GreyEnergy Malware Research Paper: Maldoc to Backdoor

nozominetworks.com/2019/02/12/blog/greyenergy-malware-research-paper-maldoc-to-backdoor/

By February 12, 2019

As a security researcher I believe it's important for those defending critical and industrial infrastructure to share knowledge and stay up-to-date on malware tradecraft.

So, when the GreyEnergy Advanced Persistent Threat (APT) was unveiled by ESET last year, I put my reverse engineering skills to work to analyze one of the malware's infection techniques. This was the phishing email containing a malicious Microsoft Word document (maldoc) that lead to the installation of the malware (backdoor) on a victim's network.

Today I am publishing a Research Paper that provides a comprehensive analysis of how the malware works, from the maldoc, to the custom packer and the final dropper (backdoor). This investigation is a more detailed analysis than what I put forward in a blog article in November 2018. And, the deepest analysis is done on the packer, an executable that decrypts and decompresses another executable inside itself.

This article provides a summary of the techniques used by the packer to conceal its true functionality and provides a link to download my full Research Paper, *GreyEnergy: Dissecting the Malware from Maldoc to Backdoor, Comprehensive Reverse Engineering Analysis*.

Once the GreyEnergy malware infects a system, it does a very good job of using anti-analysis techniques to conceal its true functionality.

GreyEnergy Anti-Analysis Techniques Conceal its Suspected "Packer" Executable

When someone opens the Word document contained in the GreyEnergy phishing email, and clicks on "Enable Content", malicious code is downloaded from a remote location.

The downloaded file is an executable which I suspected was a "packer", i.e. an executable which contains one or more executables that are compressed and encrypted. While sometimes used legitimately to protect intellectual property, packers are also used by threat actors to hide malware.

As I investigated the suspected packer executable, I found it was built using several anti-analysis techniques:

Junk code – unnecessary code that has no impact on the suspected packer's code, and whose purpose is to confuse the reverse engineer. I determined that GreyEnergy contains a massive amount of junk code.

Overlapping instructions – GreyEnergy uses JMP instructions that function as overlapping instructions, where the same sequence of bytes can be interpreted as different instructions, depending on the exact byte in which execution starts.

JMP-based execution code – the execution flow of the suspected GreyEnergy packer is almost completely based on the use of JMP instructions, instead of sequential instructions. This makes it very hard to identify the true executable, hidden in a sea of junk code.

Furthermore, the binary file of the suspected packer appeared to have **overlay data.** This is data appended at the end of the file that includes an additional executable component, and is decrypted during run-time.

Entropy – this is an assessment of a file's randomness. Using one measure of entropy, with a scale of 0 to 8, where results of 7 or more indicate encryption, GreyEnergy has a score of 7.994. This is a strong indicator that the overlay data is encrypted.

Dynamic Analysis Reveals the Malware

After assessing the above aspects of the malware, I had a strong suspicion that I was dealing with a packer, but lacked solid proof. I decided to switch to a dynamic analysis approach to order to speed up the investigation. I then discovered several interesting attributes of the suspected packer file:

Hardcoded imports – the *WinAPIs* called by the suspected packer are not contained in the PE import table, but loaded at runtime and pushed ono the stack using a *mov* instruction, without any kind of obfuscation technique.

String overwrite – the suspected packer overwrites all strings with zeros, after the strings have been loaded into memory.

By now, there are multiple indicators that strongly suggest that the binary is a packer:

- Apparently encrypted overlay
- Anti-analysis techniques
- APIs manually resolved by parsing the PE header
- Strings hardcoded inside the code and overwritten with *0x00s* after use

Accessing the overlay – the malware uses a series of steps to identify where the overlay starts and the exact size of its own executable, and allocates space for itself inside the memory. My analysis reveals exactly how the malware identifies the right offset for the overlay.

