Hash: e142a1e51ce0e8d28fd852683b65688dcc97a6b705e8adc799d5af0bdefefcf

Malware family name: Ramnit
Classification: Worm/Trojan

Tools

Static Analysis Tools

Strings – Extracted strings to identify potential API calls, registry paths, persistence mechanisms, and referenced filenames
CFF Explorer – Analysis the PE structure, including headers, imports, and exported functions
VirusTotal, HybridAnalysis, CAPE Sandbox – Performed hash-based analysis across multiple antivirus engines

Dynamic Analysis Tools

X32dbg – Debugged the sample at runtime to observe execution flow and confirm process creation attempts
RegShot – Compared registry states before and after execution to identify persistence-related modifications
AutoRuns – Inspected startup locations post-reboot to identify persistence mechanisms
Process Explorer – Monitored running processes post-reboot to verify whether the malware executed in memory

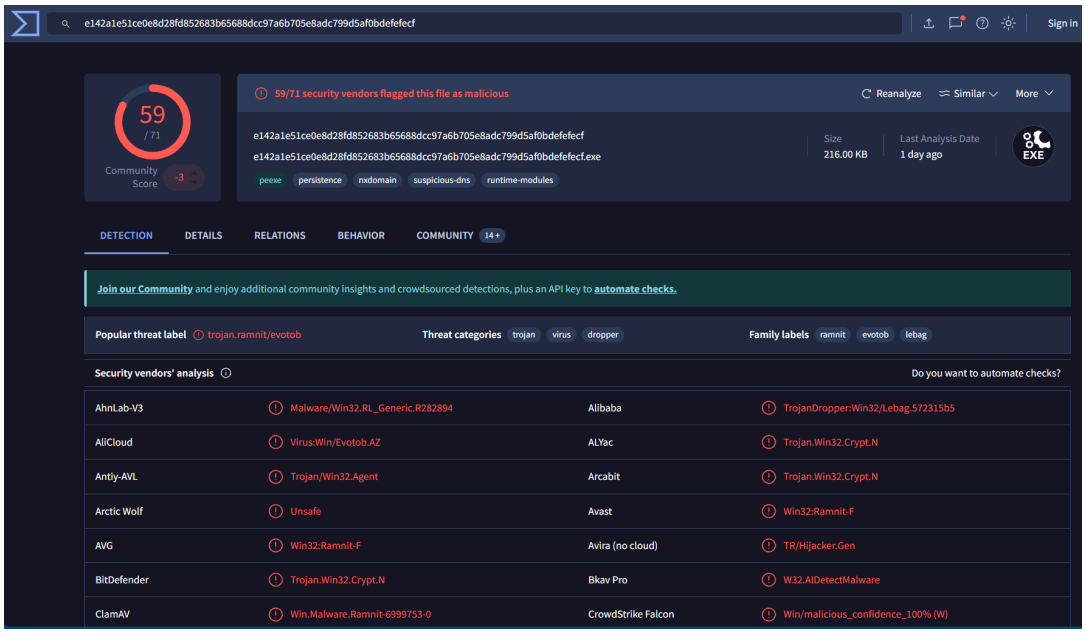
Executive Summary

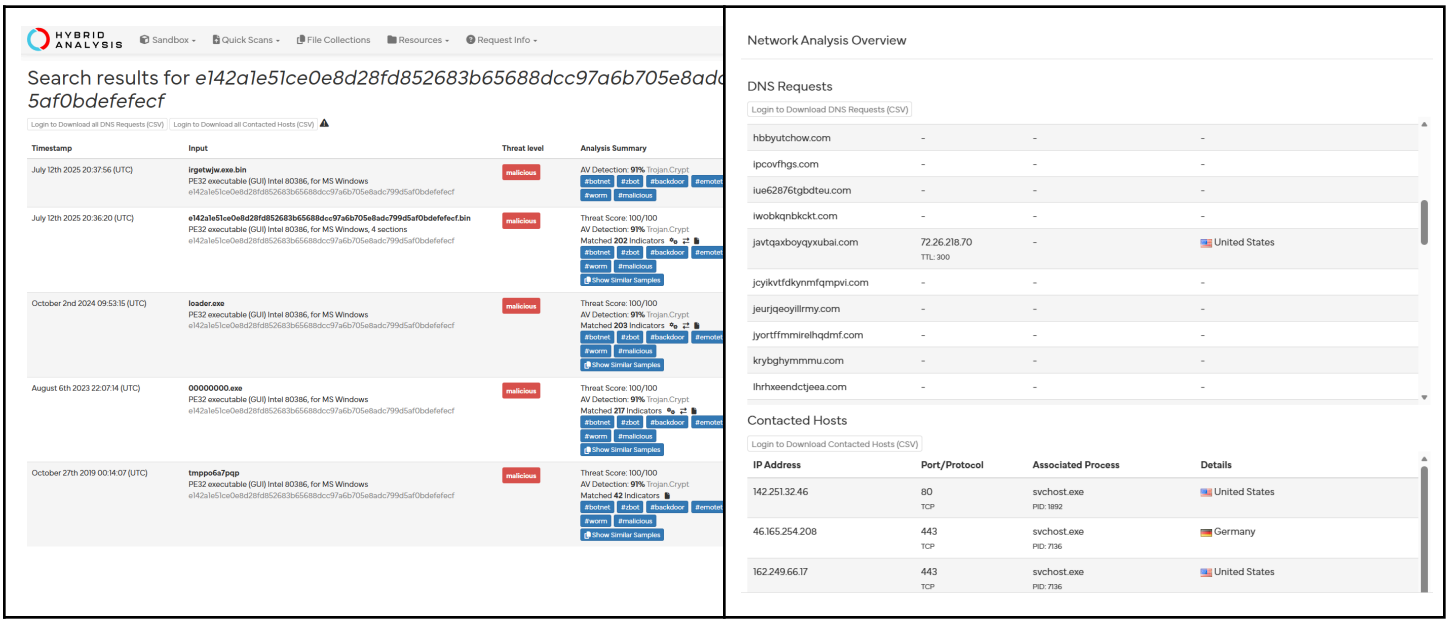
The executable is a Windows-based malware sample that demonstrates multiple malicious behaviors, including unauthorized process creation, registry modification, privilege escalation attempts, and persistence establishment. Static analysis revealed extensive use of Windows API functions related to memory manipulation, registry access, and process execution. Dynamic analysis confirmed that the malware attempts to spawn additional processes and modify system configuration settings, triggering User Account Control prompts during execution. Registry analysis and startup inspection showed that the malware successfully configured multiple persistence mechanisms; however, post-reboot analysis using AutoRuns and Process Explorer indicated that the malware did not successfully execute after system restart. This was likely due to environmental constraints such as disabled network connectivity preventing retrieval or execution of a secondary payload.

