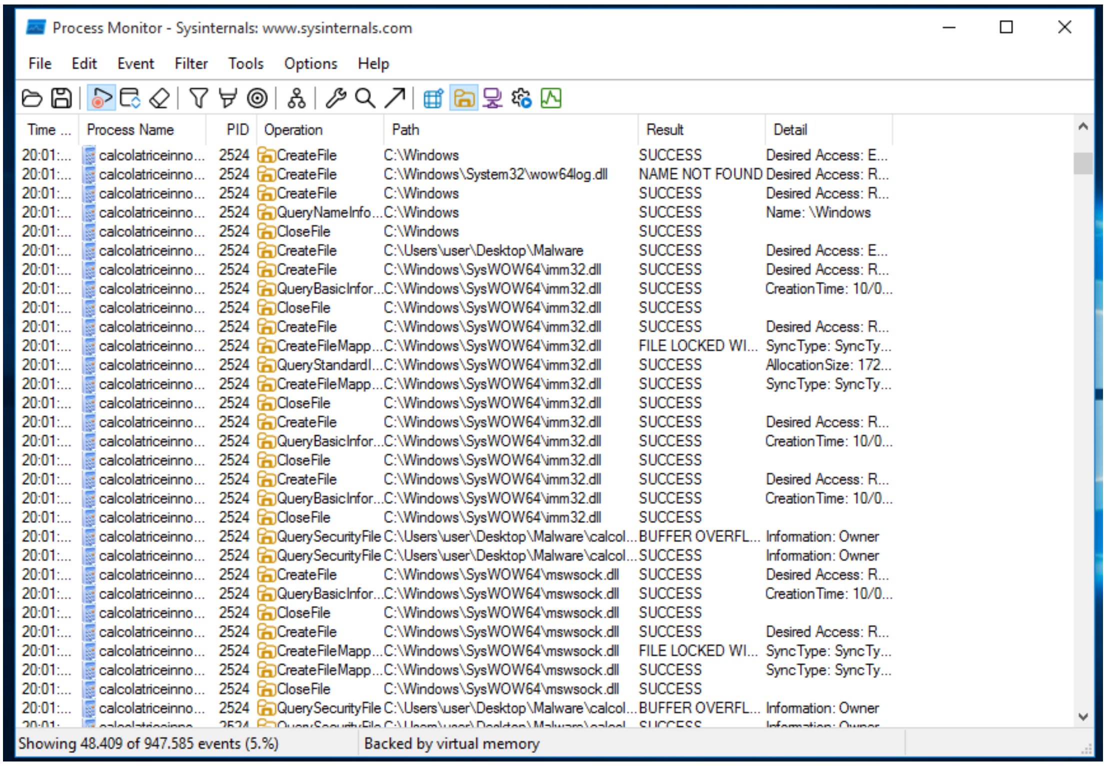
W22D1 – Analisi dinamica di base

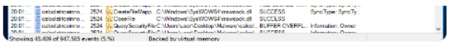
Esercizio obbligatorio

Ho aperto Process Monitor ed ho eseguito il malware per dare inizio all'analisi dinamica. Filtrando i risultati, ho trovato tutte le azioni del malware sul file system.



Ho esaminato quanto trovato da Process Monitor ed ho visto che il malware viene eseguito dalla cartella C:\Users\user\Desktop\Malware e tenta di accedere a vari file e librerie DLL. Infatti tramite C:\Windows\SysWOW64 cerca di accedere e caricare numerose librerie DLL come apphelp.dll, kernel32.dll, user32.dll, winmm.dll e ws2_32.dll al fine di svolgere le sue attività malevole senza essere notato, sfruttando funzioni legittime. Esegue anche operazioni di lettura su alcune dll come apphelp.dll e ws2_32.dll e crea numerosi file mapping per provare una iniezione di codice o per manipolare la memoria, grazie anche alle librerie che aveva caricato e che erano evidenti anche nell'analisi statica. Indaga anche con delle query la sicurezza dei file, magari per modificarne delle impostazioni o i permessi e potrebbe tentare delle connessioni di rete, magari verso IP malevoli, grazie alle librerie ws2_32.dll e mswsock.dll. Con il caricamento delle librerie imm32.dll e user32.dll potrebbe cercare di intercettare l'input utente, cosa che fa pensare magari ad un keylogger e il malware in generale sta raccogliendo informazioni riguardo al sistema host con delle query sui file di sistema. Il malware accede anche a librerie DLL di sistema per cercare di nascondere la sua natura malevola e cerca di agganciarsi a file di sistema (file locked) per non essere trovato dai sistemi di difesa. Con l'operazione CreateFileMapping il malware potrebbe cercare di iniettare del codice, di manipolare la memoria oppure potrebbe tentare un DLL hijacking, con il quale va a sostituire una libreria legittima con una libreria malevola per eseguire dei suoi payload. Con le query riguardo alle informazioni di sicurezza dei file il malware potrebbe cercare di dare origine ad un attacco di buffer overflow, con la relativa possibilità di provare una privilege escalation. Il file wow64log.dll, che non è stato trovato (name not found), rappresenta un altro elemento preoccupante, per cui il malware potrebbe tentare di caricare una propria libreria DLL, magari per eseguire un payload, oppure per verificare se l'esecuzione sta avvenendo in una sandbox. In generale, il malware è ancora in una fase iniziale dell'esecuzione, per cui sta caricando delle librerie utili con cui poi in un secondo momento catturare input utente e stabilire connessioni verso l'esterno, comportandosi come un trojan o essendo parte di un attacco più strutturato. Potenzialmente il malware potrebbe anche tentare esso stesso diversi attacchi, ad esempio un DDL hijacking o un buffer overflow ed utilizza molte tecniche per non essere rilevato come malware, anche con l'esecuzione veloce delle varie istruzioni.

Ho fatto analizzare lo screenshot di ProcMon delle attività del malware con il file system da ChatGPT e questo è il risultato che ho ottenuto.



I am performing a basic dynamic analysis of this malware using Process Monitor. Describe in details the actions of the malware impacting the file system.

From the Process Monitor (ProcMon) screenshot, we can extract several key insights about the behavior of the malware impacting the file system. Here's a detailed breakdown of the malware's actions:

1. File Creation and Modification

- The process, `calcolatricinema`, is actively creating, accessing, and interacting with various files.
- **File Creation/Modification**: The malware creates and writes to files in several key locations. For example:
  - Files under `C:\Windows\SysWOW64\` like `imm32.dll`, `mswsock.dll`, and `winmm.dll` are being accessed.
  - It is also creating a file at `C:\Users\user\Desktop\Malware\calcol...`.

This suggests that the malware is attempting ↓ nteract with system libraries and possibly dropping its payloads or modifying legitimate files.

2. Suspicious File Locking

- Some files are locked, as indicated by the event: `FILE LOCKED WITH ONLY READERS` for `imm32.dll` and `mswsock.dll`.
- This behavior suggests that the malware is attempting to access or lock system files to maintain persistence or avoid detection. By locking a file, the malware could be preventing other processes from modifying or deleting it, ensuring that its malicious code remains active.

3. DLL and System File Access

- The malware is heavily interacting with key system DLLs:
  - `imm32.dll`, `winmm.dll`, and `mswsock.dll` are part of the Windows system, typically related to input (keyboard/mouse), multimedia operations, and network socket operations respectively. The fact that these files are being opened and accessed can indicate that the malware is trying to hook into system functions to perform various tasks (e.g., keylogging, network traffic interception).
  - `CreateFileMapping`, `QueryBasicInfoFile`, `CreateFile`, and `CloseFile` operations further reinforce that the malware is manipulating these libraries.

