



# Malware Analysis

TRACCIA FINALE W24D4

DOCENTE:  
Niko

STUDENTE:  
Simone Cisbaglia

# Agenda

01

ANALISI STATICÀ

02

ANALISI STATICÀ AVANZATA

03

ANALISI DINAMICA FOCUS 1

04

ANALISI DINAMICA FOCUS 2

05

ANALISI DINAMICA FOCUS 3

# 01

## ANALISI STATICÀ

Con l'analisi statica si ispeziona il codice sorgente o binario di un malware per identificare funzionalità e minacce senza eseguirlo, a differenza dell'analisi dinamica dove si esegue il codice in sandbox per osservarne il comportamento.

# Strategia

Utilizzo tool dedicati come VirusTotal, CFF Explorer, Exeinfo PE e IDA Pro.

## VirusTotal

Analizza file, URL, domini e indirizzi IP sospetti per rilevare malware e altre minacce.



## CFF Explorer

Controlla le funzioni importate ed esportate da un malware.

È un tool da installare su macchine virtuali dedicate all'analisi dei malware.

## VirusTotal

## Exeinfo PE

Verifica i file .exe e controlla tutte le loro proprietà.



## IDA Pro

## IDA Pro

Per ottenere informazioni su variabili locali e parametri della funzione main(), IDA Pro è uno strumento avanzato di reverse engineering che offre capacità di disassemblaggio, debugging e analisi statica.



# Prima analisi

## STRUTTURA

Ha 5255 entry points, 4 sezioni e importa 2 librerie:

- **KERNEL32.DLL**: gestione di memoria e processi.
- **ADVAPI32.DLL**: funzioni di sicurezza e gestione account.

3

## REPUTAZIONE

Prima di procedere all'analisi, mi assicuro che il file sia effettivamente un malware. Per farlo, ne estraggo l'hash con **MD5DEEP** e controllo la sua reputazione su **VirusTotal**, basandomi sui riscontri di vari software antivirus.

1

## CATEGORIA

In questa prima analisi, posso vedere che il virus è noto: si tratta di un malware di tipo **Trojan** compilato in data 11-06-2011 in C++. Questo malware è progettato per colpire macchine Intel 386 e processori successivi/compatibili.

2

# HASH CON MD5DEEP64

```
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3>cd md5deep-4.3  
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3>md5deep6  
4 Malware Build Week U3.exe  
[a9c55bb87a7c5c3c923c4fa12940e719] C:\Users\user\Desktop\Software Malware analysi  
s\md5deep-4.3\md5deep-4.3\malware_Build_Week_U3.exe
```

## VIRUSTOTAL



! 52/71 security vendors and no sandboxes flagged this file as malicious [Reanalyze](#) [Similar](#) [More](#)

57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d... Lab3.exe Size 52.00 KB Last Modification Date 1 hour ago EXE

peexe spreader armadillo checks-user-input

DETECTION

DETAILS

RELATIONS

BEHAVIOR

TELEMETRY

COMMUNITY 10

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label [trojan.doina/totbrick](#)

Threat categories [trojan](#)

Family labels [doina](#) [totbrick](#) [genericrcxq](#)

# VIRUSTOTAL

Portable Executable Info ⓘ

**Compiler Products**

- [C++] VS98 (6.0) SP6 build 8804 count=1
- [ C ] VS98 (6.0) SP6 build 8804 count=55
- [---] Unmarked objects count=54
- [RES] VS98 (6.0) SP6 cvtres build 1736 count=1

**Header**

Target Machine	Intel 386 or later processors and compatible processors
Compilation Timestamp	2011-11-06 18:55:06 UTC
Entry Point	5255
Contained Sections	4

**Sections**

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	22086	24576	6.23	6bb361ab84e6ea32f545b12825db9c07	304301.84
.rdata	28672	2478	4096	3.77	23fde5162e5b17a6e440a468b942c3b8	290048.5
.data	32768	16040	12288	0.6	e433b4c400efc11a593220e77ab72779	2826623.75
.rsrc	49152	6768	8192	4.15	9d561586eeb5ecda6c3214cd6a35d6f3	573983.75

**Imports**

- + KERNEL32.dll
- + ADVAPI32.dll

**Contained Resources By Type**

BINARY	1
--------	---

# CFF EXPLORER:

Se mi sposto su «Import Directory», posso controllare le librerie e le funzioni importate. In «Section Headers» vedo le sezioni presenti nel file eseguibile:

**.text:** contiene le istruzioni che la CPU eseguirà all'avvio del malware.

**.rdata:** include informazioni su librerie e funzioni importate/esportate, lette durante l'esecuzione.



**.data:** contiene i dati e le variabili globali modificabili del programma.



**.rsrc:** include risorse come icone, immagini, menu e stringhe utilizzate dall'eseguibile, non parte del codice stesso.



# CFF EXPLORER:

CFF Explorer VIII - [Malware\_Build\_Week\_U3.exe]

File Settings ?

File: Malware\_Build\_Week\_U3.exe

- xe
- Dos Header
- Nt Headers
- File Header
- Optional Header
- Data Directories [x]
- Section Headers [x]**
- Import Directory
- Resource Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address
Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	00005646	00001000	00006000	00001000	00000000
.rdata	000009AE	00007000	00001000	00007000	00000000
.data	00003EA8	00008000	00003000	00008000	00000000
.rsrc	00001A70	0000C000	00002000	0000B000	00000000

Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Asc

00000030 00 00 00 00 00 00 00 00 00 00 E0 00 00 00 00 00 . . . .

# CFF EXPLORER:

CFF Explorer VIII - [Malware\_Build\_Week\_U3.exe]

File Settings ?

File: Malware\_Build\_Week\_U3.exe

- Dos Header
- Nt Headers
  - File Header
  - Optional Header
    - Data Directories [x]
- Section Headers [x]
- Import Directory
- Resource Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

# CFF EXPLORER:

CFF Explorer VIII - [Malware\_Build\_Week\_U3.exe]

File Settings ?

File: Malware\_Build\_Week\_U3.exe

- Dos Header
- Nt Headers
  - File Header
  - Optional Header
    - Data Directories [x]
- Section Headers [x]
- Import Directory
- Resource Directory
- Address Converter
- Dependency Walker
- Hex Editor
- Identifier
- Import Adder
- Quick Disassembler
- Rebuilder
- Resource Editor
- UPX Utility

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
0000769E	N/A	000074EC	000074F0	000074F4	000074F8	000074FC
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
00007632	00007632	0295	SizeofResource
00007644	00007644	01D5	LockResource
00007654	00007654	01C7	LoadResource
00007622	00007622	02BB	VirtualAlloc
00007674	00007674	0124	GetModuleFileNameA
0000768A	0000768A	0126	GetModuleHandleA
00007612	00007612	00B6	FreeResource
00007664	00007664	00A3	FindResourceA

# CFF EXPLORER:

CFF Explorer VIII - [Malware\_Build\_Week\_U3.exe]

File Settings ?

File: Malware\_Build\_Week\_U3.e  
xe  
Dos Header  
Nt Headers  
File Header  
Optional Header  
Data Directories [x]  
Section Headers [x]  
Import Directory  
Resource Directory  
Address Converter  
Dependency Walker  
Hex Editor  
Identifier  
Import Adder  
Quick Disassembler  
Rebuilder  
Resource Editor  
UPX Utility

Module Name Imports OFTs TimeStamp ForwarderChain Name RVA FTs (IAT)

Module Name	Imports	OFTs	TimeStamp	ForwarderChain	Name RVA	FTs (IAT)
000076D0	N/A	00007500	00007504	00007508	0000750C	00007510
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

OFTs FTs (IAT) Hint Name

OFTs	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000076AC	000076AC	0186	RegSetValueExA
000076BE	000076BE	015F	RegCreateKeyExA

# CFF EXPLORER:

Per quanto riguarda le due librerie importate, KERNEL32.DLL e ADVAPI32.DLL, posso ipotizzare che il malware **cerchi di ottenere persistenza e modificare le chiavi di registro** (RegSetValueExA/RegCreateKeyExA) per poter essere eseguito autonomamente.

Inoltre, la presenza di funzioni come SizeofResource, LockResource, LoadResource e FindResourceA fa presupporre che sia un **Dropper**, ossia un malware che contiene al suo interno un altro malware. Utilizza queste API per localizzare, all'interno della sezione "risorse", il malware da estrarre e successivamente caricarlo in memoria per l'esecuzione.

