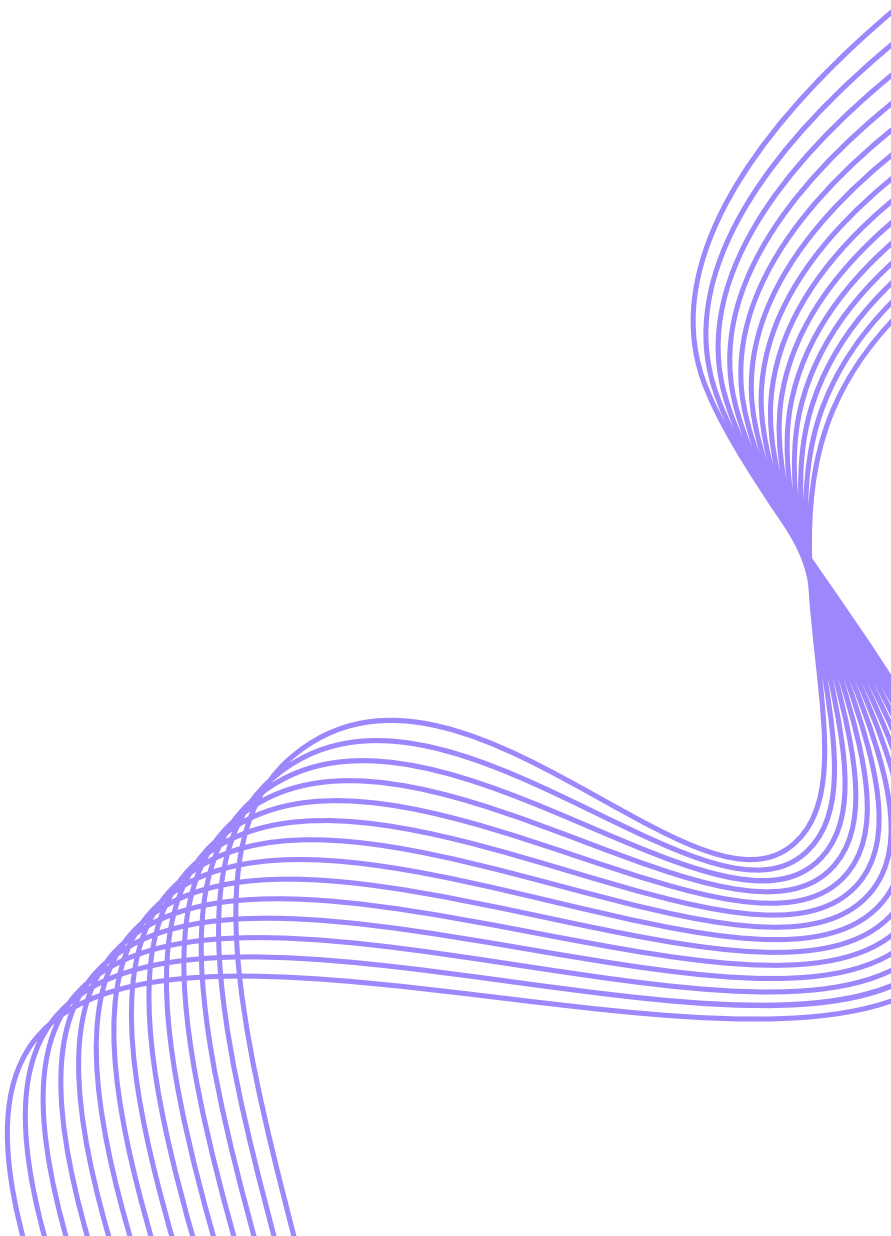
BUILD WEEK 3

MALWARE ANALYSIS AND REVERSE ENGINEERING

REPORTED BY:

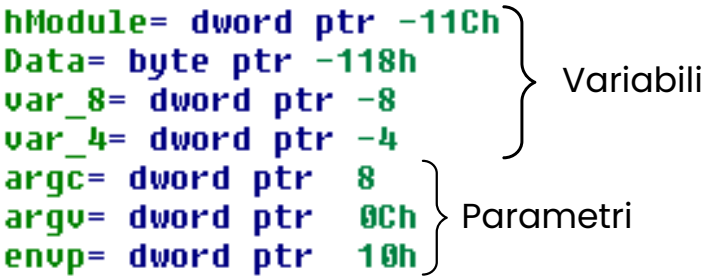
FERNANDO CATRAMBONE
ALESSANDRO MOSCETTI
MATTEO MURILLO
MICHAEL POGGIALI
BENEDETTA FORESTIERI
LUCA GALLEANI
NATALINO IMBROGNO
DAVIDE DIGLIO



Day 1

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 nella funzione **Main**()?



Utilizzando il programma "**Ida pro**" siamo andati ad analizzare il codice malevolo (Malware_Build_Week_U3) alla funzione **Main**, questo perchè è la funzione di ingresso principale del programma che viene eseguita quando viene avviato.

Abbiamo esaminato la funzione **Main()** e notato che i suoi **tre parametri** sono individuati da offset positivo rispetto al registro EBP, indicando che i valori dei parametri sono allocati a una certa distanza in avanti rispetto a EBP.

Allo stesso tempo, abbiamo rilevato la presenza di **quattro variabili** all'interno della funzione **Main()**, ciascuna identificata da un offset negativo rispetto al registro EBP. Questo suggerisce che lo spazio di memoria assegnato a queste variabili si trova a una certa distanza all'indietro rispetto a EBP.

La differenza tra **parametro** e **variabile** sta nell'utilizzo durante l'esecuzione del programma: i parametri sono valori passati a una funzione quando viene chiamata, mentre le variabili sono spazi di memoria utilizzati per conservare dati all'interno della funzione. La distinzione è evidenziata dagli offset positivi per i parametri e dagli offset negativi per le variabili rispetto al registro EBP.

Day 1

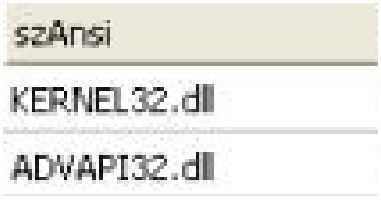
Con riferimento al file eseguibile **Malware_Build_Week_U3**, rispondere ai seguenti quesiti utilizzando i tool e le tecniche apprese nelle lezioni teoriche:

- Quali **sezioni** sono presenti all'interno del file eseguibile?
- Quali **librerie** importa il Malware?

