Cracking Cisco ASA SHA-512 Hashes with Hashcat

Troubleshooting Empire and PoshC2_Python HTTPS Connections

FEB 03

Low Privilege Active Directory Enumeration from a non-Domain Joined Host

By matt in Active Directory, PowerShell, Tools

Scenario

You have recovered Domain User credentials for a domain but have no privileged or interactive access to any targets i.e. no Domain Admin account or any account that is capable of establishing an RDP session.

Introduction

On a recent engagement I was performing an internal assessment against several untrusted Windows domains. Using Kerberos Domain Username Enumeration and subsequently performing SMB password guessing it was possible to achieve access to a number of Domain accounts.

However, it transpired that none of the identified credential sets were privileged e.g. they were not Domain Admin and additionally, none of the accounts were members of the “Remote Desktop Users” group. As a result, no interactive access to any of the target hosts was possible.

A number of different techniques exist to query Active Directory using low privileged accounts (i.e. a domain user) from our non-domain joined pentest laptop and I will discuss a few options for doing this in this post.

The ultimate goal of this enumeration is to:

- Enumerate all Domain accounts
- Enumerate privileged accounts to target i.e. Domain Admins or members of the Remote Desktop Users group
- Enumerate the Domain’s password policy
- Enumerate further avenues of attack

Once this enumeration is complete accounts can be subject to further password guessing attempts.

The domain user credentials used in the following examples are username = `ops`, password = `Pa55word`

windapsearch

The first of the tools I will discuss is windapsearch:

As the tool's author describes: "windapsearch is a Python script to help enumerate users, groups and computers from a Windows domain through LDAP queries"

Using our prerequisite/previously guessed domain user account the following syntax can be used to query the remote domain for all users within the domain:

```
windapsearch --dc-ip [IP_ADDRESS] -u [DOMAIN]\\USERNAME -p [PASSWORD] -U
```

The following figure shows the tool enumerating all users in the domain (-U switch):

NOTE: Output has been cleaned up a little with grep & cut

```
windapsearch --dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word -U | grep cn: | cut -d " " -f 2
```

```
root@cyclops:/# windapsearch --dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word -U | grep cn: | cut -d " " -f 2
matt
Guest
krbtgt
test123
bob
alice
hacker123
DOMAIN2$
JACK.KELLY
THOMAS.GLOVER
JAMES.LEES
JOSHUA.GIBSON
MATTHEW.REES
RYAN.LANE
JOSEPH.CURTIS
SAMUEL.CROSS
LIAM.FOWLER
JORDAN.FROST
LUKE.FISHER
CONNOR.BURGESS
BENJAMIN.ADAMS
HARRY.WYATT
WILLIAM.MATTHEWS
```

Using the `--da` switch we can also enumerate Domain Admins:

`windapsearch -dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word --da | grep cn: | cut -d " " -f 2`

```
root@cyclops:/# windapsearch --dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word --da | grep cn: | cut -d " " -f 2
matt
hacker123
JACK.KELLY
mradmin
sqladmin
admin1
```

Using the `-m` switch we can enumerate members of the “Remote Desktop Users” group:

```
windapsearch --dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word -m "Remote Desktop Users" | grep CN=
```

```
root@cyclops:/# windapsearch --dc-ip 192.168.5.1 -u mydomain\\ops -p Pa55word -m "Remote Desktop Users" | grep CN=
[+]     Using DN: CN=Remote Desktop Users,CN=Builtin,DC=MYDOMAIN,DC=TEST
CN=JACK.KELLY,CN=Users,DC=MYDOMAIN,DC=TEST
CN=matt,CN=Users,DC=MYDOMAIN,DC=TEST
```