4. Potential Code Injection or DLL Hijacking

- Operations like `CreateFileMapping` and `ySecurityFile` for DLL files (`imm32.dll`, `mswsock.dll`, etc.) may suggest that the malware is attempting to manipulate how these

4. Potential Code Injection or DLL Hijacking

- Operations like `CreateFileMapping` and `QuerySecurityFile` for DLL files (`imm32.dll`, `mswsock.dll`, etc.) may suggest that the malware is attempting to manipulate how these libraries are loaded or injected. By using `CreateFileMapping`, the malware could be trying to inject code into these DLLs in memory or modify them for persistence.
- **DLL Hijacking**: The access pattern could also indicate a possible DLL hijacking technique, where the malware tries to replace or alter legitimate DLLs with malicious versions.

5. BUFFER OVERFLOW Flags

- The `BUFFER OVERFLOW` result, especially when querying security information (`QuerySecurityFile`) on `calcol...`, is indicative of the malware requesting more data than expected or probing for vulnerabilities.
- This may be a method to exploit system processes or escalate privileges by causing a buffer overflow within the security query context, potentially leading to system compromise.

6. Name Not Found for `wow64log.dll`

- The absence of `wow64log.dll` (NAME NOT FOUND) indicates that the malware may be attempting to load a non-existent or custom DLL that could serve as part of its payload or assist in its execution. This could also point to a ti-debugging technique, where malware checks for specific files to detect if it's running in a sandboxed environment.

7. File Access Timing

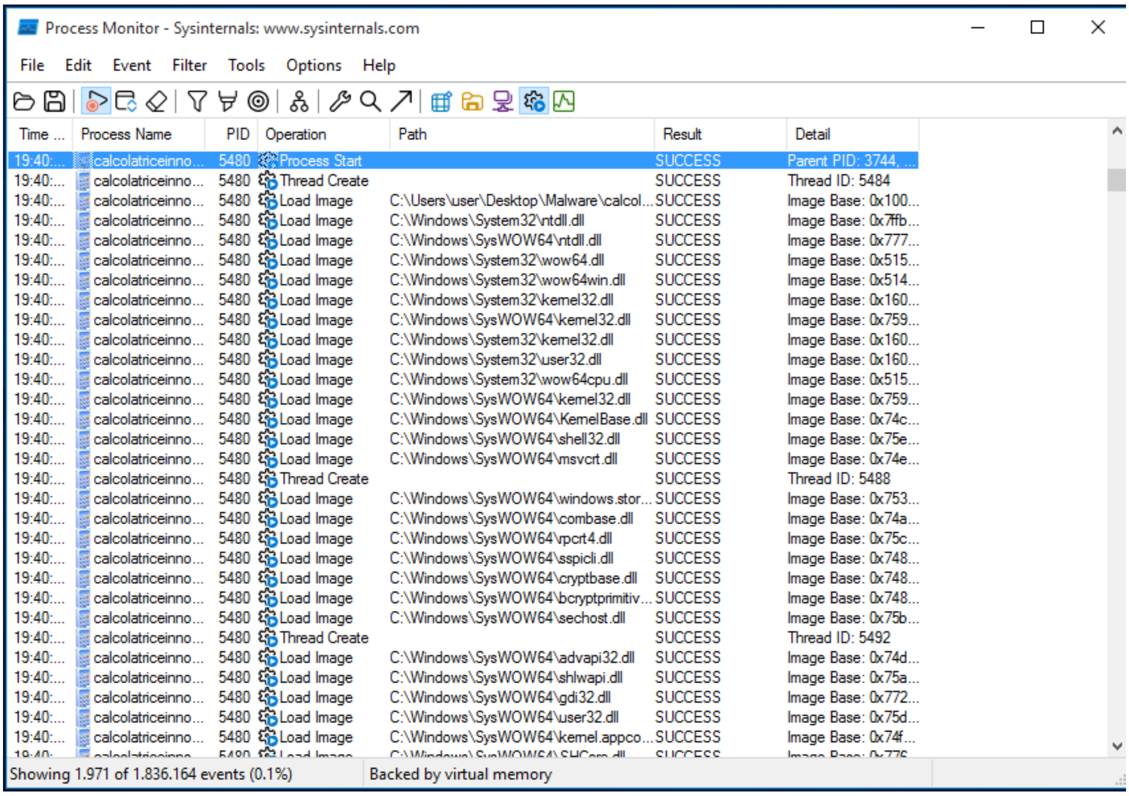
- The timestamps indicate that these file system operations are being carried out in quick succession, which aligns with typical malware behavior, rapidly attempting to establish persistence, inject code, or modify system libraries before being detected.

Summary:

- **Suspicious File Creation**: The malware creates new files, likely to drop payloads.
- **System File Access**: Key DLLs in `SysWOW64` are accessed for potential manipulation (hooking, injection).
- **File Locking**: Critical DLLs are being locked, which may prevent their modification by other processes.
- **Potential Exploit Tactics**: `BUFFER OVERFLOW` events and security queries indicate the malware might be trying to exploit vulnerabilities.
- **Persistence and Anti-Analysis**: The `NAME NOT FOUND` event, along with attempts to interact with critical DLLs, suggests that the malware may be engaging in anti-analysis techniques and trying to maintain persistence.

To conclude, the malware appears to be engaging in multiple behaviors typical of malicious software—modifying system files, interacting with DLL, possible code injection, and employing file-locking mechanisms to maintain control over compromised resources. Further analysis should

Ho controllato le attività del malware relative ai processi e ai thread con Process Monitor ed ho ottenuto il seguente risultato.



Time ...	Process Name	PID	Operation	Path	Result	Detail
19:40:...	calcolatriceinno...	5480	Process Start		SUCCESS	Parent PID: 3744, ...
19:40:...	calcolatriceinno...	5480	Thread Create		SUCCESS	Thread ID: 5484
19:40:...	calcolatriceinno...	5480	Load Image	C:\Users\user\Desktop\Malware\calcol...	SUCCESS	Image Base: 0x100...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x7ffb...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\ntdll.dll	SUCCESS	Image Base: 0x777...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x515...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\wow64win.dll	SUCCESS	Image Base: 0x514...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x160...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x759...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x160...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x160...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\System32\wow64cpu.dll	SUCCESS	Image Base: 0x515...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\kernel32.dll	SUCCESS	Image Base: 0x759...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\KernelBase.dll	SUCCESS	Image Base: 0x74c...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\shell32.dll	SUCCESS	Image Base: 0x75e...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\msvcrt.dll	SUCCESS	Image Base: 0x74e...
19:40:...	calcolatriceinno...	5480	Thread Create		SUCCESS	Thread ID: 5488
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\windows.stor...	SUCCESS	Image Base: 0x753...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\combase.dll	SUCCESS	Image Base: 0x74a...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\rpcrt4.dll	SUCCESS	Image Base: 0x75c...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\sspicli.dll	SUCCESS	Image Base: 0x748...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\cryptbase.dll	SUCCESS	Image Base: 0x748...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\bcryptprimitiv...	SUCCESS	Image Base: 0x748...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\sechost.dll	SUCCESS	Image Base: 0x75b...
19:40:...	calcolatriceinno...	5480	Thread Create		SUCCESS	Thread ID: 5492
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\advapi32.dll	SUCCESS	Image Base: 0x74d...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\shlwapi.dll	SUCCESS	Image Base: 0x75a...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\gdi32.dll	SUCCESS	Image Base: 0x772...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\user32.dll	SUCCESS	Image Base: 0x75d...
19:40:...	calcolatriceinno...	5480	Load Image	C:\Windows\SysWOW64\kernel.appco...	SUCCESS	Image Base: 0x74f...

