# Tricks and Treats: GHOSTPULSE’s new pixel- level deception

elastic.co/security-labs/tricks-and-treats

Subscribe

19 October 2024•Salim Bitam

The updated GHOSTPULSE malware has evolved to embed malicious data directly within pixel structures, making it harder to detect and requiring new analysis and detection techniques.

13 min read

Malware analysis

## Update

This research covers an update to stage 2 of GHOSTPULSE, originally disclosed by Elastic Security Labs in October 2023.

## Key takeaways

1. GHOSTPULSE has shifted from using the IDAT chunk of PNG files to embedding its encrypted configuration and payload within the pixel structure.
2. Recent campaigns involve tricking victims with creative social engineering techniques, such as CAPTCHA validations that trigger malicious commands through Windows keyboard shortcuts.
3. Elastic Security has enhanced its YARA rules and updated the configuration extractor tool to detect and analyze both the old and new versions of GHOSTPULSE.

## Preamble

The GHOSTPULSE malware family (also known as HIJACKLOADER or IDATLOADER) has continuously evolved since its discovery in 2023, evading detection with increasingly developed techniques.

In its earlier iterations, GHOSTPULSE abused the IDAT chunk of PNG files to hide malicious payloads, as detailed in a previous article from Elastic Security Labs. However, recent analysis has uncovered a significant change in its algorithm. Instead of extracting the payload from the IDAT chunk, the latest version of GHOSTPULSE now parses the pixels of the image to retrieve its configuration and payload. This new approach involves embedding malicious data directly within the pixel structure.

In this research publication, we’ll explore this new pixel-based algorithm and compare it with the previous IDAT chunk technique with updated detection rules.

## Introduction

Recently, we've observed several campaigns involving LUMMA STEALER using GHOSTPULSE as its loader, a topic also explored by HarfangLab. These campaigns stand out due to their creative social engineering tactics. Victims are tricked into validating a CAPTCHA, but the website instructs them to execute a series of Windows keyboard shortcuts instead of the usual process. These shortcuts trigger a command copied to the clipboard by malicious JavaScript. This leads to a PowerShell script being executed, initiating the infection chain by downloading and executing a GHOSTPULSE payload.

In previous versions of GHOSTPULSE, it was delivered as part of a multi-file package. This package typically contained a benign executable, an infected DLL loaded by the executable, and a PNG file storing the encrypted configuration.

However, in the latest version, GHOSTPULSE has streamlined its deployment. Now, the entire package consists of a single file—a benign but compromised executable that includes the PNG file within its resources section.

## Technical analysis

The updated second stage of the malware retains much of its previous structure, including using the same hashing algorithm for resolving Windows API names. However, the most significant change is in how the malware now locates its configuration, which holds both the payload and critical instructions for its deployment.

The following is a screenshot showing the pseudocode of both implementations:

In earlier versions, GHOSTPULSE would parse a PNG file for an encrypted data blob, which was divided into chunks and stored sequentially. The malware's parsing process was straightforward: it would search for a specific marker within the file—in this case, the IDAT string. Once found, the malware would check for a 4-byte tag that followed the string. The encrypted chunk would be extracted if this tag matched the expected value. This process continues for every occurrence of the IDAT string that comes after until the full encrypted payload is collected.

In the new version, the encrypted configuration is stored in the pixels of the image. The malware constructs a byte array by extracting each pixel's `RED`, `GREEN`, and `BLUE` (RGB) values sequentially using standard Windows APIs from the GdiPlus(GDI+) library. Once the byte array is built, the malware searches for the start of a structure that contains the encrypted GHOSTPULSE configuration, including the XOR key needed for decryption. It does this by looping through the byte array in 16-byte blocks. For each block, the first 4 bytes represent a CRC32 hash, and the next 12 bytes are the data to be hashed. The malware computes the CRC32 of the 12 bytes and checks if it matches the hash. If a match is found, it extracts the offset of the encrypted GHOSTPULSE configuration, its size, and the 4-byte XOR key, and then XOR decrypts it.

