

# 1. ¿Se encuentran empaquetados los ejecutable? ¿Cómo lo puede determinar? (en caso positivo, desempaquetelos)

sample\_vg655\_25th.exe  
no está empaquetado

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4890]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\marce>"C:\Users\marce\Downloads\upx-4.2.4-win64\upx.exe" -t "C:\Users\marce\Downloads\examples\sample_vg655_25th.exe"
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2024
UPX 4.2.4 Markus Oberhumer, Laszlo Molnar & John Reiser May 9th 2024

upx: C:\Users\marce\Downloads\examples\sample_vg655_25th.exe: NotPackedException: not packed by UPX

Tested 0 files.

C:\Users\marce>
```


```
Users > marce > Desktop > ejecutable.py > ...
1 import magic
2
3 file_path = "C:\\Users\\marce\\Downloads\\examples\\sample_vg655_25th.exe"
4 file_type = magic.from_file(file_path)
5 print(f"Tipo de archivo: {file_type}")
6
```

```
File Edit Selection View Go Run Terminal Help
t-with-langgraph (1).ipynb day-3-function-calling-with-the-gemini-api.ipynb day-2-embeddings-and-similarity-scores (4).ipynb day-4-fine-tuning-a-custom-model (1).ipynb ejecutable.py x dia2. ...
C:\Users\marce\Desktop>ejecutable.py
1 import magic
2
3 file_path = "C:\\Users\\marce\\Downloads\\examples\\sample_vg655_25th.exe"
4 file_type = magic.from_file(file_path)
5 print(f"Tipo de archivo: {file_type}")
6

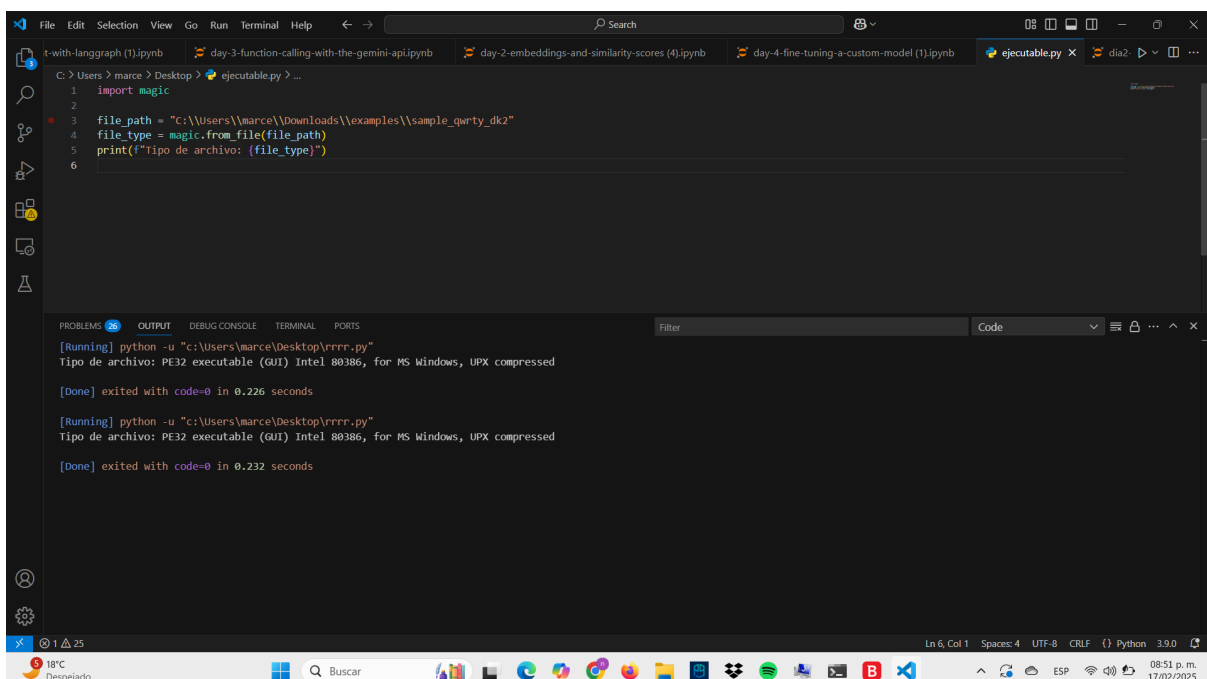
[Running] python -u "c:\Users\marce\Desktop\rrrr.py"
Tipo de archivo: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
[Done] exited with code=0 in 0.226 seconds