L'analisi condotta attraverso **CFFExplorer** ha consentito di ottenere una panoramica più dettagliata delle attività che può effettuare il malware.
Le **sezioni** da cui è composto il malware sono:

- **.text**: contiene le istruzioni che la CPU eseguirà una volta che il software sarà avviato.
- **.data**: contiene i dati e le variabili globali del programma eseguibile, che devono essere disponibili da qualsiasi parte del programma.
- **.rdata**: contiene i dati disponibili in sola lettura come librerie o funzioni importate o esportate dal programma.
- **.rsrc**: include le risorse utilizzate dall'eseguibile come ad esempio icone, immagini, menu e stringhe che non sono parte dell'eseguibile stesso.

Il malware utilizza funzioni provenienti da due **librerie**:



- **Kernel32.dll**: contiene le funzioni principali per l'interazione con il sistema operativo.
- **Advapi32.dll**: sono presenti le funzioni necessarie per interagire con il registro di Windows.

In questo modo, il malware sfrutta le risorse di tali librerie per eseguire operazioni specifiche, coinvolgendo sia il sistema operativo che il registro di Windows.

Day 1

Ipotesi del comportamento del **Malware_Build_Week_U3** dalle informazioni trovate con **CFFExplorer**

1. L'analisi delle librerie di questo malware ha rivelato una serie di funzioni chiave che indicano un comportamento potenzialmente dannoso e orientato all'attacco.

   - L'uso di **GetProcAddress** indica una dinamicità nel caricamento di funzioni, suggerendo che il malware vada a caricare altre librerie e funzioni.

   - Le funzioni **RegSetValueExA** e **RegCreateKeyExA** suggeriscono che il malware potrebbe cercare di persistere nel sistema attraverso la modifica del Registro di Sistema.

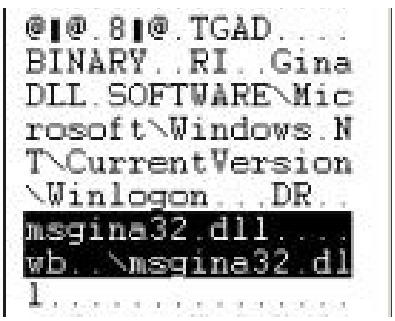
     

   - L'impiego di **LoadResource**, **LockResource**, e **SizeofResource** indica un interesse verso la manipolazione delle risorse presenti nell'eseguibile del malware.

Dalle funzioni emerse nelle librerie possiamo supporre che si tratti di un malware della famiglia dei **Dropper**.
Inoltre analizzando la sezione **.data,** che contiene i dati necessari al programma per funzionare, abbiamo individuato un file di nome **msgina32.dll** e un path che riguarda **winlogon**, che è un processo Windows che riguarda il logon interattivo.
Da questi elementi possiamo supporre che il malware tramite un componente malevolo interferisca con l'accesso per rubare le credenziali.

.data	00003EA8	00006000
.rsrc	00001A70	0000C000



Day 2

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**

```
push    0               ; lpSecurityAttributes
push    0F003Fh         ; samDesired
push    0               ; dwOptions
push    0               ; lpClass
push    0               ; Reserved
push    offset SubKey   ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
push    80000002h       ; hKey
call    ds:RegCreateKeyExA
```

I valori necessari per questa funzione vengono trasmessi attraverso lo stack di memoria mediante l'operazione "**push**". Prima di eseguire la funzione, i parametri vengono posti nello stack in sequenza, e la funzione li preleva da lì durante l'esecuzione.
L'indirizzo **00401017** nel codice contiene la chiave il cui valore viene fornito come argomento a **RegCreateKeyExA**.
Questa funzione sta ad indicare che il programma sta cercando di creare o aprire una chiave del Registro di Sistema per scrivere o leggere informazioni al fine di manipolare il Registro di Sistema.
L'**oggetto** nella loc **00401017** contiene il percorso della chiave del Registro di Sistema che si desidera creare o aprire.

- Spiegare il significato delle istruzioni comprese tra gli indirizzi **00401027** e **00401029**
- Tradurre il codice Assembly nel corrispondente **costrutto C.**

```
00401027        test    eax, eax
00401029        jz      short loc_401032
```

L'istruzione **test eax, eax** è simile all'operatore logico **AND** ma a differenza che non memorizza il risultato in **eax.** In particolare, effettuerà un test bit a bit tra il registro **eax** e se stesso impostando cosi i flag di zero (**ZF**) a 0.
L'istruzione **jz short loc_401032** a questo punto effettuerà un salto alla locazione solo se il flag di zero (**ZF**) sarà impostato a zero.
Questo potrebbe essere una sua rappresentazione in **costrutto C**:

```
if (eax==0){
    nome_registro="GinaDLL"
}
else {
    return 1;
}
```

Day 2

Con riferimento al **Malware** in analisi, spiegare:

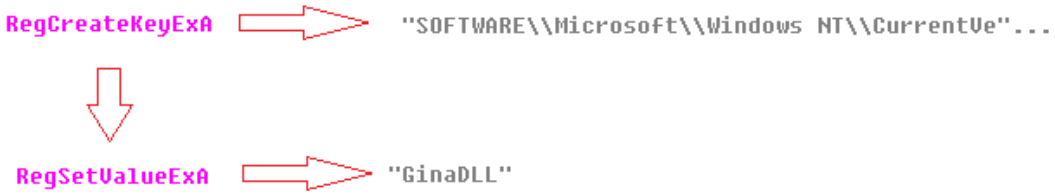
```
push    0               ; lpSecurityAttributes
push    0F003Fh         ; samDesired
push    0               ; dwOptions
push    0               ; lpClass
push    0               ; Reserved
push    offset SubKey   ; "SOFTWARE\\Microsoft\\Windows NT\\CurrentVe"...
push    80000002h       ; hKey
call    ds:RegCreateKeyExA
```

```
40103E          push    offset ValueName ; "GinaDLL"
401043          mov     eax, [ebp+hObject]
401046          push    eax             ; hKey
401047          call    ds:RegSetValueExA
```

