LetsDefend

Introduction

Static malware analysis is a method of examining malicious software **without** actually executing it. It involves analyzing the code and other characteristics of the malware file without running the program. This type of analysis is typically conducted on the static properties of a file, such as its **code structure, file headers, and embedded resources.**

Key aspects of **static malware analysis** include:

1. Code Inspection: Analysts examine the actual code of the malware to understand its functionality, identify specific malicious behaviors, and uncover any vulnerabilities it may exploit.

2. File Structure Analysis:This involves looking at the overall structure of the malware file, including headers, sections, and any embedded data. This information can provide insights into the file's origin, purpose, and potential evasion techniques.

3. Resource Examination: Some malware may include additional resources such as images, configuration files, or encrypted data. Analysts inspect these resources to gain a more comprehensive understanding of the malware's capabilities.

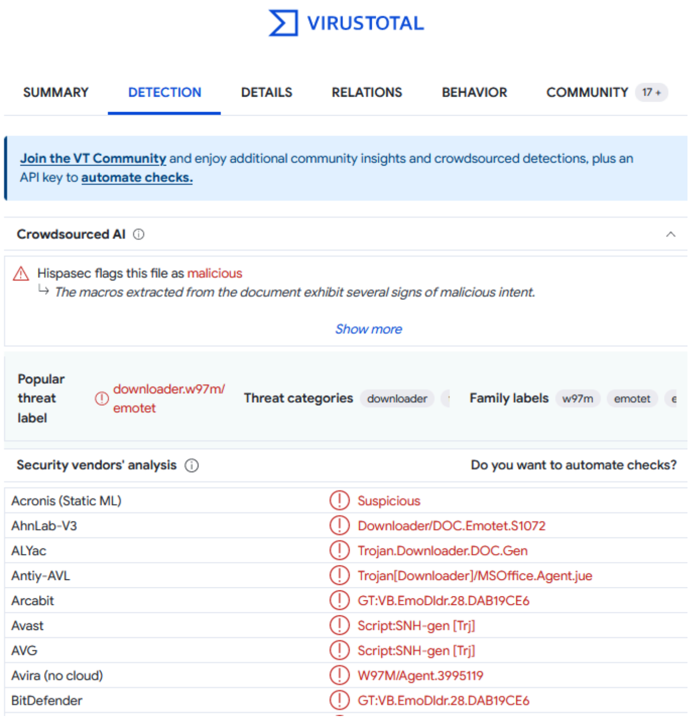
4. String Analysis: Strings, which are sequences of characters, can reveal important information about the malware, such as hardcoded IP addresses, URLs, or encryption keys. Analysts analyze these strings to identify potential indicators of compromise (IOCs).

5. API Calls and Function Analysis: Examining the use of application programming interface (API) calls and functions within the code can help analysts understand how the malware interacts with the operating system and other software components.

Static analysis is advantageous because it doesn't involve running the actual malware, reducing the risk of inadvertently triggering malicious activities. However, it has limitations, as certain malware may employ techniques to evade static analysis, such as code obfuscation or encryption. To overcome these limitations, a combination of static and dynamic analysis (analyzing the malware in a controlled environment by executing it) is often used for a more comprehensive understanding of the threat.

The provided file appears to be potentially malicious. Exercise caution when handling it. The file is named "PO-465514-180820.doc.zip" and is password-protected with the password "infected."

- First, let's verify the file's integrity by checking its **MD5** checksum through the command line. (d7e6921bfd008f707ba52dee374ff3db)
- Next, let's verify the **SHA256** checksum using the command line. (044aa7e93ec81b297b53aaebad9bbac1a9d754219b001aaf5d4261665af30bc7)
- Now that we have obtained the hash, let's switch to VirusTotal to gather details on the files.



Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Crowdsourced AI

Hispasec flags this file as malicious
↳ *The macros extracted from the document exhibit several signs of malicious intent.*