[Running] python -u "c:\Users\marce\Desktop\rrrr.py"
Tipo de archivo: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
[Done] exited with code=0 in 0.232 seconds

[Running] python -u "c:\Users\marce\Desktop\ejecutable.py"
Tipo de archivo: PE32 executable (GUI) Intel 80386, for MS Windows
[Done] exited with code=0 in 0.2 seconds
```

 sample\_qwrty\_dk2

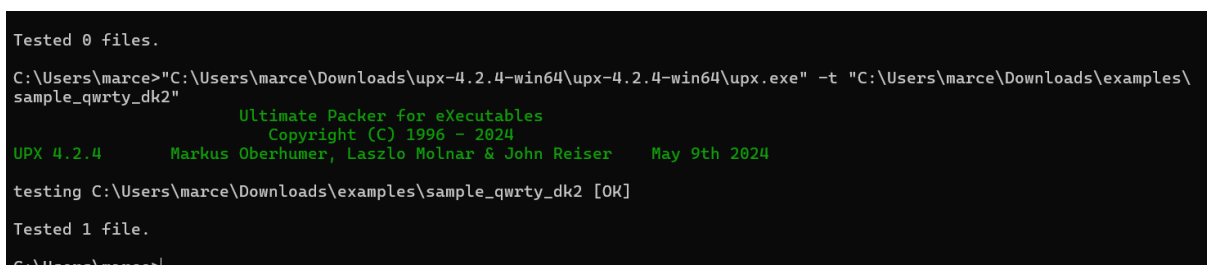
está empaquetado con UPX



```
File Edit Selection View Go Run Terminal Help
t-with-langgraph (1).ipynb day-3-function-calling-with-the-gemini-api.ipynb day-2-embeddings-and-similarity-scores (4).ipynb day-4-fine-tuning-a-custom-model (1).ipynb ejecutable.py x dia2
C:\Users\marce> Desktop > ejecutable.py > ...
1 import magic
2
3 file_path = "C:\\Users\\marce\\Downloads\\examples\\sample_qwrty_dk2"
4 file_type = magic.from_file(file_path)
5 print(f'Tipo de archivo: {file_type}')
6

[Running] python -u "C:\\Users\\marce\\Desktop\\rrrr.py"
Tipo de archivo: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
[Done] exited with code=0 in 0.226 seconds

[Running] python -u "C:\\Users\\marce\\Desktop\\rrrr.py"
Tipo de archivo: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
[Done] exited with code=0 in 0.232 seconds
```



```
Tested 0 files.

C:\Users\marce>"C:\Users\marce\Downloads\upx-4.2.4-win64\upx-4.2.4-win64\upx.exe" -t "C:\Users\marce\Downloads\examples\sample_qwrty_dk2"

      Ultimate Packer for eXecutables
      Copyright (C) 1996 - 2024
UPX 4.2.4      Markus Oberhumer, Laszlo Molnar & John Reiser      May 9th 2024

testing C:\Users\marce\Downloads\examples\sample_qwrty_dk2 [OK]

Tested 1 file.

C:\Users\marce>
```

DESEMPAQUETAR  sample\_qwrty\_dk2

```

Tested 0 files.

C:\Users\marce>"C:\Users\marce\Downloads\upx-4.2.4-win64\upx-4.2.4-win64\upx.exe" -t "C:\Users\marce\Downloads\examples\sample_qwrty_dk2"

          Ultimate Packer for eXecutables
          Copyright (C) 1996 - 2024
UPX 4.2.4      Markus Oberhumer, Laszlo Molnar & John Reiser   May 9th 2024

testing C:\Users\marce\Downloads\examples\sample_qwrty_dk2 [OK]

Tested 1 file.