# EXEINFOPE:

# EXEINFOPE:

Header info : [ Malware\_Build\_Week\_U3.exe ] - Size of Code: 006000h - decimal : 24 KB

Directory Info :	RVA	SIZE	
Export :	00000000	00000000	>> Not used 1970-01-01
Import :	000074EC	0000003C	>> ( 02 ) .rdata 1970-01-01
Resource :	0000C000	00001A70	12 % of exe Nr of ID : 0
Exception :	00000000	00000000	1970-01-01
Security :	00000000	00000000	not Signed
Base Reloc :	00000000	00000000	

From header :	Very often :
Size of headers :	00001000 400 or 1000
Size of optional header :	00E0 00E0
Number of Dirs :	0010 0010h
Base of Code :	00001000 00001000
Image Base :	00400000 00400000
Magic optional header :	010B 010B 32bit
Debugger Info - size :	No

Imports :					
DllName	OriginalFirstThunk	TimeDateStamp	ForwarderChain	Name	FirstThunk
KERNEL32.dll	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	00007528	00000000	00000000	000076D0	00007000

# EXEINFOPE:

Exeinfo PE ha confermato gli stessi dati rilevati precedentemente. Le informazioni riguardanti le librerie importate, le funzioni utilizzate e le sezioni del file eseguibile corrispondono a quanto già analizzato.

Questo conferma ulteriormente che il malware **tenta di ottenere persistenza** modificando le **chiavi di registro** e utilizza funzioni specifiche che suggeriscono che si tratti di un **Dropper**. Questi dati rafforzano l'ipotesi iniziale sull'intento e la struttura del malware.

# IDA Pro

Strumento per il reverse engineering avanzato, disassemblaggio, debugging e analisi statica.

**Variabili locali:** Allocate nello stack tramite istruzioni PUSH o SUB che aumentano il puntatore dello stack (SP) per creare spazio.

Nell'assembly, le etichette che iniziano con VAR\_ indicano variabili locali.

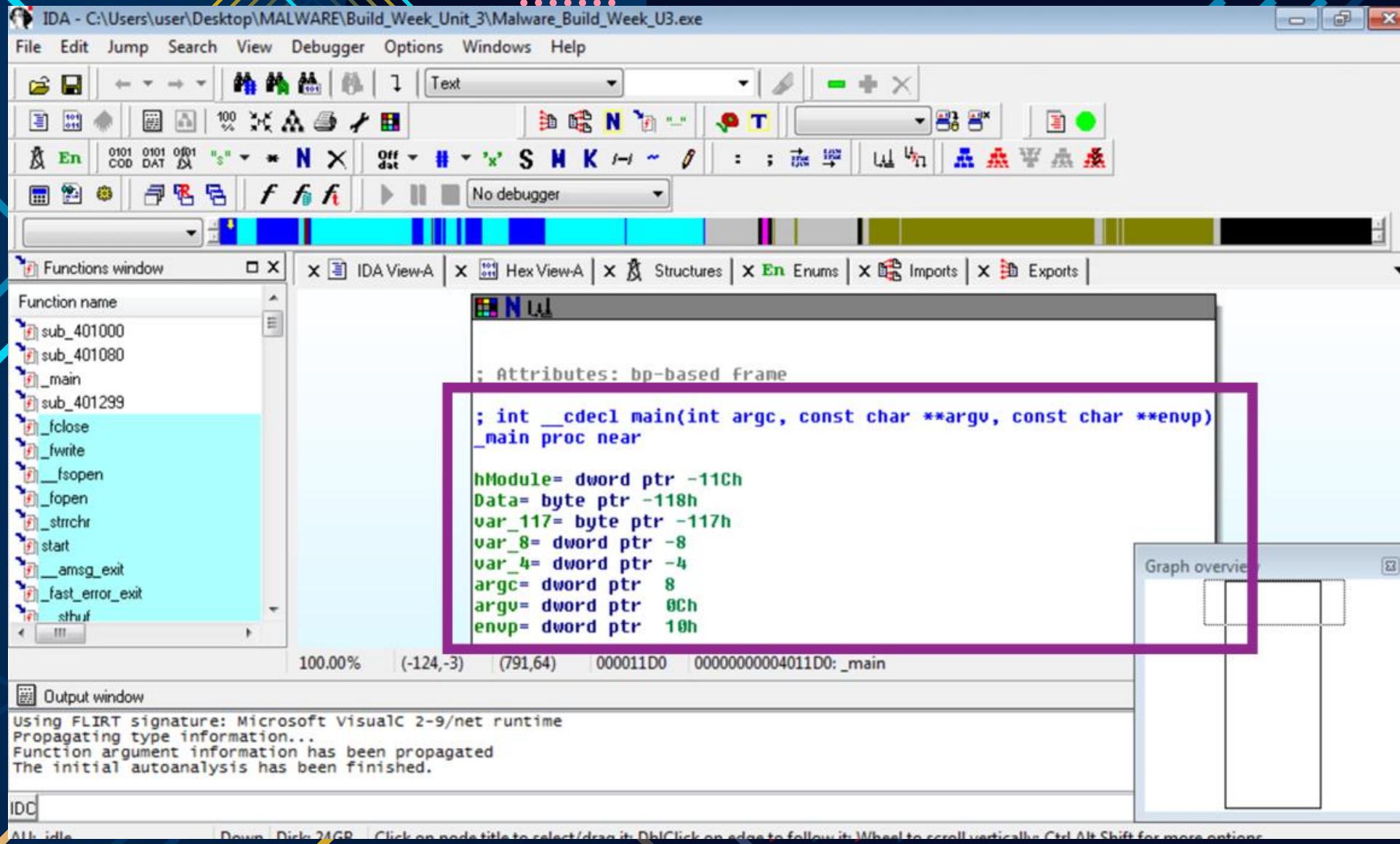
**Parametri:** Solitamente passati ai registri o tramite lo stack prima della chiamata della funzione. I commenti nel codice possono fornire indicazioni sui parametri. Le etichette ARG\_ indicano parametri passati alla funzione.

**Valori offset:** Gli offset (es. VAR\_54H) sono utilizzati per accedere a dati specifici nello stack. Gli offset negativi rispetto all'indirizzo base del frame (EBP su x86) indicano variabili locali, mentre gli offset positivi indicano parametri.

In questo caso, posso identificare cinque variabili locali:

hModule, Data, var\_117, var\_8 e var\_4. I parametri visibili sono tre: argc, argv e envp.

# IDA Pro



# Prime conclusioni:

In conclusione, attraverso l'analisi statica avanzata del malware utilizzando strumenti dedicati come **VirusTotal**, **CFF Explorer**, **Exeinfo PE** e **IDA Pro**, ho potuto identificare diverse caratteristiche chiave del malware. Confermando che si tratta di un **Trojan**, noto e compilato in C++ nel 2011, progettato per colpire macchine Intel 386 e processori successivi.

Le librerie importate, **KERNEL32.DLL** e **ADVAPI32.DLL**, indicano che il malware cerca di ottenere persistenza e modificare chiavi di registro per eseguire in autonomia. La presenza di funzioni specifiche suggerisce che il malware potrebbe essere un **Dropper**, utilizzando API per localizzare e caricare in memoria un altro malware contenuto nelle risorse.

Attraverso IDA Pro, abbiamo analizzato più in dettaglio le variabili locali e i parametri della funzione principale, identificando le etichette che indicano variabili locali e parametri. Le variabili locali rilevate includono hModule, Data, var\_117, var\_8 e var\_4, mentre i parametri sono argc, argv e envp.

## 02

# ANALISI STATICÀ AVANZATA

L'analisi statica avanzata presuppone la conoscenza dei fondamenti del reverse engineering per identificare il comportamento di un malware a partire dall'analisi delle istruzioni che lo compongono. Questo passaggio è essenziale per comprendere esattamente cosa fa il malware a livello di istruzioni della CPU. Inoltre, posso estrarre stringhe di testo, URL, chiavi di cifratura e altre risorse dal codice del malware, che possono indicarne il comportamento o l'intento. Esaminando il codice relativo alla rete, posso capire come il malware comunica.

# Focus con IDA Pro:

Proseguendo con l'analisi statica del codice del malware utilizzando IDA Pro, ho trovato la funzione **RegCreateKeyExA** alla locazione di memoria 00401021. Questa è una funzione delle API di Windows che permette alle applicazioni di interagire con il registro di Windows, creando nuove chiavi o aprendo chiavi esistenti per modificarne i valori.

Il malware può utilizzare **RegCreateKeyExA per ottenere persistenza**, creando nuove chiavi di registro o modificando chiavi esistenti, assicurandosi che il codice malevolo venga eseguito ad ogni avvio del sistema. Ad esempio, potrebbe aggiungere una voce nella chiave RUN per eseguire automaticamente il malware all'avvio del sistema.