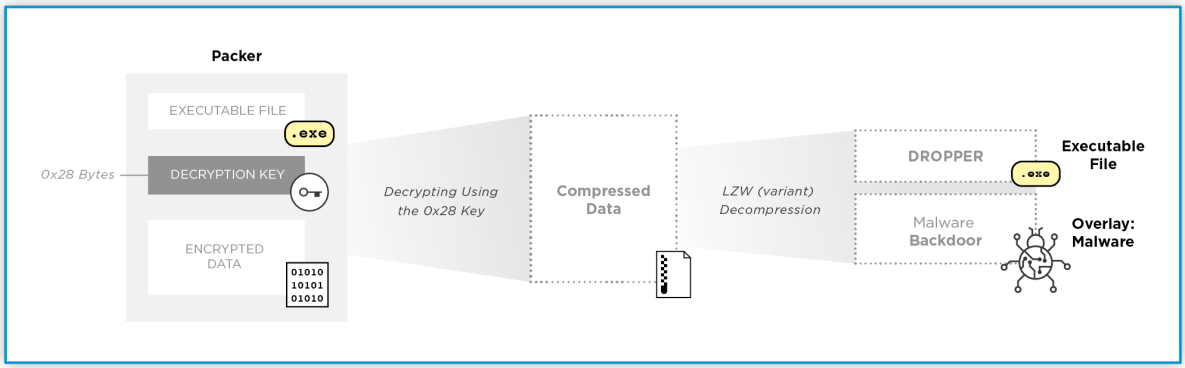
Decryption algorithm – the malware uses a custom algorithm to hide its malicious components. When the decryption algorithm is applied, it is clear the data contains an executable. However, there are several unexpected bytes between the recognized patterns, indicating that the data is not yet complete. I suspected that the data is compressed somehow.

Decompression algorithm – my suspicion is quickly confirmed, and after decompression, the new buffer contains a valid PE header.

The original entry point (OEP) – next the packer points to the uncompressed buffer, parses the PE header and iterates all sections again. Once it accesses the overlay data, a second PE header is revealed, which is the real malicious component (backdoor), waiting to be installed inside the victim's systems.

It's now possible to identify two specific components from the unpacked data – the dropper and the backdoor.

The suspected packer executes the dropper in-memory without storing it inside the filesystem. This step confirms that the binary is a packer, because it has just demonstrated all the primary characteristics of packers.



The flow executed by the Packer includes decryption and decompression of the Dropper and Backdoor. (Click to enlarge)

GreyEnergy – A Stealthy Infection Requiring Proactive Defenses

Once complete, my analysis showed that the GreyEnergy packer is robust and capable of significantly slowing down the reverse engineering process. The techniques used are not new, but both the tools and the tactics employed were cleverly selected. The threat actors' broad use of anti-forensic techniques underlines their attempt to be stealthy and ensure that the infection would go unnoticed.

I urge you to download my full Research Paper, containing all the details related to reverse engineering the packer, as well as my analysis of the malicious Word document and the dropper.

Based on how well the malware disguises itself once it infects a system, the best way for industrial organizations to protect themselves from the GreyEnergy APT is to train employees on the dangers of email phishing campaigns, including how to recognize malicious emails and attachments.

In addition, critical infrastructure networks should always be monitored with dedicated cyber security systems to proactively detect threats present on the network.

Free Tools to Help the Security Community Defend Against GreyEnergy

As a direct outcome of this analysis, I developed tools to help analysts dissect this piece of malware. The ***GreyEnergy Yara Module***, is high-performing code for compiling with the Yara engine. It adds a new keyword that determines whether a file processed by Yara is the GreyEnergy packer or not.

