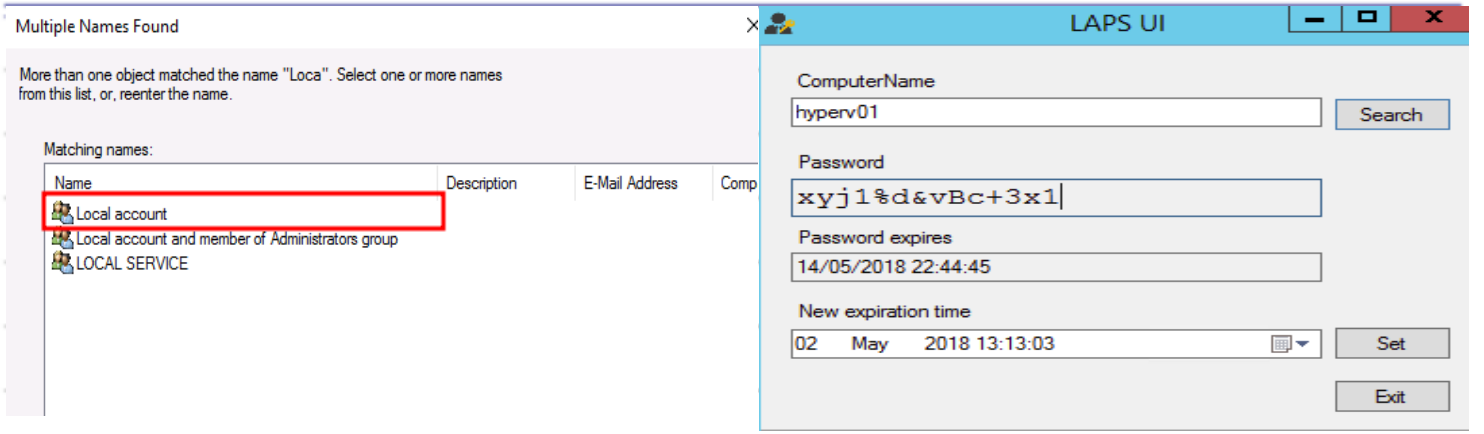
Download <u>Mimikatz</u>: https://github.com/gentilkiwi/mimikatz

- Duo deployment things the built-in Local Administrator account password is (often, like 95%) the same across the entire network for every user.
- This account should only be used in Disaster Recovery scenario's.
- Do not get breached through this attack pls.
- Attackers often will use mimikatz to perform this attack and perform lateral movement inside the network.

- **<u>Assume Breach</u>**
- Attacker compromise a user with local admin privileges on the workstation.
- Local SAM gets dumped of the victim
- The hash of the BUILTIN\Administrator is available.
- Now since most companies do not have proper measures against this attack.
- The attacker can use the stolen hash to authenticate to different users with PsExec for example.

Pass-The-Hash → Countermeasures

Make sure **workstations** have the following setting(s) and use **LAPS**



Deny access to this computer from the network	Local account
Deny log on as a batch job	Local account
Deny log on as a service	Local account
Deny log on locally	Not Defined
Deny log on through Remote Desktop Services	Local account

Workstations Do NOT NEED To Talk To Each other

Kevin Beaumont
@GossiTheDog

Following

Replying to @_HuyKha

Yep. It's really good. And you can manage centrally via Group Policy. At Crabbers we had it enabled company wide, was a huge mitigation for WannaCry etc - nothing could move laterally through company as inbound SMB was blocked (except from trusted jump stations).

Golden Ticket → KRBTGT

What is Golden Ticket?

- A Golden Ticket can be created when the intruder has the right privileges to do so. **Golden Ticket has the same value as Enterprise Admin.**
- Golden Ticket provides the intruder complete access to EVERYTHING in the domain.
- When the attacker controls the KRBTGT account by stealing the hash of it. It allows the attacker to generate Ticket Granting Tickets (TGTs) for any account in the Active Directory domain. And with valid TGTs, the attacker can request from the Ticket Granting Service (TGS) access to any resource/system on its domain.

