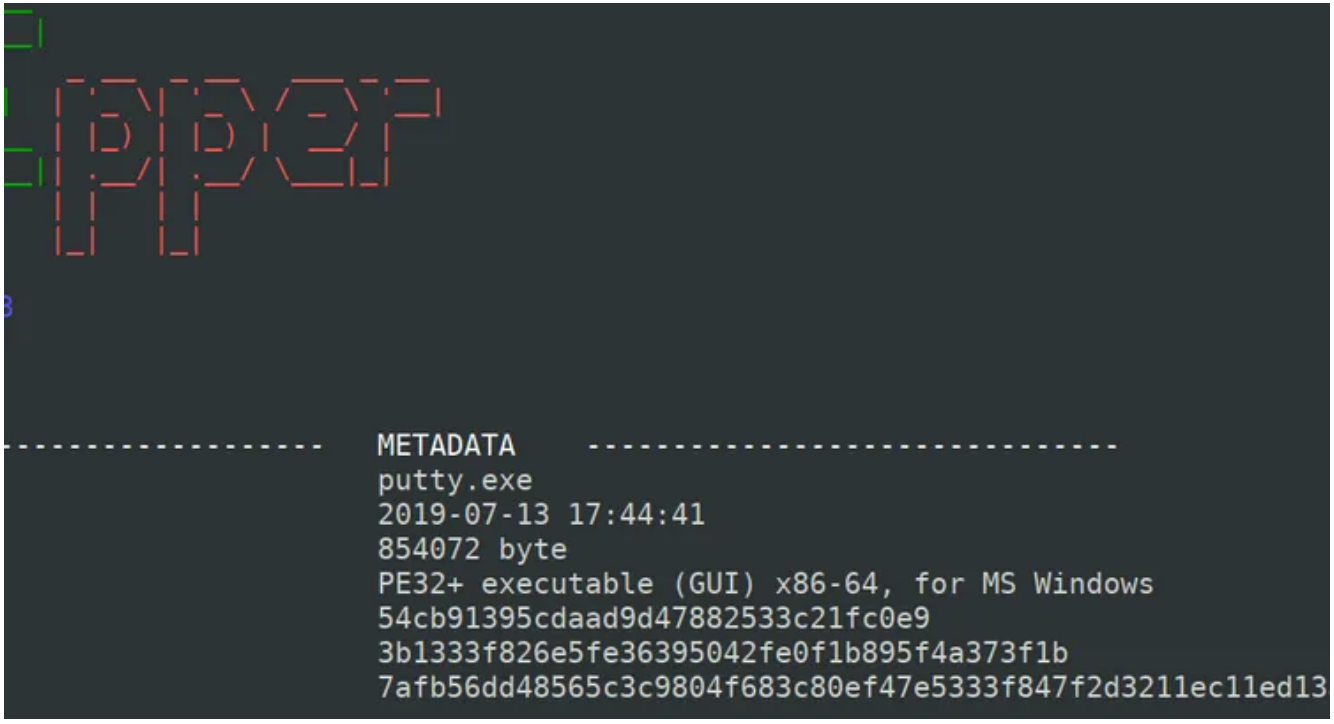
So Long, and Thanks for All the Fish

JUST SOME RANDOM THOUGHTS ABOUT THE MEANING OF LIFE, THE UNIVERSE, AND EVERYTHING

MENU



PEpper: a python script to perform malware static analysis on Portable Executable format

RECENT POSTS

#WIBattack: Not only S@T Browser, but also WIB SIM toolKit is vulnerable to SimJacker attacks

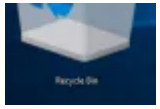
September 28, 2019

Checkm8: a new 'unpatchable' jailbreak for all iOS devices from iPhone 4s to iPhone X

September 27, 2019

How the progress bar keeps you sane, by Daniel Engber

September 27, 2019



Windows Forensics: analysis of Recycle bin artifacts

September 26, 2019

CATEGORIES

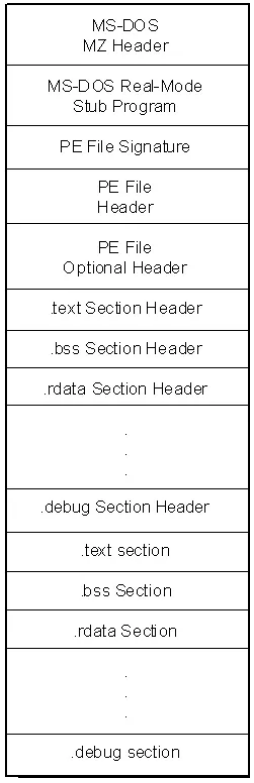
Written by Andrea Fortuna • on September 25, 2019 • in Cybersecurity, Malware Analysis

A useful tool: fast and easy to use.

What is Portable Executable

The **P**ortable **E**xecutable format is the standard file format for executables, object code and DLLs used in 32- and 64-bit versions of Windows operating systems.

The PE file format is organized as a linear stream of data.
It begins with an MS-DOS header, a real-mode program stub, and a PE file signature.
Following is a PE file header and optional header. Beyond that, all the section headers appear, followed by all of the section bodies.
Closing out the file are a few other regions of miscellaneous information, including relocation information, symbol table information, line number information, and string table data.



PEpper

PEpper is a python script, developed to perform *malware static analysis* on **P**ortable **E**xecutable.

Static analysis checks for malware without inspecting the actual code or instructions. It applies various techniques and tools to swiftly decide whether the file is malicious or not.
The indicators collected using static analysis may comprise the file name, file type, file size, and MD5 checksums or hashes recognized by antivirus detection tools.
PEpper is able to extract this set of information:

- **Suspicious entropy** ratio
- **Suspicious name** ratio
- Suspicious **code size**
- Suspicious **debugging time-stamp**
- Number of **export**
- Number of **anti-debugging** calls
- Number of **virtual-machine detection** calls
- Number of **suspicious API** calls
- Number of **suspicious strings**
- Number of **YARA** rules matches
- Number of **URL** found
- Number of **IP** found
- *Cookie on the stack* (**GS**) support
- *Control Flow Guard* (**CFG**) support
- *Data Execution Prevention* (**DEP**) support
- *Address Space Layout Randomization* (**ASLR**) support
- *Structured Exception Handling* (**SEH**) support

exploited by surveillance companies for espionage operation

Andrea Fortuna on How to make a “Ultra-Geek” Linux Workstation

- *Thread Local Storage* (**TLS**) support
- Presence of **manifest**
- Presence of **version**
- Presence of **digital certificate**
- **Packer** detection
- **VirusTotal** database detection
- **Import hash**

The installation is fast and simple: just clone the git repository, resolve the dependencies and install the tool using PIP:

```
$ git clone https://github.com/Th3Hurrican3/PEpper/
$ cd PEpper
$ pip3 install -r requirements.txt
$ python3 pepper.py ./malware_dir
```

References and downloads

- https://github.com/Th3Hurrican3/PEpper

Related posts

Like this:

Loading...

TAGS

CYBERSECURITY MALWARE ANALYSIS PYTHON

PRINT

Andrea Fortuna

How to install latest Widevine plugin on Chromium

COMMENTS

Enter your comment here...

This site uses Akismet to reduce spam. Learn how your comment data is processed.