Analysis Findings

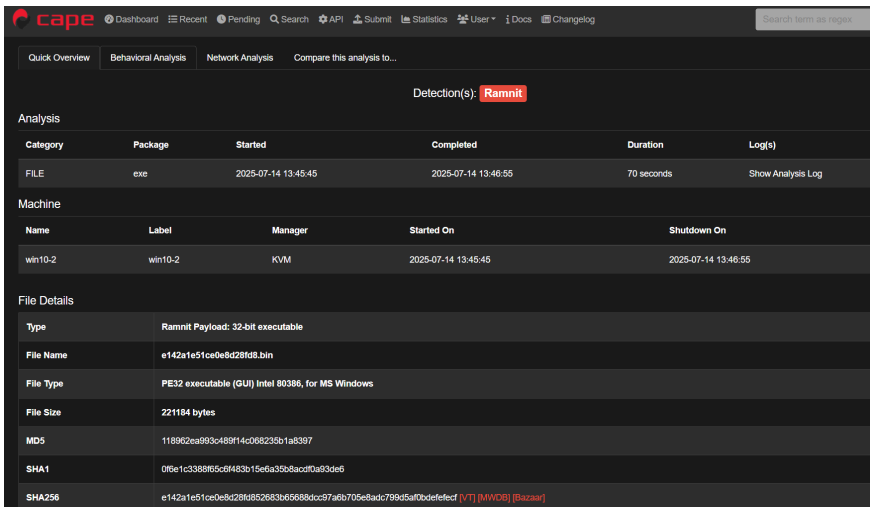
After submitting the hash sample to VirusTotal, Hybrid Analysis, and CAPE Sandbox, multiple security engines flagged the hash as malicious. Sandbox reports associated the hash with behaviors including registry modifications, process execution, and persistence-related activity.

VirusTotal and HybridAnalysis indicate the file as malicious, with HybridAnalysis linking it to a file named “loader.exe”. HybridAnalysis detected many DNS queries for domains that cannot be resolved. Since the file is named “loader”, it’s possible that this malware’s intention is simply to weaken defenses and establish persistence within the system, before then reaching the network to download a second payload without the user’s knowledge.





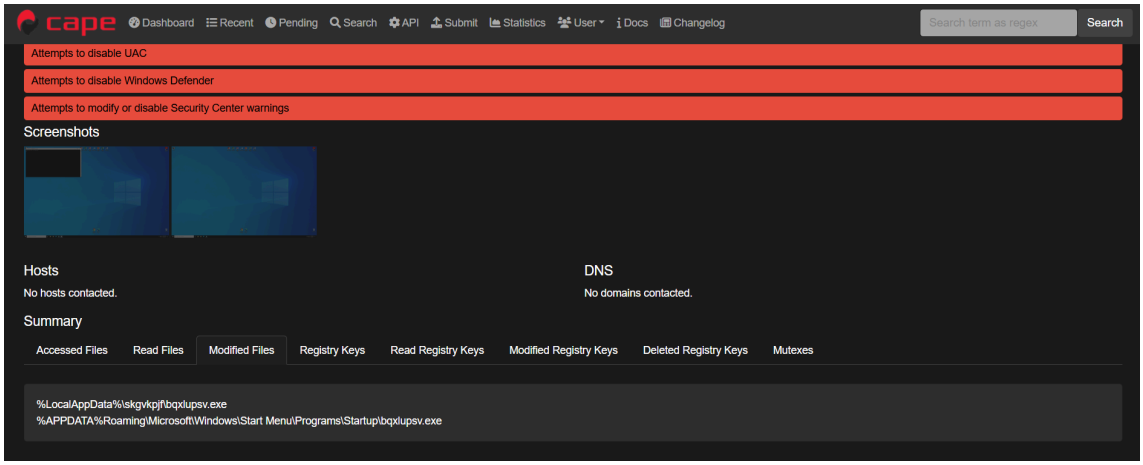
CAPE Sandbox also flagged the file as malicious, under the Ramnit family. PE information on CAPE indicates loader.exe as the exported DLL name.



