

Beatrice Folino

[M6 - Progetto Finale]

EPICODE - CYBERSECURITY CLASS

05 maggio 2024



TRACCIA

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.

Malware Analysis

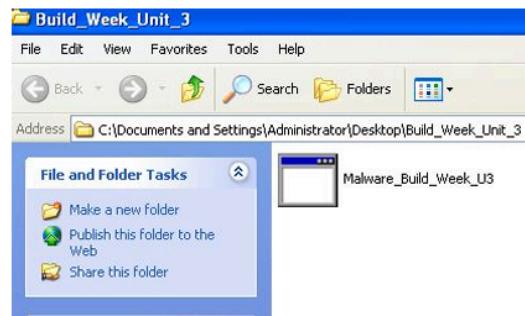
Con riferimento al Malware in analisi, spiegare:

- 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

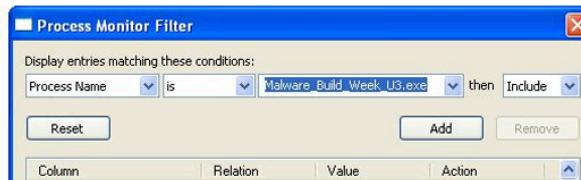
Preparate l'ambiente ed i tool per l'esecuzione del Malware (suggerimento: avviate principalmente Process Monitor ed assicurate di eliminare ogni filtro cliccando sul tasto «reset» quando richiesto in fase di avvio). Eseguite il Malware, facendo doppio click sull'icona dell'eseguibile



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.

INTRODUZIONE

Esistono diversi tipi di analisi:

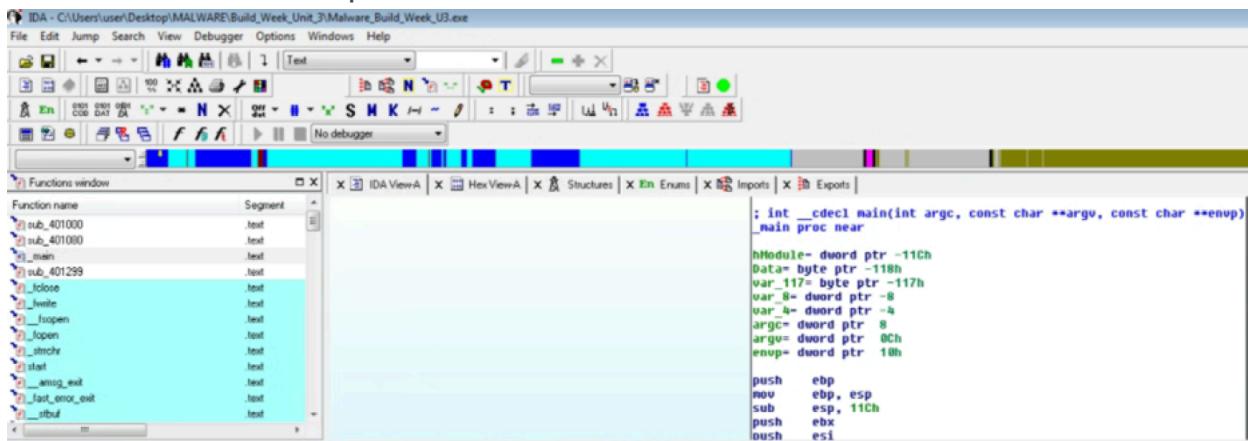
- Analisi Statistica: utilizza tecniche e strumenti per carpire il comportamento del software malevolo senza attivarlo, fornendo informazioni sulle minacce
- Analisi Dinamica: osserva il comportamento del malware sotto copertura scoprendo quali dati modifica e come agisce sul sistema

Esse si suddividono in ulteriori due livelli:

- Analisi Basica: esegue il malware in un ambiente sandbox per osservarne il comportamento e tentare di neutralizzarlo
- Analisi Avanzata: identifica il comportamento del malware analizzando le sue istruzioni. Questo tipo di analisi impiega debugger per monitorare lo stato del programma durante l'esecuzione, raccogliendo informazioni dettagliate sulle sue azioni e interazioni con il sistema.

PROCEDIMENTO ANALISI STATICÀ

Utilizzando IDA Pro, possiamo notare che i parametri della funzione main() sono tre, un int e due char, mentre il numero delle variabili dichiarate sono cinque.