Showing 1.971 of 1.836.164 events (0.1%) | Backed by virtual memory

Dall’analisi delle azioni su processi e threads, si nota che il malware carica numerose librerie di sistema Windows DLL e carica file da diverse cartelle di sistema come C:\Windows\System32 e C:\Windows\SysWOW64. Il malware viene eseguito e da subito si può analizzare il PID del processo padre, per risalire a come è stato eseguito il malware, magari per fini di analisi forense. Il malware crea numerosi thread con successo, per svolgere operazioni parallele e lo fa per nascondersi, sembrare un software legittimo e svolgere più operazioni contemporaneamente. Con l’esecuzione di thread multipli riesce a svolgere più operazioni contemporaneamente, carica componenti di sistema e probabilmente è in grado di rimanere persistente sul sistema. Alcuni dei thread aperti hanno il PID 5484 e 5492 e i vari thread creati caricano librerie DLL utili per funzioni di sistema, come ntdll.dll, kernel32.dll e wow64.dll e vi è il potenziale pericolo di una process injection, specie con librerie come ntdll.dll, kernel32.dll e mswsock.dll.

Ho fatto analizzare gli screenshot di ProcMon anche a ChatGPT e riporto il risultato dell’analisi effettuata.

Describe in details the actions of the malware involving processes and threads. Consider the screenshot of the analysis using ProcessMonitor

From the new screenshot of Process Monitor, we can observe the **process and thread activity** of the malware named `calcolatricinema`, showing key indicators of how it behaves in relation to system processes and threads. Here is a detailed breakdown of its actions:

1. Process Creation

- **Process Start (PID: 5480)**: The malware initiates a new process (`calcolatricinema.exe`) with PID 5480.
  - **Parent Process (PID: 3744)**: This is crucial for understanding how the malware is started. The parent process (PID 3744) suggests that another process triggered the execution of `calcolatricinema`. This could be either a legitimate application compromised by the malware or another malicious component responsible for starting this process.

2. Thread Creation

- Multiple **Thread Create** events are visible, [illegible]d to different Thread IDs such as `5484` and `5492`. These threads are associated with specific actions or interactions with system DLLs.

- The **Thread Create** operations are critical for understanding the internal execution behavior. By spawning multiple threads, the malware can handle different tasks simultaneously, such as interacting with the network, reading/writing files, or injecting code into other processes.

3. DLL Loading (Image Load)

The malware loads several key system DLLs into its memory space. This is a common action as malware often needs to leverage legitimate Windows libraries to perform various system functions or abuse them for malicious purposes. Here are some significant DLLs being loaded:

- **Core System DLLs**:
  - `ntdll.dll`: A core Windows system file involved in handling system calls and various low-level operations. The loading of `ntdll.dll` is crucial because many malware use it to access native Windows API functions.
  - `kernel32.dll` & `kernelbase.dll`: These are core libraries that handle many low-level functions such as memory management, process/thread creation, and I/O operations. Malware may use these for process injection, resource allocation, and managing I/O operations to persist in the system.
  - `wow64.dll`, `wow64cpu.dll`, `wow64win.dll`: These DLLs handle the execution of 32-bit applications on a 64-bit Windows system. The loading of these indicates that the malware may be a 32-bit application running on a 64-bit Windows environment, leveraging these DLLs for compatibility.
- **Multimedia and Network-related DLLs**:
  - `winmm.dll`: Handles multimedia tasks, like audio. Although not directly harmful, malware sometimes interacts with multimedia APIs to mask its activities or steal data.
  - `mswsock.dll`: This DLL handles network socket communication. The fact that this is being loaded suggests the malware might have networking capabilities, such as connecting to command-and-control (C2) servers, performing data exfiltration, or communicating with other infected hosts.
- **Security-related DLLs**:
  - `bcryptprimitives.dll`: Provides cryptographic services like hashing and encryption. Loading this DLL might indicate that the malware is performing encryption, possibly related to hiding its communication or even ransomware activity.
  - `advapi32.dll`: This DLL is used for advanced Windows API services, including access to the Windows Registry, user accounts, and process security tokens. This suggests the malware may be manipulating security settings or the Registry to maintain persistence or escalate privileges.
- **COM and Shell DLLs**:
  - `combase.dll` and `shell32.dll`: These DLLs are associated with COM (Component Object Model) functions and shell-related tasks like file explorer integrations. Malware often uses

these for file and directory manipulation or to interface with Windows Explorer for malicious purposes.

The sequential loading of these DLLs suggests the malware is preparing its environment by accessing critical system functionality and ensuring it has the necessary tools (DLLs) to operate.

4. Process Injection Potential

- By loading these system DLLs, especially ones like `ntdll.dll`, `kernel32.dll`, and `mswsock.dll`, the malware is likely preparing for or engaging in **process injection**. This is where the malware could inject code into legitimate processes using the loaded DLLs to hide its presence and evade detection by security tools.

5. Thread ID Assignment

- Each thread creation event shows a **Thread ID** like `5484` or `5492`. These threads likely represent distinct parts of the malware's functionality running in parallel. For example:
  - One thread may handle file system interactions (such as creating or modifying files).
  - Another thread might manage networking (like connecting to C2 servers or sending stolen data).
  - A different thread could be dedicated to maintaining persistence by monitoring or modifying system processes.

Summary of Actions:

1. **Process Creation**:
   - The malware process (`calcolatricinema.exe`) is started, with a parent process (PID: 3744), indicating that another process is responsible for its execution.
2. **Multiple Thread Creation**:
   - Several threads are created, each likely responsible for specific tasks such as file system manipulation, network communication, or injecting code into other processes.
3. **Loading System DLLs**:
   - The malware loads critical system DLLs related to system calls, threading, networking, security, and cryptographic operations. This suggests the malware is preparing for sophisticated actions that may involve network communication, security manipulation, or process injection.
4. **DLLs related to WOW64**:
   - DLLs like `wow64.dll` and `wow64cpu.dll` indicate that the malware is running in a 32-bit process on a 64-bit Windows system. This is common for backward compatibility but could also imply that the malware targets both 32-bit and 64-bit systems.
5. **Potential Process Injection**:
   - The combination of `ntdll.dll` and `kernel32.dll` being loaded may indicate that the

