

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

8/2/2024

# W24D4

Progetto mese 6

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Michele Garzoni

## Malware Analysis

Il Malware da analizzare è nella cartella Build\_Week\_Unit\_3 presente sul desktop della macchina virtuale dedicata.

### Analisi statica

Con riferimento al file eseguibile Malware\_Build\_Week\_U3, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

- Quanti parametri sono passati alla funzione Main()?
- Quante variabili sono dichiarate all'interno della funzione Main()?
- Quali sezioni sono presenti all'interno del file eseguibile? Descrivete brevemente almeno 2 di quelle identificate
- Quali librerie importa il Malware? Per ognuna delle librerie importate, fate delle ipotesi sulla base della sola analisi statica delle funzionalità che il Malware potrebbe implementare. Utilizzate le funzioni che sono richiamate all'interno delle librerie per supportare le vostre ipotesi.

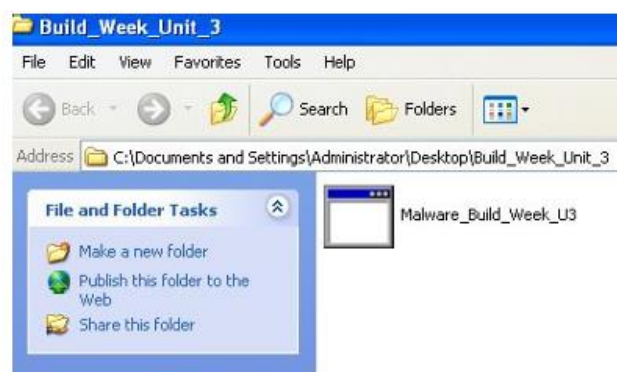
Con riferimento al Malware in analisi, spiegare:

- ☐ Lo scopo della funzione chiamata alla locazione di memoria **00401021**
- ☐ Come vengono passati i parametri alla funzione alla locazione **00401021**;
- ☐ Che oggetto rappresenta il parametro alla locazione **00401017**
- ☐ Il significato delle istruzioni comprese tra gli indirizzi **00401027** e **00401029**.
- ☐ Con riferimento all'ultimo quesito, tradurre il codice Assembly nel corrispondente costrutto C.
- ☐ Valutate ora la chiamata alla locazione **00401047**, qual è il valore del parametro «ValueName»?

## Malware Analysis

### Analisi dinamica

Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile



## Malware Analysis

- Cosa notate all'interno della cartella dove è situato l'eseguibile del Malware? Spiegate cosa è avvenuto, unendo le evidenze che avete raccolto finora per rispondere alla domanda

Analizzate ora i risultati di Process Monitor (consiglio: utilizzate il filtro come in figura sotto per estrarre solo le modifiche apportate al sistema da parte del Malware). Fate click su «ADD» poi su «Apply» come abbiamo visto nella lezione teorica.



## Malware Analysis

Filtrate includendo solamente l'attività sul registro di Windows.

- Quale chiave di registro viene creata?
- Quale valore viene associato alla chiave di registro creata?

Passate ora alla visualizzazione dell'attività sul file system.

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

Unite tutte le informazioni raccolte fin qui sia dall'analisi statica che dall'analisi dinamica per delineare il funzionamento del Malware.

L'esecuzione dell'esercizio richiede la combinazione di varie analisi viste durante il modulo 6 del corso.

Per questa prima parte servirà l'applicazione dell'analisi statica basilica ed avanzata.

L'analisi statica si riferisce all'ispezione del codice sorgente o del codice binario di un programma (in questo caso, un malware) per identificarne la funzionalità, le caratteristiche e le potenziali minacce senza eseguirlo, questo approccio si contrappone all'analisi dinamica, dove il codice viene eseguito in un ambiente controllato (sandbox) per osservare il suo comportamento.

L'analisi statica basilica consiste nell'esaminare un eseguibile senza vedere le istruzioni che lo compongono e la sua funzione è confermare se un dato file è malevolo e fornire informazioni generiche circa le sue funzionalità. L'analisi statica avanzata presuppone la conoscenza dei fondamenti di «reverse-engineering» al fine di identificare il comportamento di un malware a partire dall'analisi delle istruzioni che lo compongono. Questo passaggio è essenziale per capire esattamente cosa fa il malware a livello di istruzioni della cpu.

Si possono inoltre estrarre stringhe di testo, url, chiavi di cifratura, e altre risorse dal codice del malware, che possono indicare il suo comportamento o intento e se ne può esaminare il codice relativo alla rete per comprendere come il malware comunica.

Prima di procedere all'analisi mi assicuro che sia di fatto un malware, estraendone l'hash con md5deep e controllando su virustotal la sua reputazione che si basa su vari riscontri di software antivirus.

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++, analizzato l'ultima volta in data 17 aprile 2024. è un malware progettato per colpire la macchina intel 386 e processori successivi/compatibili. ha 5255 entry points, 4 sezioni ed importa 2 librerie:

**kernel32.dll** - una delle librerie fondamentali di windows, contiene numerose funzioni che gestiscono la memoria, i processi e i thread. i malware la utilizzano per manipolare i processi e per accedere a diverse api di sistema ed ottenere persistenza.

**advapi32.dll** - fornisce funzioni relative alla sicurezza e alla gestione di account, che i malware possono sfruttare per modificare permessi, accedere a token di sicurezza e alterare il registro di sistema, ad esempio per essere avviati all'avvio del sistema operativo. andrò a confermare tutti questi dati tramite tool dedicati come cffexplorer/exeinfope e ida pro.

```
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3>cd md5deep-4.3
C:\Users\user\Desktop\Software Malware analysis\md5deep-4.3\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
```

57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d12e5ba0384056d7

52 / 71

Community Score

52/71 security vendors and no sandboxes flagged this file as malicious

Reanalyze

Similar

More

57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d...

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

genericrcrq

Security vendors' analysis

Do you want to automate checks?

AhnLab-V3	Trojan.Win32.Agent.C39204	Alibaba	Trojan:Win32/Totbrick.db39e83f
AliCloud	Backdoor	ALYac	Gen:Variant.Doina.65814
Antiy-AVL	Trojan.Win32.Agent	Arcabit	Trojan.Doina.D10116
Avast	Win32:Trojan-gen	AVG	Win32:Trojan-gen
Avira (no cloud)	TR/Agent.53248.465	BitDefender	Gen:Variant.Doina.65814
BitDefenderTheta	Gen:NN.ZedlaF.36802.aq4@a0clrOb	Bkav Pro	W32.AIDetectMalware
ClamAV	Win.Trojan.Agent-595082	CrowdStrike Falcon	Win/malicious_confidence_100% (W)
Cylance	Unsafe	Cynet	Malicious (score: 99)

57d8d248a8741176348b5d12dcf29f34c8f48ede0ca13c30d12e5ba0384056d7

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  
id: 0xe, version: 7299 count=15  
id: 0x13, version: 8034 count=5  
id: 0x15, version: 9782 count=1