The screenshot shows the IDA Pro interface with the assembly view selected. The assembly code for the main function is displayed:

```
; int __cdecl main(int argc, const char **argv, const char **envp)
_main proc near
hModule= dword ptr -11Ch
Data= byte ptr -11Bh
var_117= byte ptr -117h
var_B= dword ptr -8
var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

push ebp
mov ebp, esp
sub esp, 11Ch
push ebx
push esi
```

- **.text** è una sezione fondamentale di un file eseguibile PE di Windows. Contiene il codice macchina effettivo che viene eseguito dal processore quando il programma viene avviato. La sezione .text può essere utilizzata per comprendere il funzionamento di un programma e per identificare potenziali vulnerabilità di sicurezza.
- **.rdata** è una sezione importante del file eseguibile che contiene dati in sola lettura utilizzati dal programma durante l'esecuzione. I dati nella sezione .rdata possono includere costanti di programma, tabelle di dati, risorse del programma e informazioni sul debug.



Malware_Build_Week_U3.exe									
Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	00005646	00001000	00006000	00001000	00000000	00000000	0000	0000	60000020
.rdata	000009AE	00007000	00001000	00007000	00000000	00000000	0000	0000	40000040
.data	00003EA8	00008000	00003000	00008000	00000000	00000000	0000	0000	C0000040
.rsrc	00001A70	0000C000	00002000	0000B000	00000000	00000000	0000	0000	40000040

Le librerie importate dal malware sono la KERNEL32 e ADVAPI32.

The screenshot shows the IDA Pro interface with the file "Malware_Build_Week_U3.exe" open. The top menu bar includes File, Edit, Search, View, Debugger, Options, Windows, and Help. The main window displays the Imports table and the Functions window. The Imports table lists various functions and their corresponding addresses, ordinals, names, and libraries. The Functions window on the left lists the internal functions of the malware. The status bar at the bottom indicates "Line 85 of 117" and "Line 1 of 53".

Address	Ordinal	Name	Library
0000000000407000		RegSetValueExA	ADVAPI32
0000000000407004		RegCreateKeyExA	ADVAPI32
000000000040700C		SizedResource	KERNEL32
0000000000407010		LockResource	KERNEL32
0000000000407014		LoadResource	KERNEL32
0000000000407018		VirtualAlloc	KERNEL32
000000000040701C		GetModuleFileNameA	KERNEL32
0000000000407020		GetModuleHandleA	KERNEL32
0000000000407024		FreeResource	KERNEL32
0000000000407028		FindResourceA	KERNEL32
000000000040702C		CloseHandle	KERNEL32
0000000000407030		GetCommandLineA	KERNEL32
0000000000407034		GetVersion	KERNEL32
0000000000407038		ExitProcess	KERNEL32

In base alle funzioni richiamate all'interno delle librerie, si può ipotizzare che si tratti di un dropper, ovvero un programma malevolo che al suo interno contiene un malware.

I dropper rappresentano una tipologia di malware con caratteristiche ben precise che li distinguono da altri tipi di minacce informatiche. La loro funzione primaria è quella di distribuire e attivare altri malware all'interno di un sistema infetto. Per compiere il compito, i dropper sfruttano diverse tecniche, tra cui l'utilizzo di specifiche API per estrarre il malware contenuto al loro interno.

LE API CHIAVE PER L'ESTRAZIONE DEI MALWARE

- **FindResource()**: funzione che permette di identificare la risorsa contenente il malware all'interno del file eseguibile del dropper
- **LoadResource()**: individuata la risorsa, la carica in memoria, rendendola disponibile per l'estrazione
- **LockResource()**: funzione che blocca la risorsa in memoria, impedendo la modifica o la sovrascrizione durante il processo di estrazione
- **SizeOfResource**: determina la dimensione della risorsa contenente il malware, garantendo che venga estratta la quantità corretta di dati