Per quanto riguarda il metodo di passaggio dei parametri alla funzione, la convenzione di chiamata più comune nei sistemi operativi su architettura x86 è quella di "pushare" i parametri sullo stack prima della chiamata (CALL) alla funzione RegCreateKeyExA. I parametri vengono letti dalla funzione in ordine inverso, partendo da hKey fino a lpdwDisposition.

# Focus con IDA Pro:

.text:00401017

push offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\Cu

Alla locazione di memoria 00401017, ho trovato l'istruzione PUSH offset SubKey, che spinge l'indirizzo della sottochiave di registro nello stack. Il nome vicino al codice suggerisce che potrebbe essere il nome della sottochiave che verrà creata dal processo hKey in WINLOGON, un componente del sistema operativo Windows responsabile della gestione delle sessioni di accesso (login) e disconnessione (logout) degli utenti.

.text:00401027

test

eax, eax

.text:00401029

jz

short loc 401032

Per quanto riguarda il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029, vediamo:

- Un'istruzione condizionale TEST, simile all'istruzione AND logico bit a bit, ma senza modificare il contenuto degli operandi (in questo caso EAX e se stesso). Modifica invece il flag ZF (Zero Flag) del registro EFLAGS, che viene impostato a 1 solo se il risultato dell'AND è 0 ( $0 \text{ AND } 1 = 0 * 1 = 0 \rightarrow \text{ZeroFlag} = 1$ ). Viene utilizzato per controllare se un valore è zero o meno. Se TEST è zero, lo ZF è 1.
- Un conditional jump di tipo JZ, che nel flusso di controllo salta a una determinata locazione di memoria (in questo caso 00401032) se ZF è pari a uno. Se il valore contenuto nel registro EAX non è 0, il salto non avverrà.

# Focus con IDA Pro:

```
.text:00401032 loc_401032:  
.* .text:00401032          mov     ecx, [ebp+cbData] ; CODE XREF: sub_401000+29↑j
```

L'operazione mostrata nell'immagine equivale a un ciclo if in C come il seguente (dove eax rappresenta una variabile in C che contiene il valore che era nel registro EAX):

```
if (eax == 0) {  
    // Vai a loc_401032  
    // Codice equivalente a quello trovato nella locazione 00401032  
    ecx = cbData; // Assegna a ecx il valore di cbData  
}  
else {  
    // Riprova a fare un'operazione (ad esempio..)  
    // Codice per gestire il caso in cui eax non sia zero  
}
```



# Focus con IDA Pro:

```
.text:0040103E  
.text:00401043  
.text:00401046  
.text:00401047
```

```
push    offset ValueName ; "GinaDLL"  
mov     eax, [ebp+hObject]  
push    eax             ; hKey  
call    ds:RegSetValueExA
```

Per quanto riguarda la chiamata alla locazione 00401047, posso vedere che si tratta di una chiamata alla funzione **RegSetValueExA**, una funzione delle API di Windows che imposta il valore di una voce nel registro di sistema. Il prefisso DS: indica che l'indirizzo della funzione è preso dal registro DS, che è il segmento dati.

Questa funzione, quindi, impedisce istruzioni affinché l'offset ValueName abbia valore "GinaDLL" in una specifica chiave di registro identificata da hKey.



# Focus con IDA Pro:

The image shows the IDA Pro interface with the title bar "IDA - C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\Malware\_Build\_Week\_U3.exe". The menu bar includes File, Edit, Jump, Search, View, Debugger, Options, Windows, and Help. The toolbar contains various icons for file operations, search, and analysis. The status bar at the bottom shows "AU: idle", "Down", and "Disk: 24GB".

The main window displays assembly code in the "Functions window" tab. A specific function, `text-00401002`, is highlighted with a green border. The assembly code for this function is as follows:

```
push    ecx
push    0          ; lpdwDisposition
lea     eax, [ebp+hObject]
push    eax         ; phkResult
push    0           ; lpSecurityAttributes
push    0F003Fh    ; samDesired
push    0           ; dwOptions
push    0           ; lpClass
push    0           ; Reserved
push    offset SubKey ; "SOFTWARE\Microsoft\Windows N
push    80000002h   ; hKeu
call    ds:RegCreateKeyExA
test   eax, eax
jz     short loc_401032
mov    eax, 1
```

The output window at the bottom shows the following messages:

```
Pattern is not found
Command "JumpEnter" failed
Command "JumpEnter" failed
Retrieving information from the database... ok
```

The IDC window at the bottom is currently empty.

# Focus con IDA Pro:

The image shows the IDA Pro interface with the title bar "IDA - C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\Malware\_Build\_Week\_U3.exe". The menu bar includes File, Edit, Jump, Search, View, Debugger, Options, Windows, and Help. The toolbar contains various icons for file operations, search, and analysis. The main window displays assembly code in the middle pane, with the current instruction at address 00401021 highlighted. The left pane shows the Functions window listing several subroutines and their addresses. The bottom pane shows the Output window with messages about pattern searching and database retrieval. The IDC (IDC) window is also visible at the bottom.

```
File Edit Jump Search View Debugger Options Windows Help
Address 00401021
Functions window
IDA View-A Hex View-A Structures Enums Imports Exports
Function name
sub_401000
sub_401004
sub_401006
sub_401009
sub_40100A
sub_40100C
sub_401011
sub_401013
sub_401015
text:00401017
sub_40101C
sub_401021
sub_401027
sub_401029
sub_40102B
push    ecx
push    0
lea     eax, [ebp+hObject]
push    eax
push    0
push    0F003Fh
push    0
push    0
push    0
push    0
push    offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\Cu
push    80000002h
call   ds:RegCreateKeyExA
test   eax, eax
jz    short loc_401032
mov    eax, 1
00001017 | 0000000000401017: sub_401000+17
Output window
Pattern is not found
Command "JumpEnter" failed
Command "JumpEnter" failed
Retrieving information from the database... ok
IDC
AU: idle Down Disk: 24GB
```

# Focus con IDA Pro:

The image shows the IDA Pro interface with the title bar "IDA - C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\Malware\_Build\_U3.exe". The menu bar includes File, Edit, Jump, Search, View, Debugger, Options, Windows, and Help. The toolbar contains various icons for file operations, search, and analysis. The main window displays assembly code in the "Functions window" and "IDA View-A" tab. The assembly code for function .text:00401003 is as follows:

```
push  ecx
push  0
lea   eax, [ebp+hObject]
push  eax
push  0
push  0F003Fh
push  0
push  0
push  0
push  offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\Cu
push  80000002h
call  ds:RegCreateKeyExA
test  eax, eax
jz    short loc_401032
mov   eax, 1
```

The output window at the bottom shows:

```
Pattern is not found
Command "JumpEnter" failed
Command "JumpEnter" failed
Retrieving information from the database... ok
```

The IDC window at the bottom is empty.

# Focus con IDA Pro:

IDA - C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\Malware\_Build\_Unit\_3.exe

File Edit Jump Search View Debugger Options Windows Help

Address: 00401021

No debugger

Functions window

Function name

- sub\_401000
- sub\_401080
- \_main
- sub\_401299
- \_fclose
- \_fwrite
- \_fopen
- \_open
- \_strchr

; Alignment : default

; Imports from ADVAPI32.dll

=====

; Segment type: Externs

; \_idata

; LSTATUS \_\_stdcall RegSetValueEx(HKEY hKey, LPCSTR lpValueName, DWORD Reserved, DWORD dwType, const BYTE \*lpData, DWORD cbD

extrn RegSetValueExA:dword ; CODE XREF: sub\_401000+47↑p

Command "JumpEnter" failed

Retrieving information from the database... ok

IDC

AU: idle Down Disk: 24GB

```
.text:00401032 loc_401032:    ; CODE XREF: sub_401000+29↑j
.text:00401032        mov     ecx, [ebp+cbData]      ; cbData
.text:00401035        push    ecx
.text:00401036        mov     edx, [ebp+lpData]      ; lpData
.text:00401039        push    edx
.text:0040103A        push    1                  ; dwType
.text:0040103C        push    0                  ; Reserved
.text:0040103E        push    offset ValueName ; "GinaDLL"
.text:00401043        mov     eax, [ebp+hObject]
.text:00401046        push    eax
.text:00401047        call    ds:RegSetValueExA
test    eax    eax
```

# Conclusioni

Durante l'analisi del malware con IDA Pro, ho individuato la funzione **RegCreateKeyExA** alla locazione di memoria 00401021. Questa funzione delle API di Windows permette al malware di interagire con il registro di Windows **per creare o modificare chiavi, ottenendo persistenza.**

Alla locazione 00401017, l'istruzione PUSH offset SubKey spinge l'indirizzo della sottochiave di registro nello stack, suggerendo la creazione di una sottochiave tramite hKey in WINLOGON, responsabile della gestione delle sessioni di accesso e disconnessione degli utenti.