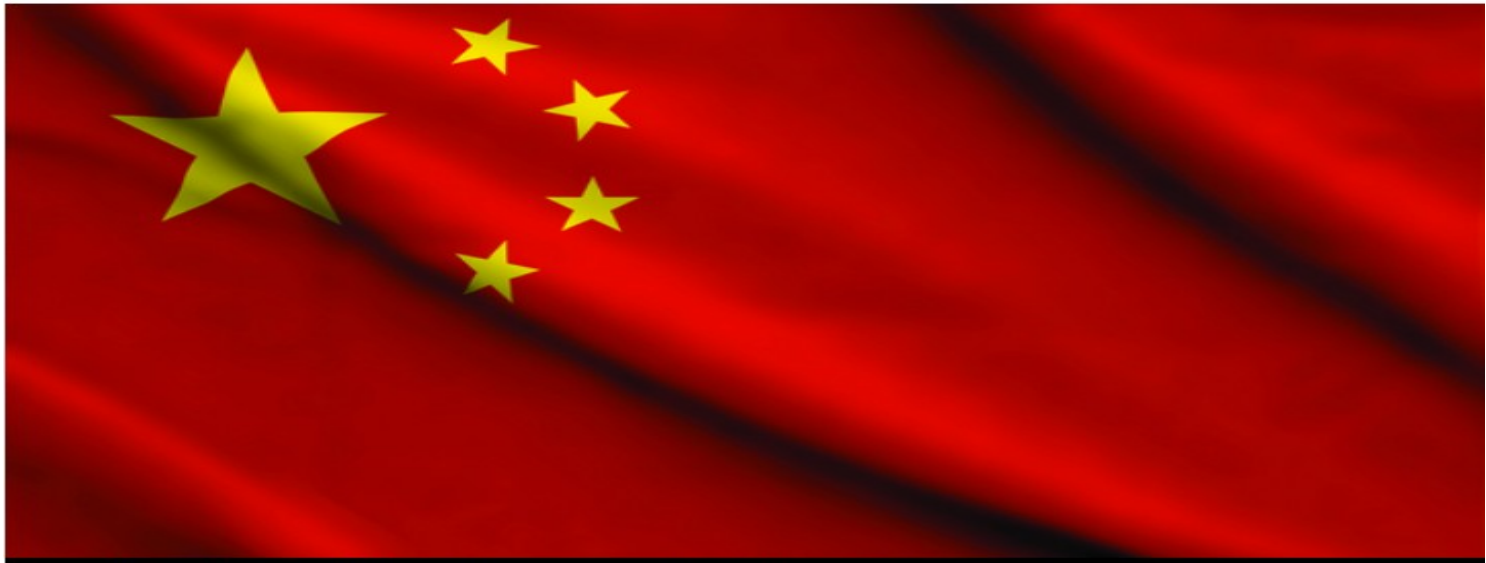
```
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 12/29/2015 11:53:15 PM
Object Security ID   : S-1-5-21-2222611480-1876485831-1594900117-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 95a11f7d93fa3a5a61073662e6bd8468
    ntlm- 0: 95a11f7d93fa3a5a61073662e6bd8468
```

KRBTGT account is the most powerful account in AD. If the hash gets stolen of it, you might have a huge problem.

APT15 has created Golden Tickets

Chinese APT15 hacking group 'infiltrated UK gov contractor and stole military secrets'

Google Cloud used by hackers as part of nation-state attack, claims NCC



Chinese APT15 hacking group 'infiltrated UK gov contractor and stole military secrets'

"APT15 was also observed using Mimikatz to dump credentials and generate Kerberos golden tickets. This allowed the group to persist in the victim's network in the event of remediation actions being undertaken, such as a password reset."

Who can create Golden Tickets?