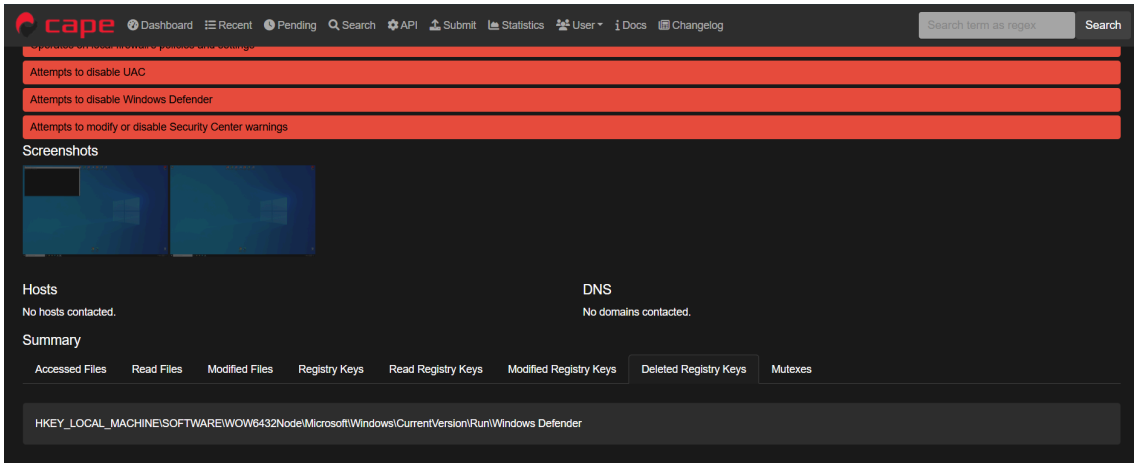
PE Information

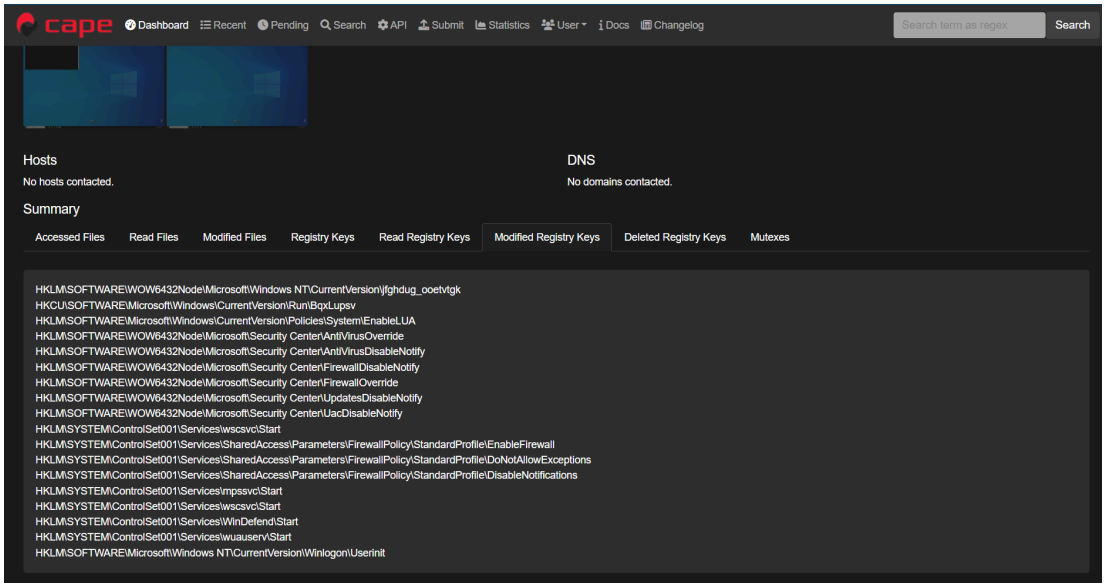
Image Base	Entry Point	Reported Checksum	Actual Checksum	Minimum OS Version	Compile Time	Import Hash	Icon	Icon Exact Hash	Icon Similarity Hash	Icon DHash	Exported DLL Name
0x15190000	0x00029 7b5	0x00000000	0x0003c2e4	4.0	2015-11-11 17:56:40	eb21f550e104c8827a04ad90c1f193b6		8d59bc82d181b41db7131ab4a5664584	3871c0b3f90370465c84376cbc084f50	74f4c898b6b698d0	loader.exe

A file within the Startup folder was modified, which shows that the malware is attempting to establish persistence within the system.