Header

Target MachineIntel 386 or later processors and compatible processors

Compilation Timestamp2011-11-06 18:55:06 UTC

Entry Point5255

Contained Sections4

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

BINARY1

Contained Resources By Language

# CFF EXPLORER

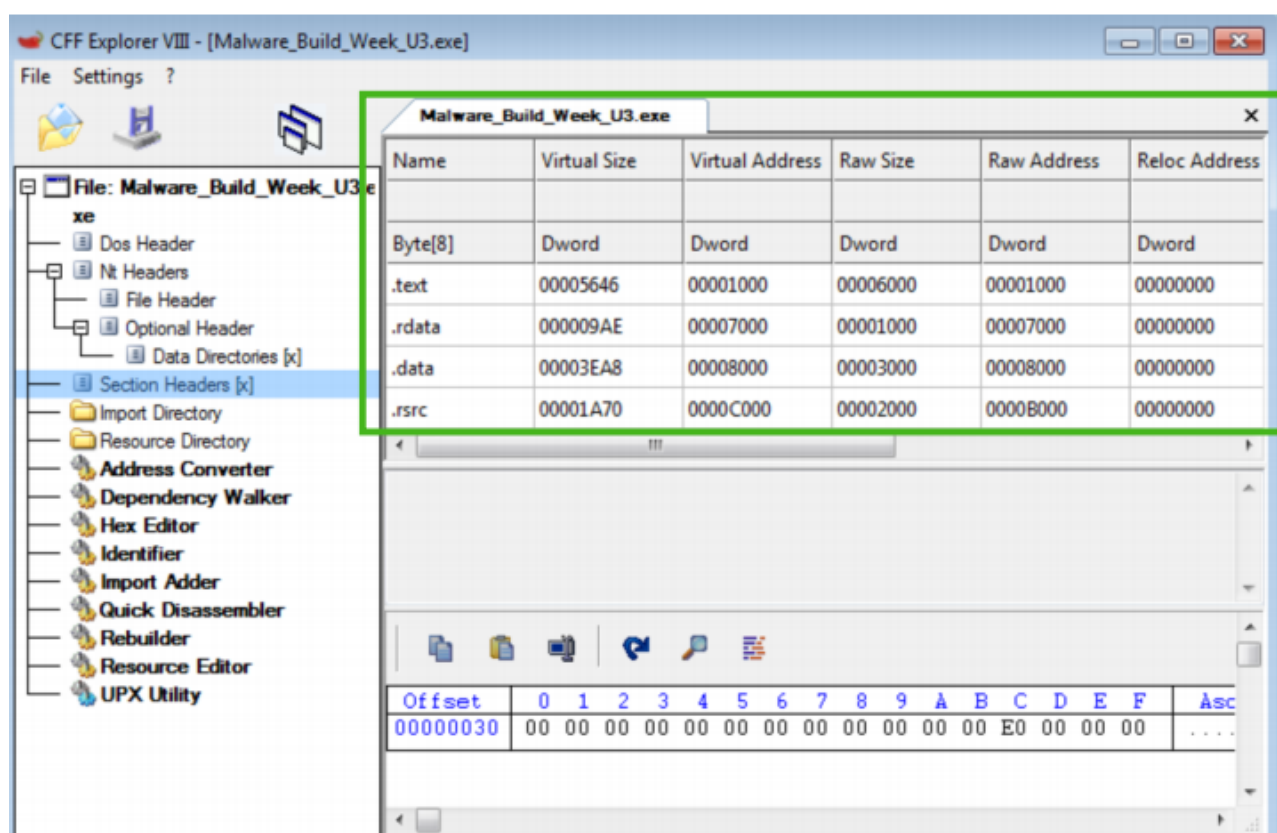
Per controllare le funzioni importate ed esportate da un malware, si può utilizzare cff explorer, un tool da installare sulle macchine virtuali dedicate all'analisi dei malware. Se mi sposto su «import directory» posso controllare le librerie e le funzioni importate, mentre su «section headers» posso vedere le sezioni presenti all'interno del file eseguibile:

.text - contiene le righe di codice, istruzioni, che la cpu eseguirà all'avvio del malware.

.rdata - sezione che include le informazioni circa le librerie e le funzioni importate ed esportate dall'eseguibile, dati che il programma legge mentre è in funzione.

.data - contiene i dati / le variabili globali del program-ma eseguibile, che devono essere disponibili da qualsiasi parte del programma e possono essere modificati.

.rsrc - include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.



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

File Settings ?

File: Malware\_Build\_Week\_U3.exe

- File: Malware\_Build\_Week\_U3.exe
  - 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

Malware\_Build\_Week\_U3.exe

Module Name	Imports	OFTs	TimeDateStamp	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 VIII - [Malware\_Build\_Week\_U3.exe]**

File Settings ?

**Malware\_Build\_Week\_U3.exe**

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 VIII - [Malware\_Build\_Week\_U3.exe]**

File Settings ?

**Malware\_Build\_Week\_U3.exe**

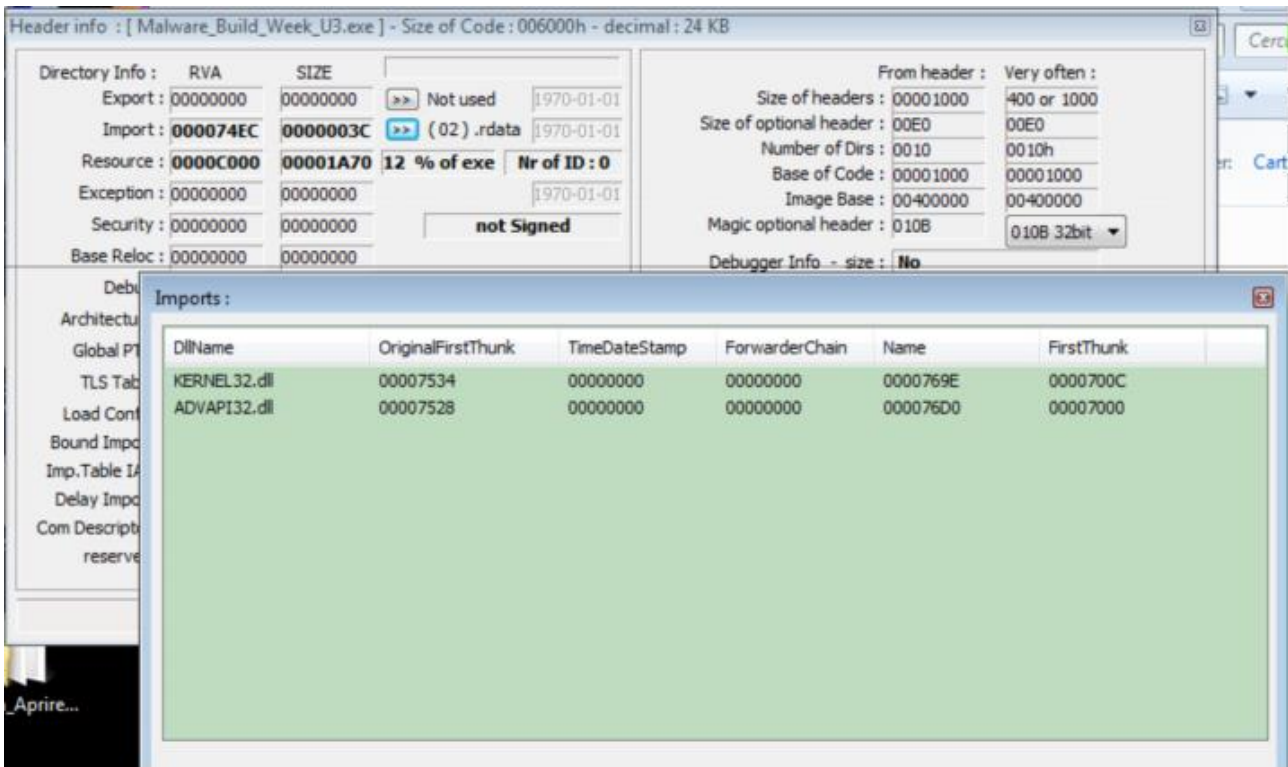
Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000076D0	N/A	00007500	00007504	00007508	0000750C	00007510
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
KERNEL32.dll	51	00007534	00000000	00000000	0000769E	0000700C
ADVAPI32.dll	2	00007528	00000000	00000000	000076D0	00007000