- **Built-in groups**
- Domain Admins
- Enterprise Admins
- Schema Admins
- Administrators

- **Domain Root**
- Users with the following settings:
- Replication Directory Change
- Replication Directory Change all

- **Good to know**
- Users or Groups with ‘’Full control’’ ‘’WriteDacl’’ ‘’WriteOwner’’ for *This object and all Descedant objects* on the Domain Root. Can perform a DCSync attack as well and grab the hash of the KRBTGT account.
- If you’re running an older version of Exchange on prem.
- **Exchange Windows Permission** has WriteDacl on the Domain Root / Special and can perform DCSync to grab the KRBTGT hash. You might move it away from the group :)

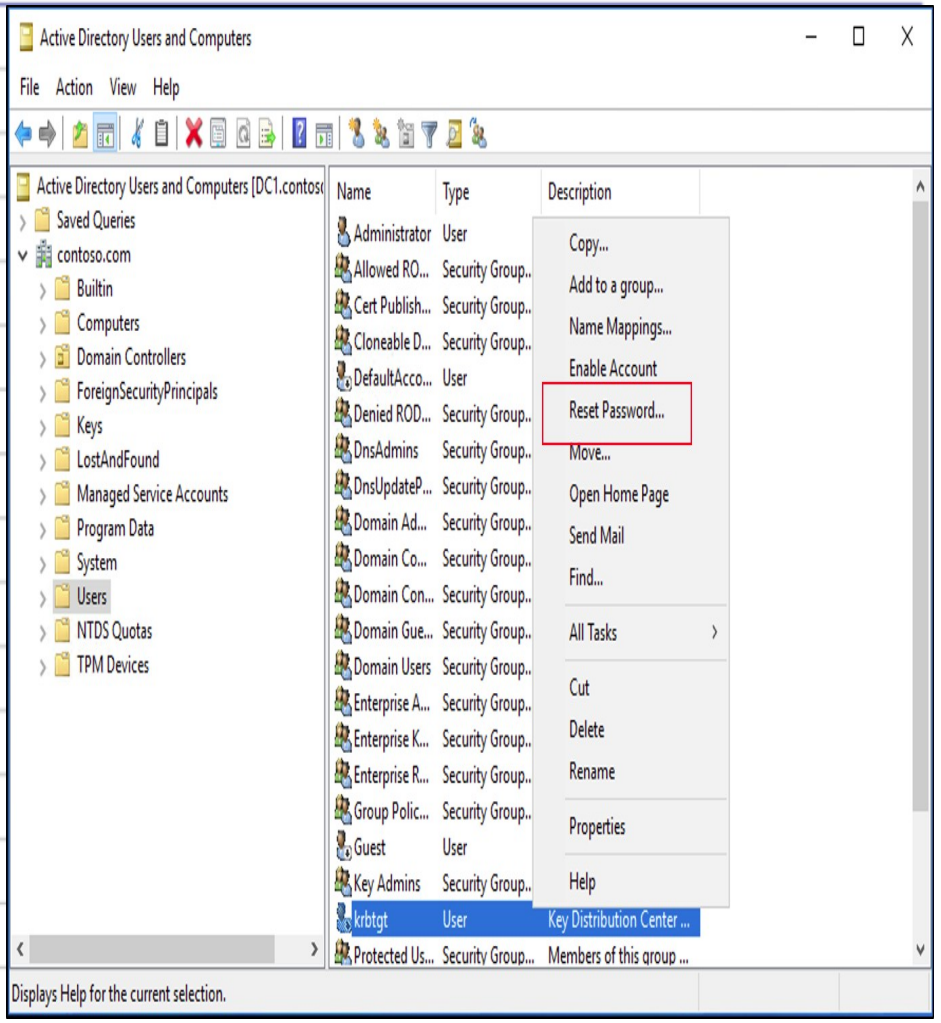
Golden Ticket → Reset 2x KRBTGT

Reset the password of the KRBTGT account twice every half year.