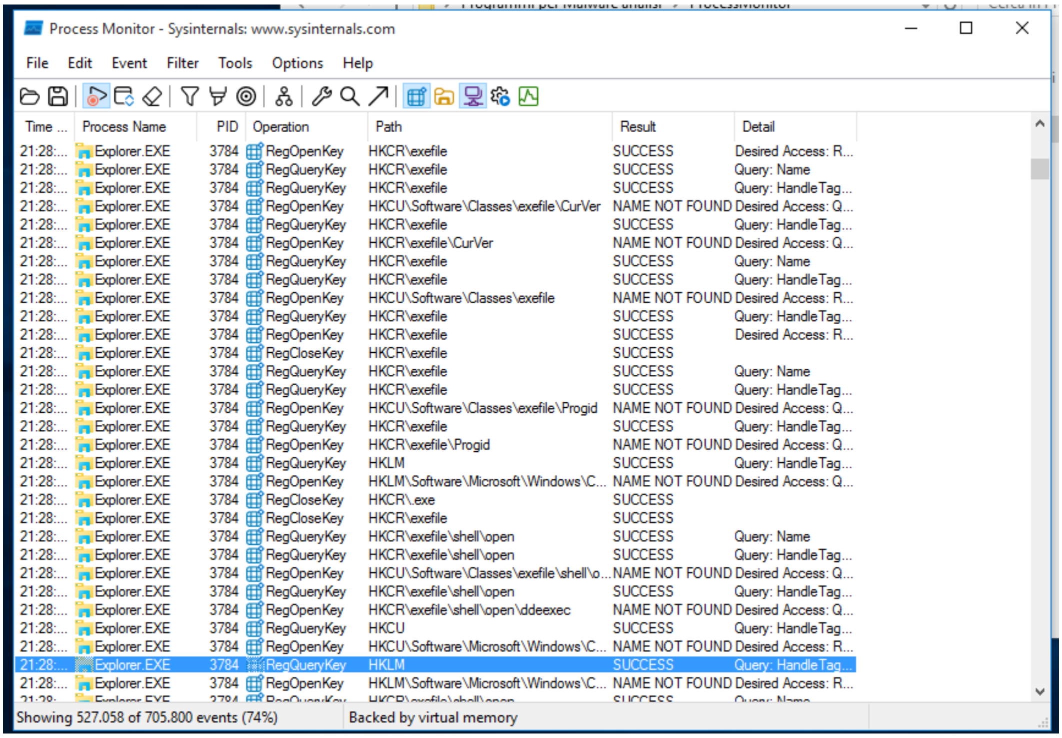
5. **Potential Process Injection**:
   - The combination of `ntdll.dll` and `kernel32.dll` being loaded may indicate that the malware is using these functions to inject code into other processes, hiding its malicious activities within legitimate processes.

Conclusion:

The malware (`calcolatricinema`) exhibits complex behavior involving process and thread management. It starts its process, spawns multiple threads to handle various malicious tasks, and loads numerous system DLLs. These DLLs give it access to core system functionality, enabling activities such as process injection, network communication, and potential cryptographic operations. The process appears to be carefully setting up its environment to remain hidden, evade detection, and persist on the infected system.

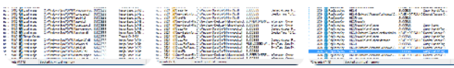
Esercizio facoltativo

Per completare l'analisi del malware con Process Monitor, ho analizzato anche le interazioni del malware con i registri di Windows e gli eventi legati alla rete.



Vi sono molte interazioni tra il malware e i registri di Windows con il processo Explorer.exe e i registri con cui interagisce sono HKCR\exefile, che serve per interagire con i file eseguibili, e i registri relativi alla shell di Windows come shell\open. Quest'ultimo potrebbe servire per lavorare con la shell e magari cambiare i path e procedere con un hijacking e in entrambi i casi il malware potrebbe tentare di modificare i registri per rimanere persistente sul sistema. Riassumendo il comportamento del malware, esso viene avviato dalla cartella C:\Users\user\Desktop\Malware e poi inizia a caricare varie librerie DLL, cercando di mascherarsi come un processo legittimo. L'analisi del processo padre legittimo, cioè Explorer.exe, fa pensare ad un tentativo di social engineering, per cui l'utente ha scaricato il malware pensando di scaricare una nuova calcolatrice innovativa per Windows. Il malware, poi, accede al registro di sistema ed opera nelle chiavi relative alle connessioni di rete e a WinSock, per conoscere e possibilmente modificare le impostazioni di rete. Anche il caricamento delle librerie ws2_32.dll e mswsock.dll fa pensare a delle operazioni di rete. Il malware, poi, accede al file system per aprire e leggere dei file di sistema e crea dei thread aggiuntivi per eseguire dei processi paralleli contemporaneamente e magari nascondersi dietro ad un processo legittimo aprendo nuovi thread. I numerosi thread possono anche far pensare ad un tentativo di collegarsi ad un server command-and-control. Le tempistiche di esecuzione sono rapide e il malware è particolarmente capace nell'offuscamento, perché non esegue operazioni particolarmente sospette o pericolose, però ha la potenzialità di fare molti danni e potrebbe anche trattarsi di un malware di ricognizione, in grado di studiare un sistema senza eseguire subito azioni malevole. Con "file locked with only readers" il malware accede a dei file di Sistema contenuti nelle librerie già esaminate e poi evita una successiva modifica dei file alterati, per non interferire con la sua attività malevola. L'interazione con un processo legittimo di sistema come Explorer.exe potrebbe rappresentare la chiave per il malware per diventare persistente sul sistema e potrebbe sfruttare la vulnerabilità del buffer overflow.

Ho interrogato anche ChatGPT riguardo alle ulteriori caratteristiche del malware che emergono da questa analisi dinamica tramite Process Monitor e questa è la risposta che è stata generata.



Tell me additional considerations about this malware based on the screenshots of the basic dynamic analysis using ProcessMonitor

Based on the three Process Monitor screenshots you've shared, there are additional considerations and insights that can be drawn regarding the malware's behavior. Here's a more detailed analysis of the key components that point to potential objectives, risks, and techniques used by this malware:

1. Registry Activity

The third screenshot shows extensive **Registry Key operations** by the process `Explorer.EXE` (PID 3784), interacting with various registry hives such as:

- **HKCR\exefile** (stands for "executable file") keys: These keys are responsible for defining how Windows handles executable files (`.exe`). The frequent querying of these registry keys may indicate that the malware is attempting to manipulate how `.exe` files are executed on the system, potentially to either redirect executions or alter behaviors.
- **HKLM\Software\Microsoft\Windows\CurrentVersion\shell**: These keys are associated with the Windows shell and how it interacts with file types and commands. Querying or modifying these keys might suggest an attempt to hijack execution paths for common operations, giving the

malware the ability to maintain persistence by hooking into standard operations like opening files or launching applications.

Consideration: Persistence and Hijacking Executables

- By interacting with the `HKCR\exefile` registry keys, the malware could be attempting to modify the system's handling of `.exe` files, a classic tactic for gaining persistence by controlling how files are opened (i.e., redirecting `.exe` files to launch the malware instead).
- This is reinforced by querying `shell\open`, where modifications could alter what happens when a user double-clicks files in the Windows Explorer shell, possibly redirecting execution to the malware or its payloads.

2. **Parent Process Analysis**

From the first and third screenshots, we can analyze the **Parent-Child Process Relationships**:

- **Parent Process (PID 3744) is** `Explorer.EXE`: The first screenshot reveals that `calcolatricinema` (PID 5480) is launched by `Explorer.EXE` (PID 3744). This is important because it suggests that the malware might be masquerading or disguising itself as a legitimate application or could be initiated through user interaction with the system (e.g., executing a file through the Windows Explorer interface).

Consideration: Social Engineering or User Interaction

- Since the parent process is `Explorer.EXE`, it indicates that this malware might have been triggered by user interaction—such as double-clicking on a malicious executable or downloading an infected file. This aligns with scenarios where malware is dropped through phishing campaigns or malicious file downloads.

3. **DLL Load Behavior and System Component Manipulation**

Across the first and second screenshots, numerous **DLL loading operations** occur during the malware's execution. The dynamic analysis shows:

- Key **system libraries** (`ntdll.dll`, `kernel32.dll`, `wow64.dll`) are loaded, and these are involved in system-level functions such as memory management, file access, and inter-process communication.
- The frequent interaction with `wow64` **DLLs** (which facilitate the execution of 32-bit processes on 64-bit systems) implies that this malware might be designed for cross-architecture compatibility, running smoothly on both 32-bit and 64-bit versions of Windows.

