MALWARE ANALYSIS

I parametri passati alla funzione Main() sono 3: int argc, const char **argv, const char **envp.
Le variabili dichiarate all'interno del main sono 8: hModule(word), Data(byte), var_117(byte), var_8(word), var_4(word), argc(word), argv(word), envp(word).
Il malware importa le librerie:

- **KERNEL32:** Dll che espone agli applicativi la maggiorparte delle API base di Win32.
- **ADVAPI32:** Dll che provvede alle chiamate funzioni per manipolare il registro di sistema

Le sezioni all'interno dell'exe sono:

- .text: Contiene istruzioni che la CPU esegue ad avvio exe.
- .data: Include le variabili globali del programma.
- .rdata: Include generalmente le informazioni circa le librerie e le funzioni importate ed esportate dall'exe.
- .rsrc: Risorse aggiuntive non codice, come icone, menu, immagini.

La chiamata all'indirizzo di memoria 00401021 chiama dal Data Segment la funzione **RegCreateKeyExA**, ovvero una funzione di creazione di una chiave di registro specifica. Se la chiave già esiste, la funzione provvederà invece ad aprirla. La funzione fa parte dell'header winreg.h.
La chiave di registro da creare è specificata poco sopra, alla voce: "push offset Subkey"(Indirizzo 00401017) e comanda la creazione di una chiave "SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion**Winlogon**", ovvero una chiave che si occupa di caricare le configurazioni dei singoli utenti di una macchina, e crea dei desktop separati per funzione: Controllo funzioni(CTRL+ALT+DEL), default utente e screensaver. Processi richiamati da Winlogon sono caricati ancora prima del login ed inoltre possono avere accesso a notifiche di eventi all'interno della macchina, tramite determinati plugins.
Il malware quindi aggiunge un notevole livello di persistenza e controllo nella macchina, e non può essere semplicemente soppresso con un riavvio.
La chiave viene creata all'interno di **HKEY_LOCAL_MACHINE**, come specificato dal push di sopra: H80000002 è la costante corrispondente alla cartella di registro sopra menzionata.

```
.text:0040101C                push    80000002h       ; hKey
.text:00401021                call    d; char SubKey[]
.text:00401027                test    e
.text:00401029                jz      s
.text:0040102B                mov     eax, 1

SubKey          db 'SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon',0
                                        ; DATA XREF: sub_401000+17↑o
```