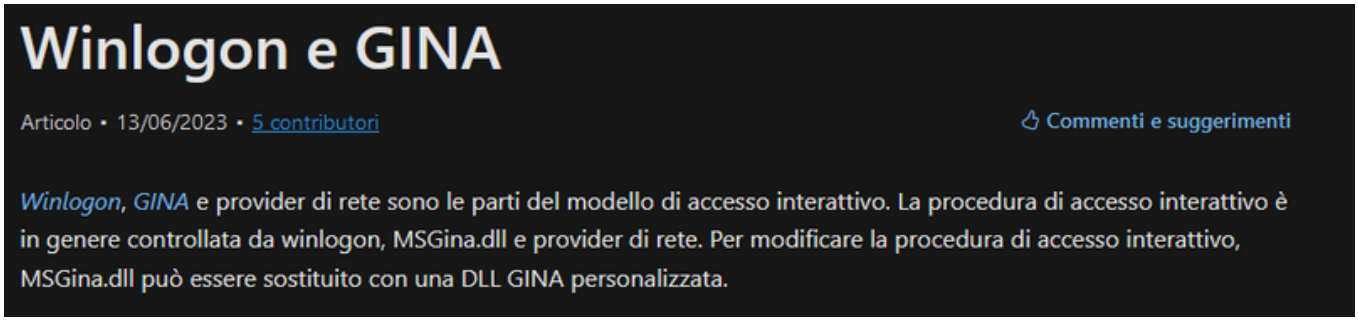
- Qual è il valore di «**ValueName**» alla locazione **00401047**

Il valore del parametro ValueName è "**GinaDLL**"

- Spiegate quale **funzionalità** sta implementando il Malware in queste sezione.



In queste sezione il malware sta creando una nuova chiave di registro **RegCreateKeyExA** e sta settando il suo nome a: "**GinaDLL**" utilizzando la funzione **RegSetValueExa** . Come possiamo vedere GINA e Winlogon servono per gestire la procedura di accesso.

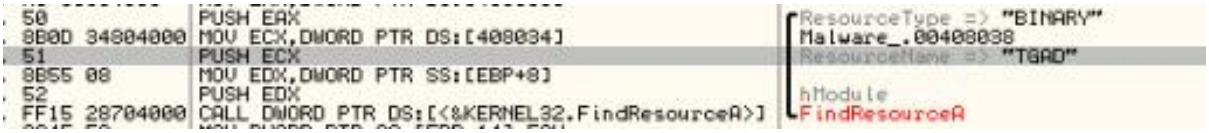


https://learn.microsoft.com/it-it/windows/win32/secauthn/winlogon-and-gina

Day 3

Analizzando le routine tra le locazioni di memoria **00401080** e **00401128**:

- Qual è il valore del parametro «**ResourceName**» passato alla funzione



Tramite l'utilizzo del software **"OllyDBG"** abbiamo individuato che il valore del parametro **"ResourceName"** è **"TGAD"**

- Il susseguirsi delle chiamate di funzione che effettua il Malware in questa sezione di codice che **funzionalità** sta implementando?

Dalle chiamate di funzione presenti in questa sezione di codice abbiamo una conferma che il malware sia un **dropper**, ovvero un software malevolo che svolge il ruolo di un trasportatore di altri malware, facilitando la loro introduzione e esecuzione nell'ambiente del computer infetto

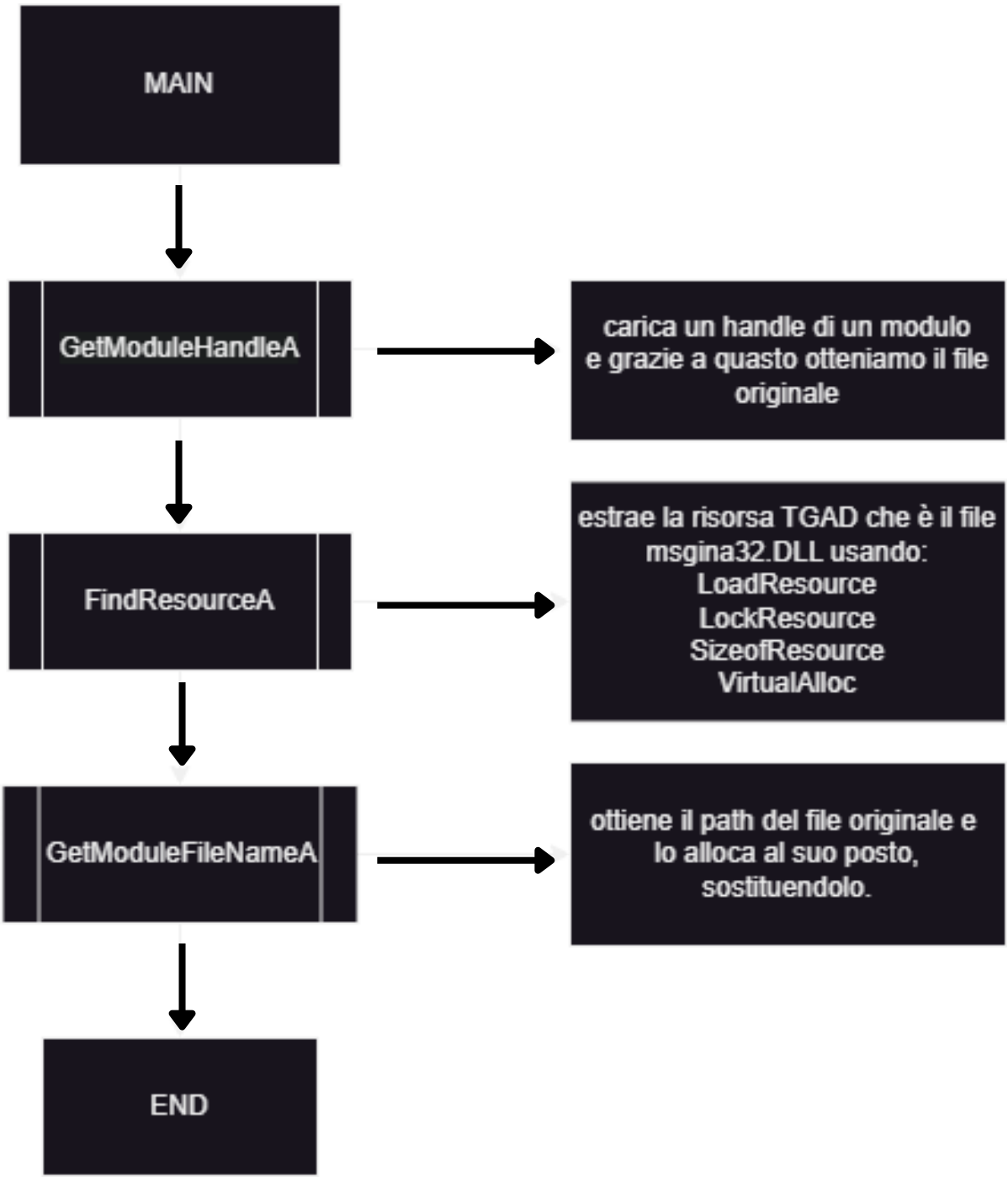
```
loc_4010DF:                             ; CODE XREF: sub_401080+56↑j
                mov     eax, [ebp+hResInfo]
                push    eax             ; hResInfo
                mov     ecx, [ebp+hModule]
                push    ecx             ; hModule
                call    ds:LoadResource
                mov     [ebp+hResData], eax
                cmp     [ebp+hResData], 0
                jnz     short loc_4010FB
                jmp     loc_4011A5
; ---------------------------------------------------------------------------

loc_4010FB:                             ; CODE XREF: sub_401080+74↑j
                mov     edx, [ebp+hResData]
                push    edx             ; hResData
                call    ds:LockResource
                mov     [ebp+var_8], eax
                cmp     [ebp+var_8], 0
                jnz     short loc_401113
                jmp     loc_4011A5
; ---------------------------------------------------------------------------

loc_401113:                             ; CODE XREF: sub_401080+8C↑j
                mov     eax, [ebp+hResInfo]
                push    eax             ; hResInfo
                mov     ecx, [ebp+hModule]
                push    ecx             ; hModule
                call    ds:SizeofResource
```

