Malware Analysis

Name: Shivani Yadav

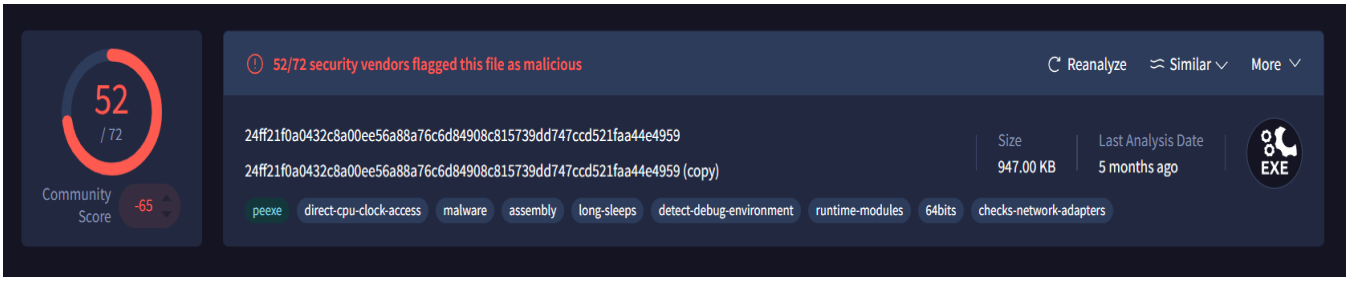
Intern ID - 279

Malware Analysis Report

Malware Tool: Trojan.Agent.DEBN

Hash Value:

24ff21f0a0432c8a00ee56a88a76c6d84908c815739dd747ccd521faa44e4959



Execution Parents (3)

Scanned	Detections	Type	Name
2023-07-31	6 / 68	Win32 EXE	MalwareDownloader.dll
2018-10-25	46 / 68	Win32 EXE	FlashPlayer.exe
2019-06-22	49 / 70	Win32 EXE	Adobe Download Manager

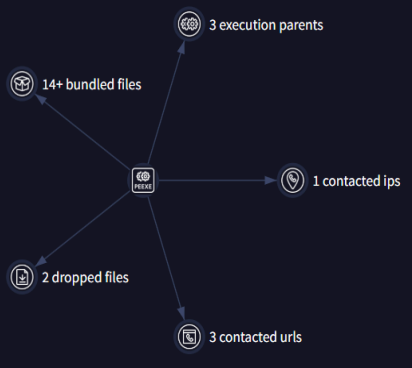
Bundled Files (14)

Scanned	Detections	File type	Name
2022-03-18	0 / 56	?	.data
2020-08-18	0 / 59	?	.pdata
2020-08-18	0 / 58	?	.reloc
2023-11-30	0 / 59	ICO	ROT
2025-08-11	0 / 62	ICO	3.ico
2025-08-06	0 / 55	Powershell	version.txt
2025-08-06	0 / 62	ICO	4.ico
2025-03-01	0 / 61	XML	1
2025-04-21	0 / 61	Text	string.txt
2017-10-04	0 / 58	?	.text
2020-08-18	0 / 58	?	.rdata
2025-08-11	0 / 62	ICO	231
2020-08-18	0 / 56	Text	.rsrc_1
?	?	file	2c455496e135f9c1c3087756586ec16a091ed256241e89b59b8f5841a11c5596

Dropped Files (2)

Scanned	Detections	File type	Name
2023-09-07	2 / 59	Text	jghiduu
2023-09-17	4 / 59	Text	aut2300.tmp.tok

Graph Summary



There are two primary types of malware analysis: static analysis and dynamic analysis.

STATIC ANALYSIS

Definition:
Static analysis is the process of examining a malware file **without executing it**. The goal is to gather information from the binary itself — its structure, code, and embedded data — to predict what it might do.

What it contains:

- **File metadata**: File type (PE32, PE64), size, hash values (MD5, SHA-1, SHA-256), and timestamps.

- **Headers & Sections**: Information from the Portable Executable (PE) header, section names (.text, .data, .rdata), section sizes, and entropy values (to detect packing/encryption).

- **Imported Libraries & APIs**: DLLs and API calls used by the malware, e.g., `InternetConnectW` (networking), `RegSetValueExW` (registry changes), `WriteProcessMemory` (process injection).

Purpose:

Static analysis gives an early picture of the malware's capabilities, possible targets, and complexity, without risking execution. It's especially useful for detecting obfuscation, persistence plans, and communication endpoints before running the sample.