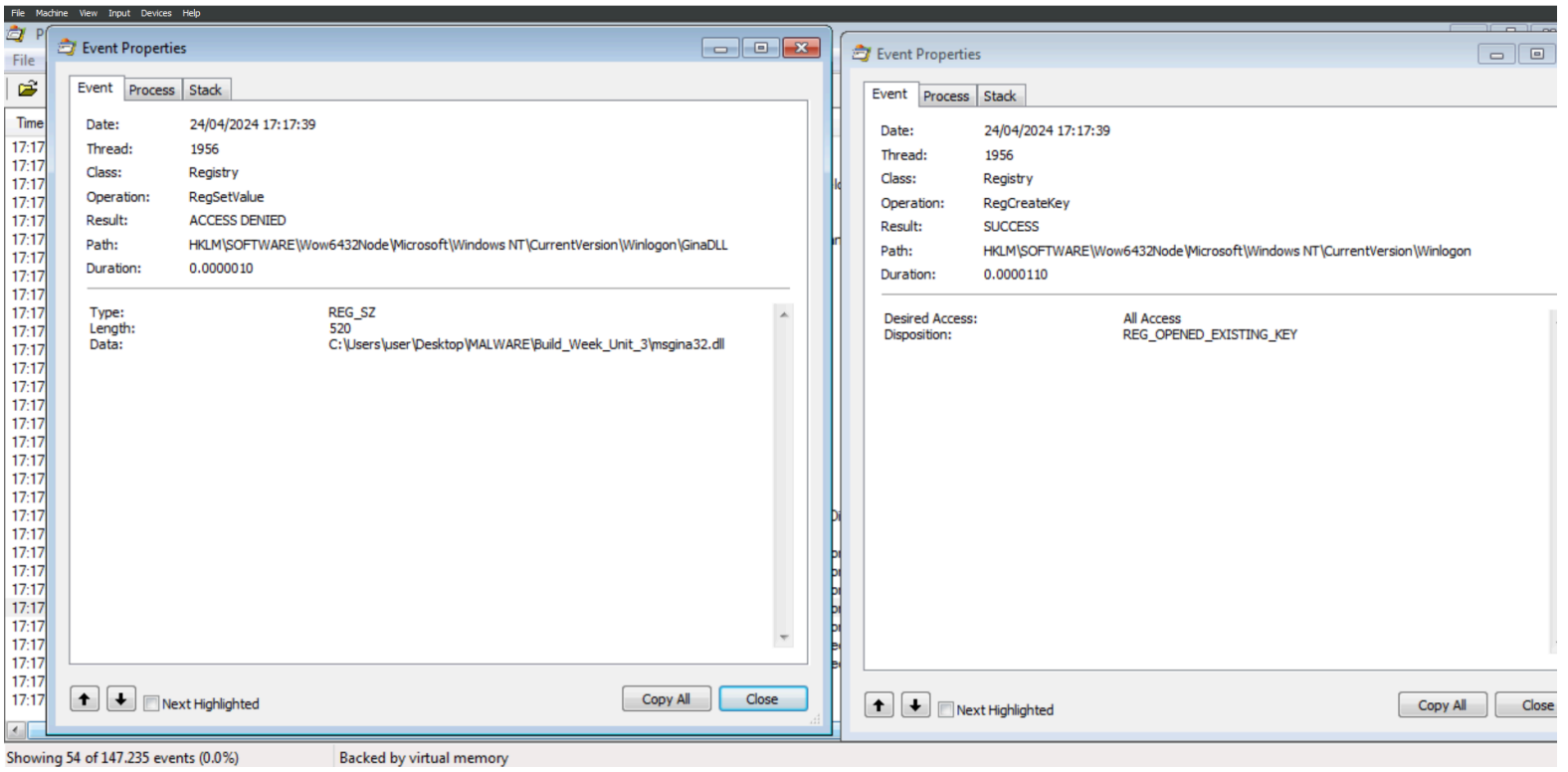
L'istruzione 00401027, test eax eax, non fa altro che comparare le entries del registro eax con se stessa tramite un'operazione di AND e setta le condizioni per il Jump Zero di 00401029 settando lo zero flag se entrambi i registri sono uguali. In quanto stiamo

comparando lo stesso registro, il risultato attiverà sempre la zero flag ed effettuerà il salto alla loc_401032.
“Short” indica che il salto viene effettuato all’interno dello stesso modulo.
Una traduzione in C sarebbe:

```
While(true)
        {
                goto loc_401032
        }
```

Il valore del parametro “ValueName” è: **GinaDLL**

Avviando il programma, il malware prova ad installare una versione infetta di GINA, tramite la msgina32.dll che crea all’interno della cartella e registrandola nel registro come “GinaDLL”.



GINA, o “Graphical identification and authentication”, era un componente delle vecchie versioni di Windows che forniva servizi di logon utenti ed avvio predefinito dei processi. Gli utenti erano in grado di creare proprie versioni customizzate del servizio, creando un’entry nel registro di Winlogon contenente il nome GinaDLL e la locazione della dll modificata, esattamente le stesse operazioni rilevate durante l’analisi statica su IDA.

```
.text:00401245        mov     byte ptr [eax], 0
.text:00401248        mov     edi, offset aMsgina32_dll ; "\\msgina32.dll"
.text:0040124D        lea     edx, [ebp+Data]
```

```
.text:00401161        push    offset Mode     ; "wb"
.text:00401166        push    offset aMsgina32_dll_0 ; "msgina32.dll"
.text:0040116B        call    _fopen
```

GINA era in grado di leggere tutte le credenziali inserite nel sistema per le fasi di autenticazione. Sottoponendo una dll modificata al registro, si può di fatto intercettare tutti dati inseriti in fase di login e quindi rubare tutte le credenziali. Il malware inizia a funzionare dal primo riavvio dalla sua installazione.

Il malware deve includere delle funzioni di logging per avere qualsiasi effetto. Un'analisi della DLL ci porta a scoprire esattamente questo tramite le funzioni di WLX che usano il file msutil32.sys per segnare tutte le credenziali usate.

```
.text:10001490A0                push    esi
.text:10001490A1                push    edi
.text:10001490A2                push    offset aWlxloggedout_0 ; "WlxLoggedOutSAS"
.text:10001490A7                call    sub_10001000
.text:10001490AC                push    64h             ; unsigned int
```

Infine, la chiamata di sistema che ha creato la DLL infetta.