This deletes the current and previous TGT.

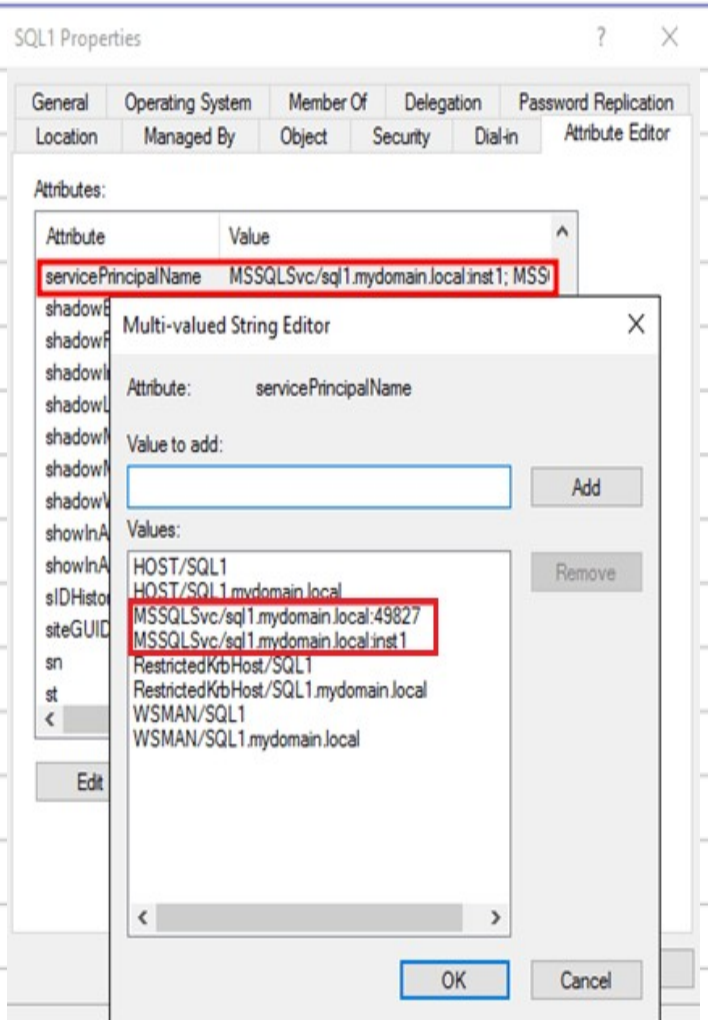
Log on DC → Reset password of krbtgt → Wait 24 hours → Log on DC again → Do the second password reset on the krbtgt account

<u>Warning</u>
<u>I</u>f you reset the password rapidly 2x, some services might break.



Silver Ticket → Service Accounts

- Silver Tickets can be obtained for services that use Kerberos as an authentication mechanism and are used to generate tickets to access that particular resource and the system that hosts the resource (e.g., SharePoint).
- Attacker needs to find services account that has those servicePrincipalName (SPN) values and use Kerberoasting to get the plain-text to create a Silver Ticket to remain persistence for that specific ‘’service’’



Silver Ticket + Unconst. Kerberos = DCSync

DC Computer Accounts **ARE NOT** the same as Service Accounts

- **Assume Breach**

- Attacker compromised the entire domain or a server with Unconstrained Kerberos Delegation. Let's say he did the second one.
- Attacker exploits the MS-RPRN (Spoolsv.exe) to force the DC to connect to the compromised server
- Attacker dives into the memory of the (compromised) server and retrieves the hash of the **DC COMPUTER ACCOUNT.**
- Attacker can now leverage from Silver Ticket to perform a DCSync attack to grab the KRBTGT hash and create Golden Tickets, even if company resetted the password twice. (Enterprise password reset is hard)

kerberos::golden /admin:username /domain:example.local /sid-S-1-5-21-2578996962-4185879466-36960401 /target:dc1.example.local /rc4:<hash> /service:LDAP /ptt

```
[DC] 'krbtgt' will be the user account

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 12/29/2015 11:53:15 PM
Object Security ID   : S-1-5-21-2222611480-1876485831-1594900117-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 95a11f7d93fa3a5a61073662e6bd8468
    ntlm- 0: 95a11f7d93fa3a5a61073662e6bd8468
```