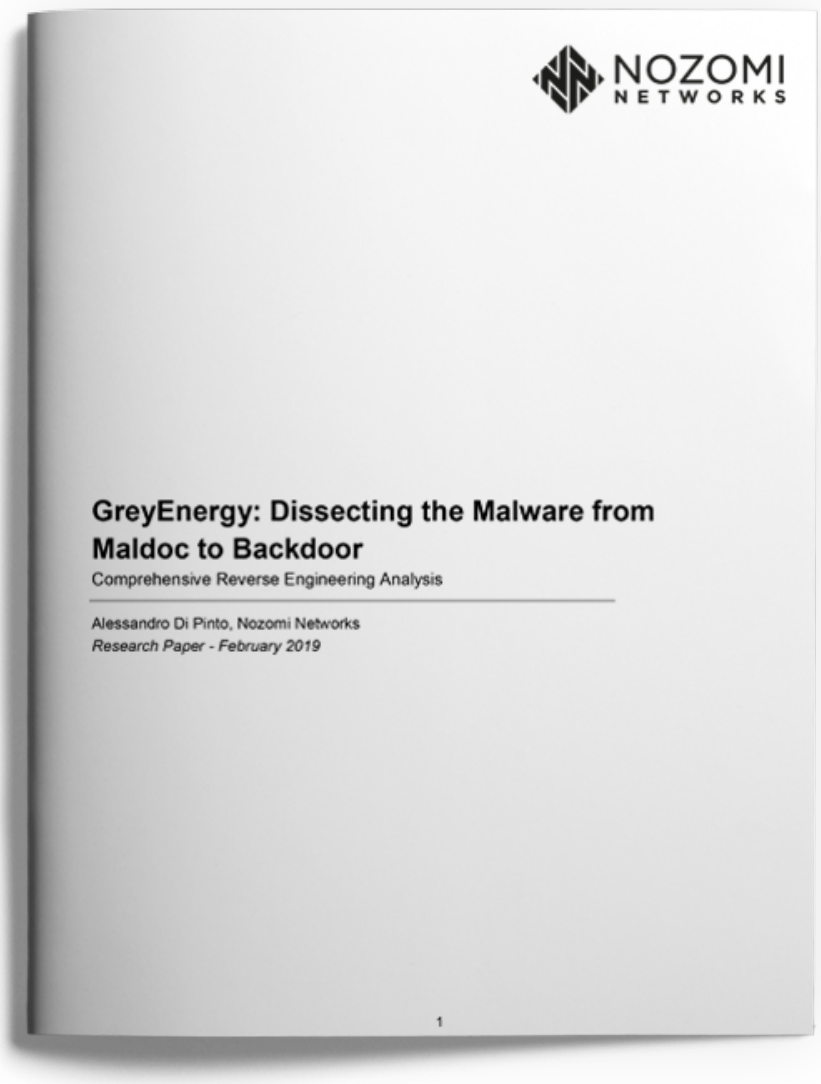
This tool, combined with the previously published ***GreyEnergy Unpacker*** (a Python script that automatically unpacks both the dropper and the backdoor, extracting them onto a disk), saves other security analysts the reverse engineering work explained in this paper.

I hope that these tools, along with my findings, facilitate further GreyEnergy analysis and help the security community better defend critical infrastructure systems in the future.

Related Content to Download

RESEARCH PAPER

GreyEnergy: Dissecting the Malware from Maldoc to Backdoor
Comprehensive Reverse Engineering Analysis



Read this paper to learn:

- The high-level flow of the GreyEnergy phishing campaign
- How the malware disguises itself and its functionality
- How each stage of the malware works:
    - Stage 0 – Malicious Word Document
    - Stage 1 – Packer
    - Stage 2 – Dropper
- About two new tools for further GreyEnergy analysis

DOWNLOAD

Alessandro Di Pinto
Security Research Manager, Nozomi Networks
@adipinto

Alessandro Di Pinto is an Offensive Security Certified Professional (OSCP) with an extensive background in malware analysis, ICS/SCADA security, penetration testing and incident response. He holds GIAC Reverse Engineering Malware (GREM) and GIAC Cyber Threat Intelligence (GCTI) certifications. Alessandro co-authored the research paper “TRITON: The First ICS Cyber Attack on Safety Instrument Systems” and “Analyzing the GreyEnergy Malware: from Maldoc to Backdoor”.