Le istruzioni tra 00401027 e 00401029 includono un'istruzione condizionale TEST e un conditional jump JZ, utilizzate per controllare se un valore è zero e saltare a una locazione di memoria (00401032), equivalente a un ciclo if in C.

Alla locazione 00401047, la chiamata alla funzione RegSetValueExA imposta il valore di **una voce** nel registro di sistema. Il prefisso DS: indica che l'indirizzo della funzione è preso dal registro DS. Questa funzione imposta l'offset ValueName al valore "GinaDLL" in una chiave di registro identificata da hKey.

## 03

# ANALISI DINAMICA FOCUS 1

Durante l'analisi dinamica, eseguo il codice in un ambiente controllato. Questo mi permette di osservare il comportamento del malware in modo sicuro, monitorando le sue interazioni con il sistema operativo, i file di sistema, le chiavi di registro e la rete. Attraverso questa tecnica, posso identificare le azioni malevoli che il malware tenta di eseguire e capire meglio come si comporta in un contesto reale.

# Preparazione ambiente

Prima di iniziare, ricreo un'istantanea della macchina Windows su **VirtualBox** per poterla ripristinare in caso di problemi. Poiché andrò ad eseguire dei malware, mi assicuro di rispettare i seguenti accorgimenti:



Disattivo il controller USB.



Disattivo la comunicazione con la rete (impostazione solo INT).

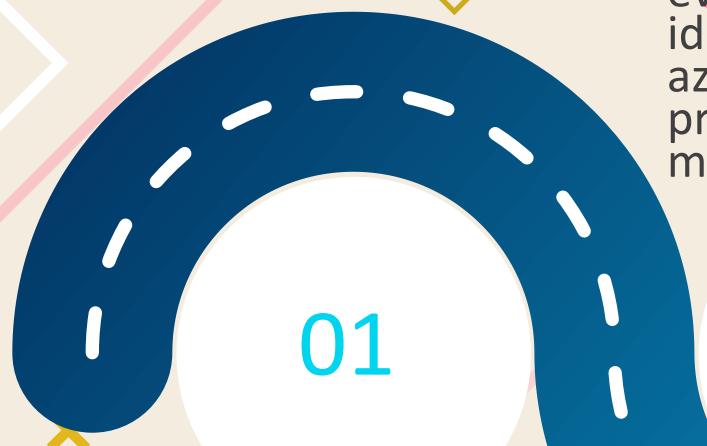


Disabilito la condivisione delle cartelle.



Disabilito gli appunti condivisi (copia/incolla).

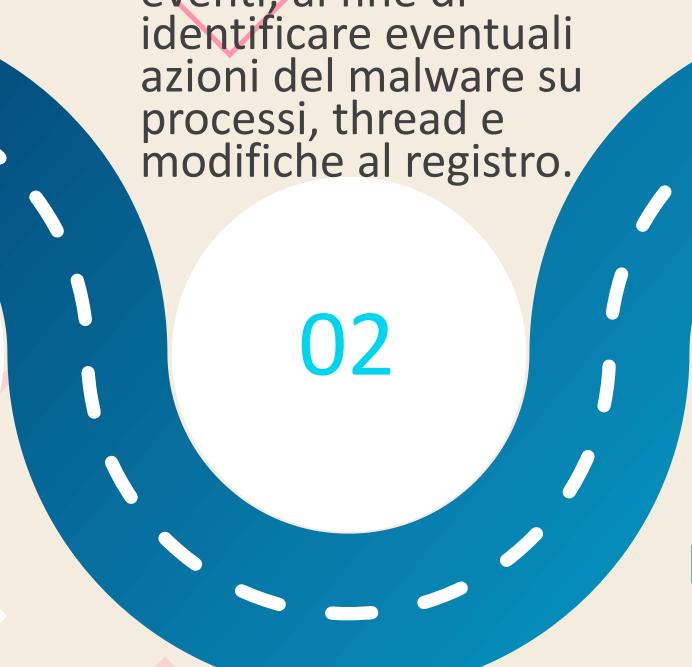
# Preparazione ambiente



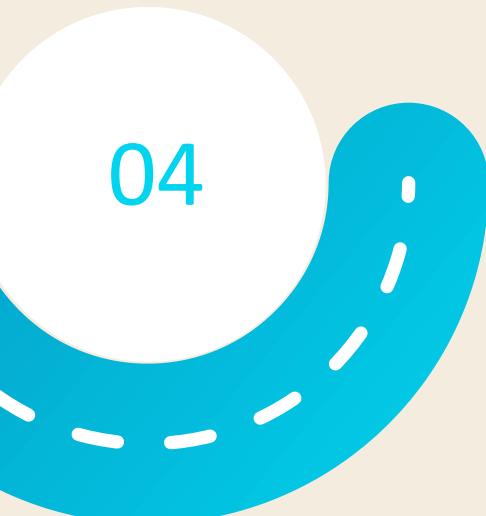
Copia ambiente

▷▷▷  
Copia dell'ambiente  
virtuale dove eseguire il  
malware in sicurezza

**Procmon**  
per catturare tutti gli  
eventi, al fine di  
identificare eventuali  
azioni del malware su  
processi, thread e  
modifiche al registro.

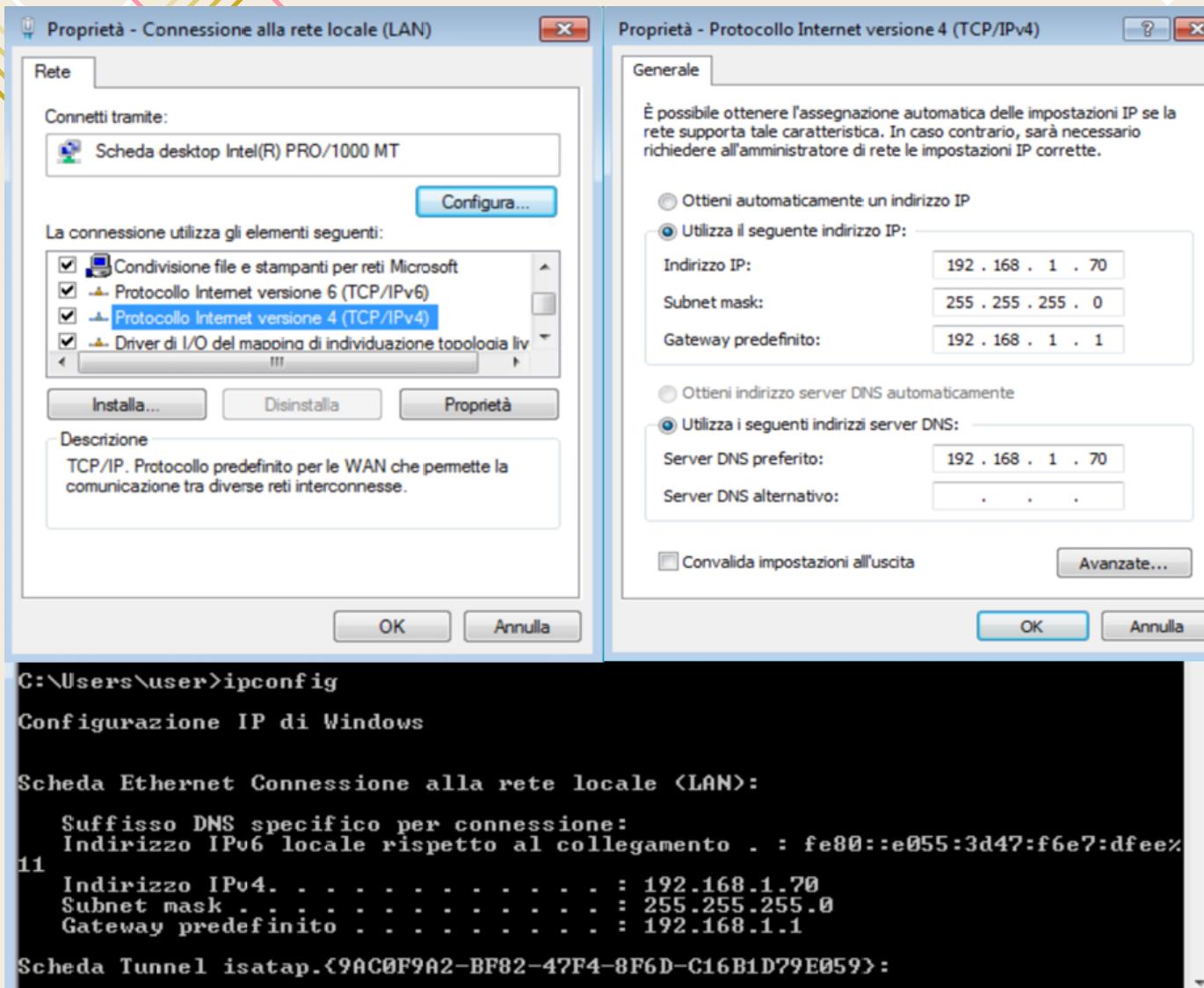


**Regshot**  
per confrontare le  
modifiche a livello di chiavi  
di sistema tramite  
screenshot (prima e dopo).



**ApateDNS**  
Imposto un IP statico  
per monitorare, tramite  
ApateDNS, le chiamate ✘  
che il malware farà nel  
web.  
✘  
✘