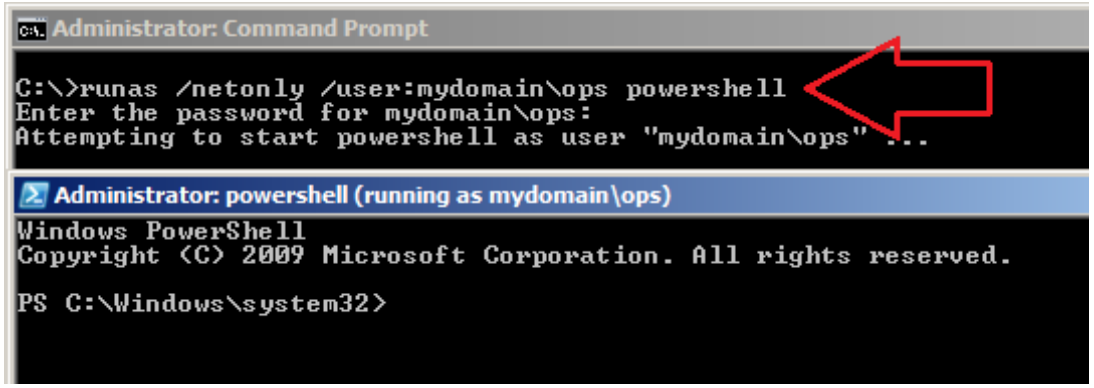
PowerView

The excellent PowerView from harmj0y probably offers us the best options for AD enumeration in our Domain User / non-Domain joined context.

PowerView is thoroughly and eloquently discussed in harmj0y's multiple blog posts (see references), but I'll just discuss a couple of options that can be useful in our scenario.

Initially, we establish a PowerShell session on our non-domain joined Windows host using `runas` and `/netonly` i.e. credentials are specified for remote access only:

`runas /netonly /user:mydomain\op powershell` (we are subsequently prompted for the password):



Note: I've already installed PowerSploit (which provides PowerView) in the following path:

```
C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev
```

Once we have imported PowerSploit via:

```
Import-module .\PowerSploit.psd1
```

We can query the domain for all domain users:

```
Get-DomainUser -Domain mydomain.test -DomainController 192.168.5.1 | findstr samaccountname
```

```
powershell (running as mydomain\ops)
PS C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev> import-module .\PowerSploit.psd1
PS C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev> Get-DomainUser -Domain mydomain.test -DomainContr
oller 192.168.5.1 | findstr samaccountname
samaccountname          : matt
samaccountname                : Guest
samaccountname                : krbtgt
samaccountname                : test123
samaccountname          : bob
samaccountname          : alice
samaccountname          : hacker123
samaccountname          : JACK.KELLY
samaccountname          : THOMAS.GLOVER
samaccountname          : JAMES.LEES
samaccountname          : JOSHUA.GIBSON
samaccountname          : MATTHEW.REES
samaccountname          : RYAN.LANE
samaccountname          : JOSEPH.CURTIS
samaccountname          : SAMUEL.CROSS
samaccountname          : LIAM.FOWLER
samaccountname          : JORDAN.FROST
samaccountname          : LUKE.FISHER
samaccountname          : CONNOR.BURGESS
samaccountname          : BENJAMIN.ADAMS
samaccountname          : HARRY.WYATT
samaccountname          : WILLIAM.MATTHEWS
```

We can also query the domain for a list of Domain Admins:

```
Get-DomainGroupMember -identity "Domain Admins" -Domain mydomain.test -
DomainController 192.168.5.1 | findstr MemberName
```

```
powershell (running as mydomain\ops)
PS C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev> Get-DomainGroupMember -identity "Domain Admins" -
Domain mydomain.test -DomainController 192.168.5.1 | findstr MemberName
MemberName              : admin1
MemberName              : sqladmin
MemberName              : mradmin
MemberName              : JACK.KELLY
MemberName              : hacker123
MemberName              : matt
```

Next we query the domain for the members of the “Remote Desktop Users” group:

```
Get-DomainGroupMember -identity "Remote Desktop Users" -Domain mydomain.test -DomainController 192.168.5.1 | findstr MemberName
```

```
powershell (running as mydomain\ops)
PS C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev> Get-DomainGroupMember -identity "Remote Desktop Users" -Domain mydomain.test -DomainController 192.168.5.1 | findstr MemberName
MemberName              : JACK.KELLY
MemberName              : matt
```

We can also query AD for a list of all available shares that our current user context is able to access:

```
Find-DomainShare -CheckShareAccess -Domain mydomain.test -DomainController 192.168.5.1
```

```
powershell (running as mydomain\ops)
PS C:\Windows\System32\WindowsPowerShell\v1.0\Modules\PowerSploit-dev> Find-DomainShare -CheckShareAccess -Domain mydomain.test -DomainController 192.168.5.1

Name         Type Remark                  ComputerName
----         ---- ------                  ------------
NETLOGON        0 Logon server share       W2K8-DC.MYDOMAIN.TEST
Open_Share      0 Passwords & Secret Stuff W2K8-DC.MYDOMAIN.TEST
SYSVOL          0 Logon server share       W2K8-DC.MYDOMAIN.TEST
SECRET_SHARE    0                          WIN8-MYDOMAIN.MYDOMAIN.TEST
```