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

Per quel che riguarda le due librerie importate, che sono kernel32.dll (1) e advapi32.dll (2), posso ipotizzare che il malware cerchi di ottenere persistenza e modificare le chiavi di registro (regsetvalueexa/regcreatekeyexa) per poter essere eseguito in autonomia. Inoltre la presenza di funzioni come sizeofresource/lockresource/loadresource/findresourcea fa presupporre che sia un dropper, cioè un malware che contiene al suo interno un altro malware, ed utilizza queste api che permettono di localizzare all'interno della sezione «risorse» il malware da estrarre, e successivamente da caricare in memoria per l'esecuzione.



Si confermano gli stessi dati di CFF



# IDAPRO

Ora per informazioni su variabili locali e parametri della funzione main (), userò ida pro (interactive disassembler professional) che è uno strumento avanzato di reverse engineering software che offre capacità di disassemblaggio, debugging e analisi statica.

**Variabili locali:** le variabili locali in una funzione assembly sono tipicamente allocate nello stack. Questo è spesso fatto attraverso istruzioni di tipo push all'inizio di una funzione o con un'istruzione sub che aumenta il puntatore dello stack (sp) per creare spazio.

**Parametri:** solitamente i parametri sono passati ai regi-stri o attraverso l'uso dello stack prima della chiama-ta della funzione. I commenti nel codice possono fornir-e indicazioni su dove e quanti parametri sono passati.

Nell'assembly, le etichette che iniziano con var\_ tendono a indicare variabili locali, mentre quelle che iniziano con arg\_ indicano argomenti o parametri passati alla funzione.

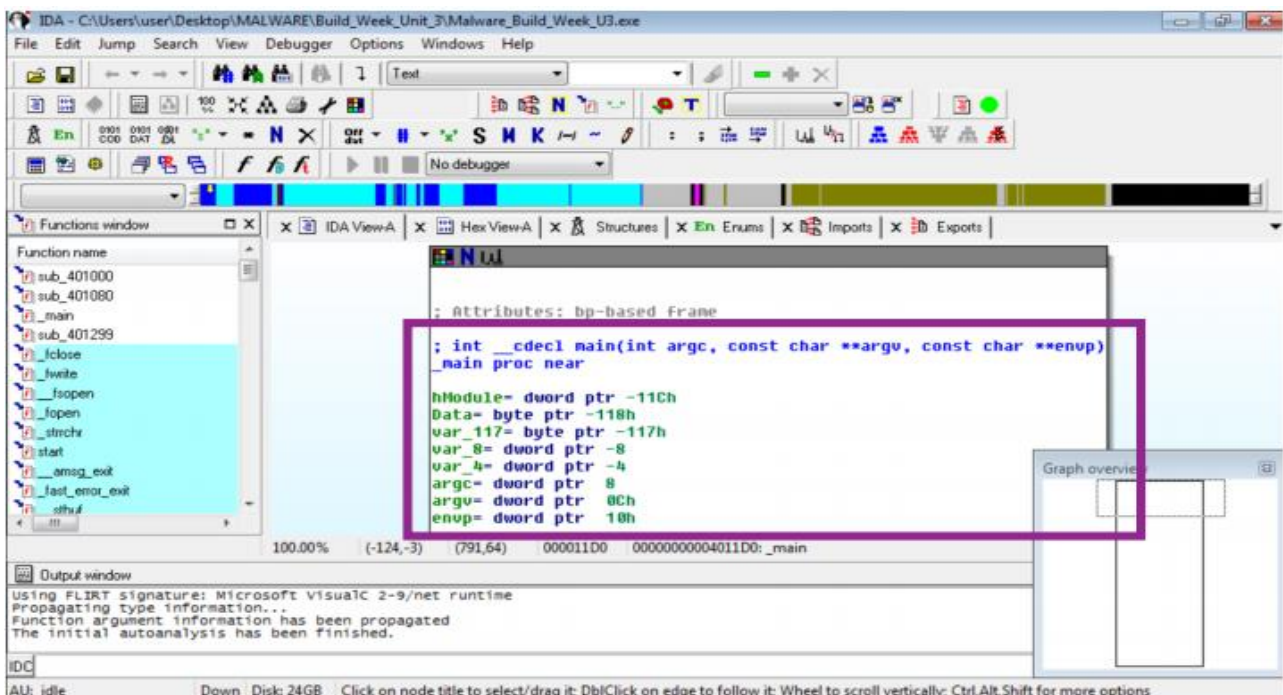
**Valori offset:** gli offset (come var\_54h) sono utilizzati per accedere a dati specifici sullo stack. Gli offset negativi rispetto all'indirizzo base del frame della funzione (ad esempio ebp su x86) di solito indicano variabili locali, mentre gli offset positivi indicano parametri.

In questo caso le variabili locali in evidenza che possiamo rilevare ad un'occhiata sono 5:

1. hmodule
2. data
3. var\_117
4. var\_8
5. var\_4.

Mentre i parametri sono 3

1. argc
2. argv
3. envp.



Proseguendo con l'analisi statica del codice del malware grazie all'utilizzo di ida pro, troviamo alla locazione di memoria 00401021 la funzione regcreatekeyexa, che è una funzione nota che fa parte delle api di windows che permettono alle applicazioni di interagire con il registro di windows.

Queste funzioni sono utilizzate per creare nuove chiavi di registro o per aprire chiavi esistenti per modificare i loro valori. Nello specifico il malware può utilizzare regcreatekeyexa per ottenere persistenza creando nuove chiavi di registro o modificando chiavi esistenti ed assicurandosi che il codice malevolo venga eseguito ogni volta che il sistema viene avviato; per esempio, potrebbe aggiungere una voce nella chiave run per eseguire automaticamente il malware all'avvio del sistema.