# Preparazione ambiente



# APATEDNS

**ApateDNS**

Capture Window DNS Hex View

Time	Domain Requested	DNS Returned
18:44:53	teredo.ipv6.microsoft.com	FOUND

[+] Using 192.168.1.70 as return DNS IP!  
[+] DNS set to 127.0.0.1 on Scheda desktop Intel(R) PRO/1000 MT.  
[+] Sending valid DNS response of first request.  
[+] Server started at 18:44:51 successfully.

DNS Reply IP (Default: Current Gateway/DNS):

# of NXDOMAIN's:

Selected Interface:

**Regshot 1.9.0 x64 Unicode**

Compare logs save as:  
 Plain TXT  HTML document

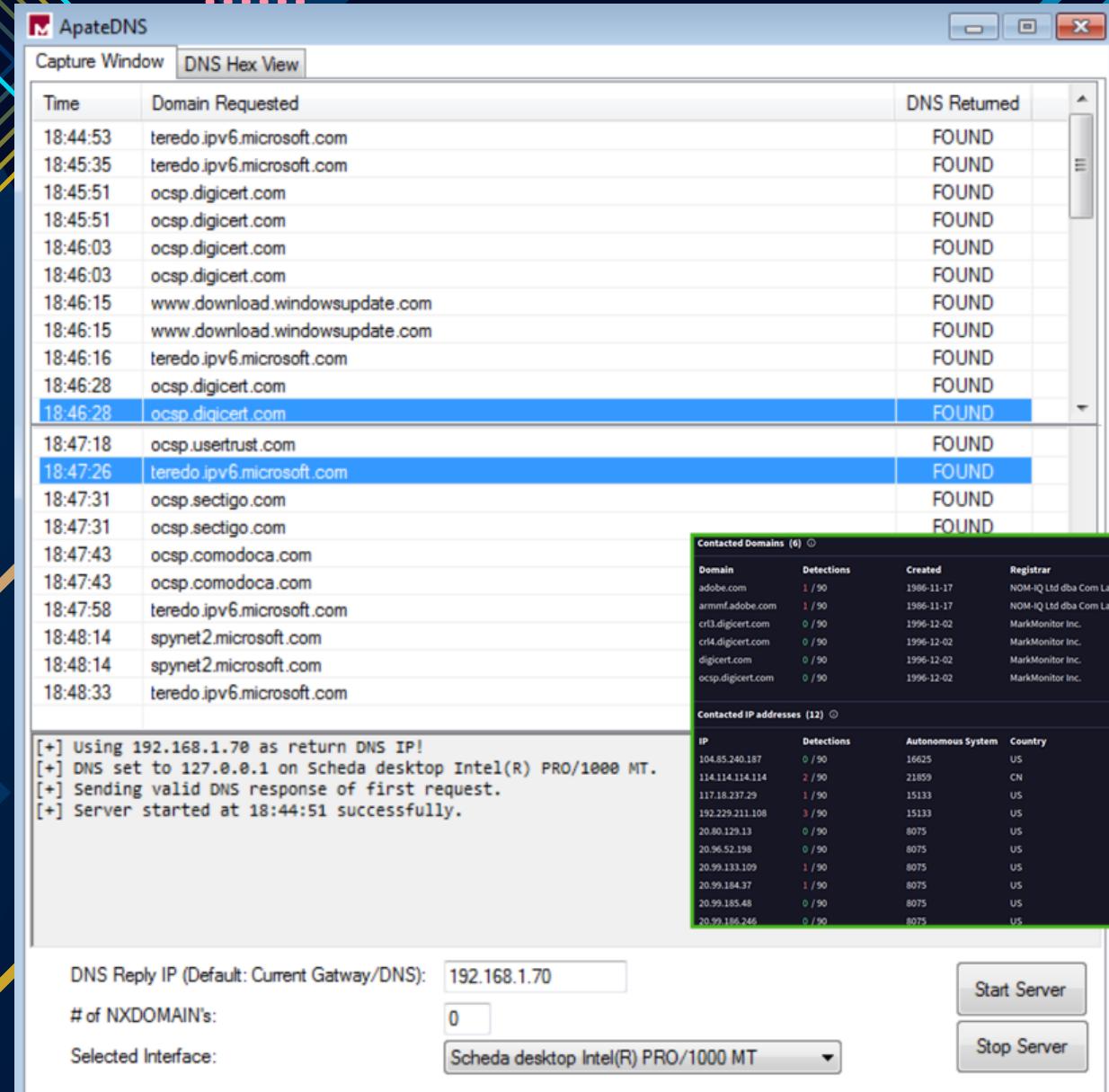
Scan dir 1[;dir2;dir3;...;dir nn]:  
C:\Windows

Output path:  
C:\Users\user\AppData\Loc

Add comment into the log:

Keys: 226266 Values: 439912 Time: 3s687ms

# CHIAMATE INTERCETTATE



# Conclusioni

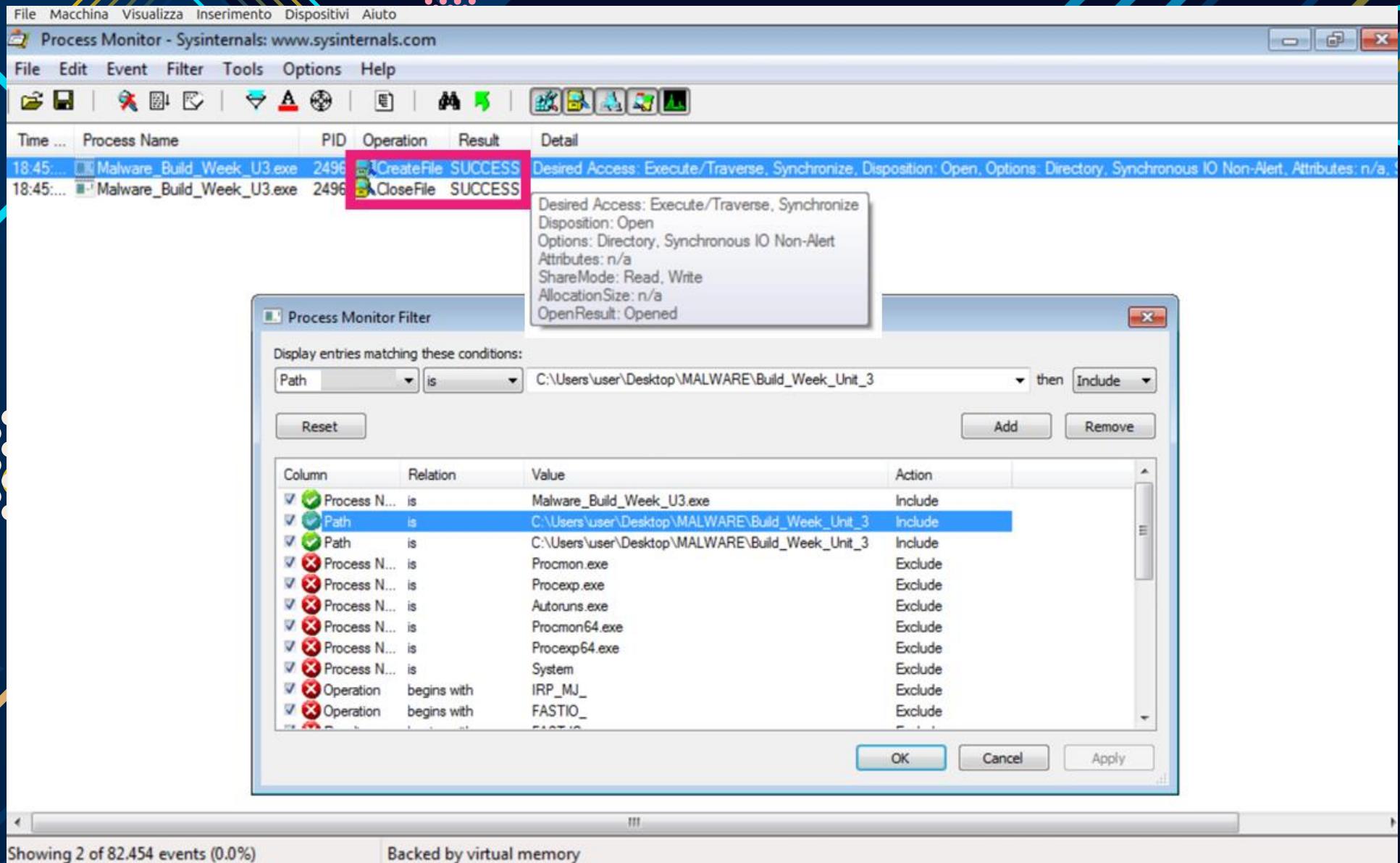
Si nota subito che all'interno della cartella dove è situato l'EXE del malware viene creato un file, ovvero **MSGINA.DLL** (acronimo di "Microsoft Graphical Identification and Authentication").