Microsoft Remote Server Administration Tools (RSAT)

Microsoft RSAT is designed to allow administrators to manage Windows Servers from a remote computer. RSAT provides another option for us to enumerate domains from our low privileged, non-connected domain context:

Initially, RSAT proves useful for the enumeration of the remote Window Domain's password policy. Again, we do this from a `runas`, `/netonly` initiated PowerShell session (see PowerView above for details):

```
Get-ADDefaultDomainPasswordPolicy -Server 192.1685.5.1
```

powershell (running as mydomain\ops)

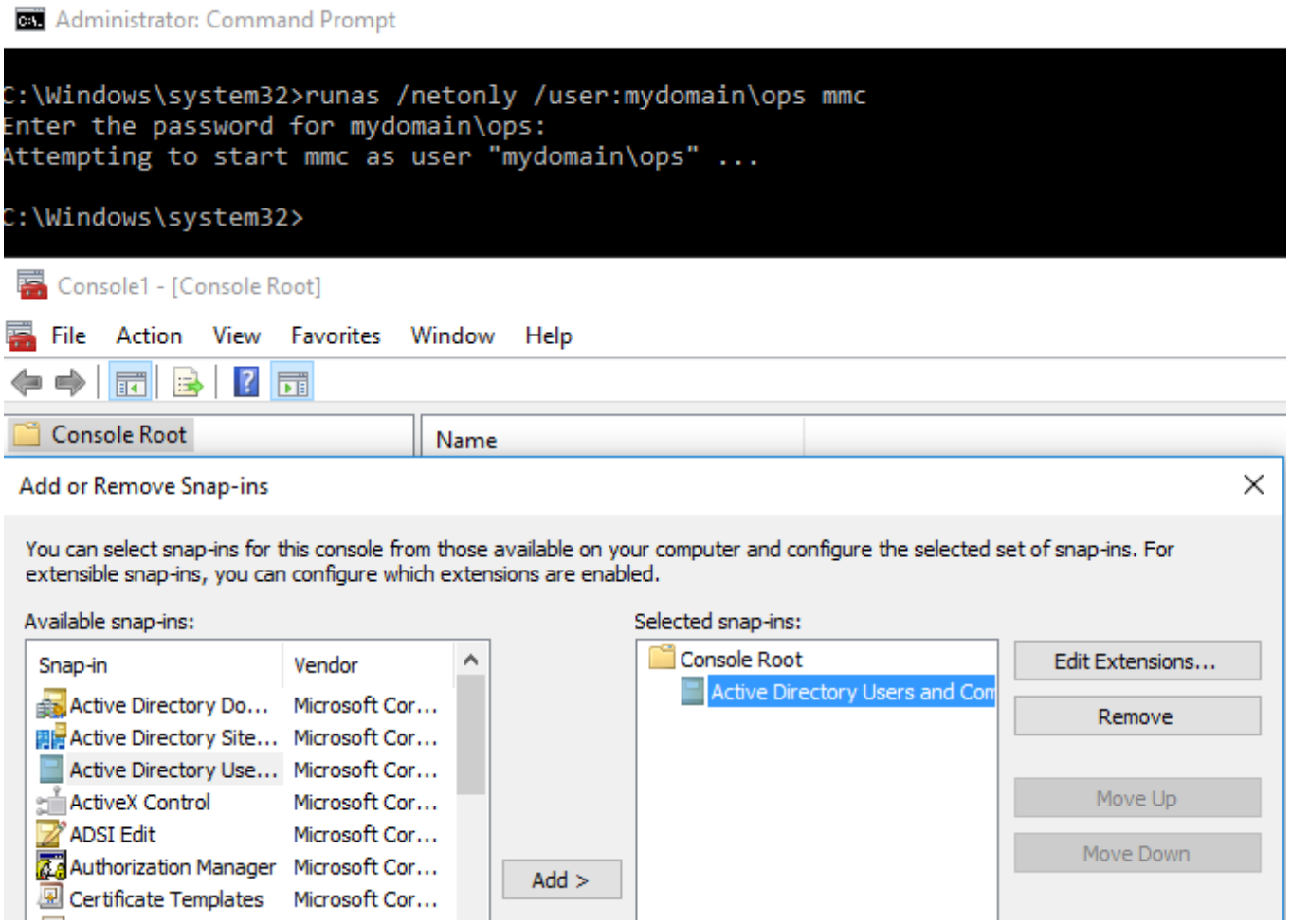
```
PS C:\> Get-ADDefaultDomainPasswordPolicy -Server 192.168.5.1


ComplexityEnabled           : False
DistinguishedName           : DC=MYDOMAIN,DC=TEST
LockoutDuration             : 00:30:00
LockoutObservationWindow    : 00:30:00
LockoutThreshold            : 0
MaxPasswordAge              : 00:00:00
MinPasswordAge              : 1.00:00:00
MinPasswordLength           : 6
objectClass                 : {domainDNS}
objectGuid                  : 7d454d80-f0f0-44c6-9a7f-5ff9db6eac0c
PasswordHistoryCount        : 23
ReversibleEncryptionEnabled : False
```