- È possibile identificare questa funzionalità utilizzando l'analisi **statica basica**? (elencare le evidenze a supporto).

Già dall'analisi **statica basica** avevamo intuito il possibile funzionamento del malware tramite la presenze delle funzioni **LoadResouce**, **Lock Resource** **SizeOfResource** all'interno della libreria **KERNEL32.dll** oltre che alla presenza della sezione **.rsrc** che contiene le ulteoriori risorse che il **dropper** va a caricare nella macchina vittima.

Day 3

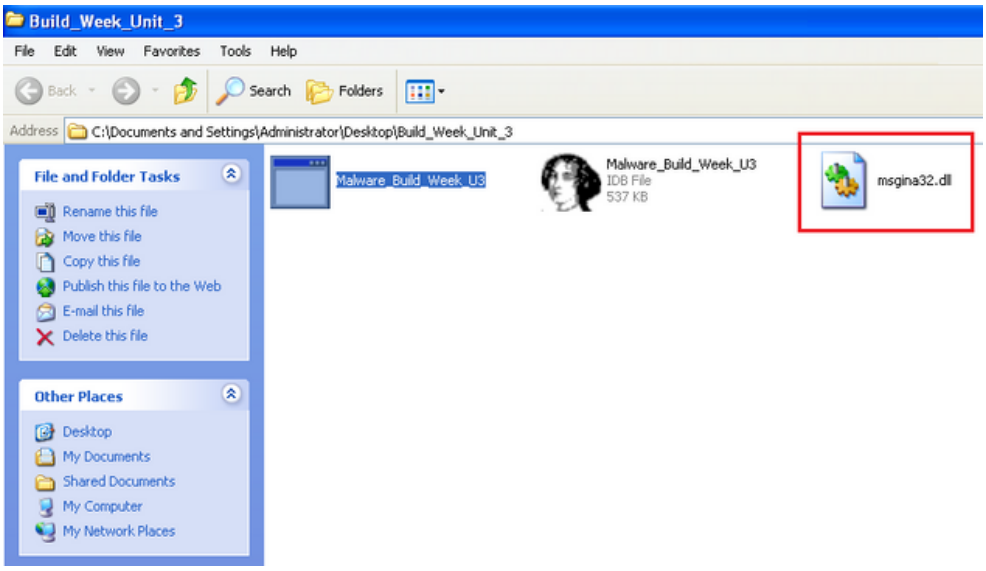
- Disegnare un diagramma di flusso che comprenda le tre funzioni che descrivono le funzionalità appena viste del malware.



Come si può vedere in figura questo diagramma semplificato mostra il comportamento di queste tre funzioni all'interno del **main** ed il loro utilizzo.

Day 4

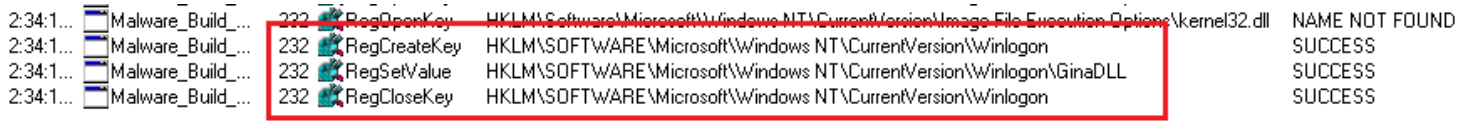
- Preparate l'ambiente ed i tool per l'esecuzione del Malware
- Cosa notate all'interno della cartella dove è situato l'eseguibile del Malware?



Una volta preparato il nostro laboratorio, isolando la nostra macchina virtuale abbiamo avviato il malware, all'interno della stessa cartella viene creato un file di nome "**msgina32.dll**". Questa evidenza ci da conferma sulle ipotesi precedentemente fatte ovvero che si tratta di un **dropper** che estrae un file **.dll**

Filtrate includendo solamente l'attività sul registro di Windows.

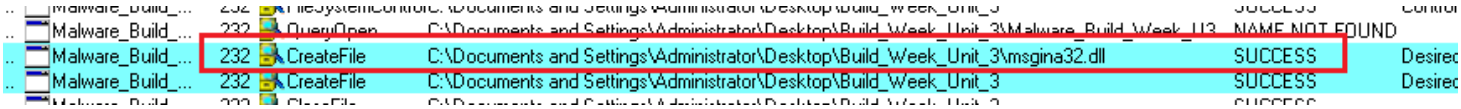
- Quale chiave di registro viene creata?
- Quale valore viene associato alla chiave di registro creata?



Viene creata una chiave di registro all'indirizzo di **Winlogon**, a questa chiave di registro viene associato il valore **"GinaDLL"** che abbiamo già visto nelle giornate precedenti.

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?



La funzione chiamata è **"CreateFile"** come possiamo vedere in figura crea il nostro file **msgina32.dll**.