La funzione chiamata all'indirizzo di memoria **00401021** sembra essere responsabile della creazione di una chiave di registro e utilizza la funzione **RegCreateKeyExA** della libreria ADVAPI32.DLL.

```
IDA - C:\Users\user\Desktop\MALWARE\Build_Week_3\Malware_Build_Week_U3.exe
File Edit Jump Search View Debugger Options Windows Help
File Edit Jump Search View Debugger Options Windows Help
Functions window x IDA ViewA x Hex ViewA x Structures x En Enums x Imports x Exports
Function name
sub_401000
sub_401080
main
sub_401299
close
write
fopen
open
strchr
stat
ams_exit
fast_error_exit
stbuf
fbuf
sub_401679
.cbData = dword ptr 0Ch
.text:00401000 push    ebp
.text:00401000 mov     ebp, esp
.text:00401001 push    ecx
.text:00401003 push    0
.text:00401004 push    eax, [ebp+hObject]
.text:00401006 push    eax, [phkResult]
.text:00401009 push    0
.text:0040100A push    0
.text:0040100C push    0
.text:00401011 push    0
.text:00401013 push    0
.text:00401015 push    0
.text:00401017 push    offset SubKey ; "SOFTWARE\Microsoft\Windows NT\CurrentVersion\Run\"
.text:0040101C push    80000002h ; hKey
.text:00401021 call    ds:RegCreateKeyExA
.text:00401027 test    eax, eax
.text:00401029 jz     short loc_401032
```

I parametri passati dalla funzione sono:

- hKEY
- SubKey
- dwFlags (dwOptions)
- lpSecurityAttributes
- Reserved
- ipdwDisposition

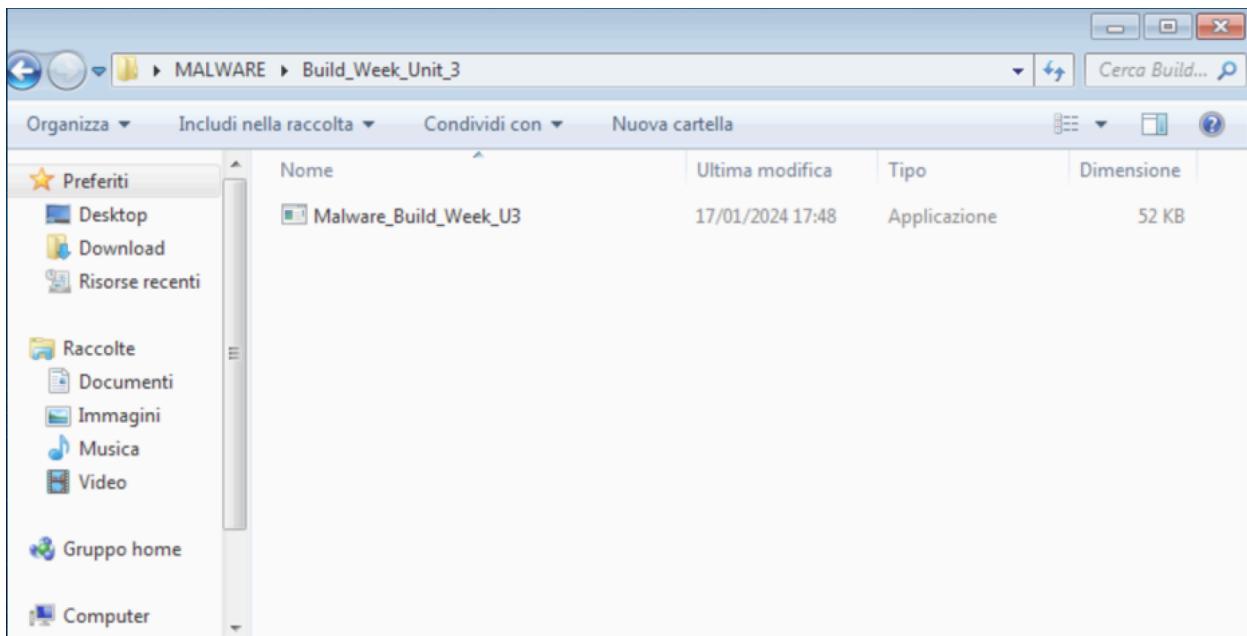
L'oggetto rappresentato dal parametro alla locazione di memoria 00401017 è molto probabilmente una stringa.

Essa è con tutta probabilità il percorso della chiave di registro che la funzione alla locazione 00401021 sta tentando di creare ed è probabilmente memorizzata come stringa null-terminata, ovvero terminante in un carattere nullo (\0).