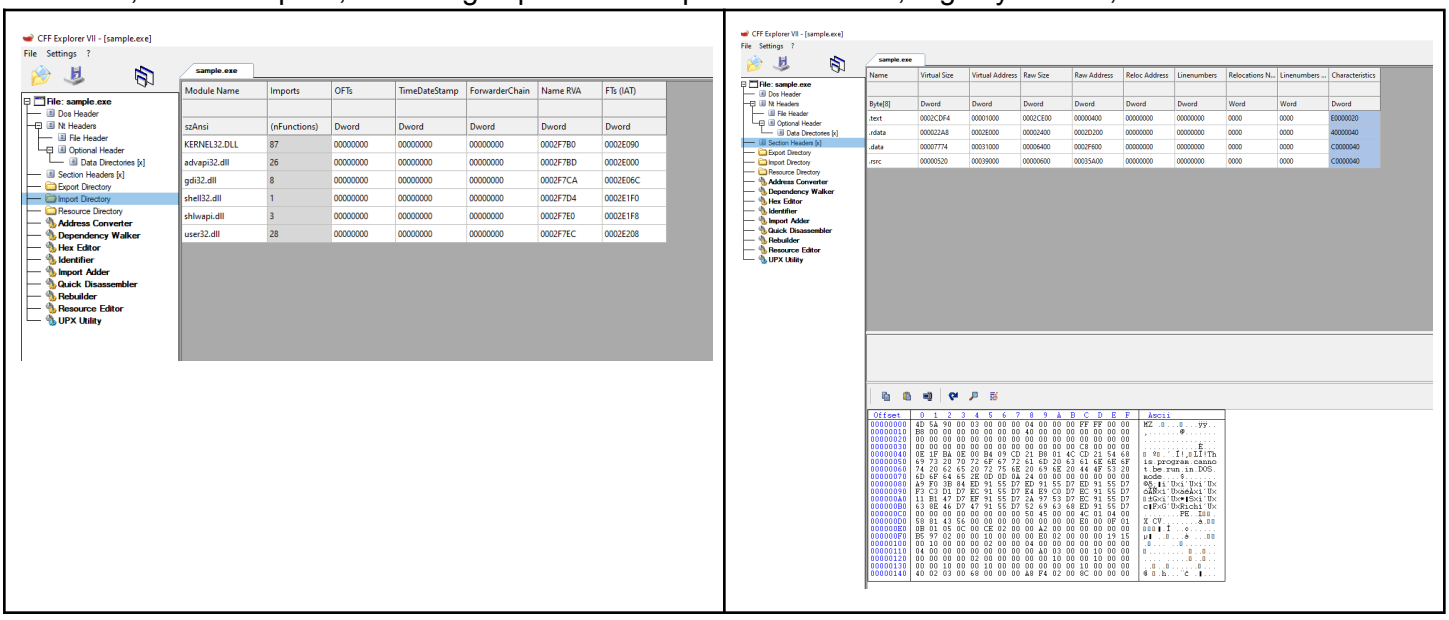


To further establish persistence, the registry key for Windows Defender was deleted, and registry keys for Windows Defender, firewall policy, and anti virus were modified.





The file is a valid 32-bit WIndows Portable Executable (PE32) with standard sections (.text, .rdata, .data, .rsrc) and no evidence of packing. Import table analysis showed use of kernel32.dll, advapi32.dll, shell32.dll, user32.dll, and shlwapi.dll, indicating capabilities for process creation, registry access, and shell interaction.



Export table analysis revealed executable export functions named “_ApplyExploit@4” and “_CheckBypassed@0”. The presence of exploit-related function names within an executable suggests that the sample may be intended to function as an exploit helper or secondary component invoked by another process.

CFF Explorer VII - [sample.exe]

File Settings ?

sample.exe

- File: sample.exe
  - Dos Header
  - Nt Headers
    - File Header
    - Optional Header
      - Data Directories [x]
  - Section Headers [x]
  - Export Directory
  - Import Directory
  - Resource Directory
  - Address Converter
  - Dependency Walker
  - Hex Editor
  - Identifier
  - Import Adder
  - Quick Disassembler
  - Rebuilder
  - Resource Editor
  - UPX Utility

Member	Offset	Size	Value
Characteristics	0002F440	Dword	00000000
TimeDateStamp	0002F444	Dword	56438158
MajorVersion	0002F448	Word	0000
MinorVersion	0002F44A	Word	0000
Name	0002F44C	Dword	0003027C
Base	0002F450	Dword	00000001
NumberOfFunctions	0002F454	Dword	00000002
NumberOfNames	0002F458	Dword	00000002
AddressOfFunctions	0002F45C	Dword	00030268

Ordinal	Function RVA	Name Ordinal	Name RVA	Name
(nFunctions)	Dword	Word	Dword	szAnsi
00000001	0002DC9E	0000	00030287	_ApplyExploit@4
00000002	0002DBCE	0001	00030297	_CheckBypassed@0

Extracted strings revealed references to process execution functions (CreateProcessA, ShellExecuteExA, runas), memory allocation and protection APIs (VirtualAlloc, VirtualProtect, ZwAllocateVirtualMemory), registry persistence locations (Software\Microsoft\Windows\CurrentVersion\Run, Winlogon), Windows Defender exclusion paths, and HTTP-related strings suggesting potential outbound communication. Strings analysis also revealed references to loader.exe, suggesting that the analyzed sample may function as a loader or helper component rather than a complete standalone payload.

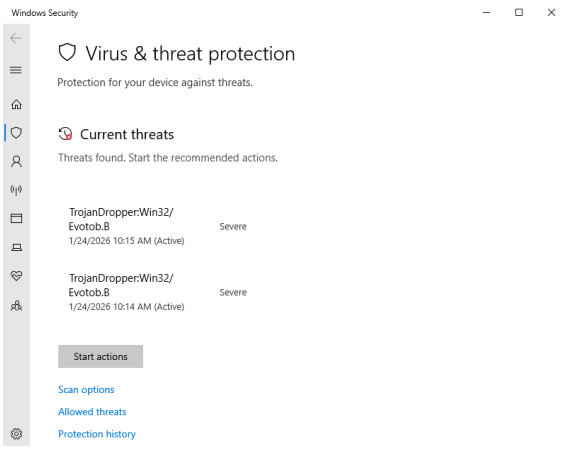
```
Line 1045: %APPDATA%
Line 2060: %APPDATA%
Line 2149: %APPDATA%\Apple Computer\Safari\Cookies\Cookies.plist
Line 2150: %APPDATA%\Mozilla\Firefox\
Line 2158: 1APPDATA%\Macromedia\Flash Player\#SharedObjects
Line 2159: %APPDATA%\Opera\

Line  945: GetTempFileNameA
Line  946: GetTempPathA
Line 1858: GetTempFileNameA
Line 1859: GetTempPathA
Line 2666: %temp%\..\..\LocalLow\cmd.%username%.bat
Line 2738: PsInitialSystemProcess
Line 2801: GetTempPathA
Line 2802: GetTempFileNameA

Line 2684: system32
Line 2687: \system32\sdbinst.exe"

Line  926: CreateProcessA
Line 1832: CreateProcessA
Line 2788: CreateProcessA

Line 1965: ShellExecuteA
Line 2873: ShellExecuteExA

Line  958: LoadLibraryA
Line 1871: LoadLibraryA
Line 2791: LoadLibraryA
Line 2830: LoadLibraryExA

Line  943: GetProcAddress
Line 1855: GetProcAddress
Line 2803: GetProcAddress

Line  980: VirtualAlloc
Line  981: VirtualFree
Line  982: VirtualProtect
Line 2698: ZwAllocateVirtualMemory
```

```
Line 2739: NtFreeVirtualMemory
Line 2741: NtAllocateVirtualMemory
Line 2770: VirtualAlloc
Line 2771: VirtualAllocEx
Line 2772: VirtualFree
Line 2773: VirtualProtect
Line 2774: VirtualProtectEx
Line 2815: VirtualQueryEx
Line 2816: VirtualFreeEx
Line 2932: ZwProtectVirtualMemory
Line 2934: ZwFreeVirtualMemory
Line 2965: ZwWriteVirtualMemory

Line  976: Sleep
Line 1891: Sleep
Line 2767: Sleep
```