Consideration: Process Injection and Exploitation

- The loading of `ntdll.dll` and `kernel32.dll` suggests the possibility of **process injection**, where the malware might attempt to inject itself into legitimate processes to evade detection. These DLLs contain low-level API functions used for thread/process creation, memory allocation, and communication with the OS, making them likely candidates for exploitation.

- The **locking** of `imm32.dll` and `mswsock.dll` (in the second screenshot) suggests that the malware is trying to control key system components related to input handling and networking, possibly hooking into these DLLs to intercept user input (for keylogging) or manipulate network traffic.

4. Suspicious File Locking and Interaction with System Files

In the second screenshot, there are multiple instances of **FILE LOCKED WITH ONLY READERS** and attempts to create or access files like `imm32.dll`, `mswsock.dll`, and `winmm.dll`.

Consideration: Anti-Debugging and Tampering with System Files

- By locking system files, the malware could be **preventing their modification** by other processes, particularly security tools or system repairs. This could help the malware avoid detection and maintain control over infected resources.
- The interaction with critical system libraries like `imm32.dll` (related to input) and `mswsock.dll` (related to network communication) indicates that the malware is not only modifying the file system but potentially altering how these key components behave, which could lead to data exfiltration (such as capturing keystrokes or network traffic).

5. BUFFER OVERFLOW Flag

The **BUFFER OVERFLOW** result from the second screenshot (involving `QuerySecurityFile`) suggests

The **BUFFER OVERFLOW** result from the second screenshot (involving `QuerySecurityFile`) suggests
the malware is testing or probing for weaknesses in file security.

Consideration: Vulnerability Exploitation

- The BUFFER OVERFLOW event could be the malware attempting to trigger a vulnerability. Malicious software often uses buffer overflow techniques to escalate privileges, execute arbitrary code, or crash security mechanisms. This might point to a potential exploit being deployed against the system or file-level security.

6. Process and Thread Management

The first screenshot shows a significant number of **thread creation** events and **process start** operations, indicating that the malware is multi-threaded and capable of handling multiple tasks concurrently.

Consideration: Multi-Tasking Capabilities

- By creating multiple threads, the malware can efficiently manage various tasks such as file system operations, communication with command-and-control (C2) servers, process injection, or monitoring user activities without being easily detected. This implies a high level of sophistication and modularity.

7. Network Activity Potential

Given that the malware loads `mswsock.dll`, a library responsible for handling network sockets, there is a strong possibility that the malware includes **networking capabilities** such as:

- Communicating with a remote C2 server (for receiving instructions, downloading additional payloads, or exfiltrating data).
- Creating backdoors or listening sockets on the infected system, allowing attackers to establish remote control or lateral movement within the network.

8. Explorer.EXE Interaction

The repeated interaction with `Explorer.EXE` in the third screenshot shows how deeply the malware integrates with the Windows shell, likely to manipulate or monitor standard user activities.

Consideration: Shell Interaction for Persistence

- The use of the Explorer process to launch and interact with system resources might be part of the malware's **persistence mechanism**. By embedding itself in such a core Windows process, it can mask its presence and avoid easy detection by security tools, which often trust `Explorer.EXE`.

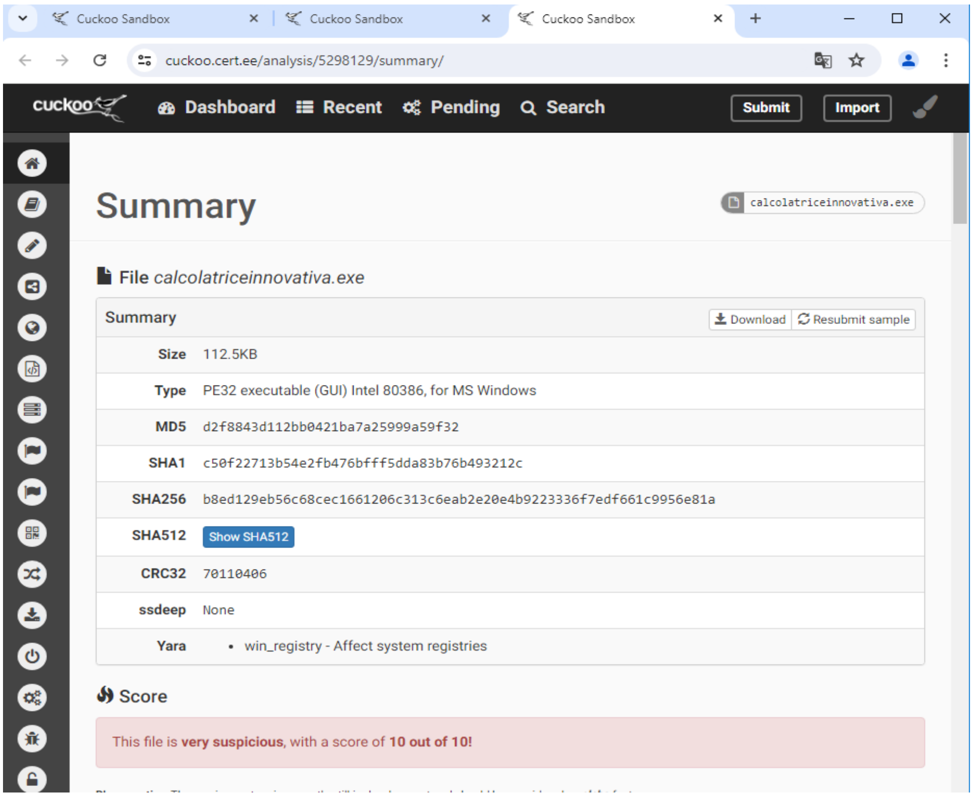
Final Considerations

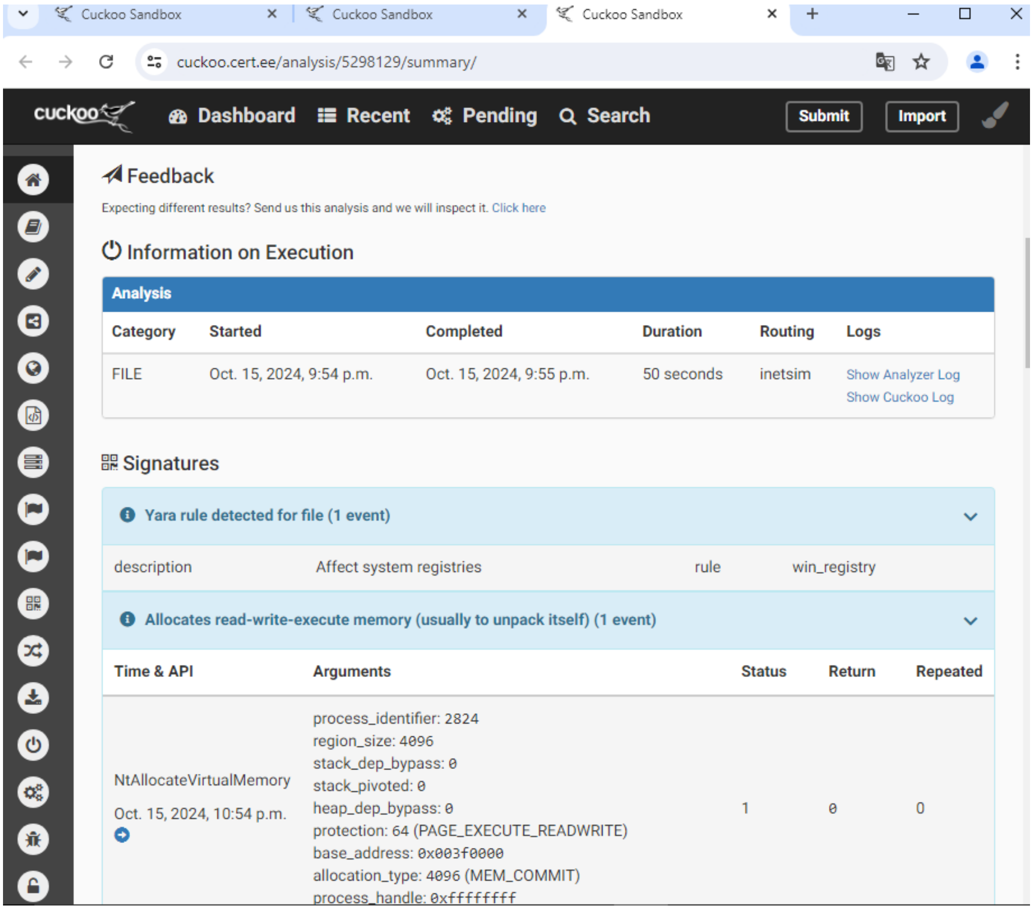
- **Persistence Mechanisms**: The malware likely utilizes multiple techniques to ensure persistence, including registry key manipulation (`exefile`, `shell\open`), file system locks, and potentially even process injection into `Explorer.EXE`.
- **Evasion Tactics**: The BUFFER OVERFLOW event, combined with file locking and extensive use of DLLs related to system-level operations, suggests the malware is actively working to evade detection and maintain control over system resources.
- **Exploitation and Privilege Escalation**: The BUFFER OVERFLOW and registry key access indicate possible attempts to exploit system weaknesses and escalate privileges.
- **Multi-Tasking Nature**: The presence of multiple threads highlights the malware's ability to manage various tasks simultaneously, such as C2 communication, data theft, persistence maintenance, and even attempts at spreading laterally within a network.