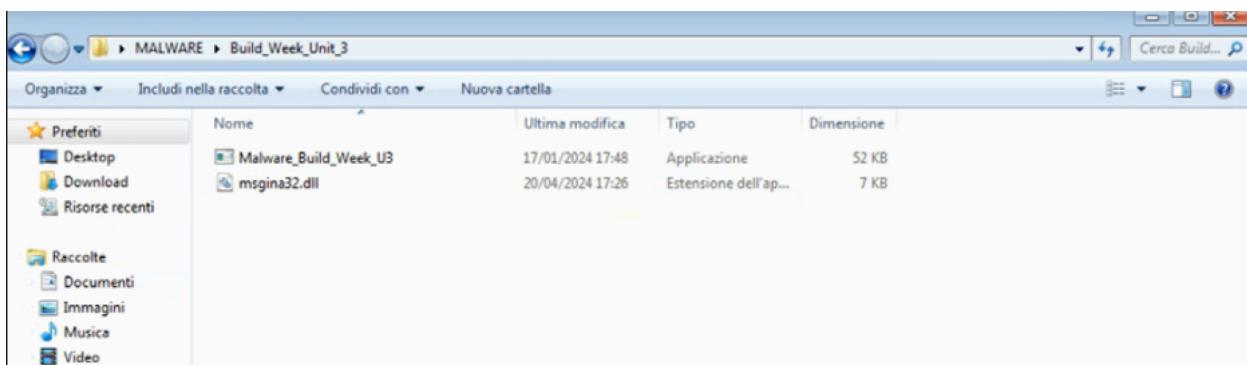
Le istruzioni comprese tra gli indirizzi 00401027 e 00401029 controllano il valore di ritorno della chiamata alla funzione RegCreateKeyExA e saltano a una routine di gestione degli errori se la chiamata fallisce.

- **00401027**: esegue un confronto tra il valore contenuto nel registro ExA e il valore zero
- **00401029**: istruzione condizionale di salto
- **istruzione successiva**: se il valore è uguale a zero, il programma salterà all'istruzione successiva situata a otto byte di distanza. Se il valore è diverso da zero, il programma continuerà a eseguire l'istruzione successiva a questa

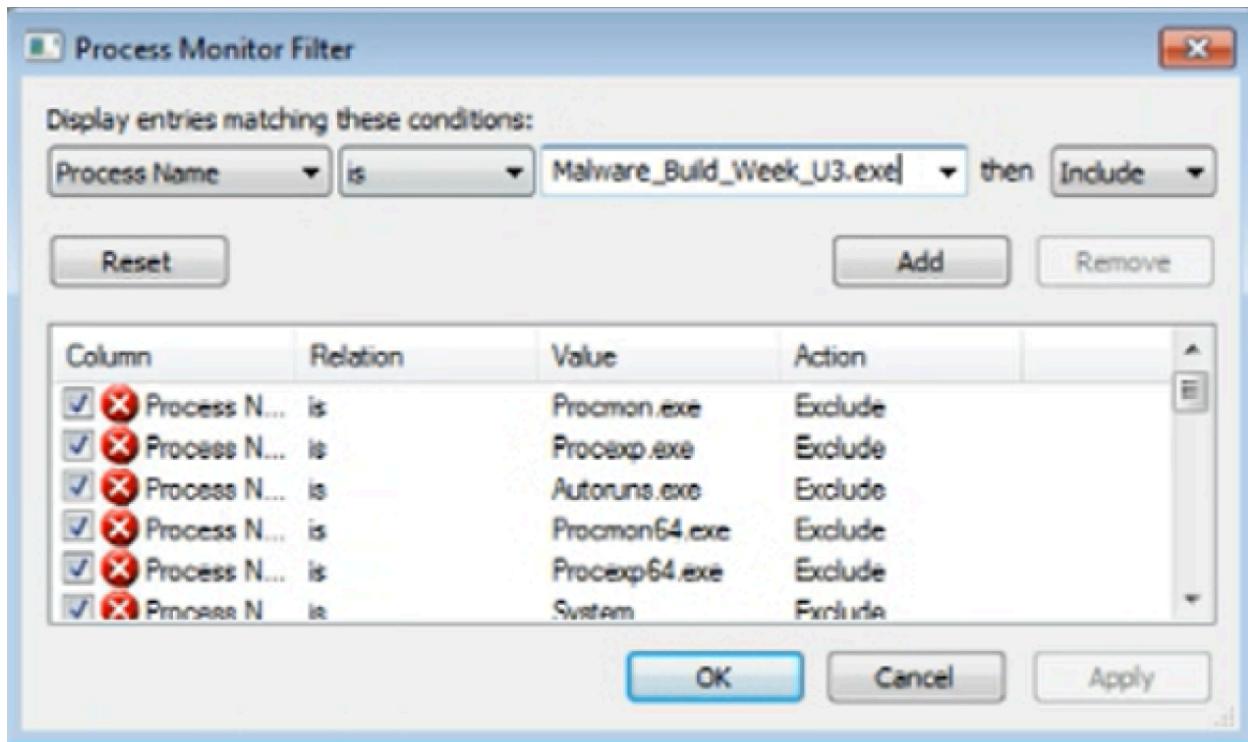
ANALISI DINAMICA



Dopo aver fatto il doppio click è apparsa l'estensione dell'app msgina32.dll, questo file è un componente critico del processo di accesso a Windows.