Day 5

GINA (Graphicauthentication& authentication) è un componente di Windows che permette l'autenticazione degli utenti tramite interfaccia grafica, ovvero permette agli utenti di inserire **username** e **password** nel classico riquadro Windows, come quello in figura.

- Cosa può succedere se il file **.dll lecito** viene sostituito con un file **.dll malevolo** che intercetta i dati inseriti?

Sapendo che GINA è un componente di Windows che permette l'autenticazione tramite interfaccia grafica nel caso in cui un .dll lecito venga sostituito con un file .dll malevolo c'è il rischio che le credenziali degli utenti vengano rubate.

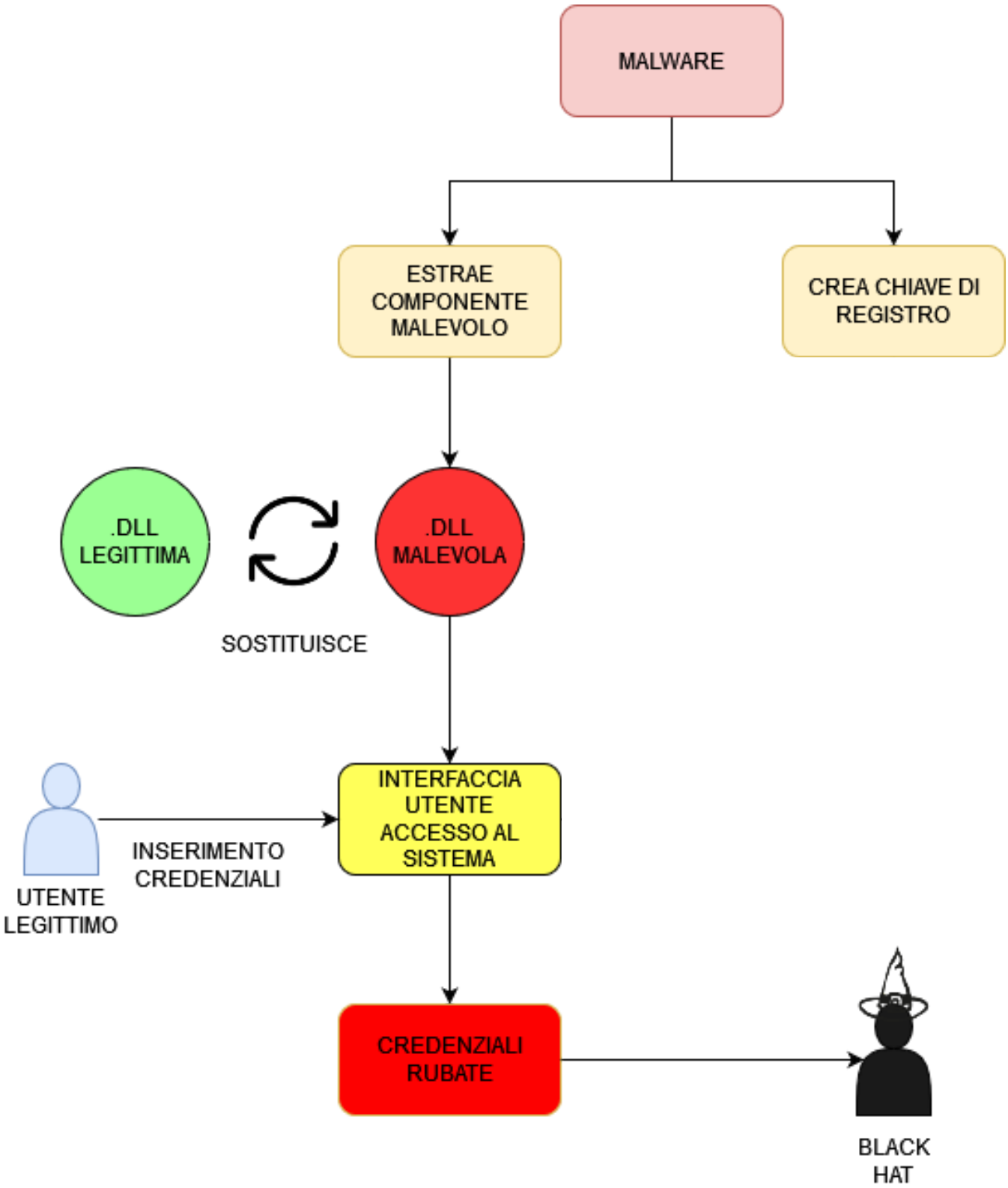
- Delineate il **profilo** del Malware e delle sue funzionalità.

Possiamo concludere che il comportamento del malware è così descritto:

- Estrae una componente malevola di nome "GinaDLL" dalle sue risorse all'interno della cartella dove si trova il malware.
- Crea una chiave di registro "Winlogon" che è parte del sistema operativo Windows e serve per l'autenticazione interattiva.
- Sostituisce il componente .dll legittimo con quello malevolo estratto precedentemente.
- Il componente malevolo viene utilizzato per rubare le credenziali.

Day 5

- Unite tutti i punti visti fino ad esso per creare un grafico che ne rappresenti lo scopo ad alto livello.



Bonus 1

- Spiegare l'analisi di **Any.rune** del file malevolo

Dall'analisi di **Any.run** abbiamo scoperto che questo è un file malevolo, infatti utilizza un malware firmato **Agent Tesla**, questo è uno **spyware** che raccoglie informazioni sulle azioni delle sue vittime registrando le sequenze di tasti e le interazioni fatte dall'utente.
Può essere contratto attraverso campagne di **phishing**
(L'aggressore invia i file malevoli attraverso **email** mascherandosi da sito legittimo)
Una volta **scaricato/cliccato** il malware si comporterà dunque come uno **stealer**, un software dannoso destinato a **ottenere l'accesso non autorizzato** alle informazioni degli utenti e **trasferirle all'aggressore** come **file** e **password**.
I ladri dunque sono in grado di **spiare la macchina vittima** e registrare ogni tasto premuto dall'utente acquisendo cosi tutte le informazioni sensibili.