C:\Users\marce>"C:\Users\marce\Downloads\upx-4.2.4-win64\upx-4.2.4-win64\upx.exe" -d "C:\Users\marce\Downloads\examples\sample_qwrty_dk2" -o "C:\Users\marce\Downloads\examples\sample_qwrty_dk2_unpacked.exe"

          Ultimate Packer for eXecutables
          Copyright (C) 1996 - 2024
UPX 4.2.4      Markus Oberhumer, Laszlo Molnar & John Reiser   May 9th 2024

-----
File size      Ratio      Format      Name
-----
8192 <-      5632      68.75%     win32/pe     sample_qwrty_dk2_unpacked.exe

Unpacked 1 file.

C:\Users\marce>

```

## 2. Análisis Comparativo de Funciones con la Tabla 3 del Artículo

Muestra una lista de las DLLs y funciones utilizadas. Lea el [artículo](#)

[Descargar artículo](#)

"Malware classification based on API calls and behaviour analysis." Compare la tabla 3 contra las funciones de los ejecutables. ¿Hay indicios de algún comportamiento malicioso en base a la comparación? Justifique su respuesta.

```

import pefile
from datetime import datetime

|
|
|
executables = {
    "sample_qwrty_dk2_unpacked.exe": "C:\\Users\\marce\\Downloads\\examples\\sample_qwrty_dk2_unpacked.exe",
    "sample_vg655_25th.exe": "C:\\Users\\marce\\Downloads\\examples\\sample_vg655_25th.exe"
}

for exe_name, exe_path in executables.items():
    print(f"\n{'='*30}")
    print(f" {exe_name}")
    print(f"{'='*30}")

    try:
        pe = pefile.PE(exe_path)

        compile_timestamp = pe.FILE_HEADER.TimeDateStamp
        compile_date = datetime.datetime.fromtimestamp(compile_timestamp)
        print(f"\n[+] Fecha de compilación: {compile_date}")

        print("\n[+] DLLs Importadas:")
        for entry in pe.DIRECTORY_ENTRY_IMPORT:
            print(f"\nDLL: {entry.dll.decode()}")
            for imp in entry.imports:
                print(f"    {hex(imp.address)} - {imp.name.decode() if imp.name else 'Ordinal: ' + str(imp.ordinal)}")

    except Exception as e:
        print(f"\n[!] Error analizando {exe_name}: {e}")

```

## Comparación de Funciones con Indicadores de Comportamiento Malicioso

Actividad Maliciosa	Funciones Relacionadas	Coincidencia con los Ejecutables
Process Hollowing (inyección en otro proceso)	CreateProcessA, GetModuleHandleA, VirtualAllocEx, WriteProcessMemory, ResumeThread	Ambos ejecutables
Creación de hilos remotos (inyección de código)	CreateRemoteThread, OpenProcess, VirtualAllocEx, WriteProcessMemory, GetModuleHandleA, GetProcAddress	sample_vg655_25th.exe
Enumeración de Procesos (detección de procesos activos)	CreateToolhelp32Snapshot, Process32First, Process32Next, WTSEnumerateProcesses	No encontrado
Auto-eliminación después de ejecución	GetModuleFileNameA, ExitProcess, DeleteFile	sample_vg655_25th.exe
Descarga y ejecución de binarios	URLDownloadToFile, ShellExecuteExA	sample_qwrty_dk2_unpacked.exe
Manipulación del registro (persistencia)	RegCreateKeyExA, RegSetValueExA, RegQueryValueExA, RegCloseKey	sample_vg655_25th.exe
Captura de tráfico de red (sniffing)	socket, bind, WSASStartup, recvfrom	sample_qwrty_dk2_unpacked.exe
Infección de otros archivos (virus tipo Parite, Sality, Virut)	CreateFileA, WriteFile, CopyFileA, SetFileTime	sample_vg655_25th.exe

## Evaluación del Comportamiento Sospechoso

### sample\_qwrty\_dk2\_unpacked.exe

- Utiliza ShellExecuteExA y URLDownloadToFile, podría descargar y ejecutar otros archivos maliciosos.
- Usa WSASStartup, socket, send, recv, podría tener potencial comunicación con un servidor remoto (Command & Control - C2).
- Tiene funciones de manipulación de procesos (CreateProcessA), podría lanzar procesos secundarios ocultos.

### sample\_vg655\_25th.exe

- Tiene funciones de inyección de código (VirtualAllocEx, WriteProcessMemory, CreateRemoteThread), indica un posible ataque de inyección en otros procesos.
- Utiliza RegCreateKeyExA y RegSetValueExA, indica capacidad de persistencia en el sistema.
- Implementa CreateFileA, WriteFile, CopyFileA, podría ser usado para replicarse y modificar archivos en el sistema.

### 3. ¿Cuándo fueron compilados los ejecutable?

sample\_vg655\_25th.exe

**Fecha de compilación:** 2010-11-20 09:05:05

sample\_qwrty\_dk2\_unpacked.exe

**Fecha de compilación:** 2009-05-14 17:12:40

### 4. Obtenga el código ensamblador aplicando ingeniería inversa de ambos ejecutable

sample\_qwrty\_dk2\_unpacked.exe



