





Analisi di AdwereCleaner.exe



1. Preparazione dell'Ambiente

Per garantire un'analisi sicura, la prima fase ha previsto la configurazione di un ambiente controllato. Questo passaggio è fondamentale per evitare la diffusione accidentale del malware e per poter ripristinare il sistema in caso di malfunzionamenti.

-  **Isolamento della VM:** La macchina virtuale FLARE è stata isolata dalla rete per prevenire la propagazione del malware.
-  **Snapshot della VM:** È stato creato uno snapshot della VM per consentire il ripristino rapido in caso di modifiche indesiderate.

Il corretto isolamento ha permesso di procedere con l'analisi statica in totale sicurezza.



2. Analisi Statica Preliminare

L'analisi statica ha lo scopo di raccogliere informazioni sul file senza eseguirlo, al fine di identificare caratteristiche sospette.



Identificazione del File

Attraverso l'uso del comando `certutil -dump AdwereCleaner.exe > outputcertutil.txt`, sono state ottenute informazioni di base sul file:

Image Not Showing

Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

- **Firma MZ (4D 5A):** Conferma che il file è un eseguibile Windows.
- **Firma PE (50 45 00 00):** Indica che si tratta di un Portable Executable.
- **Sezioni identificate:** `.text` (codice eseguibile), `.rdata` (dati di sola lettura), `.data` (dati modificabili), `.rsrc` (risorse).



Calcolo degli Hash e Confronto su VirusTotal

♦ Calcolo degli Hash

Comandi eseguiti:

```
certutil -hashfile AdwereCleaner.exe MD5  
certutil -hashfile AdwereCleaner.exe SHA1  
certutil -hashfile AdwereCleaner.exe SHA256
```

❗ Image Not Showing

Possible Reasons

- The image was uploaded to a note which you don't have access to
- The note which the image was originally uploaded to has been deleted

[Learn More →](#)

Risultati:

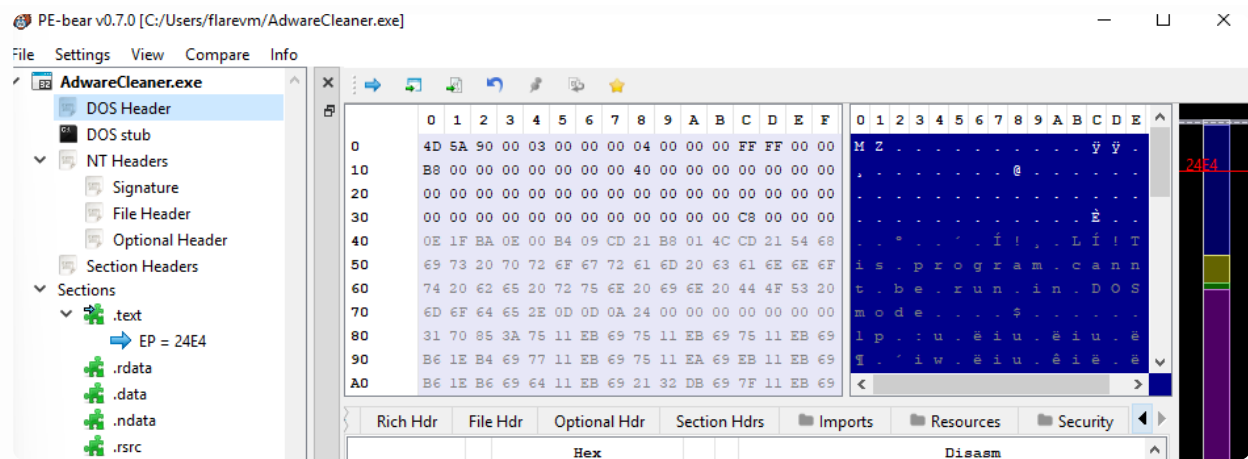
- **MD5:** 248aadd395ffa7ffb1670392a9398454
- **SHA1:** c53c140bbdeb556fca33bc7f9b2e44e9061ea3e5
- **SHA256:** 51290129cccca38c6e3b4444d0dfb8d848c8f3fc2e5291fc0d219fd642530adc
- **Risultati VirusTotal:**
 - 🚩 **53 su 70 motori antivirus** hanno rilevato il file come malevolo.
 - **Categorie di minaccia:** Trojan, FakeAV.
 - **Famiglie identificate:** porcupine, mint, boy2napig.