Per quel che riguarda il metodo in cui vengono passati i parametri alla suddetta funzione, nel modulo si è visto che la convenzione di chiamata più comune in molti sistemi operativi quando si utilizza l'architettura x86 è «pushare» i parametri sullo stack prima della chiamata (call) alla funzione regcreatekeyexa. I parametri verranno letti dalla funzione in ordine inverso, quindi a partire da hkey per finire con lpdwddisposition.

```
.text:00401017      push     offset SubKey ; "SOFTWARE\\Microsoft\\Windows NT\\Cu
```

Troviamo alla locazione di memoria 00401017 - push offset subkey - l'indirizzo della sottochiave di registro che viene spinto nello stack. Il nome vicino al codice indica che potrebbe essere il nome della sottochiave che verrà creata dal processo hkey in winlogon -> componente del sistema operativo windows che si occupa della gestione della sessione di accesso (login) e disconnessione (logout) degli utenti.

```
.text:00401027      test     eax, eax
.text:00401029      jz       short loc_401032
```

Per quel che riguarda il significato delle istruzioni comprese tra gli indirizzi 00401027 e 00401029 vediamo:

Un'istruzione condizionale test, che è simile all'istruzione and logico bit a bit, ma non va a modificare il contenuto degli operandi (cioè eax e se stesso). Modifica invece il flag zf (zero flag) del registro eflags, che viene settato ad 1 se e solo se il risultato dell'and è 0. ( $0 \text{ and } 1 = 0 * 1 = 0 \rightarrow \text{zero flag} = 1$ ). Viene di fatto 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 ad una determinata locazione di memoria (in questo caso 00401032) se zf è pari a uno. Ed a meno che il valore contenuto nel registro eax non sia 0, il salto non avverrà.

```

.text:00401032 loc_401032:                ; CODE XREF: sub_401000+29fj
.text:00401032                mov     ecx, [ebp+cbData]

```

Questa operazione equivale ad un ciclo if in c come quello seguente (qui, eax rappresenta una variabile in c che contiene il valore che era nel registro eax):

```

if (eax==0)
{
// Vai a loc_401032
}
else
{
// Riprova a fare un'operazione (ad esempio..)
}

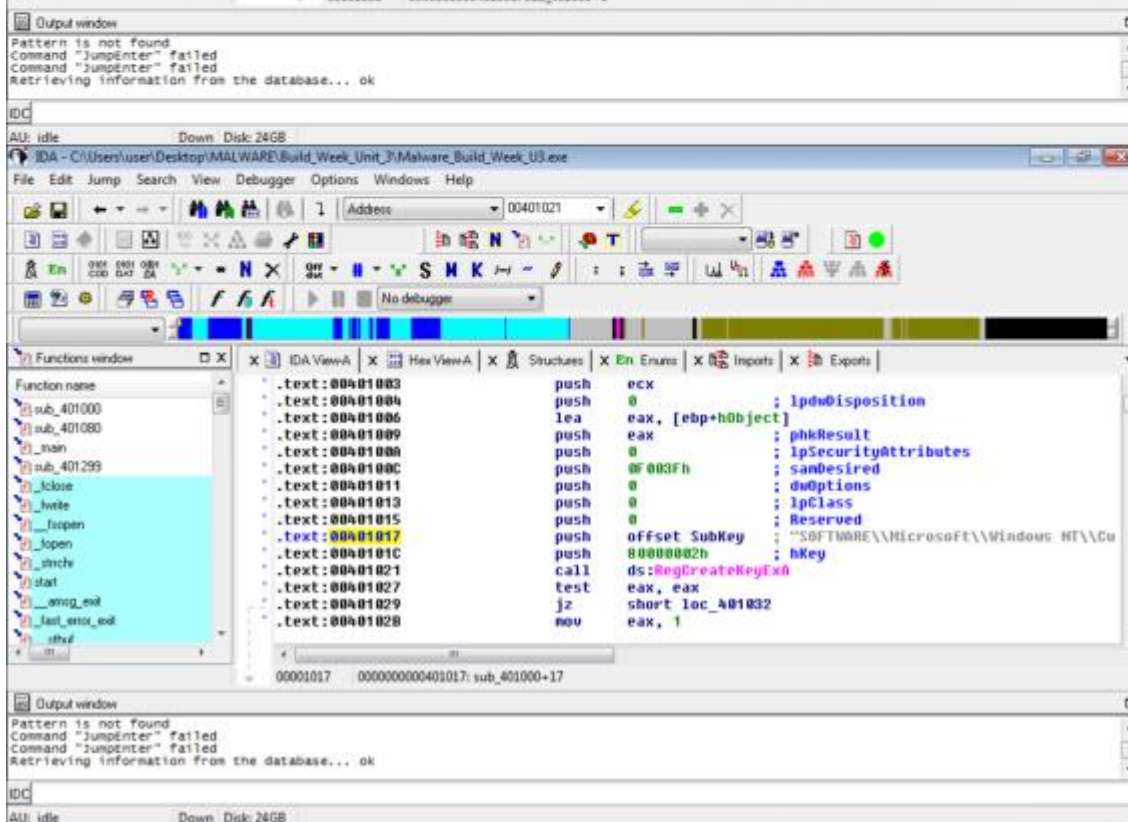
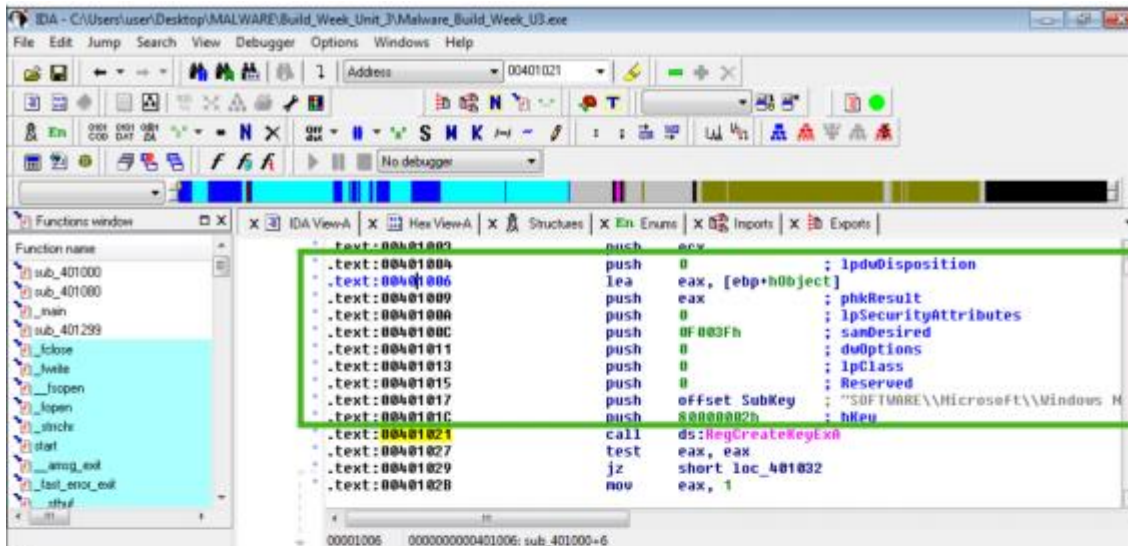
```

```

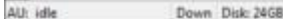
.text:0040103E                push    offset ValueName ; "GinaDLL"
.text:00401043                mov     eax, [ebp+hObject]
.text:00401046                push    eax                ; hKey
.text:00401047                call   ds:RegSetValueEx