DCSync → Information

- **<u>DCSync</u>**
- DCSync impersonates the behavior of Domain Controller and requests account password data from the targeted Domain Controller.
- DRS protocol is a necessary functionality in AD that will be used in a DCSync attack. Domain Controllers use DRS to replicate configurations, schema's, and all domain conext to other DC's.
- Attackers with the right privileges can perform DCSync to have read access to the AD database.
- **Replication Directory Change**
- **Replication Directory Change All**

- **<u>RED Team Tip</u>**
- Attackers don't need to be DA to compromise the entire domain.
- Look for users/groups with high privilege on the Domain Root through Delegated permission. Compromise those, and start using DCSync, which is a much less noisy attack than dumping the entire NTDS.DIT file.

DCSync → Detection

NOTE: Logs are on the workstation, not the DC.

- **<u>Indicator of Compromise</u>**
- Event ID: 4662
- An operation was performed on this object
- Operation Type: Object Access
- Access Mask: 0x100
- Properties: Control Access
- **{1131f6ad-9c07-11d1-f79f-00c04fc2dcd2}**
- {19195a5b-6da0-11d0-afd3-00c04fd930c9}

DCShadow → Information

Logs are not on the DC, because they are pushed through replication

- **DCShadow**
- DCShadow simulates the behavior of a DC using protocols like RPC to injects his own data.
- After an attacker has obtained DA. And what to stay under the radar. DCShadow can help.
- Mimikatz has a feature that registers the workstation as a DC in Active Directory.
- Now because AD things the workstation is a DC.
- Attacker can make changes on the workstation and push it to the legitimate DC through replication.
- **Requirements:**
- Attackers need Domain or Enterprise Admin to do this.

AD thinks now that W10-ADMIN is the DC, which is not. (Example)

Command Prompt

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\LabAdmin>hostname
W10-ADMIN