Show more

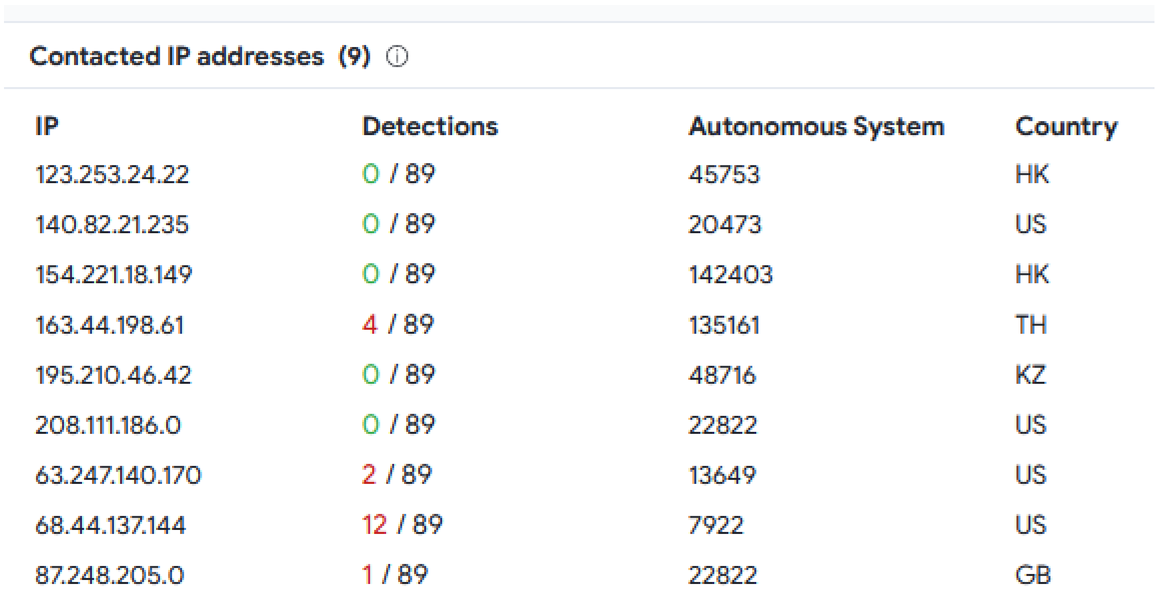
Popular threat label: downloader.w97m/emotet
Threat categories: downloader
Family labels: w97m emotet

Security vendors' analysis — Do you want to automate checks?

Acronis (Static ML)	Suspicious
AhnLab-V3	Downloader/DOC.Emotet.S1072
ALYac	Trojan.Downloader.DOC.Gen
Antiy-AVL	Trojan[Downloader]/MSOffice.Agent.jue
Arcabit	GT:VB.EmoDldr.28.DAB19CE6
Avast	Script:SNH-gen [Trj]
AVG	Script:SNH-gen [Trj]
Avira (no cloud)	W97M/Agent.3995119
BitDefender	GT:VB.EmoDldr.28.DAB19CE6

After reviewing the hash with **VirusTotal**, we have concerning news. Forty-nine security vendors and five sandboxes have flagged this file as malicious.

Websites like VirusTotal are invaluable for static analysis as they reveal the malware's behavior without the need for execution in a sandbox.



Contacted IP addresses (9)

IP	Detections	Autonomous System	Country
123.253.24.22	0 / 89	45753	HK
140.82.21.235	0 / 89	20473	US
154.221.18.149	0 / 89	142403	HK
163.44.198.61	4 / 89	135161	TH
195.210.46.42	0 / 89	48716	KZ
208.111.186.0	0 / 89	22822	US
63.247.140.170	2 / 89	13649	US
68.44.137.144	12 / 89	7922	US
87.248.205.0	1 / 89	22822	GB

Interestingly, on the Relations tab, we observe the contacted IPs, which can be considered as significant Indicators of Compromise (IOCs).

Once we finish reviewing with VirusTotal, let's return to the command line to execute **ExifTool**.

In the context of malware analysis or digital forensics, ExifTool can be employed to extract metadata from suspicious files, aiding in the understanding of the **file's origin**, creation time, and potential clues about its source or purpose. Keep in mind that while ExifTool is a valuable tool, its usage is not limited to security-related tasks and can be applied in various fields where metadata manipulation is required.

After reviewing the file with ExifTool, it confirms that this is a **DOC** file type.

While examining the information, I noticed that the **language code** indicating ‘Russia’ seemed suspicious.

Let's return to the command line and utilize the strings -n 5 [filename] command.

A checklist to examine includes:

1. IP addresses
2. Websites
3. Suspicious file locations
4. Domains
5. Potentially malicious files
6. Suspicious or malicious code (Visual Basic code or Base64)

```
Sub MaliciousMacro()

        MsgBox "Your computer is infected!", vbCritical, "Warning"

        ' Malicious actions could follow this prompt

End Sub
```

In the context of **Microsoft Office macros**, Visual Basic for Applications (VBA) code like the above may be embedded in documents to trigger malicious activities.

```
eval(atob('aW5jbHVkZSA9IGZ1bmN0aW9uKCkgew0KICAgIHNldFRpbWV
vdXQoJ2J5dGVzJywgWwo ...
```

Base64 encoding is commonly used to obfuscate malicious payloads. The actual payload is hidden within the encoded string, and it needs to be decoded before analysis.

The final tool in use will be **XORsearch**, employed with the command: `xorsearch [filename]`.

XORsearch is a tool designed to identify patterns indicative of XOR encryption in binary data. It scans a specified file for byte sequences that exhibit XOR-like characteristics, aiding in the recognition of potentially XOR-encoded content. Keep in mind that XORsearch does not automatically decrypt the data; it is a pattern recognition tool.