```

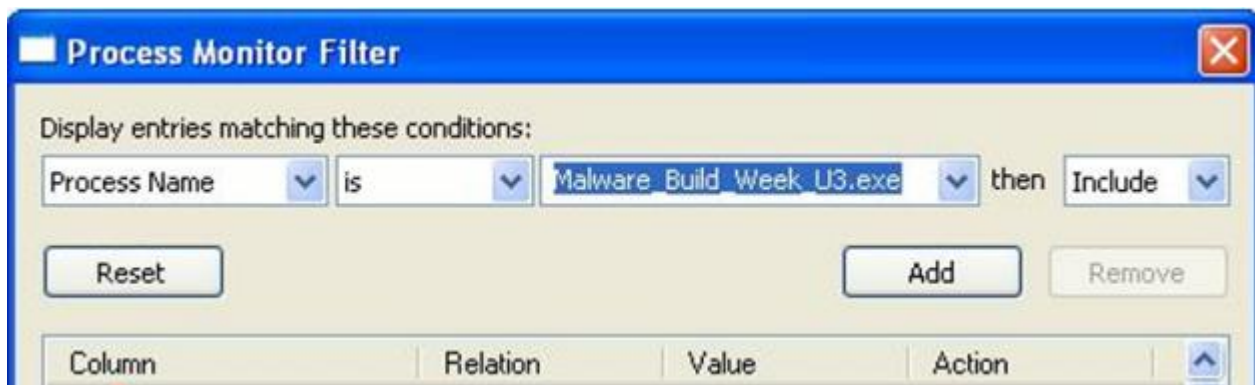
Per quel che riguarda la chiamata alla locazione 00401047 posso vedere che si tratta di una call alla funzione `regsetvalueexa`, che è una funzione dell'api di windows che imposta il valore di una voce nel registro di sistema. Il prefisso `ds:` indica che l'indirizzo della funzione è preso dal registro `ds`, che è il segmento dati. Questa funzione, quindi, impartisce istruzioni affinché `offset valuenam` abbia valore "**ginadll**" in una specifica chiave di registro identificata da `hkey`.







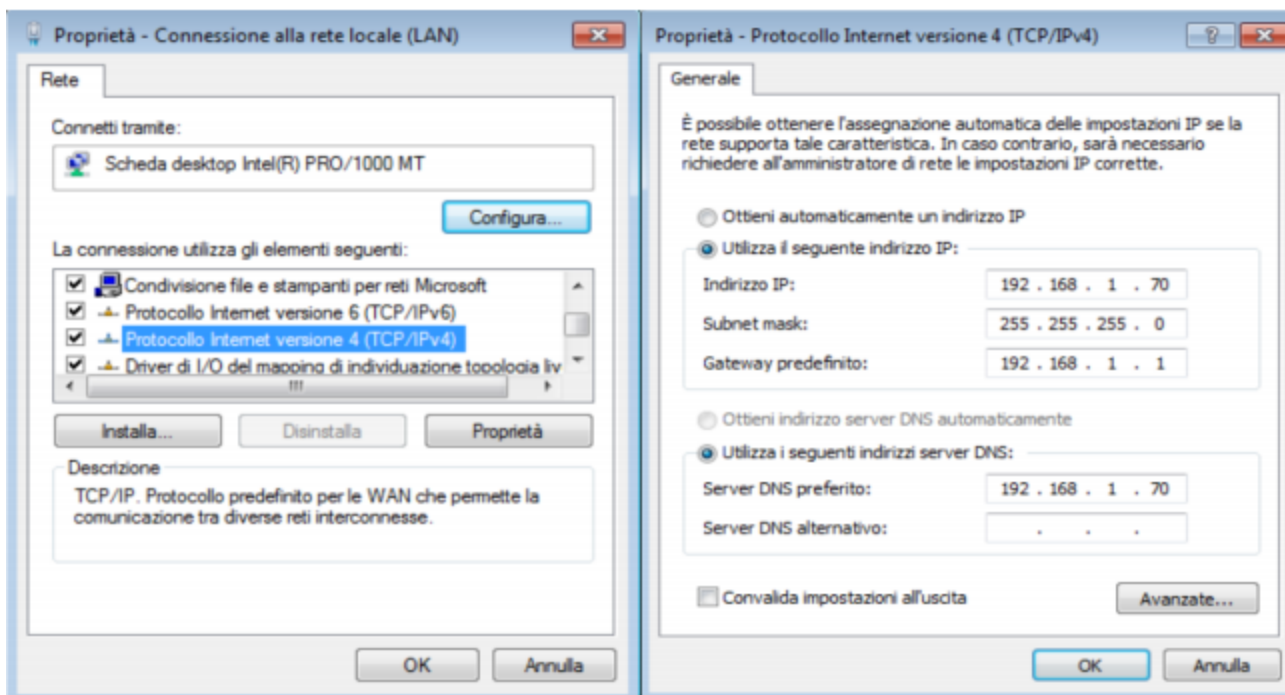
# PROCMON



Ricreo un'istantanea da virtualbox della macchina windows prima di iniziare, per poter ripristinare in caso di problemi, e visto che andrò ad eseguire il malware mi assicuro di rispettare i seguenti accorgimenti:

1. Disattivare controller usb
2. Disattivare comunicazione con la rete (solo int)
3. Disabilitare la condivisione delle cartelle
4. Disabilitare appunti condivisi (copia / incolla)

Prima di aprire il malware, mi assicuro di andare a catturare tutti gli eventi aprendo procmon, al fine di identificare eventuali azioni del malware su processi e thread, e modifiche registro; avvio regshot per confrontare tramite screenshot (prima / dopo) le modifiche che avverranno a livello di sistema, ed imposto un ip statico per vedere tramite apatedns le chiamate che il malware andrà a fare nel web (quali siti). Avendolo isolato dalla rete ovviamente non potrà eseguire tutte le sue funzioni.