```
0x1000: mov eax, dword ptr [esp + 8]
0x1004: mov edx, dword ptr [esp + 4]
0x1008: mov ecx, eax
0x100a: shr ecx, 9
0x100d: and ecx, 1
0x1010: xor ecx, edx
0x1012: je 0x1035
0x1014: mov ecx, eax
0x1016: shr ecx, 1
0x1018: xor ecx, eax
0x101a: shr ecx, 1
0x101c: xor ecx, eax
0x101e: shr ecx, 3
0x1021: xor ecx, eax
0x1023: and eax, 0x3ffff
0x1028: shr ecx, 0xd
0x102b: shl eax, 1
0x102d: test cl, 1
0x1030: je 0x1035
0x1032: xor eax, 1
0x1035: ret
0x1036: nop
0x1037: nop
0x1038: nop
```

El ejecutable posee indicadores fuertes de comportamiento malicioso, incluyendo:

Manipulación de memoria con stosd, stosw, stosb.

Llamadas indirectas (call dword ptr [dir]) para evasión de detección.

Posible inyección de código (push + call).

Posible persistencia mediante modificación del Registro.

Potencial comunicación con la red (sub esp, 0x60).

Técnicas de evasión (int3, nop).

**sample\_vg655\_25th.exe**

#### **Manipulación de SEH (Structured Exception Handling)**

- El código configura un manejador de excepciones personalizado, lo que puede ser utilizado para ocultar ejecuciones maliciosas o evadir detección.

#### **Llamadas a direcciones indirectas**

- Se observa el uso de `call dword ptr [...]`, lo que sugiere que las funciones son llamadas a través de punteros, una técnica común para evadir análisis estático.

#### **Uso de instrucciones sospechosas (int3)**

- La presencia de varias instrucciones `int3` puede indicar intentos de depuración o técnicas anti-análisis para detectar si el programa está siendo examinado en un entorno de debugging.

#### **Posible carga de librerías dinámicas o funciones de Windows**

- Se observan `push` de direcciones antes de llamadas, lo que sugiere que el código puede estar resolviendo dinámicamente nombres de funciones o librerías, lo que es común en malware para ocultar su comportamiento.

#### **Manipulación de cadenas y argumentos**

- Hay instrucciones que recorren memoria y comparan caracteres, lo que sugiere que el código podría estar analizando argumentos de línea de comandos, algo típico en dropper loaders o ejecutables que toman instrucciones dinámicas.

#### **Salto condicionales y posibles verificaciones de entorno**

- La presencia de múltiples `cmp` y `jne` indica que el código puede estar realizando comprobaciones antes de ejecutar ciertas acciones, lo que podría ser una técnica para evadir ejecución en entornos de análisis o máquinas virtuales.

#### **Uso de funciones de memoria y estructuras globales**

- Se observa manipulación de estructuras de datos en memoria (`mov dword ptr [...]`), lo que puede estar relacionado con la inyección de código en otro proceso o con la modificación de configuraciones del sistema.

## **Análisis dinámico**

5. Analice los ejecutables en la página <https://www.hybrid-analysis.com/>

[Enlaces a un sitio externo.](#)

y muestre el resultado del análisis dinámico. Incluya cualquier pantalla sobre el comportamiento de los ejecutables que la página ofrezca.

sample\_vg655\_25th.exe

Analysis Overview

Request Report Deletion

Submission name: WannaCry.exe.sample  
Size: 3.4MiB  
Type: peexe executable  