C:\Users\LabAdmin>
```

Now the attacker can push changes from the compromised workstation like reset password, set SPN, enable unconstr kerberos, delete OU's, change primay group ID for users. etc. Without getting detected!

DCShadow - Detection

- **<u>Need to know first</u>**
- Knowledge Consistency Checker (KCC) is responsible for handling the replication in the Active Directory forest.
- The DSA, which runs as Ntdsa.dll on each domain controller, provides the interfaces through which directory clients and other directory servers gain access to the directory database.
- nTDSDSA is an object that represents the DSA on the Domain Controller. DCShadow will use the nTDSDA object to create a rogue a DC and push changes through replication via KCC.
- **<u>Detection</u>**
- Event ID: 4662
- An operation was performed on the object
- Access Mask: 0x10
- **Properties: Control Access**
- **{1131f6aa-9c07-11d1-f79f-00c04fc2dcd2}**
- **{19195a5b-6da0-11d0-afd3-00c04fd930c9}**

Detect **DCShadow** & **DCSync** with DCSyncMonitor

-
- DCSyncMonitor is a great addition to every SIEM to detect DCSync & DCShadow.
- *“This tool is an application/service that can be deployed on Domain controllers to alert on Domain Controller Syncronization attempts. When an attempt is detected, the tool will write an event to the Windows Event Log. These events can be correlated in a SIEM. In addition, this tool can take a list of valid DC IP's and, in this configuration, only alert when a DC SYNC attempt comes from a non-DC ip. This tool is meant to provide Blue Teams with a way to combat DC SYNC and DC SHADOW attacks without commercial tools like Microsoft ATA or fancy IDS/IPS.”*
- **Download it here**: DCSyncMonitor

Defense Evasion - Applocker

Microsoft recommended block rules:
https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/microsoft-re

- Microsoft has identified a list of valid applications that an attacker could also potentially use to bypass Windows Defender Application Control.
- You can use Applocker to block execution of those programs and roll a GPO on workstations across the network, but test first.
- List →

- addinprocess.exe
- addinprocess32.exe
- addinutil.exe
- bash.exe
- bginfo.exe[1]
- cdb.exe
- csi.exe
- dbghost.exe
- dbgsvc.exe
- dnx.exe
- fsi.exe
- fsiAnyCpu.exe
- kd.exe
- ntkd.exe
- lxssmanager.dll
- msbuild.exe[2]
- mshta.exe
- ntsd.exe
- rcsi.exe
- system.management.automation.dll
- windbg.exe
- wmic.exe

- Security Settings
  - Account Policies
  - Local Policies
  - Windows Firewall with Advanced Sec
  - Network List Manager Policies
  - Public Key Policies
  - Software Restriction Policies
  - Application Control Policies
    - AppLocker
      - Executable Rules
      - Windows Installer Rules
      - Script Rules
      - Packaged app Rules
  - IP Security Policies on Local Compute
  - Advanced Audit Policy Configuration

Action	User	Name	Condition	Exceptions
Deny	Everyone	Block addinprocess.exe	Path	
Deny	Everyone	Block addinprocess32.exe	Path	
Deny	Everyone	Block addinutil.exe	Path	
Deny	Everyone	Block bash.exe	Path	
Deny	Everyone	Block bginfo.exe	Path	
Deny	Everyone	Block cdb.exe	Path	
Deny	Everyone	Block csi.exe	Path	
Deny	Everyone	Block dbghost.exe	Path	
Deny	Everyone	Block dbgvsc.exe	Path	
Deny	Everyone	Block dnx.exe	Path	
Deny	Everyone	Block fsi.exe	Path	
Deny	Everyone	Block fsiAnyCpu.exe	Path	
Deny	Everyone	Block kd.exe	Path	
Deny	Everyone	Block ntkd.exe	Path	
Deny	Everyone	Block lxssmanager.dll	Path	
Deny	Everyone	Block msbuild.exe	Path	
Deny	Everyone	Block mshta.exe	Path	
Deny	Everyone	Block ntds.exe	Path	
Deny	Everyone	Block rcsi.exe	Path	
Deny	Everyone	Block system.management.automation...	Path	
Deny	Everyone	Block windbg.exe	Path	
Deny	Everyone	Block wmic.exe	Path	

Defense Evasion – **Application Whitelisting Bypass**

- **APT32** has used mshta.exe for code execution.
- **APT19** used Regsvr32 to bypass application whitelisting techniques.
- Deep Panda has used regsvr32.exe to execute a server variant of Derusbi in victim networks
- **APT28** executed CHOPSTICK by using rundll32 commands such as rundll32.exe "C:\Windows\twain_64.dll". APT28 also executed a .dll for a first stage dropper using rundll32.exe. An APT28 loader Trojan saved a batch script that uses rundll32 to execute a DLL payload.

See the next slide for a config that's made by Daniel Streefkerk aka @dstreefkerk to mitigate those attacks with the <u>Windows Firewall</u>. I have added a few additional things to it.

Make sure if you are going to follow this. Block **System32** & **SysWOW64** path.

Defense Evasion – **Windows Firewall** (Outbound Connection)

Configuration can be found here: https://pastebin.com/tDtL40Gi

Windows Firewall with Advanced Security

File Action View Help

- Windows Firewall with Advanc
  - Inbound Rules
  - Outbound Rules
  - Connection Security Rules
  - Monitoring