Il rilevamento su VirusTotal ha suggerito la necessità di un'analisi approfondita tramite strumenti specifici per i file PE.

🧩 3. Analisi PE con PE-bear

Con PE-bear, l'analisi si è concentrata sulla struttura interna del file per identificare anomalie che potrebbero indicare la presenza di codice malevolo.

- **Verifica delle firme:**
 - **MZ e PE:** Conferma della struttura valida del file.



• Analisi del TimeDateStamp:

- **Data rilevata:** 25 dicembre 2013 🗓️ (possibile data falsificata per confondere gli analisti).

Disasm	General	Strings	DOS Hdr	Rich Hdr	File Hdr	Optional Hdr	Section Hdrs	Imports
Offset	Name	Value	Meaning					
CC	Machine	14c	Intel 386					
CE	Sections Count	5	5					
D0	Time Date Stamp	52ba66b5	mercoledì, 25.12.2013 05:01:41 UTC					
D4	Ptr to Symbol Table	0	0					
D8	Num. of Symbols	0	0					
DC	Size of OptionalHeader	e0	224					
DE	Characteristics	10F						
		1	Relocation info stripped from file.					
		2	File is executable (i.e. no unresolved external references).					
		4	Line numbers stripped from file.					
		8	Local symbols stripped from file.					
		100	32 bit word machine.					

• Sezione sospetta: .ndata

- Dimensione virtuale anomala.
- Potenziale presenza di codice offuscato o payload nascosti.

Disasm	General	Strings	DOS Hdr	Rich Hdr	File Hdr	Optional Hdr	Section Hdrs	Imports
Name	Raw Addr.	Raw size	Virtual Addr.	Virtual Size	Characteristics	Ptr to Reloc.	Num. of Reloc.	Num. of Symbols
> .text	400	5E00	1000	5DE2	60000020	0	0	0
> .rdata	6200	1400	7000	12DA	40000040	0	0	0
> .data	7600	400	9000	25498	C0000040	0	0	0
> .ndata	0	0	2F000	8000	C0000080	0	0	0
> .rsrc	7A00	B400	37000	B268	40000040	0	0	0

L'identificazione di una sezione anomala ha portato alla necessità di un'analisi statica avanzata tramite reverse engineering.

Analisi Statica Avanzata

Disassemblaggio

- **Strumenti Utilizzati:**

- **Ghidra** framework utilizzato per il reverse engineering del malware

- **Passaggi**

- Avvio di **Ghidra**
- Creazione di un nuovo progetto
- Importazione del malware all'interno del software

Risultati Ottenuti

L'analisi statica avanzata del malware ha rivelato diverse funzionalità malevole, evidenziate sia nel codice Assembly che nella sua decompilazione in C. L'analisi si concentra su tre aspetti fondamentali: **evasione e anti-analysis, persistenza nel sistema, manipolazione di file e directory**, oltre a potenziali azioni dannose.

1. Comparazione tra Assembly e C

L'analisi della decompilazione del malware mostra chiaramente la corrispondenza tra codice Assembly e C, rivelando il comportamento malevolo in una forma più leggibile.

1.1. Evasione e Anti-Analysis

Il malware impiega tecniche avanzate per evitare il rilevamento, come il controllo della riga di comando, il rilevamento degli ambienti di analisi e il caricamento dinamico delle API di Windows.

- **Assembly:**

- `GetCommandLineA()` → Controlla i parametri di avvio per verificare se il programma è in esecuzione in un ambiente di debugging o analisi.
- `GetTickCount()` → Misura il tempo di esecuzione per identificare l'esecuzione in una sandbox (che spesso introduce ritardi nell'esecuzione).

- `LoadLibraryA()` e `GetProcAddress()` → Caricano dinamicamente le funzioni API di Windows per nascondere chiamate sensibili.
- `GetVersion()` → Determina la versione di Windows per adattarsi a specifiche configurazioni o evitare ambienti virtualizzati.