Nomi dei processi malevoli e numero identificativo:

Uqzqkjvjt.exe (PID: 2308)
RegAsm.exe (PID: 792)

Qui di seguito ci sono elencati i vari step che che effettua il codice malevolo:

- Esecuzione manuale da parte dell'utente
- Rilascia il malware eseguibile dopo l'avvio del pc
- Il processo elimina l'eseguibile legittimo di Windows
- Mascheramento da file legittimo
- Agent Tesla in azione
- Ruba le credenziali dai browser Web
- Si connette alla porta SMTP
- Accede ai profili Microsoft Outlook
- Si connette al server CnC per trasferire le informazioni sensibili dell'utente.

https://app.any.run/tasks/444c2f53-1cce-49a9-8336-2539896df32b/

Bonus 1

- Spiegare l'analisi di **Any.rune** del file malevolo

Nel report di **Any.run** è chiaro si tratti di un malware dal nome **Rhadamanthys** scritto in linguaggio **C++** che ruba informazioni ed estrae dati sensibili.
La sua catena operativa stratificata e le tattiche di evasione avanzate lo rendono un **rischio importante** nel panorama della sicurezza informatica.
Nello specifico viene **diffuso tramite siti** che sembrano essere legittimi e dai quali viene scaricato il file malevolo che contiene il malware, una volta eseguito **ruba i dati** dell'utente vittima e li invia ad un server C&C controllato dal Black Hat.

Fa parte anch'esso della famiglia degl**i stealer** cioè un gruppo di **software dannoso** destinato ad **ottenere l'accesso non autorizzato** alle informazioni degli utenti e trasferirle all'aggressore.

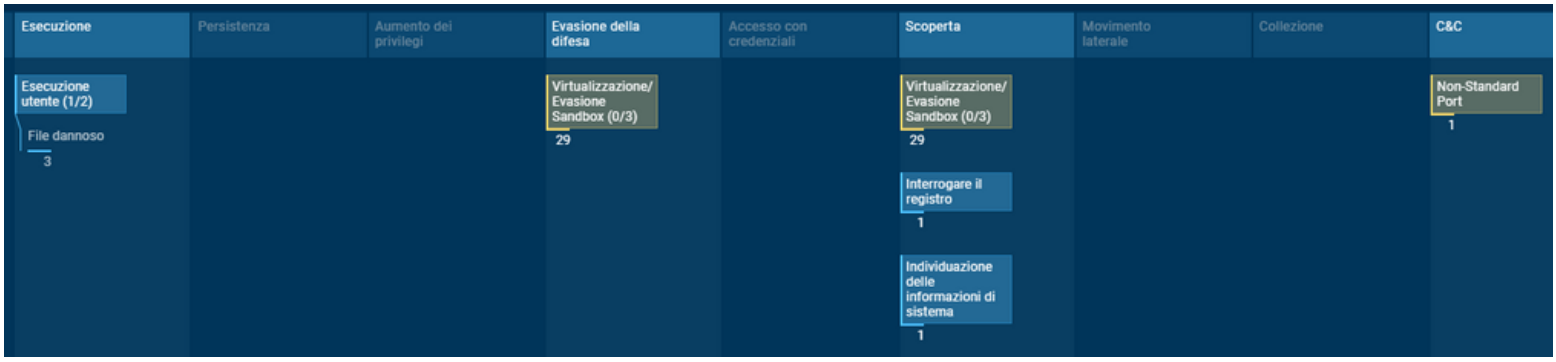
I principali metodi di distribuzione osservati per questa minaccia includono **siti Web** di software **falsi** promossi tramite **Google Ads** ed **e-mail di phishing**, che prendono di mira le vittime indipendentemente dalla loro posizione o settore.
Nel complesso, **Rhadamanthys** vanta un ampio **set di funzionalità di furto** e rappresenta una minaccia significativa.

Nomi dei processi malevoli e numero identificativo:

dialer.exe (PID: 3052)

Qui di seguito ci sono elencati i vari step che che effettua il codice malevolo:

- Esecuzione manuale da parte dell'utente
- Rilascio del file eseguibile
- L'applicazione si avvia da sola
- RHADAMANTHYS si attiva
- Il processo controlla se viene eseguito in un'ambiente virtuale
- Il processo utilizza il file che ha scaricato
- Apertura di più finestre Chrome
- Estrazione dei dati sensibili
- Si connette al server dell'aggressore per il trasferimento.



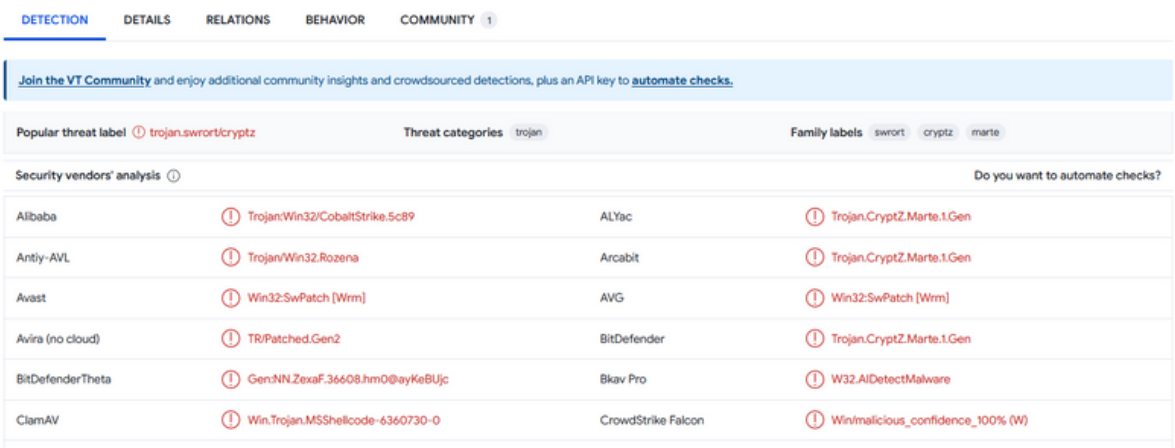
https://app.any.run/tasks/512b6efc-380b-40f5-8689-1027fa7852e2/

Bonus 2

- Analizzare il file **calcolatriceinnovativa50.exe**

Per far ciò abbiamo utilizzato diversi tool:

- **Virus Total**



- **CFFExplorer**

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
000123FC	N/A	00011FA8	00011FAC	00011FB0	00011FB4	00011FB8
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
SHELL32.dll	1	00012CA8	FFFFFFFF	FFFFFFFF	00012E42	0000109C
msvcrt.dll	26	00012DC8	FFFFFFFF	FFFFFFFF	00012F60	000011BC
ADVAPI32.dll	3	00012C0C	FFFFFFFF	FFFFFFFF	00012FFC	00001000
KERNEL32.dll	30	00012C2C	FFFFFFFF	FFFFFFFF	000131D4	00001020
GDI32.dll	3	00012C1C	FFFFFFFF	FFFFFFFF	0001320C	00001010
USER32.dll	69	00012CB0	FFFFFFFF	FFFFFFFF	000136A4	000010A4