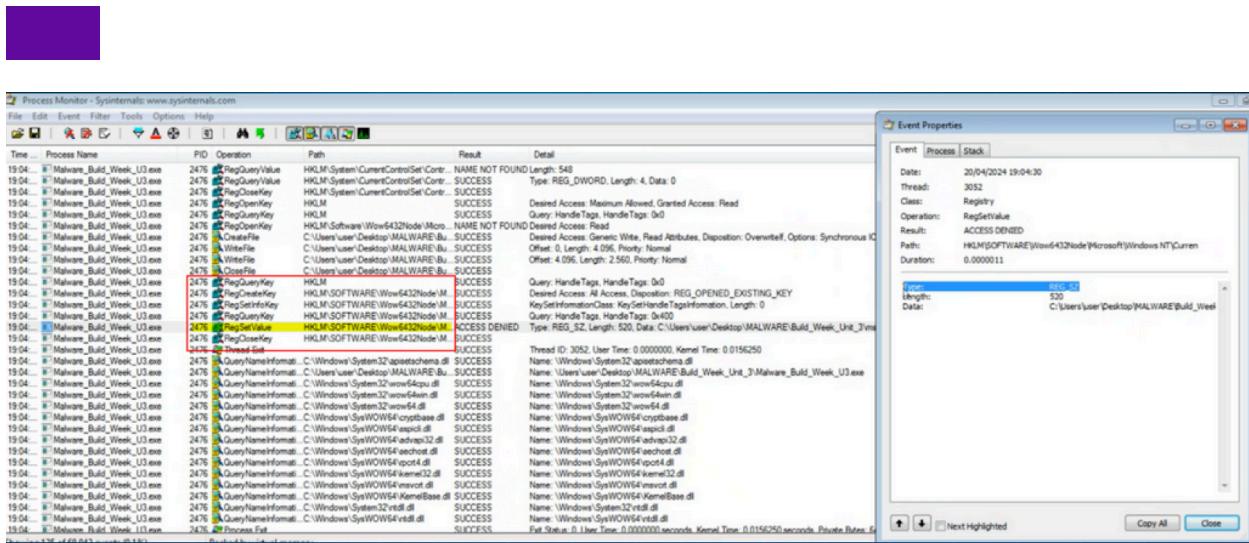


Avviamo Process Monitor ed analizziamo i risultati filtrando:



Possiamo notare che viene creata la chiave di registro HKLM, abbreviazione di HKEY_LOCAL_MACHINE, è una delle chiavi principali del registro di sistema di Windows.

Contiene informazioni di configurazione cruciali per il funzionamento del sistema operativo e dei programmi installati. A differenza di altre chiavi del registro di sistema che riguardano profili utente specifici, HKLM memorizza impostazioni globali accessibili a tutti gli utenti del computer.



Viene fatta una chiamata di sistema (CreateFile) che creala msgina.dll nella cartella del malware e a seguire vediamo la write file che inserisce il contenuto malevolo e poi la close file.

2604	RegQueryKey	HKLM
2604	RegOpenKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Diagnostics
2604	CreateFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	WriteFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	CloseFile	C:\Users\user\Desktop\MALWARE\Build_Week_Unit_3\msgina32.dll
2604	RegQueryKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegCreateKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegSetInfoKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegQueryKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	RegSetValue	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL
2604	RegCloseKey	HKLM\Software\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Winlogon
2604	Thread Exit	

CONCLUSIONI

Il comportamento globale del malware è quello della creazione e apertura della chiave di registro Winlogon con inserimento nella stessa del valore che punta alla DLL malevola creata dopo la sua esecuzione. Essendo la DLL in questione fondamentale per l'autenticazione degli utenti su Windows, possiamo presupporre che lo scopo sia di registrare le autenticazioni da parte degli utenti di sistema per poi impossessarsene.