- **C (Decompilato):**

- La funzione `entry()` chiama `GetCommandLineA()` per estrarre la riga di comando.
- `FUN_00405ebe()` richiama `GetProcAddress()` e `LoadLibraryA()` per caricare le librerie necessarie.
- `FUN_0040566a()` manipola stringhe per nascondere il percorso del file eseguibile e rendere più difficile l'analisi.

Queste tecniche di evasione permettono al malware di non essere immediatamente rilevato da ambienti di analisi automatizzati.

1.2. Persistenza nel Sistema

Per garantire la propria esecuzione dopo il riavvio del sistema, il malware modifica il registro di Windows e le variabili d'ambiente.

- **Assembly:**

- `SetEnvironmentVariableA()` → Modifica le variabili d'ambiente per influenzare il comportamento del sistema.
- `RegSetValueExA()` → Scrive chiavi di registro malevole sotto `HKCU\Software\Microsoft\Windows\CurrentVersion\Run`.
- `GetSystemDirectoryA()` e `GetWindowsDirectoryA()` → Accede a directory di sistema sensibili per copiare se stesso in percorsi nascosti.

- **C (Decompilato):**

- `FUN_00405b93()` copia stringhe nei parametri delle funzioni di sistema.
- `FUN_004030b0()` crea directory e manipola percorsi per l'esecuzione persistente.
- `FUN_00403542()` chiude e riapre handle di sistema per nascondere il proprio processo.

La modifica del registro e la creazione di directory specifiche sono indicativi di un malware con obiettivi di persistenza.

1.3. Manipolazione di File e Directory

Il malware esegue diverse operazioni su file e cartelle, spesso sovrascrivendo o copiando file eseguibili per auto-propagarsi.

- **Assembly:**

- `CreateDirectoryA()` → Crea directory temporanee per eseguire codice senza attirare attenzione.
- `CopyFileA()` → Copia file (potenzialmente se stesso) in posizioni nascoste.
- `DeleteFileA()` → Cancella file dopo averli utilizzati, per evitare il rilevamento.
- `WriteFile()` → Scrive dati su disco, potenzialmente un payload eseguibile.

- **C (Decompilato):**

- `FUN_00405bb5()` gestisce la copia di file e la scrittura su percorsi nascosti.
- `FUN_0040566a()` manipola i nomi dei file per rendere più difficile il tracciamento delle operazioni.
- `FUN_004053a7()` si occupa della gestione di messaggi di errore, utile per nascondere eventuali crash.

Queste funzioni indicano che il malware potrebbe installare copie di se stesso in diverse directory per aumentare le probabilità di esecuzione.

2. Azioni Dannose Identificate

L'analisi del codice mostra che il malware è in grado di eseguire operazioni potenzialmente dannose, tra cui:

- **Avvio di Processi Malevoli:**

- `CreateProcessA()` viene chiamata per eseguire un nuovo processo (potenzialmente una variante del malware).
- `ShellExecuteA()` può essere usata per lanciare URL malevoli o script.

- **Esecuzione di Codice Esterno:**

- `WriteFile()` scrive file eseguibili sul disco, per poi eseguirli tramite `CreateProcessA()`.
- `MoveFileExA()` e `CopyFileA()` spostano file in directory nascoste per evitare il rilevamento.

- **Spegnimento del Sistema:**

- `ExitWindowsEx(2, 0)` forza lo spegnimento o il logout dell'utente.
- `ExitProcess(2)` chiude il programma, ma può anche essere usato per terminare processi di sistema critici.

3. Analisi delle API Importate

3.1. ADVAPI32.DLL (Gestione del Registro di Windows)

- `RegOpenKeyExA()` → Apre una chiave di registro.
- `RegQueryValueExA()` → Legge un valore dal registro.
- `RegSetValueExA()` → Scrive un valore nel registro.
- `RegDeleteKeyA()` / `RegDeleteValueA()` → Cancella chiavi e valori dal registro.