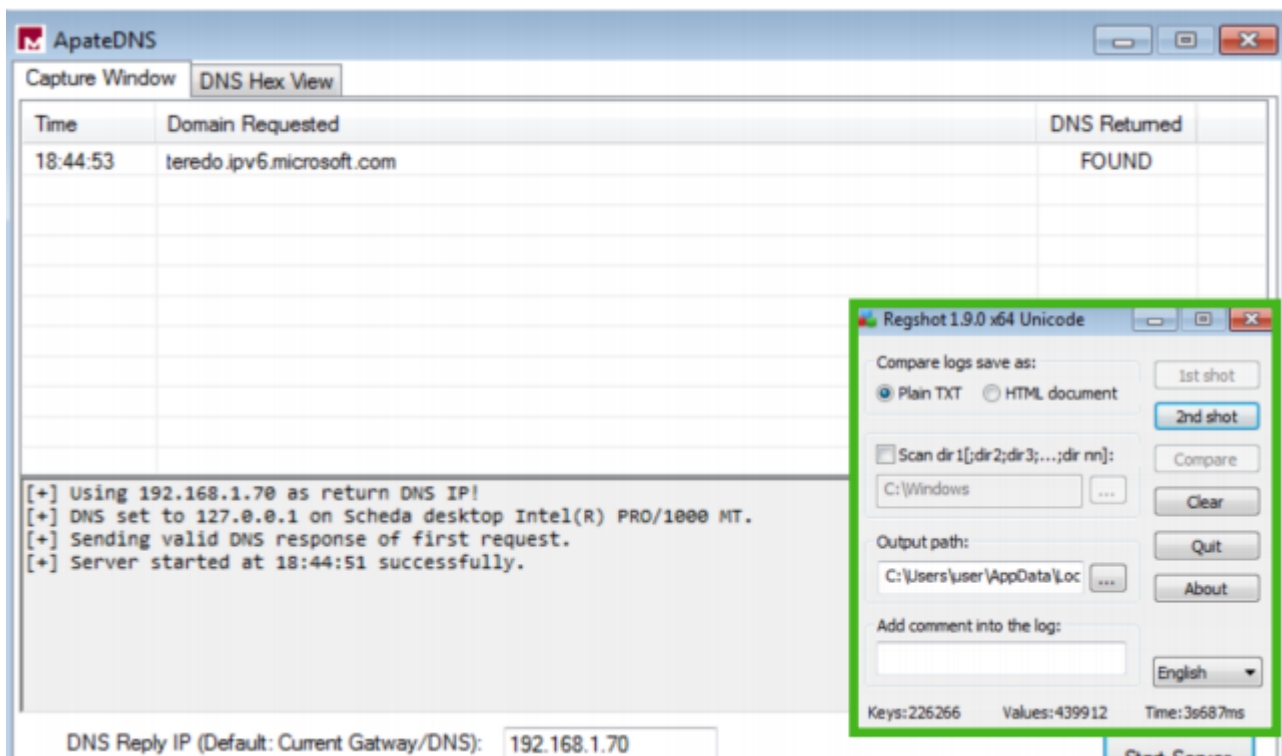
```
C:\Users\user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Connessione alla rete locale (LAN):

    Suffisso DNS specifico per connessione:
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::e055:3d47:f6e7:dfee%11
    Indirizzo IPv4. . . . . : 192.168.1.70
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.1.1

Scheda Tunnel isatap.{9AC0F9A2-BF82-47F4-8F6D-C16B1D79E059}:
```





Di seguito alcune delle chiamate intercettate:

The screenshot displays the ApateDNS application interface. The main window is titled 'ApateDNS' and has two tabs: 'Capture Window' and 'DNS Hex View'. The 'Capture Window' tab is active, showing a list of intercepted DNS requests. The table has three columns: 'Time', 'Domain Requested', and 'DNS Returned'. The data shows various domains being requested, including 'teredo.ipv6.microsoft.com', 'ocsp.digicert.com', 'www.download.windowsupdate.com', 'ocsp.usertrust.com', 'ocsp.sectigo.com', 'ocsp.comodoca.com', 'spynet2.microsoft.com', and 'teredo.ipv6.microsoft.com'. The 'DNS Returned' column shows 'FOUND' for all requests.

Below the table, there is a status bar with the following text:

```
[+] 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.
```

At the bottom of the window, there are input fields for 'DNS Reply IP (Default: Current Gateway/DNS): 192.168.1.70', '# of NXDOMAIN's: 0', and 'Selected Interface: Scheda desktop Intel(R) PRO/1000 MT'. There are also 'Start Server' and 'Stop Server' buttons.

On the right side of the window, there is a panel titled 'Contacted Domains (6)' and 'Contacted IP addresses (12)'. The 'Contacted Domains' panel shows a table with columns: 'Domain', 'Detections', 'Created', and 'Registrar'. The 'Contacted IP addresses' panel shows a table with columns: 'IP', 'Detections', 'Autonomous System', and 'Country'.

Domain	Detections	Created	Registrar
adobe.com	1 / 90	1996-11-17	NOR-IQ Ltd dba Com Laude
arnet.adobe.com	1 / 90	1996-11-17	NOR-IQ Ltd dba Com Laude
cr13.digicert.com	0 / 90	1996-12-02	MarkMonitor Inc.
cr14.digicert.com	0 / 90	1996-12-02	MarkMonitor Inc.
digicert.com	0 / 90	1996-12-02	MarkMonitor Inc.
ocsp.digicert.com	0 / 90	1996-12-02	MarkMonitor Inc.

IP	Detections	Autonomous System	Country
104.85.240.187	0 / 90	16025	US
114.114.114.114	2 / 90	21809	CN
117.18.237.29	1 / 90	15133	US
192.229.211.108	3 / 90	15133	US
20.80.129.13	0 / 90	8075	US
20.96.52.196	0 / 90	8075	US
20.99.133.109	1 / 90	8075	US
20.99.184.37	1 / 90	8075	US
20.99.185.48	0 / 90	8075	US
20.99.188.146	0 / 90	8075	US

Vediamo 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") che gestisce il processo di accesso, e nello specifico l'interfaccia utente di logon interattiva, che include la schermata di accesso classica di inserimento nome utente / 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. Ed è proprio andando a modificare questo file che il malware ottiene persistenza all'avvio del sistema operativo.

Filtrando per path su procmon vediamo la creazione del file nella cartella scelta tramite processo "createfile".



Filtrate includendo solamente l'attività sul registro di Windows.

- Quale chiave di registro viene creata?
- Quale valore viene associato alla chiave di registro creata?

Passate ora alla visualizzazione dell'attività sul file system.

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

Unite tutte le informazioni raccolte fin qui sia dall'analisi statica che dall'analisi dinamica per delineare il funzionamento del Malware.

The screenshot displays the Process Monitor (ProcMon) application window. The main pane shows a list of events, with the 'Process Name' column filtered to show only 'Malware\_Build\_Week\_U3.exe'. The 'Operation' column is filtered to show only 'RegSetValue'. The 'Path' column shows the registry path 'HKEY\_CURRENT\_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon'. The 'Result' column shows 'SUCCESS'. The 'Detail' column shows 'Desired Access: Query Value'. The 'Data' column shows 'malevol0'. The 'Event Properties' window is open on the right, showing the details of the selected event. The 'Process' tab is selected, showing the process name 'Malware\_Build\_Week\_U3.exe' and the path 'C:\Users\User\Desktop\MALWARE\_Build\_Week\_U3.exe'. The 'Operation' tab is also visible, showing the operation 'RegSetValue' and the path 'HKEY\_CURRENT\_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon'. The 'Data' field in the 'Operation' tab is highlighted with a green box, showing the value 'malevol0'.