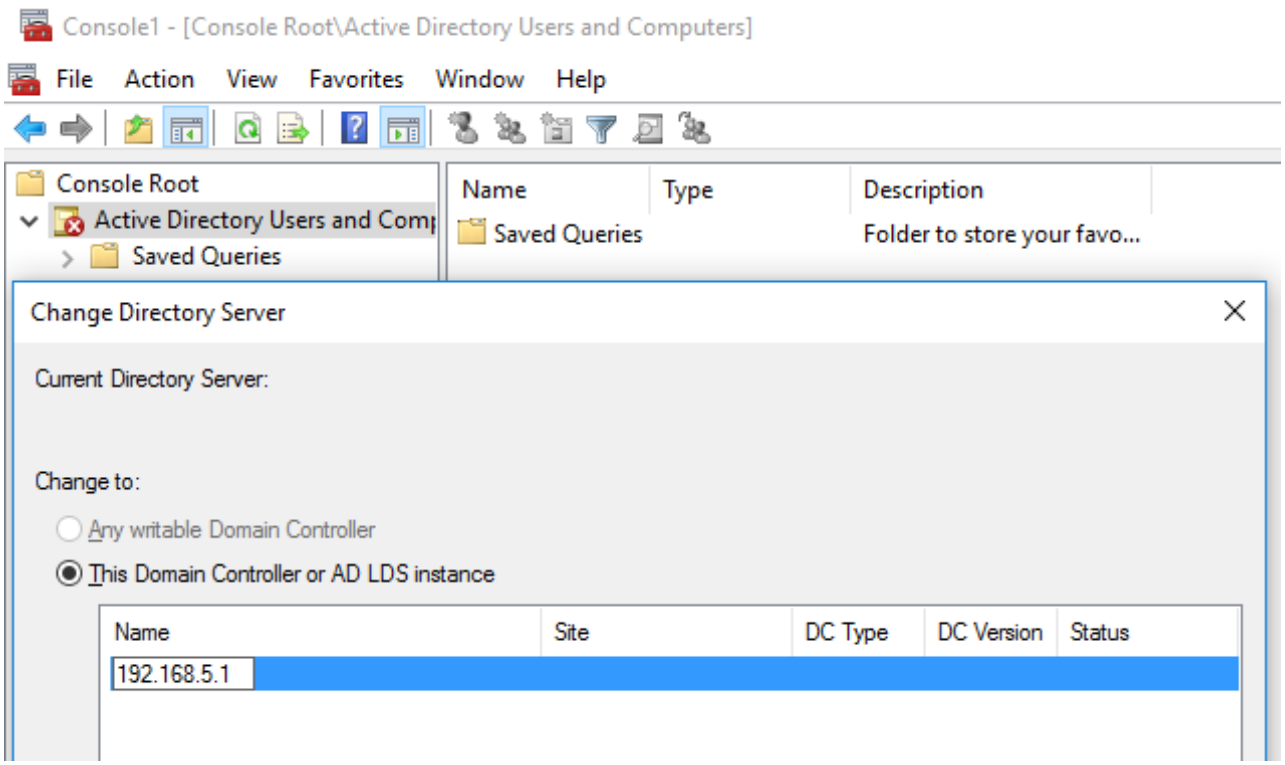
We are also able to utilise RSAT from a GUI perspective, again this is initiated via `runas`:

```
runas /netonly /user:mydomain\ops mmc
```