3.2. KERNEL32.DLL (Gestione di File, Processi e Memoria)

- `CreateFileA()` / `DeleteFileA()` / `MoveFileA()` → Accesso al file system.
- `CreateProcessA()` / `CreateThread()` → Creazione di nuovi processi e thread.
- `LoadLibraryA()` / `GetProcAddress()` → Caricamento dinamico di DLL.
- `WriteFile()` / `ReadFile()` → Scrittura e lettura di file.

3.3. SHELL32.DLL / USER32.DLL / OLE32.DLL

- `ShellExecuteA()` → Esegue file o URL.
- `MessageBoxA()` → Potrebbe mostrare messaggi all'utente.

4. Conclusione

Il malware analizzato è progettato per:

- Evadere il rilevamento con tecniche di anti-analysis e anti-debugging.
- Mantenere la persistenza nel sistema.
- Manipolare file e directory.
- Eseguire codice dannoso.

Per mitigarlo, è fondamentale un'analisi dinamica e l'uso di strumenti di monitoraggio attivi.



Analisi Dinamica del Malware: AdwCleaner.exe



Introduzione

L'analisi dinamica del malware **AdwCleaner.exe** è stata condotta per osservare il comportamento del programma in un ambiente controllato. L'obiettivo era identificare attività sospette, modifiche al sistema e tentativi di comunicazione con l'esterno. Per raggiungere questi scopi, sono stati utilizzati diversi strumenti di monitoraggio:

- **Regshot** per il confronto dello stato del registro di sistema.
- **Process Monitor** e **Process Explorer** per tracciare attività su file, processi e registro.
- **FakeNet-NG** e **Wireshark** per l'analisi del traffico di rete.
- **Autoruns** per rilevare meccanismi di persistenza.

1. Acquisizione dello Stato Iniziale del Registro

Per individuare eventuali modifiche al registro di sistema, è stato avviato **Regshot** per acquisire uno snapshot dello stato del registro prima dell'esecuzione del malware.

2. Monitoraggio delle Attività di Sistema

2.1 Configurazione di Process Monitor (Procmon)

Strumento: Procmon

Comando utilizzato:

```
.\Procmon64.exe /Quiet /Backingfile C:\Users\FLARE-VM\Desktop\procmon_log_Adwerc1
```

Spiegazione del comando:

- `Procmon64.exe` : Avvia la versione a 64 bit di Process Monitor.
- `/Quiet` : Esecuzione senza interfaccia grafica per ridurre il carico sulla VM.
- `/Backingfile` : Specifica il percorso del file di log in cui vengono salvati gli eventi registrati.

Passaggi eseguiti:

1. Avvio di Procmon.
2. Configurazione dei filtri per isolare le attività del malware.
3. Avvio del monitoraggio (Capture attivo).
4. Esecuzione del malware per osservare il comportamento.
5. Salvataggio del log per l'analisi.

2.2 Configurazione di Process Explorer

Obiettivo: Monitorare i processi in esecuzione per identificare eventuali anomalie.

Passaggi:

1. Avvio di Process Explorer.
 2. Osservazione delle proprietà e dei thread dei processi legati al malware.
-

3. Monitoraggio del Traffico di Rete

3.1 Avvio di FakeNet-NG

Prima dell'esecuzione del malware, è stato avviato **FakeNet-NG** per monitorare e simulare il traffico di rete generato dal malware.

3.2 Esecuzione del Malware

Il malware è stato avviato tramite PowerShell con il comando:

```
Start-Process .\AdwereCleaner.exe
```

4. Risultati Ottenuti

4.1 Comportamenti Identificati All'Avvio

- **Caricamento di DLL di sistema:** `kernel32.dll` , `cryptnet.dll` .
- **Creazione di thread e processi:** Funzionamento di base senza attività dannose evidenti.

4.2 Comportamenti Sospetti Dopo Aver Premuto il Tasto "Scan"

Modifiche al File System:

- **Accesso al file hosts:** Tentativo di lettura con permessi di *Generic Read* (potenziale manipolazione DNS).