Filtrando per chiavi di registro si può notare che ad un certo punto viene passata la funzione regsetvalue che è una funzione tipica della libreria importata dal malware advapi32.dll che, come detto in precedenza, fornisce funzioni relative alla sicurezza e alla gestione di 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, sempre come visto in precedenza, viene assegnato il valore di gina.dll. Questo valore "malevol0" verrà sovrascritto nella libreria msgina32.dll andandone sostituire la copia "sana". Il malware in tal modo potrà avviarsi ad ogni login nel contesto del processo winlogon.

Passate ora alla visualizzazione dell'attività sul file system.

- Quale chiamata di sistema ha modificato il contenuto della cartella dove è presente l'eseguibile del Malware?

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



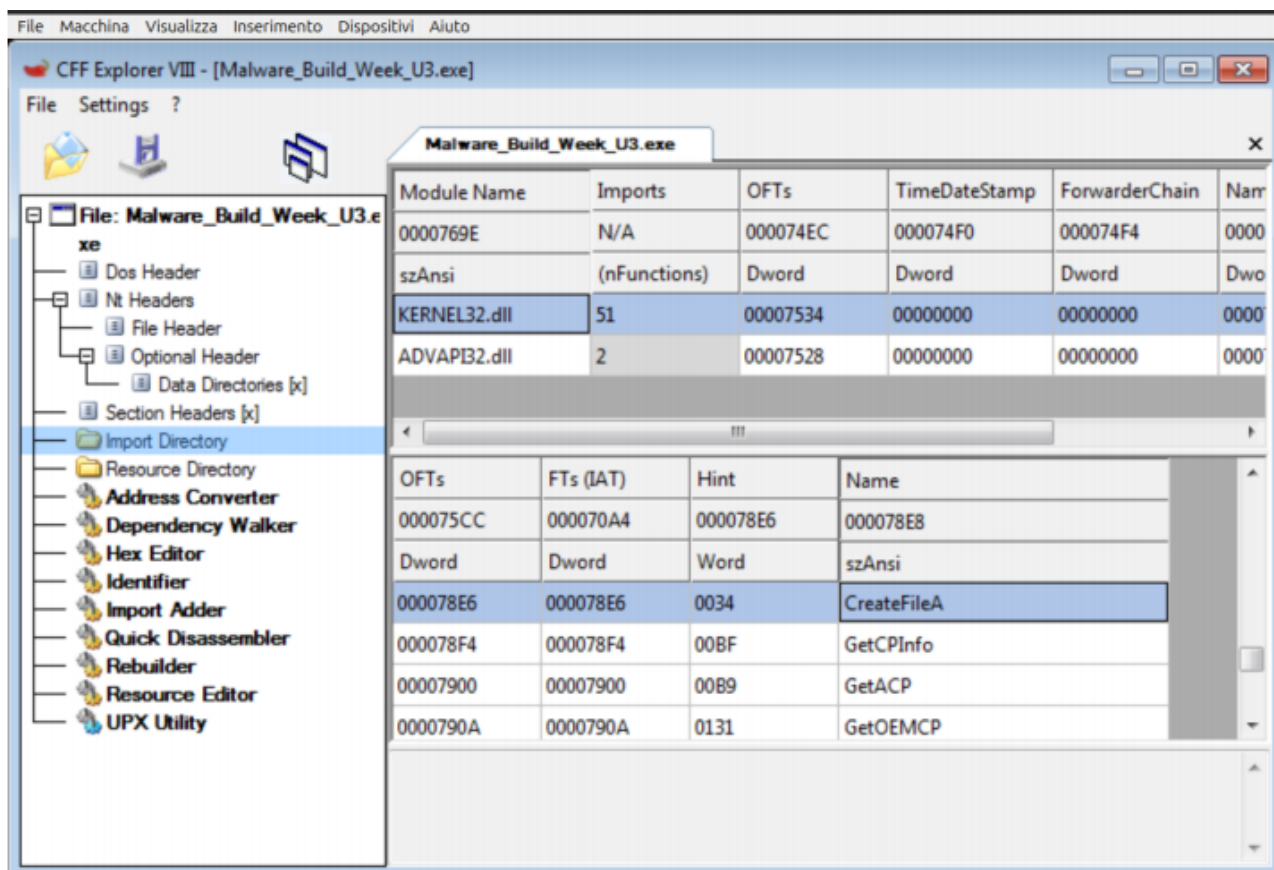
Process Name	PID	Operation	Path	Result	Detail
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows	SUCCESS	Name: \Windows
Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows	SUCCESS	
Malware_Build_Week_U3.exe	2496	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3	SUCCESS	Desired 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: 0
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, Synchronize, Disposition: Open, Options: 0
Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll	FILE LOCK	SyncType: SyncTypeCreateSection, PageProtection: PAGE_EXECUTE_READWRITE
Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll	SUCCESS	SyncType: SyncTypeOther
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\magina32.dll	SUCCESS	Desired Access: Generic Write, Read Attributes, Disposition: Open, Options: 0
Malware_Build_Week_U3.exe	2496	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\magina32.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\magina32.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\magina32.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_Build_Week_U3.exe	SUCCESS	Name: \Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Build_Week_U3.exe
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\aspicil.dll	SUCCESS	Name: \Windows\SysWOW64\aspicil.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\vpport4.dll	SUCCESS	Name: \Windows\SysWOW64\vpport4.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\KernelBase.dll	SUCCESS	Name: \Windows\SysWOW64\KernelBase.dll
Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Name: \Windows\SysWOW64\msvcrt.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

Time	Process Name	PID	Operation	Path
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows
18:45	Malware_Build_Week_U3.exe	2496	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3
18:45	Malware_Build_Week_U3.exe	2496	CreateFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryBasicInformationFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CreateFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CreateFileMapping	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\magina32.dll
18:45	Malware_Build_Week_U3.exe	2496	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\magina32.dll
18:45	Malware_Build_Week_U3.exe	2496	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\magina32.dll
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\magina32.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\apisetschema.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\Malware_Build_Week_U3.exe
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64cpu.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64win.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\wow64.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\cryptbase.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\aspicil.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\sechost.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\vpport4.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\advapi32.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\KernelBase.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\msvcrt.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\System32\ntdll.dll
18:45	Malware_Build_Week_U3.exe	2496	QueryNameInformationFile	C:\Windows\SysWOW64\ntdll.dll
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Windows
18:45	Malware_Build_Week_U3.exe	2496	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3