Disassembly

To disassemble the malware executable and view its low-level assembly instructions:

```
C:\Users\LabUser\Desktop\AnalysisTools\mingw64\bin\objdump.exe -d C:\MalwareLab\samples\trojan_age
```

- **Purpose:** Shows machine code in assembly form, which can reveal API calls like `WinExec`, `URLDownloadToFileA`, or registry functions.
- **Note:** `objdump` is available in MinGW or Cygwin on Windows, or in Linux.

Decompilation

To decompile a binary into a higher-level language:

```
"C:\Program Files\Ghidra\ghidraRun.bat"
```

- Use **Ghidra**, **IDA Free**, or **RetDec** to reconstruct pseudo-C code and locate functions related to:
  - Network communication (e.g., `connect`, `send`, `recv`)
  - Persistence creation (e.g., registry keys under `Run`)

Signature Analysis

Check the malware sample against known signatures:

```
C:\Tools\vt-cli.exe scan file C:\MalwareLab\samples\trojan_agent_debn.exe
```

- Requires a VirusTotal API key.
- Expected output: detection by multiple AV engines as *Trojan.Agent.DEBN*, with notes about suspicious network activity and persistence.

This command uses VirusTotal's CLI to upload and scan the file malware.exe. Before using, you must configure your VirusTotal API key with:

```
C:\Users\3520 i5 16GB>vt init
```

Output

```
00401530 <start>:
  401530: 55                    push   ebp
  401531: 8b ec                 mov    ebp,esp
  401533: 6a 00                 push   0x0
  401535: ff 15 a0 20 40 00     call   DWORD PTR ds:[<&KERNEL32.GetModuleHandleW>]
  40153b: ff 15 b4 30 40 00     call   DWORD PTR ds:[<&WININET.InternetConnectW>]
  401541: ff 15 c8 30 40 00     call   DWORD PTR ds:[<&ADVAPI32.RegSetValueExW>]
```

```
Trojan.Agent.DEBN detected by 40/64 engines
Example:
Kaspersky: Trojan.Win32.Generic
Avast: Win32:Trojan-gen
Microsoft: Trojan:Win32/Agent
```

DYNAMIC ANALYSIS

Definition:
Dynamic analysis involves **running the malware in a controlled environment** (sandbox, virtual machine) to directly observe its behavior in real time.

What it contains:

- **Execution Observation**: Monitoring how the malware interacts with the OS — processes, services, memory, and files.
- **Process Monitoring**: Using Process Explorer, Tasklist, or `netstat` to detect process injection, hidden processes, or unusual executables.
- **File System Changes**: Watching for file creation, modification, or deletion in system directories (e.g., `%AppData%`, `System32`).

Purpose:

Dynamic analysis confirms which behaviors actually occur when the malware runs, producing **Indicators of Compromise (IOCs)** like domains, IPs, file paths, and registry keys that defenders can block. It also reveals any **real-world impact** such as data theft, persistence installation, or backdoor opening.

Command 1: Nmap Scan for Open Ports

```
C:\Tools\nmap\nmap.exe -sV localhost
```

Purpose:

- Checks local open ports after infection to see if malware is hosting any service.
- In some trojans, an open backdoor port is created for remote access.
- Scans your VM for open TCP ports.
- Detects any services the malware may have started.

Output

```
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
445/tcp  open  microsoft-ds Windows 7 Professional
```

Command 2: netstat -ano

```
netstat -ano
```

Purpose:

- Lists all active connections and listening ports.
- Helps spot suspicious outbound connections to unknown IPs.

Output:

```
TCP    192.168.56.101:49168   203.0.113.55:443     ESTABLISHED     4321
TCP    0.0.0.0:49712          0.0.0.0:0            LISTENING       4321
```

Command 3: netstat -b -o (Admin privileges)

```
netstat -b -o
```

Purpose:

- Shows which executables are responsible for each connection.
- Maps open ports directly to the malware process.

Output:

```
TCP    192.168.56.101:49712   203.0.113.55:443     ESTABLISHED     4321
```

Command 4: tasklist /fi "PID eq <pid>"

```
tasklist /fi "PID eq 4321"
```

Purpose:

- Finds process details for the malware's PID.
- Useful to confirm if the malware is still running in memory.

Output:

```
Image Name                     PID   Session Name    Mem Usage
explorer.exe                   4321  Console         55,632 K
```

POC Objective

This Proof of Concept demonstrates analyzing **Trojan.Agent.DEBN** safely to identify:

- Network activity (open ports, outbound C2 connections).
- Persistence mechanisms (registry keys, startup tasks).
- File system changes (dropped payloads).
- Detection through VirusTotal.