Accesso a DLL Critiche:

- `urlmon.dll` : Gestione delle connessioni di rete e download di contenuti web.
- `sspicli.dll` : Gestione della sicurezza e delle autenticazioni.

Caricamento di Risorse di Sistema:

- DLL caricate: `mscorlib.dll` , `webauthn.dll` , `cryptnet.dll` .

Creazione di Thread Multipli:

- **Thread ID 3244:** Attività specifiche con tempo di utilizzo in modalità utente di 0,0159250.

Traffico di Rete Monitorato con FakeNet-NG:

- **Richieste DNS:** `ctldl.windowsupdate.com` , `crl.usertrust.com` , `ocsp.comodoca.com` .
- **Traffico ICMP:** Pacchetti inviati a `192.168.34.112` .

5. Analisi delle Modifiche al Registro di Sistema (Regshot)

Chiavi Aggiunte (45 voci):

- Modifiche ai certificati di sistema e alle sessioni di esplorazione di Windows.
- Esempio:

```
HKLM\SOFTWARE\Microsoft\SystemCertificates\AuthRoot\Certificates\E12DFB4B41D7D9C32B30514BAC1D81D8385E2D46
```

Valori Aggiunti (101 voci):

- Avvio automatico di AdwCleaner:

```
HKU\...\Run\AdwCleaner
```

Valori Modificati (28 voci):

- Modifiche nella gestione audio e nei file recenti di Windows.

6. Analisi dei Processi (Process Explorer)

Processi Identificati:

- `AdwCleaner (PID 312)` e `AdwCleaner (PID 3656)` .

Analisi dei Threads:

- **Thread ID 5108 e 2720:** Attività legate alla gestione di interfacce GUI.

Call Stack:

- Funzioni principali come `NtUserWaitMessage` in `win32u.dll` .

Proprietà dei Processi:

- Interazioni con componenti di sistema tipiche di applicazioni Windows legittime.

7. Analisi del Traffico di Rete (Wireshark e FakeNet-NG)

Domini Contattati:

- `www.vikingwebscanner.com` , `ocsp.usertrust.com` , `ctldl.windowsupdate.com` .

Tipologie di Richieste:

- **DNS:** Risoluzione di nomi di dominio sospetti.
- **HTTP GET/POST:** Accesso a script esterni e possibile esfiltrazione di dati.

Possibili Indicatori di Compromissione (IoCs):

- Domini sospetti non legati a funzioni legittime.
- Uso di richieste OCSP e CRL per mascherare comunicazioni malevole.

8. Analisi della Persistenza (Autoruns)

Configurazione:

- Avvio di Autoruns e analisi delle voci di avvio automatico.

Risultati:

- Voce sospetta individuata:




`HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\AdwCleaner`

Conclusioni:

- Il malware è configurato per avviarsi automaticamente ad ogni riavvio del sistema.
- Presenza di meccanismi di persistenza tramite chiavi di registro di Windows.

Conclusioni Generali

L'analisi dinamica di **AdwCleaner.exe** ha evidenziato:

-  **Tentativi di connessione a server esterni sospetti.**
-  **Modifiche persistenti al registro di sistema.**
-  **Comportamenti che mascherano attività dannose dietro funzionalità legittime.**

Questi risultati suggeriscono la natura malevola del file analizzato, con capacità di persistenza e comunicazione esterna potenzialmente pericolose.

7. Conclusioni

L'analisi complessiva ha evidenziato:

- 🎯 **Evasione del rilevamento:** Tecniche avanzate di anti-debugging e anti-analysis.
- 🔒 **Persistenza nel sistema:** Meccanismi per garantire l'avvio automatico e la sopravvivenza del malware.
- 📁 **Manipolazione di file e registro:** Alterazioni significative del sistema operativo.
- 🌐 **Comunicazione con server remoti sospetti:** Potenziali attività di esfiltrazione dati o controllo remoto.

⚠ **Raccomandazione finale:** Sebbene non siano stati identificati payload attivi, il comportamento complesso e le tecniche utilizzate indicano un rischio significativo. È consigliata un'ulteriore analisi del file secondario `6AdwCleaner.exe` per valutare appieno il potenziale impatto.