Outbound Rules

Name	Profile	Enabled	Action	Override	Program	Local Address
Block Internet Access - cmd.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\cmd.exe	Any
Block Internet Access - cmd.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\cmd.exe	Any
Block Internet Access - conhost.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\conhost.exe	Any
Block Internet Access - cscript.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\cscript.exe	Any
Block Internet Access - cscript.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\cscript.exe	Any
Block Internet Access - dfsvc.exe - 2.0.50727	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\dfsvc.exe	Any
Block Internet Access - dfsvc.exe - 2.0.50727 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v2.0.50727\dfsvc.exe	Any
Block Internet Access - dfsvc.exe - 4.0.30319	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\dfsvc.exe	Any
Block Internet Access - dfsvc.exe - 4.0.30319 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v4.0.30319\dfsvc.exe	Any
Block Internet Access - ieexec.exe - 2.0.50727	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\IEExec.exe	Any
Block Internet Access - ieexec.exe - 2.0.50727 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v2.0.50727\IEExec.exe	Any
Block Internet Access - InstallUtil.exe - 2.0.50727	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\InstallUtil.exe	Any
Block Internet Access - InstallUtil.exe - 2.0.50727 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v2.0.50727\InstallUtil.exe	Any
Block Internet Access - InstallUtil.exe - 4.0.30319	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe	Any
Block Internet Access - InstallUtil.exe - 4.0.30319 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe	Any
Block Internet Access - MSBuild.exe - 2.0.50727	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\MSBuild.exe	Any
Block Internet Access - MSBuild.exe - 2.0.50727 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v2.0.50727\MSBuild.exe	Any
Block Internet Access - MSBuild.exe - 3.5	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v3.5\MSBuild.exe	Any
Block Internet Access - MSBuild.exe - 3.5 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v3.5\MSBuild.exe	Any
Block Internet Access - MSBuild.exe - 4.0.30319	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\MSBuild.exe	Any
Block Internet Access - MSBuild.exe - 4.0.30319 (x64)	All	Yes	Block	No	%SystemRoot%\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe	Any
Block Internet Access - msdt.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\msdt.exe	Any
Block Internet Access - msdt.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\msdt.exe	Any
Block Internet Access - mshta.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\mshta.exe	Any
Block Internet Access - mshta.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\mshta.exe	Any
Block Internet Access - powershell.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	Any
Block Internet Access - powershell.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe	Any
Block Internet Access - powershell_ise.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\WindowsPowerShell\v1.0\powershell_ise.exe	Any
Block Internet Access - powershell_ise.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell_ise.exe	Any
Block Internet Access - regsvr32.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\regsvr32.exe	Any
Block Internet Access - regsvr32.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\regsvr32.exe	Any
Block Internet Access - rundll32.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\rundll32.exe	Any
Block Internet Access - rundll32.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\rundll32.exe	Any
Block Internet Access - wscript.exe	All	Yes	Block	No	%SystemRoot%\SysWOW64\wscript.exe	Any
Block Internet Access - wscript.exe (x64)	All	Yes	Block	No	%SystemRoot%\System32\wscript.exe	Any

Logging Made Easy

Download it here: https://github.com/ukncsc/lme/

- *“Logging Made Easy is a self-install tutorial for small organisations to gain a basic level of centralised security logging for Windows clients and provide functionality to detect attacks. It's the coming together of multiple free and open-source software, where LME helps the reader integrate them together to produce an end-to-end logging capability. We also provide some pre-made configuration files and scripts, although there is the option to do it on your own.”*
- *A great addition to your security posture if you don’t have a SIEM.*

National Cyber Security Centre

a part of GCHQ

Credits to

- Sean Metcalf – ADSecurity
- ATT&CK framework
- Daniel Streefkerk - https://daniel.streefkerkonline.com/
- SpectreOps
- Authors of Mimikatz
- Insider Threat Blog
- Microsoft.com
- Github resources in the slide(s)
- NCSC
- SANS
- Articles

Thank you!!