Strings analysis also revealed multiple registry modification commands targeting Microsoft Defender and Antimalware exclusion lists. The malware attempts to exclude trusted Windows binaries such as svchost.exe, rundll32.exe, and explorer.exe, as well as entire file extensions (*.exe, *.dll), indicating a clear attempt to evade host-based security controls prior to executing additional payloads.

```
svchost.exe
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v svchost.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v consent.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v rundll32.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v spoolsv.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v explorer.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v rgjdu.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes  " /v afwqs.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions  " /v *.tmp /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions  " /v *.dll /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions  " /v *.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v svchost.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v consent.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v rundll32.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v spoolsv.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v explorer.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v rgjdu.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes  " /v afwqs.exe /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Extensions  " /v *.tmp /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Extensions  " /v *.dll /t  REG_DWORD /d 0
REG ADD "HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Extensions  " /v *.exe /t  REG_DWORD /d 0
spoolsv.exe
msseces.exe
```

Windows Defender showed severe threats before it was disabled for dynamic analysis.



Dynamic analysis confirmed that the malware executed successfully, as evidenced by process and thread creation and multiple User Account Control (UAC) elevation prompts. This analysis was conducted in an isolated Windows virtual machine using x32dbg, Process Explorer, RegShot, and AutoRuns.

When executed under x32dbg, the malware reached its entry point and executed a bypass verification, as seen by “CheckBypassed ok”. The presence of “_CheckBypassed@0” was noted earlier during static analysis in the CFF Explorer export directory. The application exited cleanly after verification.

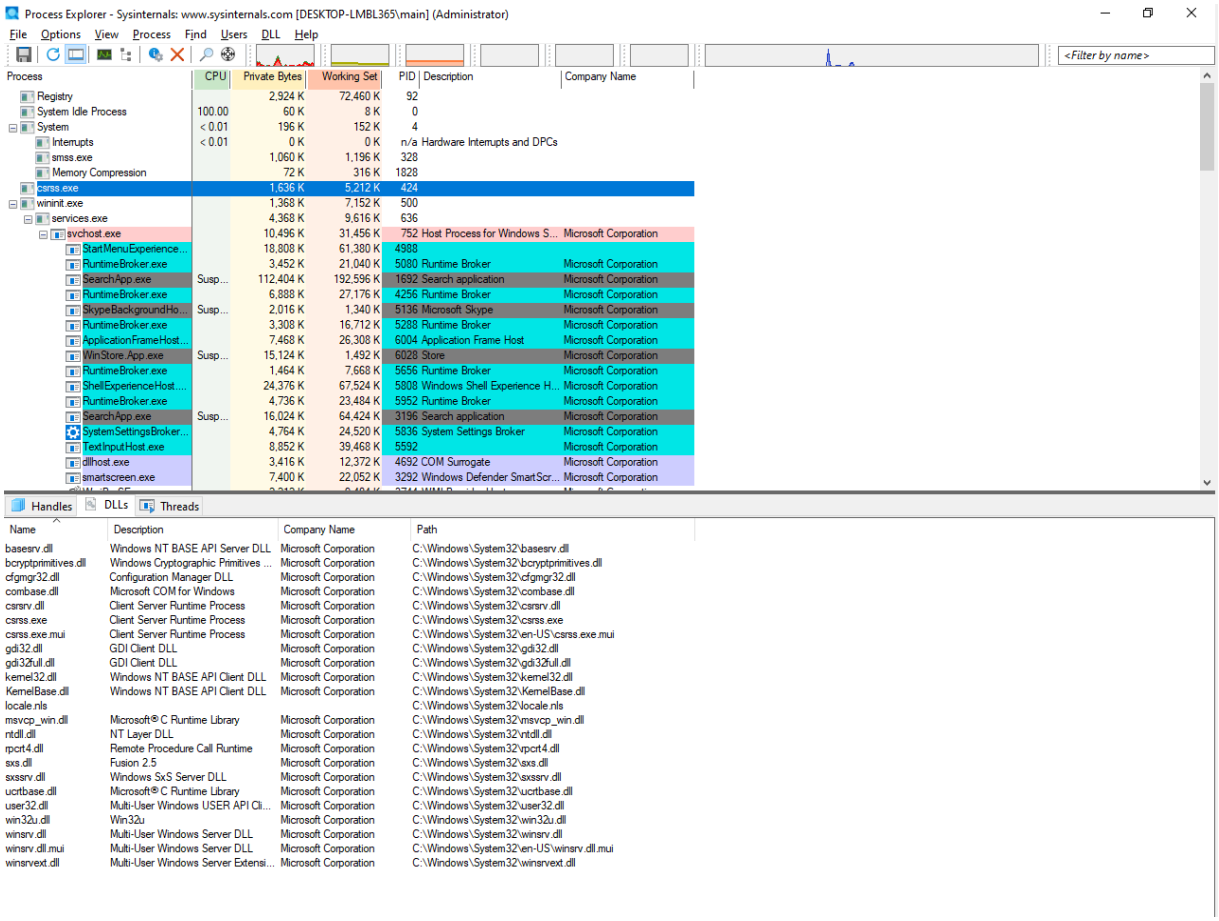
```
Process Started: 15190000 C:\Users\main\Desktop\sample.exe
  "C:\Users\main\Desktop\sample.exe"
  argv[0]: C:\Users\main\Desktop\sample.exe
Breakpoint at 151B97B5 (entry breakpoint) set!
DLL Loaded: 773E0000 C:\Windows\SysWOW64\ntdll.dll
DLL Loaded: 760C0000 C:\Windows\SysWOW64\kernel32.dll
DLL Loaded: 768B0000 C:\Windows\SysWOW64\KernelBase.dll
DLL Loaded: 743F0000 C:\Windows\SysWOW64\apphelp.dll
DLL Loaded: 76230000 C:\Windows\SysWOW64\advapi32.dll
DLL Loaded: 77310000 C:\Windows\SysWOW64\msvcrt.dll
DLL Loaded: 75DB0000 C:\Windows\SysWOW64\sechost.dll
DLL Loaded: 75E30000 C:\Windows\SysWOW64\rpcrt4.dll
DLL Loaded: 75D60000 C:\Windows\SysWOW64\gdi32.dll
DLL Loaded: 75D40000 C:\Windows\SysWOW64\win32u.dll
DLL Loaded: 75840000 C:\Windows\SysWOW64\gdi32full.dll
Thread 5412 created, Entry: ntdll.77415940, Parameter: 0045C840
Thread 2552 created, Entry: ntdll.77415940, Parameter: 0045C840
DLL Loaded: 76B00000 C:\Windows\SysWOW64\msvcp_win.dll
DLL Loaded: 75A20000 C:\Windows\SysWOW64\ucrtbase.dll
DLL Loaded: 76E00000 C:\Windows\SysWOW64\user32.dll
DLL Loaded: 762B0000 C:\Windows\SysWOW64\shell32.dll
DLL Loaded: 75930000 C:\Windows\SysWOW64\shlwapi.dll
System breakpoint reached!
DLL Loaded: 75BB0000 C:\Windows\SysWOW64\imm32.dll
INT3 breakpoint "entry breakpoint" at <sample.OptionalHeader.AddressOfEntryPoint> (151B97B5)!
DebugString: "CheckBypassed ok"
DLL Loaded: 75250000 C:\Windows\SysWOW64\ntmarta.dll
Thread 5412 exit
Thread 2552 exit
Process stopped with exit code 0x0 (0)
Saving database to C:\Tools\Debugging\release\x32\db\sample.exe.dd32 0ms
Debugging stopped!
```