Librerie importate

Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	000126B0	00001000	00012800	00000400	00000000	00000000	0000	0000	6000002
.data	0000101C	00014000	00000A00	00012C00	00000000	00000000	0000	0000	C000004
.rsrc	00008A70	00016000	00008C00	00013600	00000000	00000000	0000	0000	4000004

Sezioni headers

- **ProcessMonitor**

calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\calcolatriceinnovativa.exe
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server
calcolatriceinnovativa.exe	2676	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat
calcolatriceinnovativa.exe	2676	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Secur32.dll
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RPCRT4.dll
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ADVAPI32.dll
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
calcolatriceinnovativa.exe	2676	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\LeakTrack
calcolatriceinnovativa.exe	2676	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics
calcolatriceinnovativa.exe	2676	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\USER32.dll

Possiamo concludere che il file analizzato sia un malware.
Da **VirusTotal** possiamo notare che sia della famiglia dei **Trojan**.
Di conseguenza questo file una volta entrato nella macchina vittima andrà a **scaricare** il vero e proprio codice malevolo, un **keylogger**.
Il **Keylogger** è un tipo di malware che carpisce l'input utente, come la digitazione della tastiera o il puntatore del mouse per poi essere trasferito all'attaccante.
In questo caso però non sono presenti altre **funzionalità di rete** e quindi le informazioni ottenute dal malware non hanno modo di essere trasferite.

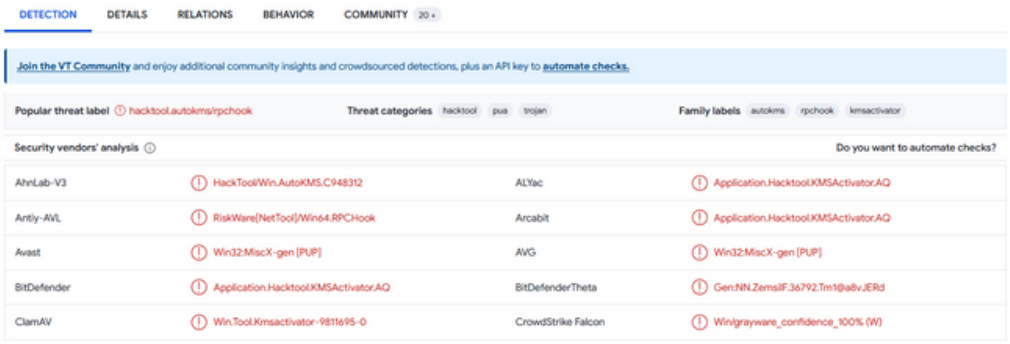
https://www.virustotal.com/gui/file/b8ed129eb56c68cec1661206c313c6eab2e20e4b9223336f7edf661c9956e81a

Bonus 2

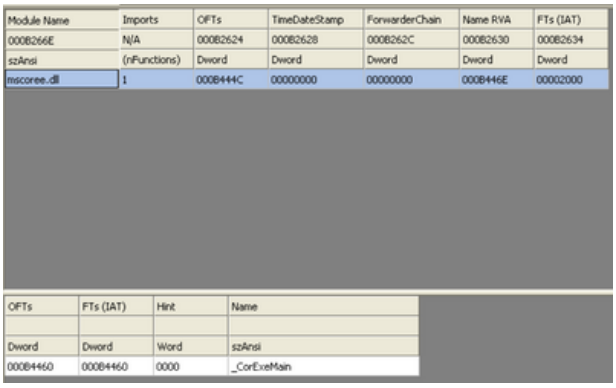
Il nostro dipendente “sveglio” dice al SOC (in questo caso noi) che ha avviato in un pc questo file innocuo **AmicoNerd.zip**
Il nostro compito è convincere il dipendente che il file sia malevolo.

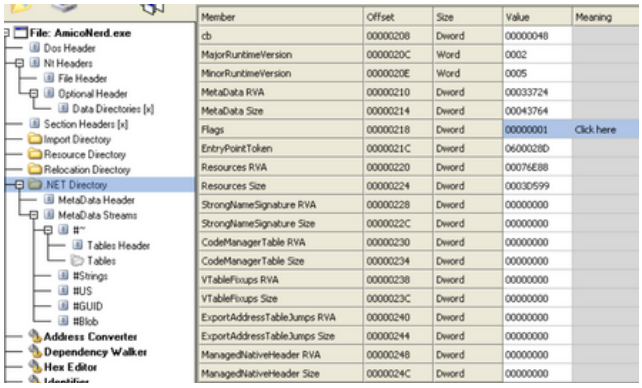
Abbiamo effettuato per questo file una serie di analisi.
Senza scendere nel dettaglio tecnico delle analisi sono state riscontrate diverse funzionalità che ci fanno ipotizzare che si tratti di una **backdoor**.
Quest’ultimi sono **malware** che stabiliscono una connessione permettendo di prendere il controllo della macchina vittima.
Per convincere il dipendente mostriamo il **report di VirusTotal** che lo segnala come malware.
Inoltre abbiamo nota questa minaccia, infatti si tratta di **KMSpico** (AutoPico), ovvero un file malevolo che **si maschera** da attivatore Windows (serve per conferire la licenza legittima ad un sistema windows che ne è sprovvisto)
che in realtà installa una **backdoor**.
Possiamo anche notare come dopo aver avviato il file viene **creata una cartella** che contiene i log delle attività del malware che immediatamente cerca di creare una **connessione ad un sito anonimo**.