Questo file gestisce il processo di accesso, nello specifico l'interfaccia utente di logon interattiva, che include la classica schermata di accesso per l'inserimento di nome utente e password.

Funziona nel contesto del processo **Winlogon** e viene caricato all'inizio del processo di avvio del sistema. È responsabile di fornire procedure personalizzabili per l'identificazione e l'autenticazione degli utenti.

Il malware ottiene persistenza all'avvio del sistema operativo modificando proprio questo file. Filtrando per path su **Procmon**, si vede la creazione del file nella cartella scelta tramite il processo **CreateFile**.

# PROCMON



## ANALISI DINAMICA FOCUS 2

Focus su chiavi registro modificate, rilievi eseguiti con i tool Procmon e Regshot

04

# PROCMON

Process Monitor - Sysinternals: www.sysinternals.com

File Macchina Visualizza Inserimento Dispositivi Aiuto

File Edit Event Tools Options Help

Time ... Process Name PID Operation Path Result Detail

18:45...	Malware_Build_Week_U3.exe	2496	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	REPARSE	Desired Access: Query Value
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	NAME NOT FOUND	Desired Access: Query Value
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	REPARSE	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Srp\GP\DLL	NAME NOT FOUND	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\Software\Wow6432Node\Policies\Microsoft\Windows\Safer\CodeIdentifiers	REPARSE	Desired Access: Query Value
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers	SUCCESS	Desired Access: Query Value
18:45...	Malware_Build_Week_U3.exe	2496	RegSetInfoKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers	SUCCESS	KeySetInformationClass: KeyValue
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers\TransparentEnabled	NAME NOT FOUND	Length: 80
18:45...	Malware_Build_Week_U3.exe	2496	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\safer\codeidentifiers	SUCCESS	
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers	NAME NOT FOUND	Desired Access: Query Value
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	REPARSE	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	SUCCESS	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegSetInfoKey	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	SUCCESS	KeySetInformationClass: KeyValue
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryValue	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions\(\Default)	SUCCESS	Type: REG_SZ, Length: 3
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	REPARSE	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegSetInfoKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	KeySetInformationClass: KeyValue
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAccompat	NAME NOT FOUND	Length: 548
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSUserEnabled	SUCCESS	Type: REG_DWORD, Length: 4
18:45...	Malware_Build_Week_U3.exe	2496	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	SUCCESS	
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM	SUCCESS	Desired Access: Maximum Allowed
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryKey	HKLM	SUCCESS	Query: HandleTags, Handled
18:45...	Malware_Build_Week_U3.exe	2496	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Diagnos	NAME NOT FOUND	Desired Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryKey	HKLM	SUCCESS	Desired Access: Maximum Allowed
18:45...	Malware_Build_Week_U3.exe	2496	RegCreateKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Granted Access: Read
18:45...	Malware_Build_Week_U3.exe	2496	RegSetInfoKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	KeySetInformationClass: KeyValue
18:45...	Malware_Build_Week_U3.exe	2496	RegQueryKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	Query: HandleTags, Handled
18:45...	Malware_Build_Week_U3.exe	2496	RegSetValue	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL	ACCESS DENIED	Type: REG_SZ, Length: 520, Data: C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
18:45...	Malware_Build_Week_U3.exe	2496	RegCloseKey	HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon	SUCCESS	
18:45...	Malware_Build_Week_U3.exe	2496	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CURRENTVERSION\Image File Execution O...	SUCCESS	

Showing 55 of 82.454 events (0.0%) Backed by virtual memory

Event Properties

Date: 23/04/2024 18:45:57  
Thread: 1456  
Class: Registry  
Operation: RegSetValue  
Result: ACCESS DENIED  
Path: HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL  
Duration: 0.0000024

Type: REG\_SZ  
Length: 520  
Data: C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\msgina32.dll

Copy All Close

# Conclusioni

Filtrando per chiavi di registro, noto che ad un certo punto viene chiamata la funzione **RegSetValue**, una funzione tipica della libreria **advapi32.dll** importata dal malware. Questa libreria fornisce funzioni relative alla sicurezza e alla gestione degli account, che i malware possono sfruttare per modificare permessi, accedere a token di sicurezza e alterare il registro di sistema.



Posso vedere la chiave di registro **REG\_SZ** (identificata da **hKey**) a cui viene assegnato il valore di **GINA.DLL**. Questo valore "malevolo" verrà sovrascritto nella libreria **msgina32.dll**, sostituendo la copia "sana". In tal modo, il malware potrà avviarsi ad ogni login nel contesto del processo **Winlogon**.



# ANALISI DINAMICA FOCUS 3

Visualizzazione dell'attività sul file system

05

# Attività malware

Filtrando per attività sul file system, vedo una **CreateFile** (funzione che modifica un file esistente, o se non esiste ne crea uno nuovo) relativa al file **msgina32.dll**. Questa funzione è tipica della libreria **kernel32.dll**, che viene appunto richiamata dal malware.