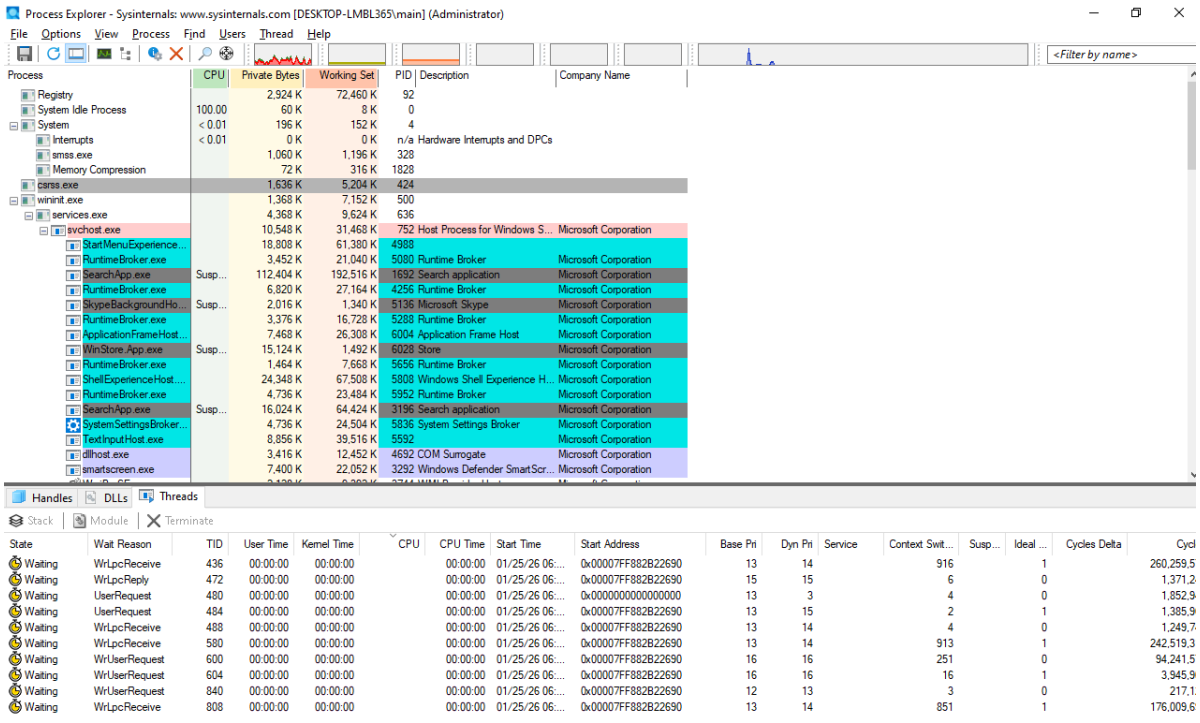
After execution, UAC prompts were triggered and a system restart was requested, indicating attempts to modify system-level or security-related settings.