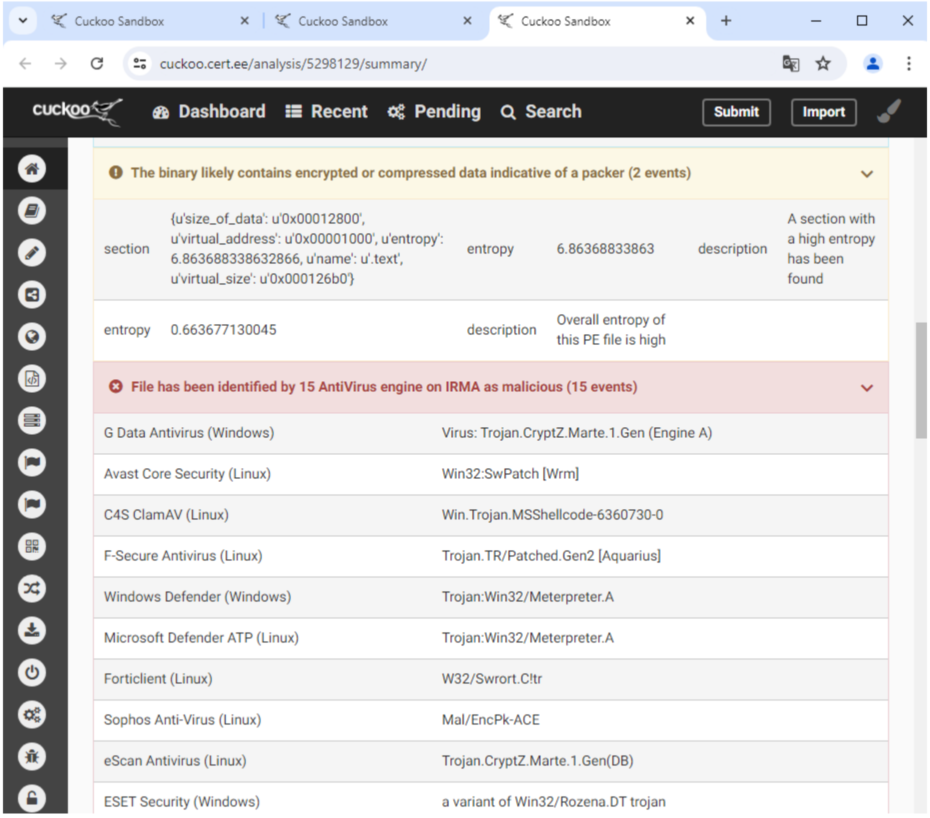
Further analysis should include **monitoring network traffic**, checking **persistence mechanisms (registry)**, and performing **memory analysis** to uncover hidden processes or injected code.

Esercizio Extra.

Ho esaminato il malware con la sandbox Cuckoo e questo è il report finale dell'analisi svolta.


Cuckoo Sandbox
cuckoo.cert.ee/analysis/5298129/summary/
Dashboard Recent Pending Search
Submit Import
Summary
calcolatriceinnovativa.exe
File calcolatriceinnovativa.exe
Summary
Download Resubmit sample
Size 112.5KB
Type PE32 executable (GUI) Intel 80386, for MS Windows
MD5 d2f8843d112bb0421ba7a25999a59f32
SHA1 c50f22713b54e2fb476bfff5dda83b76b493212c
SHA256 b8ed129eb56c68cec1661206c313c6eab2e20e4b9223336f7edf661c9956e81a
SHA512 Show SHA512
CRC32 70110406
ssdeep None
Yara win_registry - Affect system registries
Score
This file is very suspicious, with a score of 10 out of 10!


Cuckoo Sandbox
cuckoo.cert.ee/analysis/5298129/summary/
Dashboard Recent Pending Search
Submit Import
Feedback
Expecting different results? Send us this analysis and we will inspect it. Click here
Information on Execution
Analysis
Category Started Completed Duration Routing Logs
FILE Oct. 15, 2024, 9:54 p.m. Oct. 15, 2024, 9:55 p.m. 50 seconds inetsim Show Analyzer Log Show Cuckoo Log
Signatures
Yara rule detected for file (1 event)
description Affect system registries rule win_registry
Allocates read-write-execute memory (usually to unpack itself) (1 event)
Time & API Arguments Status Return Repeated
NtAllocateVirtualMemory
Oct. 15, 2024, 10:54 p.m.
process_identifier: 2824
region_size: 4096
stack_dep_bypass: 0
stack_pivoted: 0
heap_dep_bypass: 0
protection: 64 (PAGE_EXECUTE_READWRITE)
base_address: 0x003f0000
allocation_type: 4096 (MEM_COMMIT)
process_handle: 0xffffffff
1 0 0

