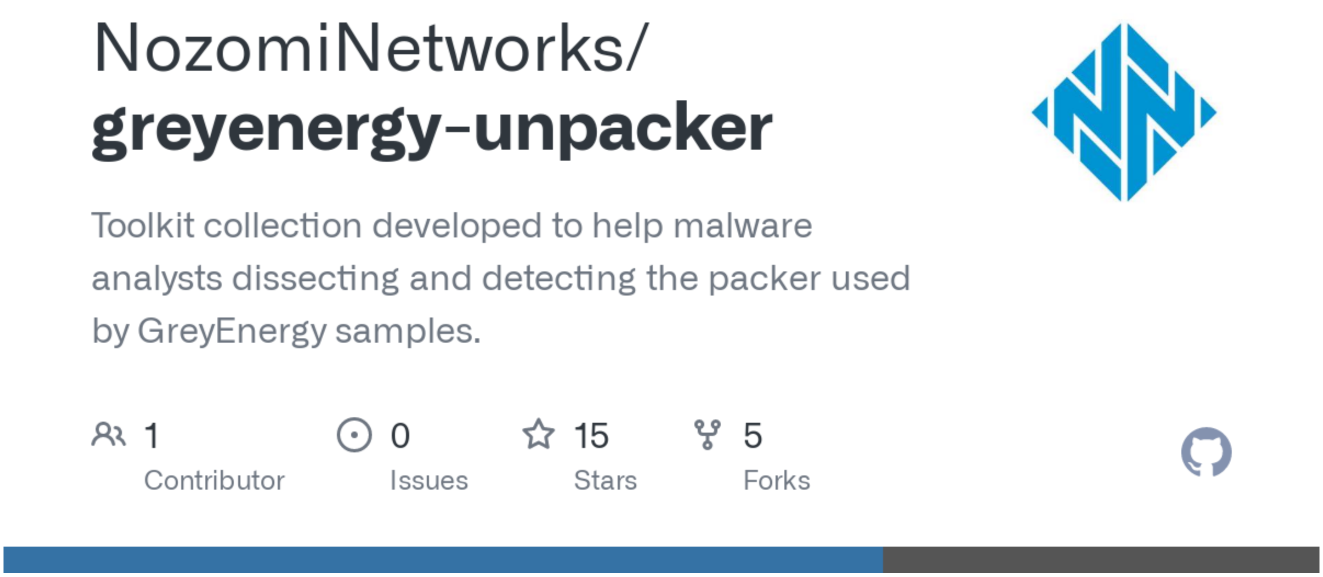
NozomiNetworks/greyenergy-unpacker: Toolkit collection developed to help malware analysts dissecting and detecting the packer used by GreyEnergy samples.

github.com/NozomiNetworks/greyenergy-unpacker

NozomiNetworks



NozomiNetworks/
greyenergy-unpacker

Toolkit collection developed to help malware analysts dissecting and detecting the packer used by GreyEnergy samples.

1 Contributor

0 Issues

15 Stars

5 Forks

Repository

Toolkit collection developed to help malware analysts dissecting and detecting the packer used by GreyEnergy samples.

Packer Overview

The GreyEnergy dropper is protected using a custom packer with the following characteristics:

- Custom decryption algorithm
- LZW (variant) decompression algorithm
- Junk code & JMP instructions (anti-analysis)
- Memory wiping (anti-forensic)
- Dynamically-resolved WinAPIs
- Overlay data payload (There are no suspicious sections in the PE header.)

Once the dropper has been decrypted/decompressed in memory, the packer performs the following steps:

- Parses the dropper's PE header, searching for the appended data
- Copies the final malware in memory, reading it from the appended data
- Resolves the dropper's imports
- Relocates the dropper's executable
- Jumps to the dropper's entry point

The dropper stores the final malware in the filesystem, establishing persistence.

greyenergy_unpacker.py

An easy-to-run tool that automatically extracts GreyEnergy packed files.

Usage

Unpacks the malware storing it on the disk.

```
python3 greyenergy_unpacker.py -f suspicious.bin
INFO : Processing the file 'suspicious.bin' (SHA256
d4e97a18be820a1a3af639c9bca21c5f85a3f49a37275b37fd012faeffcb7c4a)
INFO : Malware unpacked in 'suspicious.bin_malware_unpacked.bin' (SHA256
7e154d5be14560b8b2c16969effdb8417559758711b05615513d1c84e56be076)
```

Unpacks the malware dumping also the dropper component.

```
python3 greyenergy_unpacker.py -d -f suspicious.bin
INFO : Processing the file 'suspicious.bin' (SHA256
d4e97a18be820a1a3af639c9bca21c5f85a3f49a37275b37fd012faeffcb7c4a)
INFO : Dropper unpacked in 'suspicious.bin_dropper_unpacked.bin' (SHA256
a7d3f2b6cec72a324c375e4335e42b1f1f4d9642347d38f3de7ec706fcf25147)
INFO : Malware unpacked in 'suspicious.bin_malware_unpacked.bin' (SHA256
7e154d5be14560b8b2c16969effdb8417559758711b05615513d1c84e56be076)
```

Yara module

The file `greyenergy.c` is a YARA module developed to parse the GreyEnergy packer, decrypting only the first part of the appdata in order to confirm the detection.

After the compilation, it is possible to detect the malicious file just using the new keyword `is_packed`.

Installation

Detailed information about the compilation can be found in the official Yara documentation

The file `build.sh` contained inside the Yara's root directory configures and compiles automatically the source code. Currently it is not mentioned in the documentation, so that could be changed in the near future.

Rule example

```
import "pe"
import "greyenergy"

rule GreyEnergyPacker {
  condition:
    greyenergy.is_packed(pe.overlay.offset)
}
```

Tested Samples

Both the Yara module and the unpacker script have been successfully used to unpack the following samples (SHA-256):

```
b60c0c04badc8c5defab653c581d57505b3455817b57ee70af74311fa0b65e22
d4e97a18be820a1a3af639c9bca21c5f85a3f49a37275b37fd012faeffcb7c4a
```