Using RegShot, comparison of registry and filesystem state before and after execution revealed multiple file system changes associated with malware activity. A new directory was created at “C:\Users\main\AppData\Local\smhxfexa”, containing an executable file named “bsykpehn.exe”. Additional executables were written to user-writable locations, including “C:\Users\main\AppData\Local\Temp\ruxiloco.exe” and “C:\Users\main\AppData\Local\Temp\koqpwqnt.exe”. RegShot output also shows creation of startup persistence via placement of “bsykpehn.exe” in the user’s Startup folder.

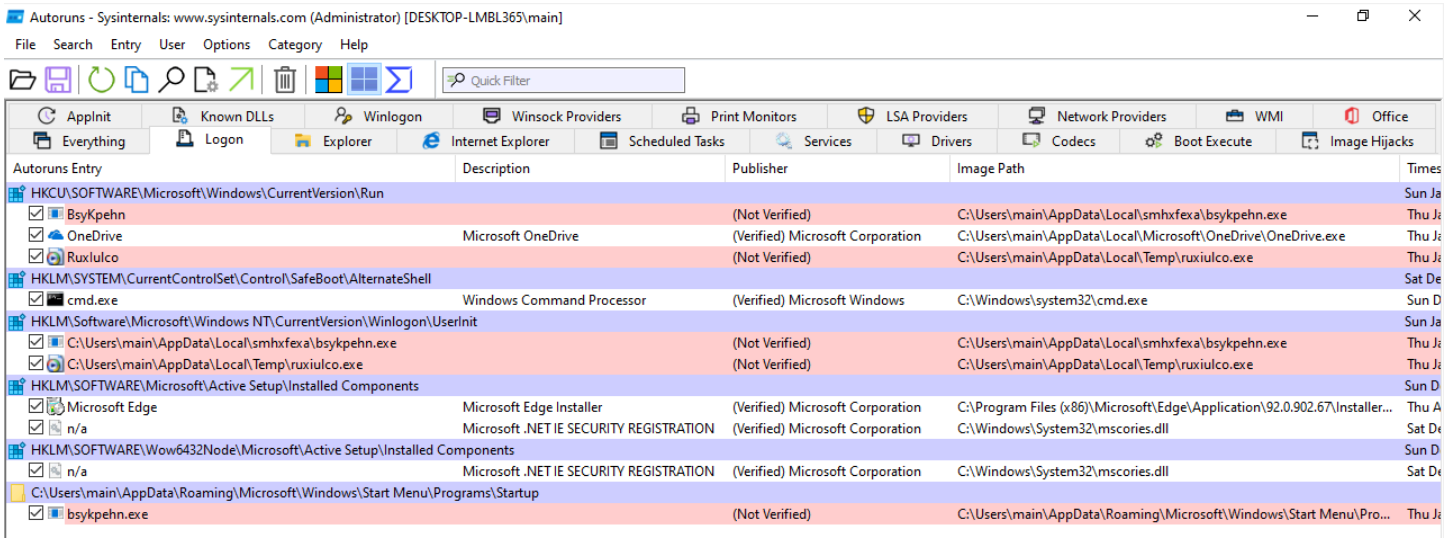
```
---------------------------------
Files added: 16
---------------------------------
C:\Windows\apppatch\CustomSDB\{f48a0c57-7c48-461c-9957-ab255ddc986e}.sdb
C:\Windows\Prefetch\CMD.EXE-AC113AA8.pf
C:\Windows\Prefetch\ISCSICLI.EXE-B5AA58D3.pf
C:\Windows\Prefetch\KOQPWQNT.EXE-D9B3D5FB.pf
C:\Windows\Prefetch\RUXIULCO.EXE-41F66901.pf
C:\Windows\Prefetch\SAMPLE.EXE-AE7C9AF2.pf
C:\Windows\Prefetch\SDBINST.EXE-9A87A34F.pf
C:\Windows\Prefetch\SVCHOST.EXE-78C2CCDD.pf
C:\Windows\Prefetch\X32DBG.EXE-8972ED E0.pf
C:\Users\main\AppData\Local\Microsoft\Windows\ActionCenterCache\windows-systemtoast-securityandmaintenance_22_0.png
C:\Users\main\AppData\Local\Temp\koqpwqnt.exe
C:\Users\main\AppData\Local\Temp\ruxiuloco.exe
C:\Users\main\AppData\Local\smhxfexa\bsykpehn.exe
C:\Users\main\AppData\Local\usenkikp.log
C:\Users\main\AppData\Local\vjlbgfrm.log
C:\Users\main\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\bsykpehn.exe

---------------------------------
Files [attributes?] modified: 17
---------------------------------
C:\Windows\bootstat.dat
C:\Windows\Logs\WindowsUpdate\WindowsUpdate.20260125.061153.467.1.etl
C:\Windows\Prefetch\7ZG.EXE-0F8C4081.pf
C:\Windows\Prefetch\CONHOST.EXE-1F3E9D7E.pf
C:\Windows\Prefetch\CONSENT.EXE-531BD9EA.pf
C:\Windows\Prefetch\DLLHOST.EXE-5E46FA0D.pf
C:\Windows\Prefetch\MPCMDRUN.EXE-F401FB84.pf
C:\Windows\Prefetch\SVCHOST.EXE-5AC380EC.pf
C:\Windows\Prefetch\SVCHOST.EXE-DCBDF9F5.pf
C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT.LOG1
C:\Windows\ServiceProfiles\NetworkService\AppData\Local\Temp\MpCmdRun.log
C:\Windows\SoftwareDistribution\DataStore\DataStore.edb
C:\Windows\SoftwareDistribution\DataStore\DataStore.jfm
C:\Windows\SoftwareDistribution\DataStore\Logs\edb.chk
C:\Windows\SoftwareDistribution\DataStore\Logs\edb.log
C:\Windows\Temp\MpCmdRun.log
C:\Users\main\AppData\Local\Microsoft\Windows\UsrClass.dat.LOG2

---------------------------------
Folders added: 1
---------------------------------
C:\Users\main\AppData\Local\smhxfexa
```

