



# WINDOWS MALWARE

# COSA FARE

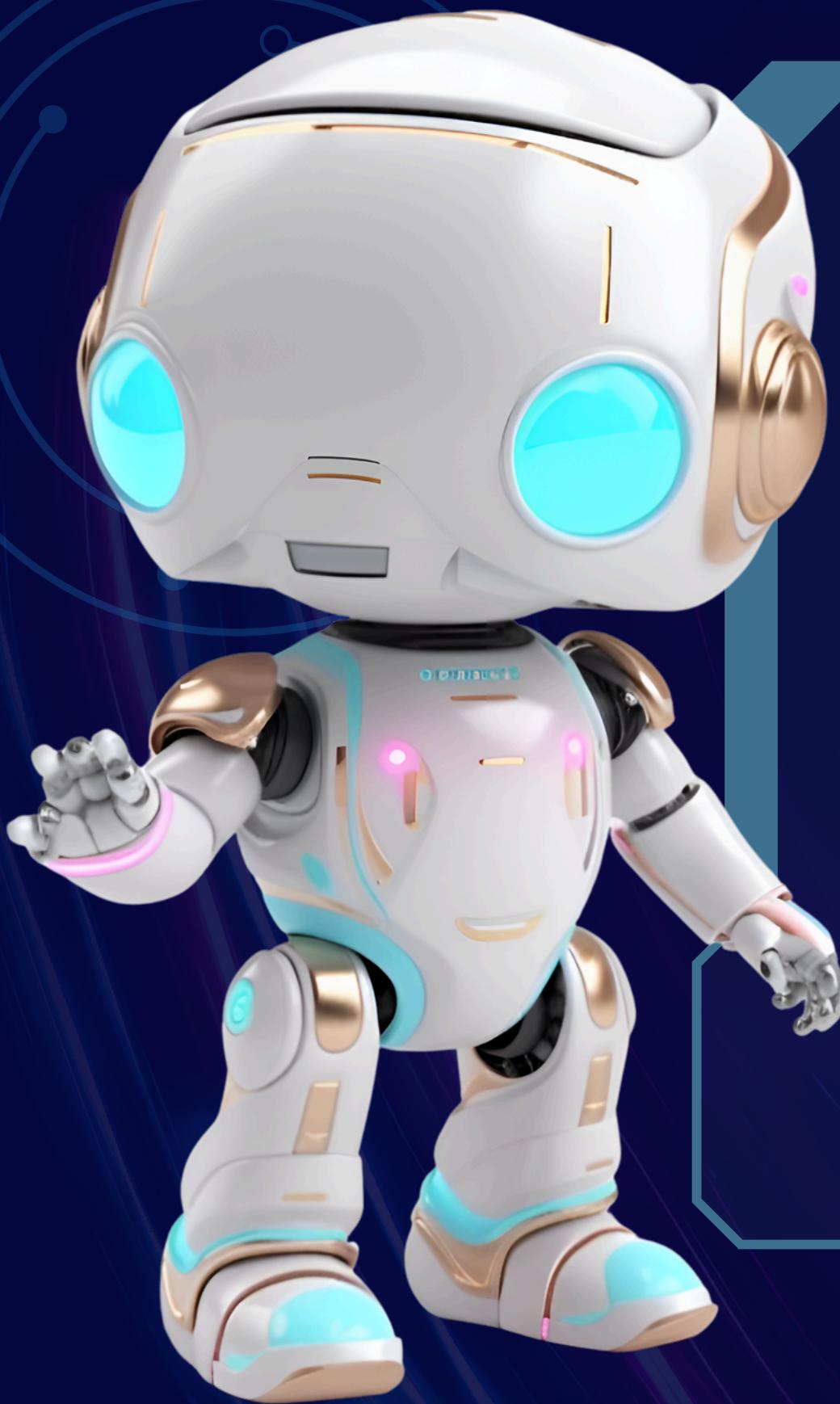
Con riferimento agli estratti di un malware reale presenti nelle prossime slide, rispondere alle seguenti domande:

## Esercizio Windows malware

- Descrivere come il malware ottiene la persistenza , evidenziando il codice assembly dove le relative istruzioni e chiamate di funzioni vengono eseguite
- Identificare il client software utilizzato dal malware per la connessione ad Internet
- Identificare l'URL al quale il malware tenta di connettersi ed evidenziare la chiamata di funzione che permette al malware di connettersi ad un URL
- BONUS: qual è il significato e il funzionamento del comando assembly "lea"



```
0040286F push    2          ; samDesired
00402871 push    eax        ; ulOptions
00402872 push    offset SubKey  ; "Software\\Microsoft\\Windows\\CurrentVersion\\Run"
00402877 push    HKEY_LOCAL_MACHINE ; hKey
0040287C call    esi ; RegOpenKeyExW
0040287E test    eax, eax
00402880 jnz     short loc_4028C5
00402882
00402882 loc_402882:
00402882 lea     ecx, [esp+424h+Data]
00402886 push    ecx        ; lpString
00402887 mov     bl, 1
00402889 call    ds:lstrlenW
0040288F lea     edx, [eax+eax+2]
00402893 push    edx        ; cbData
00402894 mov     edx, [esp+428h+hKey]
00402898 lea     eax, [esp+428h+Data]
0040289C push    eax        ; lpData
0040289D push    1          ; dwType
0040289F push    0          ; Reserved
004028A1 lea     ecx, [esp+434h+ValueName]
004028A8 push    ecx        ; lpValueName
004028A9 push    edx        ; hKey
004028AA call    ds:RegSetValueExW
```



```
.text:00401150 ; ||||||| S U B R O U T I N E |||||  
.text:00401150  
.text:00401150  
.text:00401150 ; DWORD __stdcall StartAddress(LPUOID)  
.text:00401150 StartAddress proc near ; DATA XREF: sub_401040+ECTo  
.text:00401150     push    esi  
.text:00401151     push    edi  
.text:00401152     push    0          ; dwFlags  
.text:00401154     push    0          ; lpszProxyBypass  
.text:00401156     push    0          ; lpszProxy  
.text:00401158     push    1          ; dwAccessType  
.text:0040115A     push    offset szAgent ; "Internet Explorer 8.0"  
.text:0040115F     call    ds:InternetOpenA  
.text:00401165     mov     edi, ds:InternetOpenUrlA  
.text:0040116B     mov     esi, eax  
.text:0040116D loc_40116D: ; CODE XREF: StartAddress+301j  
.text:0040116D     push    0          ; dwContext  
.text:0040116F     push    80000000h ; dwFlags  
.text:00401174     push    0          ; dwHeadersLength  
.text:00401176     push    0          ; lpszHeaders  
.text:00401178     push    offset szUrl ; "http://www.malware12COM  
.text:0040117D     push    esi          ; hInternet  
.text:0040117E     call    edi ; InternetOpenUrlA  
.text:00401180     jmp    short loc_40116D  
.text:00401180 StartAddress endp  
.text:00401180 tout:00401180
```

# PERSISTENZA

- I malware ottiene la persistenza nel sistema operativo Windows attraverso la modifica del registro di sistema, specificamente aggiungendo una voce nella sezione Run.
- Il registro di sistema di Windows è una database gerarchico che contiene le configurazioni critiche del sistema e delle applicazioni.
- La chiave Run all'interno del registro è utilizzata per specificare quali programmi devono essere avviati automaticamente all'avvio del sistema o al login dell'utente.
- Quando il malware modifica la chiave Run aggiungendo una nuova voce che punta all'eseguibile del malware (ad esempio, "C:\percorso\del\malware.exe"), esso si assicura che il proprio codice venga eseguito in modo automatico ogni volta che il sistema operativo viene avviato o che l'utente accede al sistema. Questa tecnica è fondamentale per garantire che il malware mantenga il controllo persistente del sistema senza richiedere l'interazione diretta dell'utente.
- Il risultato di questa modifica è che il malware diventa parte integrante del processo di avvio del sistema, operando in modo invisibile e rendendo più difficile la sua individuazione e rimozione da parte degli strumenti di sicurezza tradizionali.

# CODICE INTERESSATO

```
0040286F push    2          ; samDesired
00402871 push    eax        ; ulOptions
00402872 push    offset SubKey  ; "Software\Microsoft\Windows\CurrentVersion\Run"
00402877 push    HKEY_LOCAL_MACHINE ; hKey
0040287C call    esi ; RegOpenKeyExW
```

```
0040289D push    1          ; dwType
0040289F push    0          ; Reserved
004028A1 lea     ecx, [esp+434h+ValueName]
004028A8 push    ecx        ; lpValueName
004028A9 push    edx        ; hKey
004028AA call    ds:RegSetValueExW
```

Questo codice apre la chiave del registro HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run con RegOpenKeyExW, quindi imposta un valore utilizzando RegSetValueExW. Il valore aggiunto consente l'esecuzione automatica del malware all'avvio del sistema.

# CLIENT UTILIZZATO

Il malware utilizza la funzione InternetOpenA per creare un handle per la sessione di Internet. L'agente utente utilizzato è "Internet Explorer 8.0", che indica che il malware si presenta come questo browser quando si connette a Internet.

```
.text:00401154  
.text:00401156  
.text:00401158  
.text:0040115A  
.text:0040115F  
.text:00401165  
.text:0040116B
```

```
push    0          ; lpszProxyBypass  
push    0          ; lpszProxy  
push    1          ; dwAccessType  
push    offset szAgent ; "Internet Explorer 8.0"  
call    ds:InternetOpenA  
mov     edi, ds:InternetOpenUrlA  
mov     esi, eax
```

# URL

Il malware tenta di connettersi all'URL  
<http://www.malware12.COM> utilizzando InternetOpenUrlA

```
.text:00401160      push    0          ; dwContext
.text:0040116F      push    80000000h ; dwFlags
.text:00401174      push    0          ; dwHeadersLength
.text:00401176      push    0          ; lpszHeaders
.text:00401178      push    offset szUrl  ; ""http://www.malware12COM
.text:0040117D      push    esi        ; hInternet
.text:0040117E      call    edi ; InternetOpenUrlA
```

# LEA

Il comando lea (Load Effective Address) in assembly viene utilizzato per calcolare l'indirizzo effettivo di un operando e caricarlo in un registro senza eseguire un'operazione di accesso alla memoria. È utile per calcoli di indirizzi e manipolazione di puntatori.

```
04028A1 lea    ecx, [esp+434h+ValueName]
04028A8 push   ecx    ; lpValueName
```

Qui, lea calcola l'indirizzo effettivo di ValueName rispetto alla posizione dello stack e lo carica nel registro ecx.



Fabio Nobili



Antonio Perna