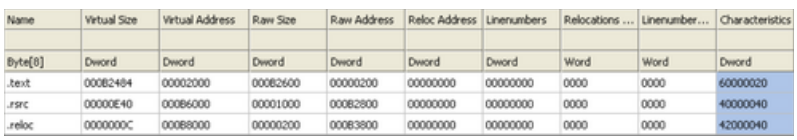
- **Virus Total**



- **CFFExplorer**



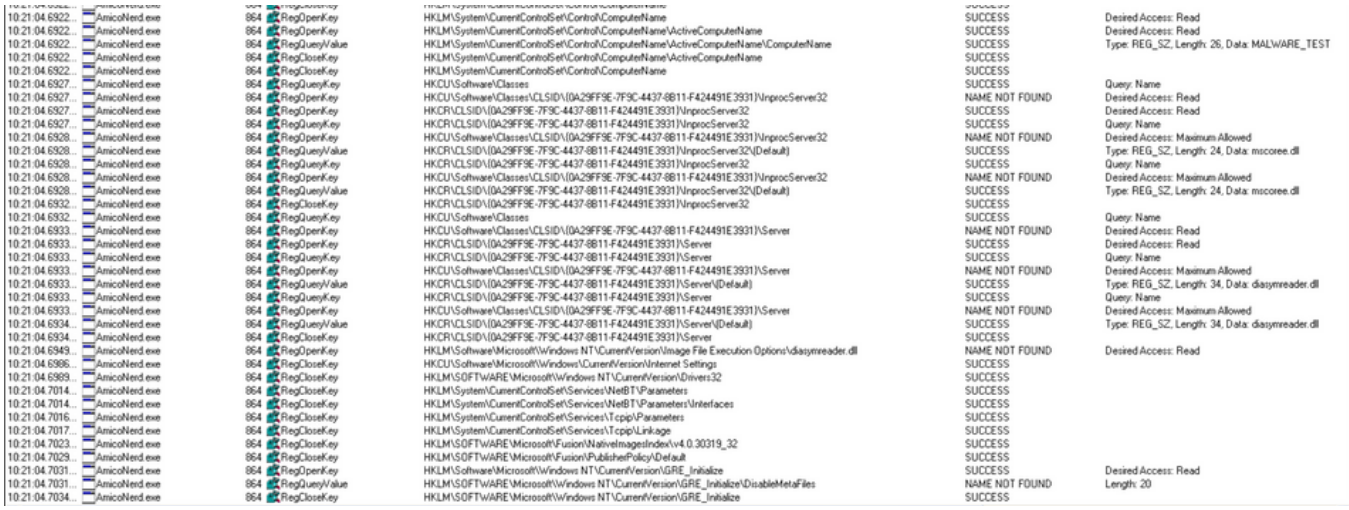


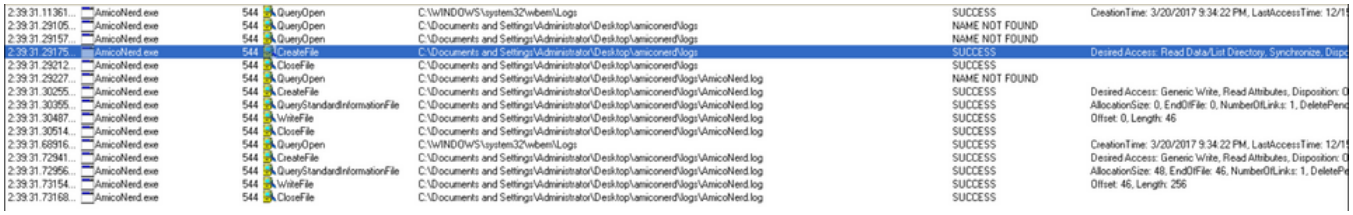


https://www.virustotal.com/gui/file/c6603d416dfc48894eda35d9a9a8523bdf9823e215ab926783ce6848aa8a62c4

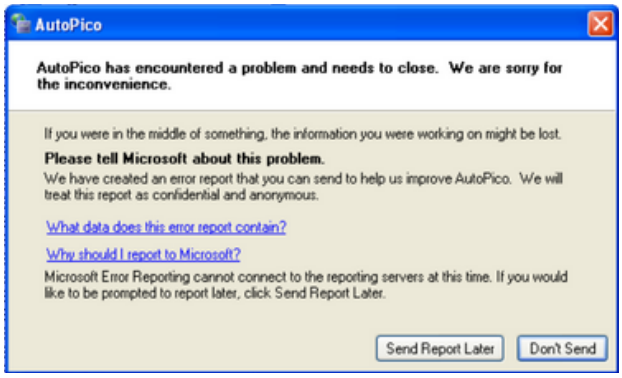
Bonus 2

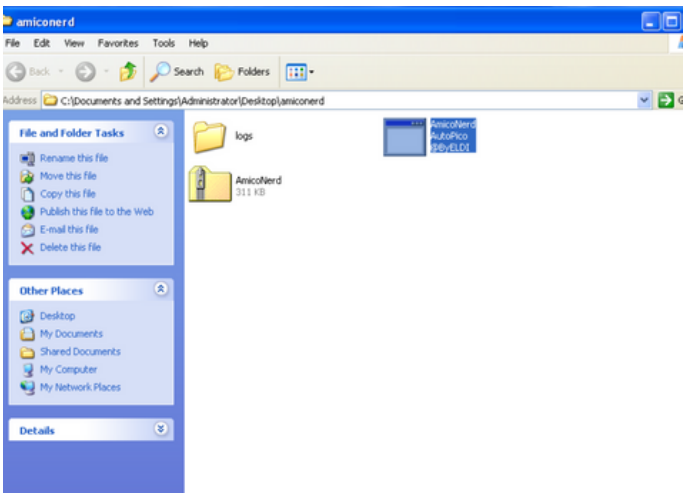
- **ProcessMonitor**

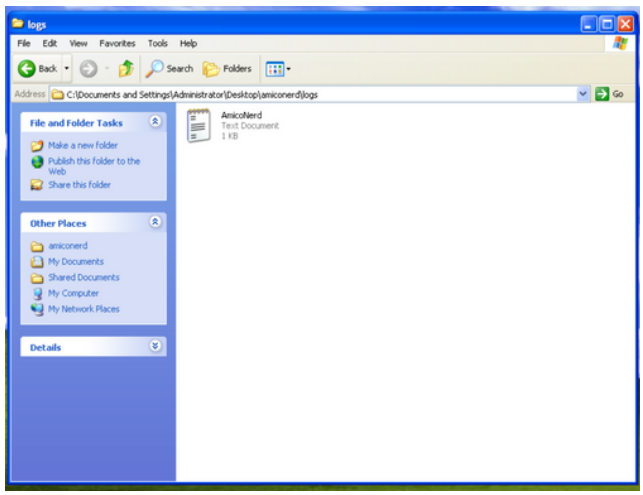




- **WindowsXP**







Thank you!