Mime: application/x-dosexec  
SHA256: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa  
Submitted At: 2017-07-05 20:35:07 (UTC)  
Last Anti-Virus Scan: 2025-02-04 04:41:45 (UTC)  
Last Sandbox Report: 2024-12-22 05:15:54 (UTC)

malicious

Threat Score: 100/100

AV Detection: 98%

Labeled As:

Trojan.Ransom.WannaCryptor

- #tag#wannacry#Worm
- #ransomware#wanacrypt0r#wcry
- #gozi#isfb#papras#ursnif
- #banker#emotet#rootkit
- #backdoor#coinminer#exploit
- #hacktool#maldoc#metasploit
- #meterpreter#plugx
- #windows-server-utility#adwind
- #agenttesla#alienspy#chanitor
- #chthonic#cridex#crimson
- #darkcomet#dofail#dridex
- #dyre#dyreza#fareit#gootkit
- #hancitor#hawkeye#infostealer
- #keylogger#lokibot#msil
- #nanocore#netwire#neutrino
- #neverquest#poisonivy#pony
- #predator#qakbot#smokeloader
- #stealer#trojan#troidesh
- #vawtrak#zbot#zeus
- X PostLinkE-Mail



Bitdefender	✗ Trojan.Ransom.WannaCryptor.A	Avira	✗ TR/Ransom.JB
Zillya!	✗ Trojan.WannaCry.Win32.2	Sophos	✗ Troj/Ransom-EMG
Vir.IT eXplorer	✗ Trojan.Win32.WannaCry.B	VirusBlokAda	✗ TrojanRansom.WannaCrypt
K7	✗ Trojan ( 0050d7171 )	McAfee	✗ Ransom-O.g
NETGATE	✗ Trojan.Win32.Malware	TACHYON	✗ Ransom/W32.WannaCry.Zen
Varist	✗ W32/Trojan.ZTSA-8671	Antiy	✗ Trojan[Ransom]/Win32.Wanna
AhnLab	✗ Trojan/Win32.WannaCryptor	CMC	✗ Win32_Filecoder_WannaCryptor_D
Lionic	✗ Trojan.Win32.Wanna.toNn	Webroot SMD	✗ Malware_40.6
Emsisoft	✗ Trojan.Ransom.WannaCryptor.A (B)	NANOAV	✗ Trojan.Win32.Ransom.eoptnj
RocketCyber	✓	Comodo	✗ Malware
ESET	✗ Win32/Filecoder.WannaCryptor.D trojan	ClamAV	✗ Win.Ransomware.Wannacryptor-9940180-0
Cylance	✗ Malware_-10		

Close

Not all reports are visible. 8 malicious and 35 error reports are hidden.

Show All As List

A  
A  
F:  
R  
In  
C  
B

Windows 11 64 bit

WannaCry.exe.sample  
December 22nd 2024 05:15:54 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
6 51 187

Characteristics:  
🔗 📁 🗑️

Windows 11 64 bit

WannaCry.exe.sample  
December 15th 2024 20:06:07 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
6 48 184

Characteristics:  
🔗 📁 🗑️

Windows 11 64 bit

WannaCry.exe.sample  
October 31st 2024 17:08:56 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
7 47 174

Characteristics:  
🔗 📁 🗑️

Windows 10 64 bit

ed01ebfbc9eb5bbea545af4d01bf5...  
October 5th 2024 13:46:11 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
6 46 161

Characteristics:  
🔗 📁 🗑️

Windows 10 64 bit

ed01ebfbc9eb5bbea545af4d01bf5...  
May 1st 2024 05:45:35 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
6 46 162

Characteristics:  
🔗 📁 🗑️

Windows 10 64 bit

owo\_im\_not\_ransomware\_xd.exe  
May 15th 2023 06:49:15 (UTC)

!

Malicious

Threat Score:  
100/100

Labeled As:  
Trojan.Ransom.Wa...

Indicators:  
10 68 179

Characteristics:  
🔗 📁 🗑️



Risk Assessment	
Remote Access	Reads terminal service related keys (often RDP related)
Ransomware	Deletes volume snapshots (often used by ransomware) Detected indicator that file is ransomware
Spyware	Accesses potentially sensitive information from local browsers Contains ability to open the clipboard Deletes volume snapshots (often used by ransomware) Found a string that may be used as part of an injection method Hooks API calls Tries to steal browser sensitive information (file access)