Showing 56 of 82,454 events (0.0%) Backed by virtual memory

Event	Process	Stack
Frame	Module	Location
8	ntoskrnl.exe	KeSynchronizeExecution + 0x3a43
9	ntdll.dll	NtCreateFile + 0xa
10	wow64.dll	Wow64EmulateAtoiThunk + 0xe697
11	wow64.dll	Wow64SystemServiceEx + 0xd7
12	wow64cpu.dll	TurboDispatchJumpAddressEnd + 0x2d
13	wow64.dll	Wow64SystemServiceEx + 0x1ce
14	wow64.dll	Wow64LdrpInitialize + 0x429
15	ntdll.dll	RtlUniform + 0x5e6
16	ntdll.dll	RtlCreateTagHeap + 0xa7
17	ntdll.dll	LdrInitializeThunk + 0xe
18	ntdll.dll	NtCreateFile + 0x12
19	KernelBase.dll	CreateFileW + 0x35e
20	kernel32.dll	CreateFileW + 0x4a
21	kernel32.dll	CreateFileA + 0x36
22	Malware_Build_Week_U3.exe	Malware_Build_Week_U3.exe + 0x4ea4
23	Malware_Build_Week_U3.exe	Malware_Build_Week_U3.exe + 0x2929
24	Malware_Build_Week_U3.exe	Malware_Build_Week_U3.exe + 0x1446
25	Malware_Build_Week_U3.exe	Malware_Build_Week_U3.exe + 0x1214
26	Malware_Build_Week_U3.exe	Malware_Build_Week_U3.exe + 0x153b
27	kernel32.dll	BaseThreadInitThunk + 0x12
28	ntdll.dll	RtlInitializeExceptionChain + 0x63
29	ntdll.dll	RtlInitializeExceptionChain + 0x36

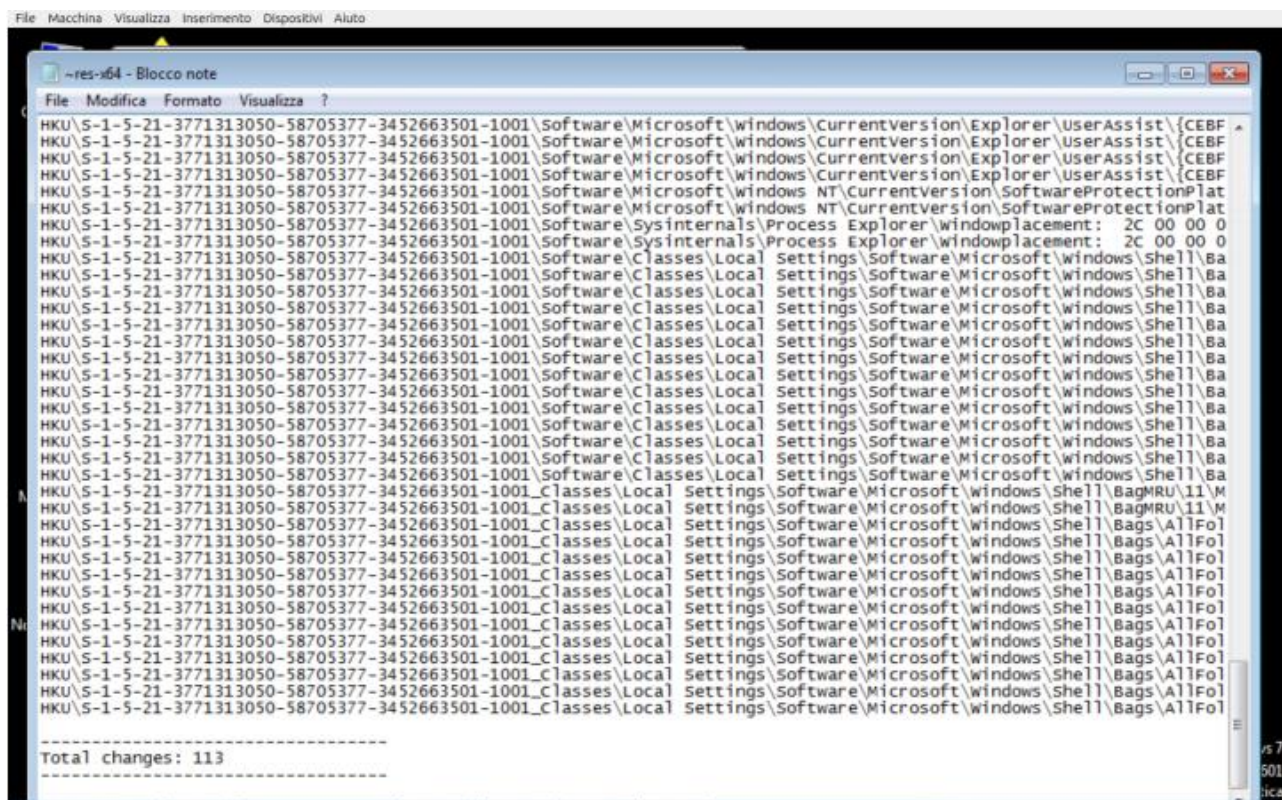


Dopo un confronto ottenuto tramite regshot posso vedere che sono stati modificati 113 elementi tra cui 10 chiavi aggiunte, 1 cancellata; valori aggiunti 63, cancellati 5, modificati 34.



File Macchina Visualizza Inserimento Dispositivi Aiuto

```
=====
value modified: 24
```







In conclusione sommando i dati ottenuti dalla analisi statica + quella dinamica posso evincere che:

- Il malware è un trojan/dropper
- All'avvio genera un file "sporco" nella cartella del suo eseguibile
- Cerca di fare un privilege escalation ed ottenere "poteri" 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 la difesa infiltrandosi in un processo comune
- Può ottenere l'accesso alle credenziali degli utenti
- Può raccogliere dati (che salva in un file chiamato msutil32.sys -> si può vederne la presenza con l'utility strings) e monitorare le utenze



## Files Written

-  C:\Documents and Settings\Administrator\Local Settings\Temp\EB93A6\msgina32.dll
-  C:\Users\user\Desktop\msgina32.dll
-  \Device\ConDrv
-  \Device\ConDrv\Connect

## Graph Summary ⓘ



Letzte

Bagno di S.

nmap -sV -p

A collage of images related to a cyber security or hacking theme. It includes a screenshot of an nmap scan output, a screenshot of a website selling steroids, a cartoon of Goofy with a safe, and the word 'Corretto'.

The nmap scan output shows a list of services available on a target IP, including IP, Sistema Operativo, Porte Aperte, Servizi in ascolto con versione, and Descrizione dei servizi. The URL <https://www.poftut.com/nmap-output/> is mentioned, along with the command `nmap -oN report1 IP`.

The steroid website screenshot shows a header 'CLENDUTEROL FOR SALE' and three main categories: 'Injectable Steroids', 'Oral Steroids', and 'HGH & Peptides'. Each category has a 'View All' button. Below these are images of various steroid products.

The cartoon image shows Goofy standing next to an open safe filled with money. He is holding a large stack of cash and a bag of money. The text '+ 3900 € GRAZIE' is written next to him.

The word 'Corretto' is written in a stylized font in the bottom right corner.

Ciao Niko! Grazie♡  
(NO MEME) Davvero