# PROCMON

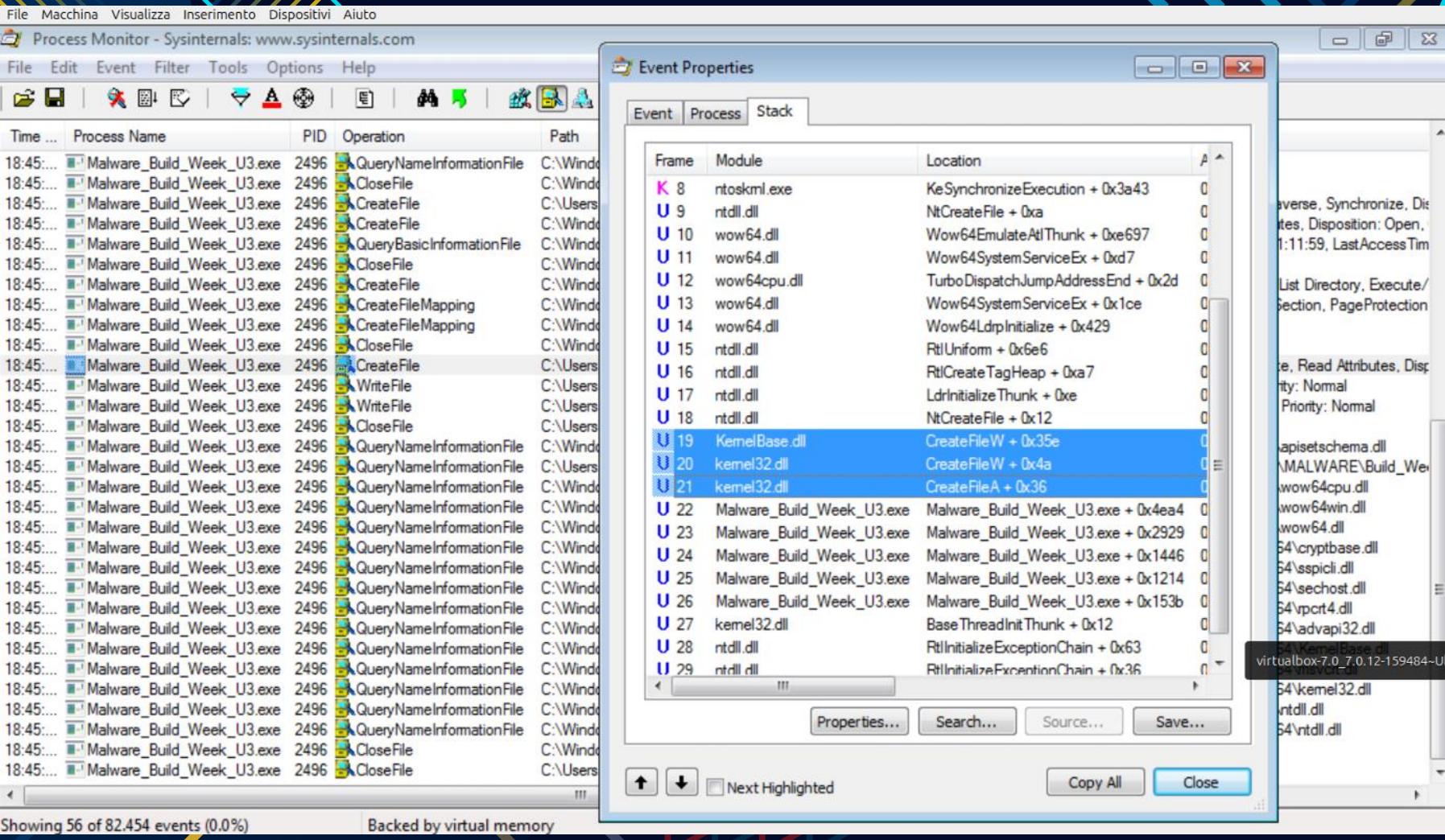
The screenshot shows the Process Monitor application interface. The main window displays a list of events captured by the monitor. The columns in the table are: Process Name, PID, Operation, Path, Result, and Detail. The 'Operation' column shows various file-related operations like QueryNameInformationFile, CreateFile, CloseFile, CreateFileMapping, WriteFile, and CloseFile. The 'Path' column shows the full path of the files being accessed, such as C:\Windows, C:\Windows\SysWOW64\sechost.dll, and C:\Users\user\Desktop\MALWARE\Build\_Week\_Unit\_3\msgina32.dll. The 'Result' column indicates the outcome of each operation, mostly 'SUCCESS'. The 'Detail' column provides more specific information about the access, including desired access rights (e.g., Execute/Traverse, Read Attributes, Read Data/List Directory), creation time, last access time, and sync types.

Process Name	PID	Operation	Path	Result	Detail
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows	SUCCESS	Name: \Windows
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows	SUCCESS	
Malware_Build_Week_U3.exe	2496	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3	SUCCESS	Desired Access: Execute/Traverse, Synchronize, Disposition: C
Malware_Build_Week_U3.exe	2496	CreateFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Desired Access: Read Attributes, Disposition: Open, Options: O
Malware_Build_Week_U3.exe	2496	QueryBasicInformationFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	CreationTime: 14/07/2009 01:11:59, LastAccessTime: 14/07/2009 01:11:59
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	
Malware_Build_Week_U3.exe	2496	CreateFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Desired Access: Read Data/List Directory, Execute/Traverse, : FILE LOCK..SyncType: SyncTypeCreateSection, PageProtection:
Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll	SUCCESS	SyncType: SyncTypeOther
Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll	SUCCESS	
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	
Malware_Build_Week_U3.exe	2496	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: O
Malware_Build_Week_U3.exe	2496	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 0, Length: 4.096, Priority: Normal
Malware_Build_Week_U3.exe	2496	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS	Offset: 4.096, Length: 2.560, Priority: Normal
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll	SUCCESS	
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\apisetschema.dll	SUCCESS	Name: \Windows\System32\apisetschema.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Bui...	SUCCESS	Name: \Users\user\Desktop\MALWARE\Build_Week_Unit_3\kali-linux-2...
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64cpu.dll	SUCCESS	Name: \Windows\System32\wow64cpu.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64win.dll	SUCCESS	Name: \Windows\System32\wow64win.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64.dll	SUCCESS	Name: \Windows\System32\wow64.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS	Name: \Windows\SysWOW64\cryptbase.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\sspicli.dll	SUCCESS	Name: \Windows\SysWOW64\sspicli.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Name: \Windows\SysWOW64\sechost.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\rpcrt4.dll	SUCCESS	Name: \Windows\SysWOW64\rpcrt4.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Name: \Windows\SysWOW64\advapi32.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\KernlBase.dll	SUCCESS	Name: \Windows\SysWOW64\KernlBase.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\msvcr7.dll	SUCCESS	Name: \Windows\SysWOW64\msvcr7.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Name: \Windows\SysWOW64\kernel32.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\ntdll.dll	SUCCESS	Name: \Windows\System32\ntdll.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Name: \Windows\SysWOW64\ntdll.dll
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows	SUCCESS	
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3	SUCCESS	

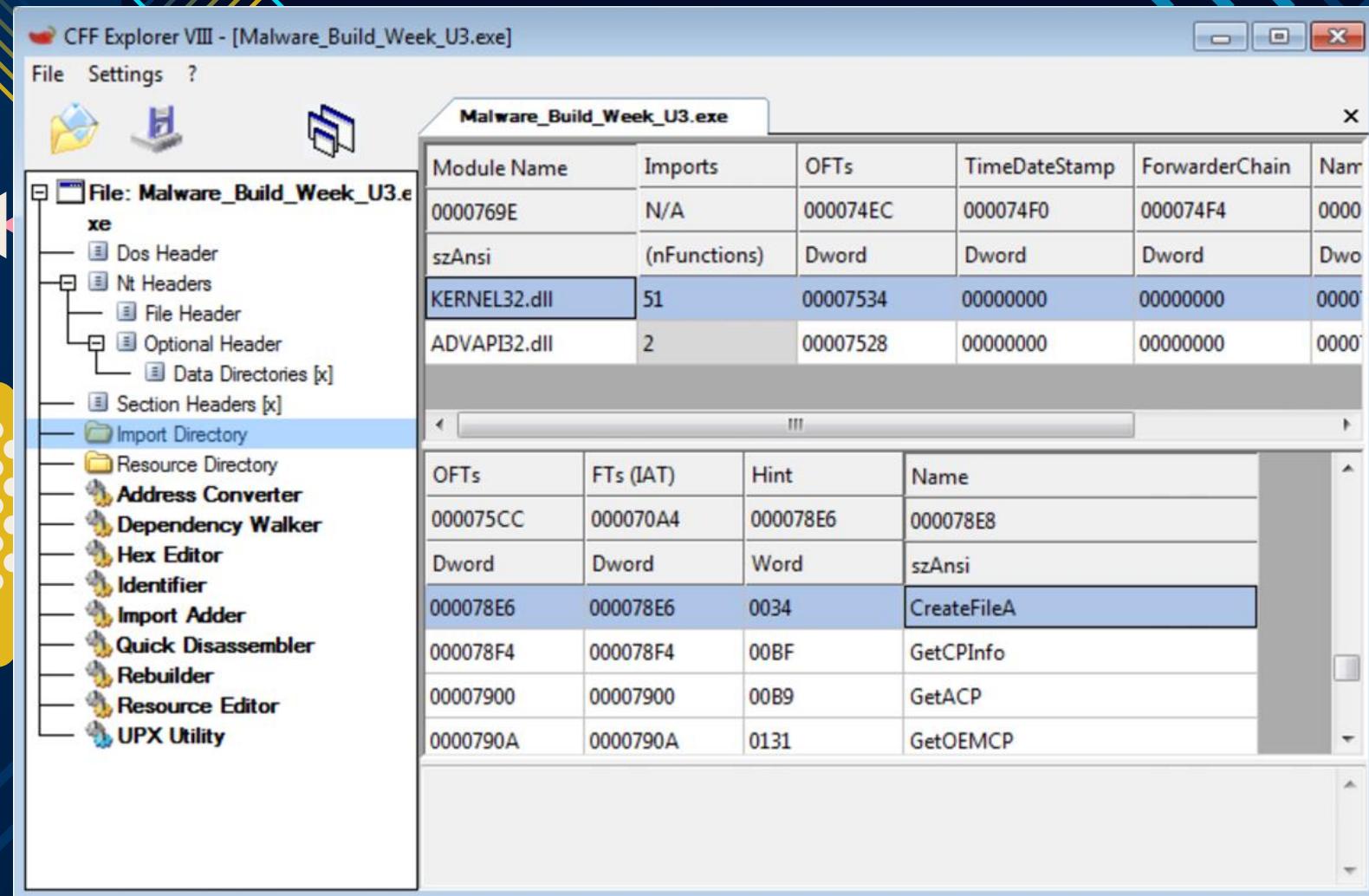
Showing 56 of 82.454 events (0.0%)

Backed by virtual memory

# PROCMON



# Attività malware



# Attività malware

Dopo aver effettuato un confronto con **Regshot**, posso vedere che sono stati modificati un totale di 113 elementi. Tra questi, ho notato che sono state aggiunte 10 nuove chiavi di registro, mentre una chiave è stata cancellata. Inoltre, sono stati aggiunti 63 nuovi valori e 5 valori sono stati cancellati. Infine, 34 valori esistenti sono stati modificati. Questi cambiamenti indicano un'ampia gamma di modifiche apportate dal malware al sistema, evidenziando l'entità del suo impatto sulla configurazione del registro di sistema.