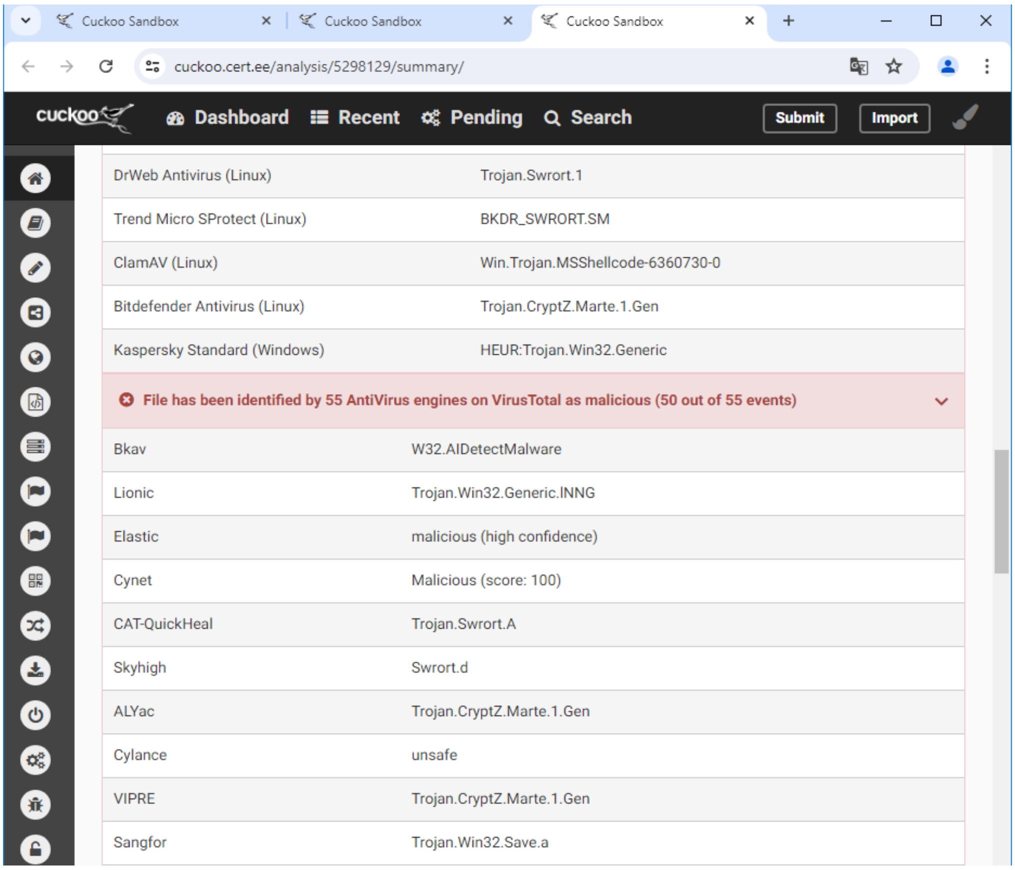


The binary likely contains encrypted or compressed data indicative of a packer (2 events)

section	{u'size_of_data': u'0x00012800', u'virtual_address': u'0x00001000', u'entropy': 6.863688338632866, u'name': u'.text', u'virtual_size': u'0x000126b0'}	entropy	6.86368833863	description	A section with a high entropy has been found
entropy	0.663677130045	description	Overall entropy of this PE file is high		

File has been identified by 15 AntiVirus engine on IRMA as malicious (15 events)

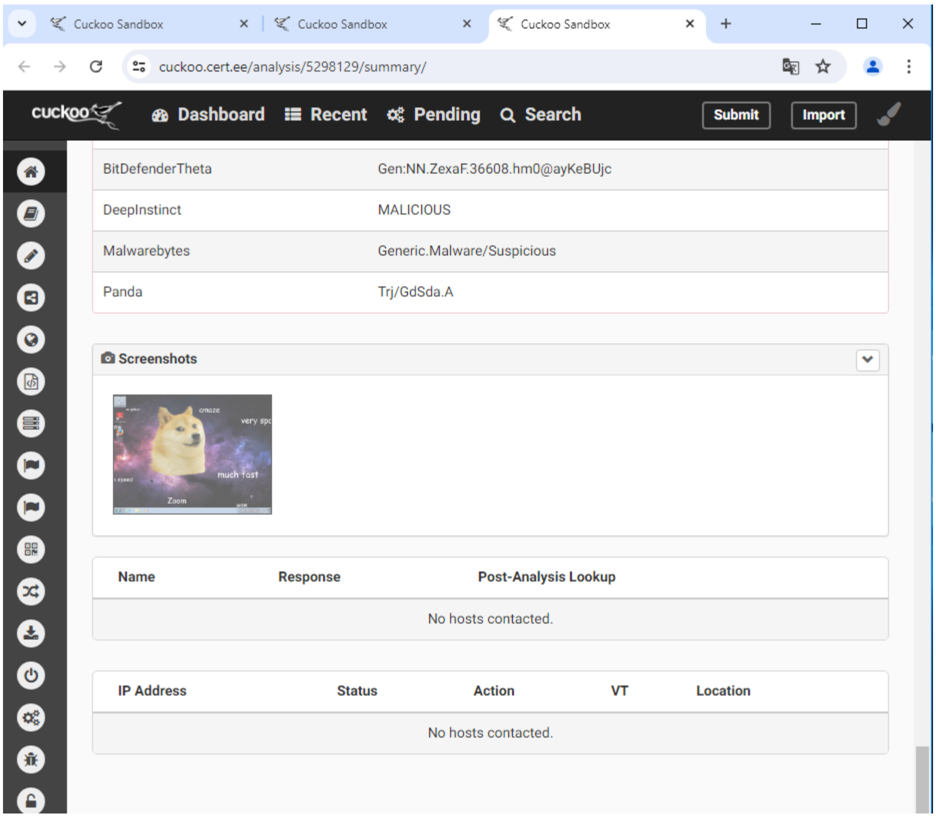
G Data Antivirus (Windows)	Virus: Trojan.CryptZ.Marte.1.Gen (Engine A)
Avast Core Security (Linux)	Win32:SwPatch [Wrm]
C4S ClamAV (Linux)	Win.Trojan.MSShellcode-6360730-0
F-Secure Antivirus (Linux)	Trojan.TR/Patched.Gen2 [Aquarius]
Windows Defender (Windows)	Trojan:Win32/Meterpreter.A
Microsoft Defender ATP (Linux)	Trojan:Win32/Meterpreter.A
Forticlient (Linux)	W32/Swrort.C!tr
Sophos Anti-Virus (Linux)	Mal/EncPk-ACE
eScan Antivirus (Linux)	Trojan.CryptZ.Marte.1.Gen(DB)
ESET Security (Windows)	a variant of Win32/Rozena.DT trojan



DrWeb Antivirus (Linux)	Trojan.Swrort.1
Trend Micro SProtect (Linux)	BKDR_SWRORT.SM
ClamAV (Linux)	Win.Trojan.MSShellcode-6360730-0
Bitdefender Antivirus (Linux)	Trojan.CryptZ.Marte.1.Gen
Kaspersky Standard (Windows)	HEUR:Trojan.Win32.Generic

File has been identified by 55 AntiVirus engines on VirusTotal as malicious (50 out of 55 events)

Bkav	W32.AIDetectMalware
Lionic	Trojan.Win32.Generic.INNG
Elastic	malicious (high confidence)
Cynet	Malicious (score: 100)
CAT-QuickHeal	Trojan.Swrort.A
Skyhigh	Swrort.d
ALYac	Trojan.CryptZ.Marte.1.Gen
Cylance	unsafe
VIPRE	Trojan.CryptZ.Marte.1.Gen
Sangfor	Trojan.Win32.Save.a



Il file è ovviamente un eseguibile Windows, ha effettuato una modifica dei registri di Windows, potrebbe essere persistente ed usa la crittografia per non essere riconosciuto come un malware. NtAllocateVirtualMemory conferma il fatto che il malware riesce a manipolare la memoria e gestire una code injection. L'alto livello di entropia suggerisce che il malware utilizza file zippati e un alto livello di offuscamento grazie alla crittografia e gli antivirus hanno riconosciuto questo sample come un file malevolo, confermando le ipotesi indicate prima che lo interpretavano come un possibile trojan o un file in grado di installare una backdoor e sottrarre dati. Tra le immagini scaricate vi è un meme e questo potrebbe rientrare nei tentativi di social engineering di distratte l'utente, facendo apparire un contenuto divertente ma che in realtà nasconde una minaccia. Al momento non sono state rilevate connessioni malevole verso l'esterno, ma questo potrebbe essere legato al rapidissimo tempo dell'analisi e al fatto che magari il malware aspetta che si verifichino delle condizioni particolari per potersi collegare.