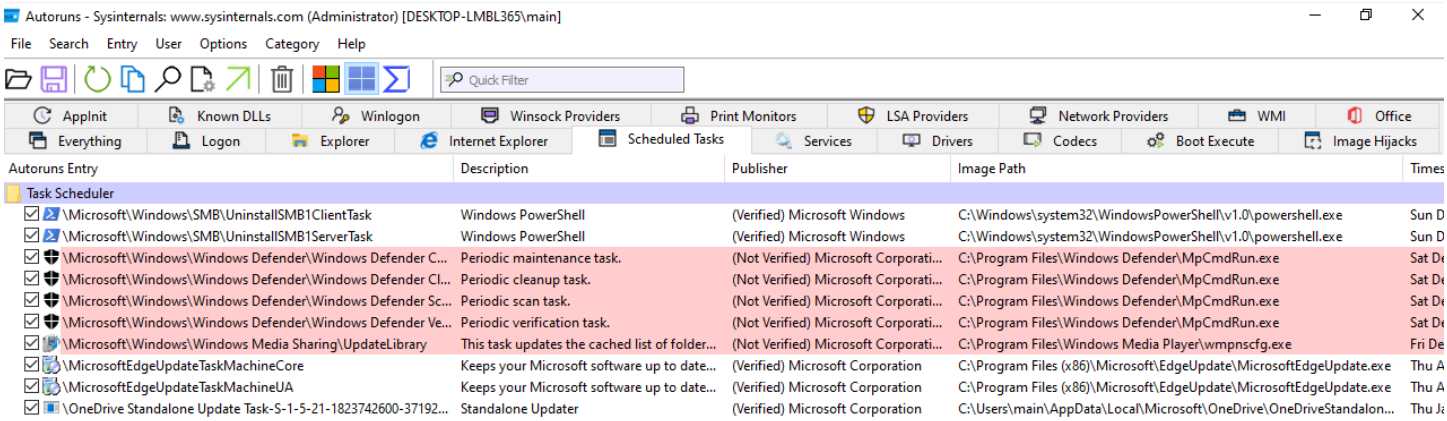
After rebooting the system to test persistence, startup entries remained present; however, Process Explorer did not show any evidence of the malware executing in memory. This suggests that while the malware successfully configured persistence mechanisms, the execution phase of persistence failed, likely due to the network being disabled.





Autoruns analysis revealed multiple persistence mechanisms established by the malware. These included a user-level Run key, modification of the Winlogon UserInit value, and placement of an executable in the user's Startup folder. All persistence entries referenced randomly named, unsigned executables located in user-writable directories. Although these mechanisms were successfully created, post-reboot process analysis did not show continued malware execution, suggesting the loader failed to retrieve or execute a secondary payload in the isolated environment.





Overall analysis indicates that the examined sample functions as a loader-stage Windows malware designed to establish persistence, weaken defenses, and prepare the system for execution of a secondary payload. Static analysis revealed indicators of defense evasion, process execution, and network-aware behavior, while dynamic analysis confirmed successful execution of environment checks and the creation of multiple persistence mechanisms. Despite these activities, post-reboot analysis showed no continued malicious execution, likely due to the absence of network connectivity preventing retrieval or activation of a second-stage payload. This sample seems to be an initial component which performs setup tasks before handing off execution to a more fully featured payload under favorable conditions.

Indicators of Compromise

File Paths

C:\Users\main\Desktop\sample.exe
C:\Users\main\AppData\Local\smhxfexa\bsykpehn.exe
C:\Users\main\AppData\Local\Temp\ruxiloco.exe
C:\Users\main\AppData\Local\Temp\koqpwqnt.exe
C:\Users\main\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\bsykpehn.exe
C:\Users\main\AppData\Local\smhxfexa\

Registry Keys/Values
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\BsyKpehn
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Ruxluco
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\UserInit
HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Processes
HKLM\SOFTWARE\Microsoft\Windows Defender\Exclusions\Extensions
HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Processes
HKLM\SOFTWARE\Microsoft\Microsoft Antimalware\Exclusions\Extensions

Associated Executables Referenced
svchost.exe
rundll32.exe
explorer.exe
consent.exe
spoolsv.exe

Behavioral Indicators
Creation of randomly named executables in user-writable directories
Multiple persistence mechanisms (Run key, Winlogon UserInit, Startup folder)
Attempts to modify Microsoft Defender exclusion lists
Repeated DNS resolution attempts prior to payload retrieval

Recommendations
If this malware were encountered in a real world environment, recommended mitigation steps would focus on rapid isolation, removal of persistence mechanisms, and prevention of reinfection. The affected system should be immediately isolated from the network to prevent further communication and potential delivery of additional payloads. All identified persistence artifacts, including malicious registry Run keys and startup folder executables, should be removed. Additionally, all dropped files in directories such as AppData and Temp should be deleted. Security controls should then be reviewed to ensure no unauthorized Microsoft Defender or antimalware exclusions remain in place, followed by a full system scan with updated signatures. As a precaution, credentials associated with the affected user should be reset. To reduce future risk, systems should be kept fully up to date, the ability to run unknown programs should be limited, and software execution from temporary or user-writable directories should be restricted. User awareness should also be reinforced about the risks of executing untrusted downloads.