# REGSHOT

File Macchina Visualizza Inserimento Dispositivi Aiuto

~res-x64 - Blocco note

File Modifica Formato Visualizza ?

Regshot 1.9.0 x64 Unicode

Comments:

Datetime: 2024/4/23 16:44:30 , 2024/4/23 16:49:02

Computer: USER-PC , USER-PC

Username: user , user

-----

Keys deleted: 1

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

-----

Keys added: 10

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Explorer\ComDlg32\openSav

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Ba

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Ba

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Ba

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Ba

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Ba

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Bags\18\Com

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Bags\18\Com

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Bags\90\Com

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\shell\Bags\90\Com

-----

values deleted: 5

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

HKU\S-1-5-21-3771313050-58705377-3452663501-1001\Software\Microsoft\windows\CurrentVersion\Internet Settings\5.0\cac

-----

Values added: 63

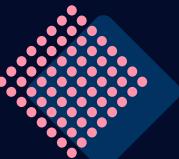
HKLM\SYSTEM\ControlSet001\services\SharedAccess\Parameters\FirewallPolicy\FirewallRules\TCP Query User{73663BDD-24D5}

HKLM\SYSTEM\ControlSet001\services\SharedAccess\Parameters\FirewallPolicy\FirewallRules\UDP Query User{9D6BBAA7-4CB9}

HKLM\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\FirewallPolicy\FirewallRules\TCP Query User{73663BDD-50}

HKLM\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\FirewallPolicy\FirewallRules\UDP Query User{9D6BBAA7-50}

# REGSHOT

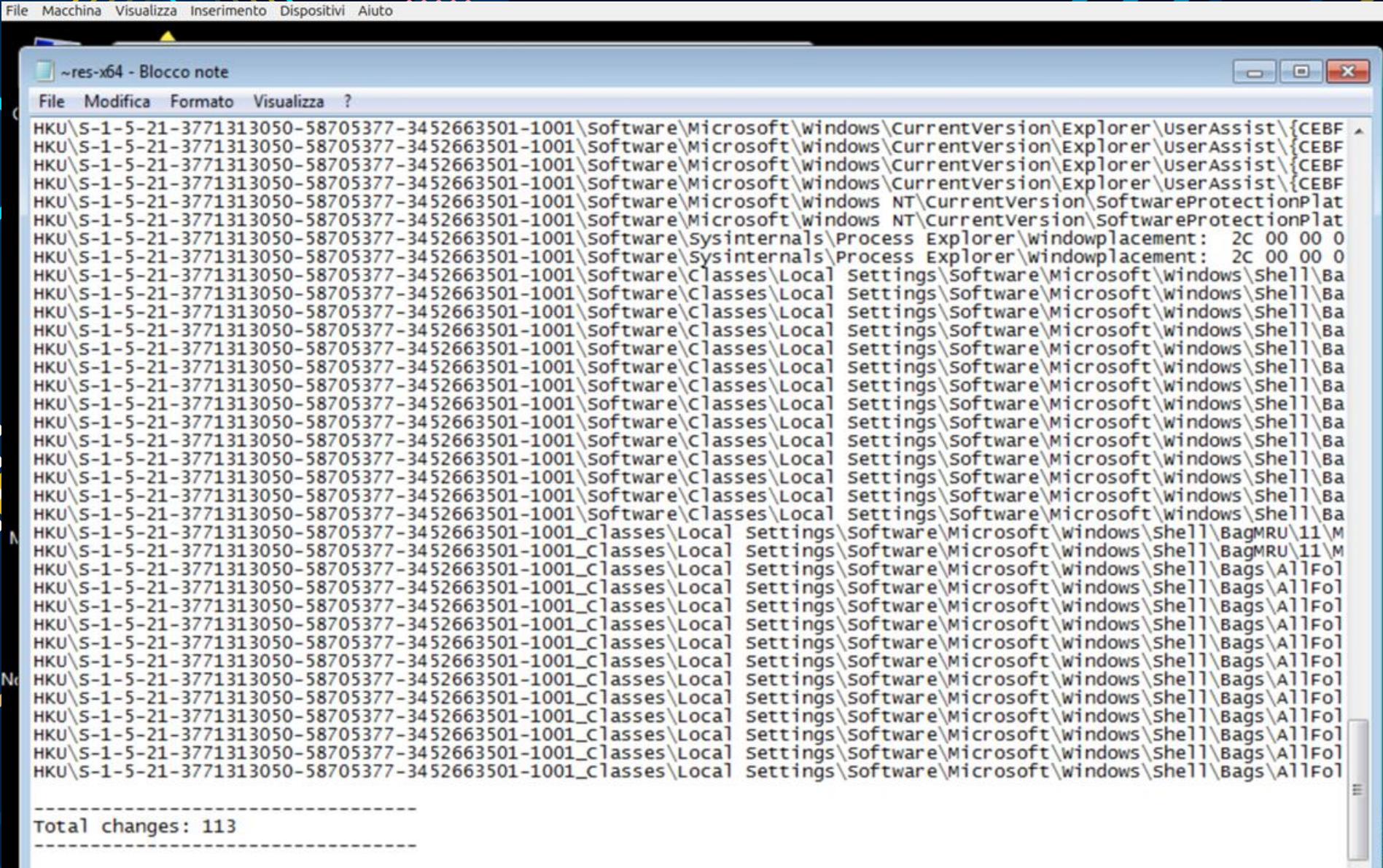


File Macchina Visualizza Inserimento Dispositivi Aiuto

values modified: 34

```
HKLM\SOFTWARE\Microsoft\SystemCertificates\AuthRoot\certificates\4EB6D578499B1CCF5F581EAD56BE3D9B6744A5E5\Blob: 03  
02 55 53 31 17 30 15 06 03 55 04 0A 13 0E 56 65 72 69 53 69 67 6E 2C 20 49 6E 63 2E 31 1F 30 1D 06 03 55 04 0B 13 1  
2D 20 46 6F 72 20 61 75 74 68 6F 72 69 7A 65 64 20 75 73 65 20 6F 6E 6C 79 31 45 30 43 06 03 55 04 03 13 3C 56 65 72  
5 04 49 E4 8D 63 47 88 3C 69 83 CB FE 47 BD 2B 7E 4F C5 95 AE 0E 9D D4 D1 43 C0 67 73 E3 14 08 7E E5 3F 9F 73 B8 33  
40 18 B9 F8 C1 ED DF DB 41 AA E0 96 20 C9 CD 64 15 38 81 C9 94 EE A2 84 29 0B 13 6F 8E DB 0C DD 25 02 DB A4 8B 19 4  
HKLM\SOFTWARE\Microsoft\SystemCertificates\AuthRoot\certificates\4EB6D578499B1CCF5F581EAD56BE3D9B6744A5E5\Blob: 19
```

# REGSHOT



# Conclusioni finali

In conclusione, sommando i dati ottenuti dall'analisi statica e da quella dinamica, posso evincere che:

- Il malware è un **Trojan/Dropper**.
- All'avvio, genera un file "sporco" nella cartella del suo eseguibile.
- Cerca di eseguire una **privilege escalation** per ottenere privilegi amministrativi.
- Ottiene persistenza (esecuzione ad ogni avvio del sistema operativo Windows) modificando una chiave di **msgina32.dll**, responsabile del login nel contesto del processo **Winlogon**.
- Evade le difese infiltrandosi in un processo comune.
- Può ottenere l'accesso alle credenziali degli utenti.
- Può raccogliere dati, che salva in un file chiamato **msutil32.sys** (la sua presenza può essere verificata con l'utilità **Strings**), e monitorare le utenze.

# Attività malware



# GRAZIE!

Simone Cisbaglia

Cyber Security Analyst

Malware Analysis