The following diagram provides a visual breakdown of this process:

## Updated configuration extractor

Based on these findings, we have updated our configuration extractor to support both versions of GHOSTPULSE. This tool takes a PNG file as input and outputs the embedded payload. You can find the updated tool in our labs-releases repository.

## Detecting GHOSTPULSE with YARA

The original GHOSTPULSE YARA rule still prevents the final stage of an infection and is built into Elastic Defend. The updated sample can be detected using the following YARA rules and will be included with Elastic Defend in a future release.

Elastic Security has updated the GHOSTPULSE YARA rules to identify this activity:

```
rule Windows_Trojan_GHOSTPULSE_1 {
    meta:
        author = "Elastic Security"
        creation_date = "2024-10-15"
        last_modified = "2024-10-15"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "GHOSTPULSE"
        threat_name = "Windows.Trojan.GHOSTPULSE"
        license = "Elastic License v2"

    strings:
        $stage_1 = { 49 63 D0 42 8B 0C 0A 41 03 CA 89 0C 1A 8B 05 ?? ?? ?? ?? 44 03 C0 8B 05 ?? ?? ?? ?? 44 3B C0 }
        $stage_2 = { 48 89 01 48 8B 84 24 D8 00 00 00 48 8B 4C 24 78 8B 49 0C 89 08 C7 44 24 44 00 00 00 00 }

    condition:
        any of them
}

rule Windows_Trojan_GHOSTPULSE_2 {
    meta:
        author = "Elastic Security"
        creation_date = "2024-10-10"
        last_modified = "2024-10-10"
        os = "Windows"
        arch = "x86"
        category_type = "Trojan"
        family = "GHOSTPULSE"
        threat_name = "Windows.Trojan.GHOSTPULSE"
        license = "Elastic License v2"

    strings:
        $a1 = { 48 83 EC 18 C7 04 24 00 00 00 00 8B 04 24 48 8B 4C 24 20 0F B7 04 41 85 C0 74 0A 8B 04 24 FF C0 89
04 24 EB E6 C7 44 24 08 00 00 00 00 8B 04 24 FF C8 8B C0 48 8B 4C 24 20 0F B7 04 41 83 F8 5C }

    condition:
        all of them
}
```

## Conclusion

In summary, the GHOSTPULSE malware family has evolved since its release in 2023, with this recent update marking one of the most significant changes.

As attackers continue to innovate, defenders must adapt by utilizing updated tools and techniques to mitigate these threats effectively. We are excited to share our newly developed configuration extractor tool, designed to analyze the older and newer versions of GHOSTPULSE. This tool empowers researchers and cybersecurity professionals by providing enhanced capabilities for understanding and combating these evolving threats. As the landscape of cyber threats changes, collaboration, and innovation remain essential for effective protection.

## Observations

All observables are also available for download in both ECS and STIX format.

The following observables were discussed in this research.

| Observable | Type | Name | Reference |
|---|---|---|---|
| 57ebf79c384366162cb0f13de0de4fc1300ebb733584e2d8887505f22f877077 | SHA-256 | Setup.exe | GHOSTPULSE sample |
| b54d9db283e6c958697bfc4f97a5dd0ba585bc1d05267569264a2d700f0799ae | SHA-256 | Setup_light.exe | GHOSTPULSE sample |

| Observable | Type | Name | Reference |
|---|---|---|---|
| winrar01.b-cdn[.]net | domain-name | | Infrastructure hosting GHOSTPULSE sample |
| reinforcenh[.]shop | domain-name | | LUMMASTEALER C2 |
| stogeneratmns[.]shop | domain-name | | LUMMASTEALER C2 |
| fragnantbui[.]shop | domain-name | | LUMMASTEALER C2 |
| drawzhotdog[.]shop | domain-name | | LUMMASTEALER C2 |
| vozmeatillu[.]shop | domain-name | | LUMMASTEALER C2 |
| offensivedzvju[.]shop | domain-name | | LUMMASTEALER C2 |
| ghostreedmnu[.]shop | domain-name | | LUMMASTEALER C2 |
| gutterydhowi[.]shop | domain-name | | LUMMASTEALER C2 |
| riderratttinow[.]shop | domain-name | | LUMMASTEALER C2 |