Persistence	Disables startup repair Grants permissions using icacils (DACL modification) Installs hooks/patches the running process Spawns a lot of processes Tries to suppress failures during boot (often used to hide system changes) Writes data to a remote process
Fingerprint	Queries kernel debugger information Queries process information Reads system information using Windows Management Instrumentation Commandline (WMIC) Reads the active computer name Reads the cryptographic machine GUID Reads the windows installation language
Evasive	Contains ability to detect virtual environment (API) Contains ability to terminate a process Input file contains API references not part of its Import Address Table (IAT) Marks file for deletion Possibly checks for the presence of an Antivirus engine Possibly tries to evade analysis by sleeping many times Possibly tries to implement anti-virtualization techniques using MAC address detection
Network Behavior	Contacts 50 hosts. <a href="#">View all details</a>

sample\_qwrty\_dk2\_unpacked.exe

Analysis Overview

Submission name: sample\_qwrty\_dk2

Size: 5.5KiB

Type: peexe executable ⓘ

Mime: application/x-dosexec

SHA256: 2f6a3ee24fa839d0fc3864609cfe40f0d78eaa950cafb3a20821a5064faa6cd ⓘ

Submitted At: 2022-03-28 19:00:59 (UTC)

Last Anti-Virus Scan: 2025-02-18 07:29:03 (UTC)

Last Sandbox Report: 2025-02-18 07:28:51 (UTC)

Request Report Deletion

malicious

Threat Score: 100/100

AV Detection: 73%

Labeled As: Win/malicious\_confidence\_100%

X Post

Link

E-Mail

-

0 Community Score ⓘ 0

Analysis Overview

Anti-Virus Scan

Falcon Sandbox

Relations

Incident Response

Community (0)

Back to top

Anti-Virus Results

Updated a while ago

File CollectionsResourcesRequest InfoIP Domain Hash

Anti-Virus Scan Results for OPSWAT Metadefender (11/24)

Last update: 2025-02-18 07:29:03 (UTC)

Bitdefender	✓	Gridinsoft	✓
Avira	✗ BDS/Small.L	Zillya!	✗ Trojan.CMDer.Win32.48
Sophos	✓	Vir.IT eXplorer	✓
VirusBlokAda	✓	K7	✗ Trojan ( 0003087a1 )
McAfee	✓	NETGATE	✓
TACHYON	✓	Varist	✗ W32/Risk.EHGV-6738
Antiy	✗ Trojan[Backdoor]/Win32.CMDer.aa	AhnLab	✗ Backdoor/Win.BackDoor
CMC	✓	Lionic	✗ Trojan.Win32.Agent.lbqx
Webroot SMD	✗ Malware_43.5	Emsisoft	✓
NANOAV	✗ Trojan.Win32.SmallJnvlej	RocketCyber	✓
Comodo	✓	ESET	✗ a variant of Win32/CMDer.AA trojan
ClamAV	✓	Cylance	✗ Malware_-10

Close

[Learn more](#)

back to top

Falcon Sandbox Reports (3)

Characteristics LegendShow All As ListSubmit

Windows 11 64 bit

sample\_qwrty\_dk2

February 18th 2025 07:28:52 (UTC)

!

Malicious

Threat Score: 100/100

Labeled As: Win/malicious\_co...

Indicators: 2 10 56

Characteristics: ⚡

Windows 10 64 bit

sample\_qwrty\_dk2

March 6th 2023 03:06:10 (UTC)

!

Malicious

Threat Score: 100/100

Labeled As: Win/malicious\_co...

Indicators: 3 9 10

Characteristics:

Windows 7 64 bit

sample\_qwrty\_dk2

March 28th 2022 19:01:01 (UTC)

?

Suspicious

Threat Score: 42/100

Labeled As: Win/malicious\_co...

Indicators: 10 1

Characteristics:

## Relations

[Back](#)

Execution Parents (1)		
Input	Threat Level	Actions
MALWR2.zip 7feaa06ed21d4670034ec43d48e3e00b14688f51003856c87999857acad67734	malicious	<a href="#">Share</a>
<a href="#">Previous</a> <b>1</b> <a href="#">Next</a>		

## Incident Response

[Previous](#) **1** [Next](#)

## Incident Response

<b>Risk Assessment</b>	
<b>Remote Access</b>	Reads terminal service related keys (often RDP related)
<b>Fingerprint</b>	Reads the windows installation language
<b>Evasive</b>	Input file contains API references not part of its Import Address Table (IAT) PE file has a section name known to be used by a packer/protector
<b>MITRE ATT&amp;CK™ Techniques Detection</b>	
We found MITRE ATT&CK™ data in 2 reports, on average each report has 10 mapped indicators. <a href="#">View all details</a>	

[Incident Response](#)  
[Anti-Virus](#)  
[Falcon S](#)  
**[Relations](#)**  
[Incident](#)  
[Communi](#)  
[Back to tr](#)

## Community

6. En base al análisis estático y dinámico, ¿considera que los ejecutables son maliciosos? Justifique su respuesta.

sample\_vg655\_25th.exe

### Justificación:

- No está empaquetado, lo que permite su análisis sin necesidad de desempaquetarlo.
- Contiene funciones relacionadas con inyección de código, como VirtualAllocEx, WriteProcessMemory, CreateRemoteThread.
- Modifica el Registro de Windows para persistencia (RegCreateKeyExA, RegSetValueExA).
- Posible replicación de archivos, usando CreateFileA, WriteFile, CopyFileA.
- Manipula el SEH (Structured Exception Handling), lo que puede ser usado para evadir detección.
- Usa llamadas indirectas (call dword ptr [...]), una técnica común en malware para ocultar su comportamiento real.
- Presencia de instrucciones int3, que pueden indicar detección de depuradores y técnicas de evasión de análisis.

- Posible ejecución de librerías dinámicas en tiempo de ejecución, sugiriendo carga dinámica de código.
- Contiene saltos condicionales (cmp y jne) para evitar ejecución en entornos controlados.

### **sample\_qwrty\_dk2\_unpacked.exe**

- Originalmente estaba empaquetado con UPX, lo que indica un intento de ocultar su contenido.
- Usa ShellExecuteExA y URLDownloadToFile, lo que sugiere que puede descargar y ejecutar otros binarios.
- Contiene funciones de manipulación de procesos (CreateProcessA), lo que podría usarse para lanzar procesos ocultos.
- Utiliza funciones de comunicación en red (WSAStartup, socket, recvfrom), lo que sugiere conexión con un servidor remoto (C2).
- Manipulación de memoria con instrucciones sospechosas (stosd, stosw, stosb), lo que indica posible ofuscación o cifrado.
- Usa llamadas indirectas (call dword ptr [dir]), una técnica para evadir detección de análisis estático.
- Posible inyección de código (push + call), lo que sugiere que podría alterar la ejecución de otros procesos.
- Persistencia en el sistema mediante modificación del Registro (RegCreateKeyExA).
- Técnicas de evasión como int3 y nop, lo que indica que podría detectar si está siendo analizado en una máquina virtual.
- Reserva memoria (sub esp, 0x60), común en malware que carga shellcode en tiempo de ejecución.