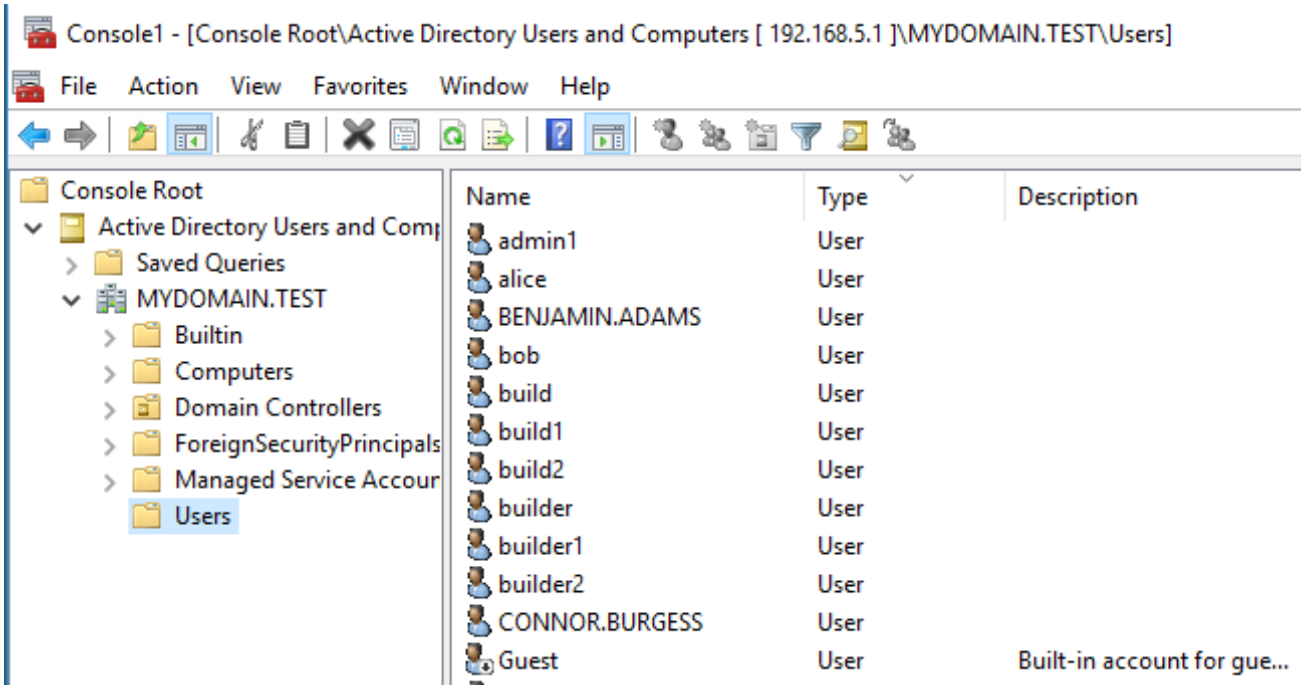
Next we add "Active Directory Users and Computers" via the new mmc console:



Changing the Domain Controller instance to our target:



We are then able to gain a graphical view of the Domain's user community:



The whole purpose behind this Domain enumeration is to provide us with further and more privileged accounts to target from a password guessing perspective. The retrieval of the Domain's password policy obviously also complements this exercise.

References:

windapearch
PowerView
PowerSploit
PowerView Cheat Sheet

1 comment

Richard on *February 11, 2018 at 08:33* # Reply

Thanks great post! Something that red teams/ pentesters miss when looking at domain password policy is determining if fine grained password policies are also being used for specific accounts. This can be a great way for blue teams to catch someone password guessing since the lockout thresholds for specific accounts can be set lower than that of the domain policy. It also serves as a way to set stronger policies for higher privileged accounts.

Leave a Reply

Your email address will not be published.

Your message

Name | Email | Website (optional)

Notify me of follow-up comments by email.

Notify me of new posts by email.

Submit Comment

Search

Follow @attackdebris

RECENT POSTS

- Troubleshooting Empire and PoshC2_Python HTTPS Connections
- Low Privilege Active Directory Enumeration from a non-Domain Joined Host
- Cracking Cisco ASA SHA-512 Hashes with Hashcat
- Kerberos Username Enumeration – Top 500 Common Usernames
- Auto-sslscan (Automatic SSL Scanning)

USEFUL LINKS

Hack, Whack and Smack
Phillips321
rewt dance
Scriptmonkey

ARCHIVES

December 2018

February 2018

January 2018

June 2017

November 2016

October 2014

February 2014

October 2013

August 2013

May 2013

September 2012

CATEGORIES