Ho fatto esaminare questo report da ChatGPT e questa è la risposta che è stata generata.


I have analyzed this malware using the sandbox Cuckoo. Assess this malware on the basis of the screenshots of the dynamic analysis using Cuckoo.

Based on the dynamic analysis results from the **Cuckoo Sandbox** as seen in the screenshots, the **malware** `calcolatriceinnovativa.exe` displays several suspicious behaviors and characteristics that confirm its malicious nature. Here is a detailed assessment of the malware:

1. File Properties and Identification (First Screenshot)

- **File Type**: The file is identified as a **PE32 executable** (Intel 80386 for Windows), indicating that it is a 32-bit executable designed for MS Windows.
- **File Hashes (MD5, SHA1, SHA256)**: These hashes provide unique fingerprints for this specific malware sample, useful for detecting the malware across different systems and environments. The **SHA256 hash** (`b8ed129eb5...`) can be used to cross-reference this malware against known databases like VirusTotal.
- **YARA Rule**: The malware has triggered a **YARA rule** related to **system registry modification**. This suggests that the malware likely affects critical registry settings, potentially for persistence or to alter system behavior.

Assessment: This confirms that the malware affects Windows registry settings, which may be a persistence mechanism. Additionally, the presence of unique cryptographic hashes enables easy identification across different threat intelligence platforms.

2. Suspicious Behavior and Execution (Second Screenshot)

- **Execution Timing**: The malware was executed for about 50 seconds in the sandbox environment, with the analysis completed without interruption.
- **YARA Rule Detection**: The YARA rule triggered highlights that the malware likely attempts to **modify system registry entries**. This is common behavior for malware that wants to establish persistence or alter system behavior, such as manipulating file associations or disabling security software.
- **Memory Manipulation (NtAllocateVirtualMemory)**: The malware allocates **read-write-execute (RWX) memory**, which is a strong indicator of **code injection** or unpacking routines. This API call (`NtAllocateVirtualMemory`) is often used by malware to **load additional malicious code into memory**, enabling it to run without writing files to disk, making detection harder.
  - The allocation is relatively small (4 KB), suggesting that the malware might be loading or injecting a small piece of code, potentially a **shellcode** or **decryption routine** for further malicious activities.

Assessment: The **RWX memory allocation** combined with **YARA rule hits** strongly indicates that this malware is likely unpacking itself in memory or attempting **code injection**. These behaviors are typical of advanced malware trying to evade detection by avoiding leaving artifacts on disk.

3. **Packed or Compressed Data Indicated by High Entropy (Third Screenshot)**

- **High Entropy Section**: The section `u_text` has **high entropy (6.86)**, which is indicative of **packed or encrypted data**. High entropy means that the data within the executable is highly randomized, often a sign of obfuscation, compression, or encryption, used by malware authors to hide the true functionality of their code.
- This confirms that the malware is **packed or compressed**, which is a common technique used to evade signature-based detection by traditional antivirus solutions.

Assessment: The malware is **likely obfuscated**, meaning the true functionality of the malware is hidden until it is unpacked during execution. This indicates that further steps such as unpacking or memory analysis would be necessary to fully understand the malware's behavior.

4. **Antivirus Detection and Classification (Third and Fourth Screenshots)**

- **Antivirus (AV) Detection**:
  - **15 AV Engines on IRMA** and **55 AV Engines on VirusTotal** flagged this sample as malicious.
  - It has been classified as **Trojan.CryptZ, Meterpreter, and Swrort** by various AV engines:
    - **Meterpreter**: This is a well-known payload used in **Metasploit**. If this is indeed a **Meterpreter payload**, it indicates that the malware provides **remote access and control** capabilities, allowing an attacker to interact with the infected machine remotely.
    - **Trojan.CryptZ**: This suggests the use of **encryption**, either to protect its payload or to perform **ransomware-like encryption** activities on the victim's machine.
    - **Swrort**: This classification suggests that the malware could act as a **backdoor**, giving attackers the ability to gain access and control over the compromised system.
- Multiple antivirus engines detect the sample as being associated with **Trojan and Backdoor** capabilities, which suggests the malware provides both **remote control functionality** and **potential data exfiltration or further exploitation** capabilities.

Assessment: The widespread detection across multiple antivirus engines, along with its classification as **Meterpreter**, **Trojan.CryptZ**, and **Backdoor**, confirms that this is a highly dangerous piece of malware capable of remote control and likely exfiltration or further exploitation of the infected system.

5. **No Network Activity Detected (Fifth Screenshot)**

- **No Hosts Contacted**: The sandbox analysis did not detect any outbound network communication during the 50-second execution time. However, this might be due to:
  - The malware **not fully executing** in the limited analysis time.

- The malware **waiting for specific triggers or conditions** (e.g., running as administrator, certain dates/times) before initiating communication.
- **C2 (Command-and-Control) obfuscation** where the malware uses covert methods to communicate with its remote server (e.g., DNS tunneling, encrypted HTTPS).

Assessment: Despite no immediate network activity, this does not rule out the possibility of remote communication. The presence of **Meterpreter** and **backdoor-related classifications** strongly implies that the malware has remote control capabilities that might not have been triggered during the sandbox session.

6. Miscellaneous Behavior: Displaying a Meme (Fifth Screenshot)

- **Screenshot showing a meme (Dogecoin image)**: The appearance of a Dogecoin-themed meme may suggest the malware is either using this as a **distraction technique** or simply has a humorous element embedded by the malware author.
- This behavior, while potentially humorous, can serve as a **smokescreen** for the actual malicious activities occurring in the background, such as process injection, registry modifications, or data exfiltration.

Assessment: Displaying a **meme** suggests the malware may have an element of **social engineering** or a tactic to deceive users while malicious activities proceed unnoticed in the background. It may also indicate **non-serious intentions**, though this is less likely given its confirmed malicious

Summary of Key Findings:

1. **Malicious File**: The file is confirmed as highly suspicious, with a **10/10 rating** on Cuckoo and detection by **55 antivirus engines** on VirusTotal.
2. **Memory Manipulation (Code Injection)**: Allocating **read-write-execute memory** confirms that the malware is likely unpacking itself in memory or injecting malicious code into processes, which is a common behavior in sophisticated malware.
3. **Registry Modification**: The malware affects the Windows Registry, likely attempting to **establish persistence** or change system behavior to facilitate its operation.
4. **Packed/Encrypted Malware**: High entropy in its sections indicates the malware is **packed or encrypted**, making static analysis difficult. It likely unpacks during execution.
5. **Remote Access Capabilities (Meterpreter)**: The detection of **Meterpreter** by antivirus engines indicates the malware is likely used to provide **remote control**, possibly for further exploitation, data theft, or system takeover.
6. **Humorous Social Engineering (Meme)**: The Dogecoin meme displayed by the malware may serve as a **distraction tactic** while malicious activities proceed in the background.

Conclusion and Recommendations:

This malware is highly dangerous, with confirmed **remote access (backdoor)** capabilities, **memory injection**, and **persistence mechanisms**. Although no immediate network activity was observed, the presence of **Meterpreter** suggests potential for **remote control and further exploitation**.

Remediation Steps:

- **Isolate the infected system** immediately to prevent further spread.
- **Analyze registry changes** for potential persistence mechanisms and revert them.
- Use **forensic tools** to inspect **memory** and determine if the malware injected any code into other processes.
- **Reinstall or restore affected systems** from a clean backup to ensure complete removal.
- **Review network logs** to check for potential stealthy C2 communications.