Dec 07 2023

NETMANAGEIT

Intelligence Report

Getting gooey with GULOADER: deobfuscating the downloader

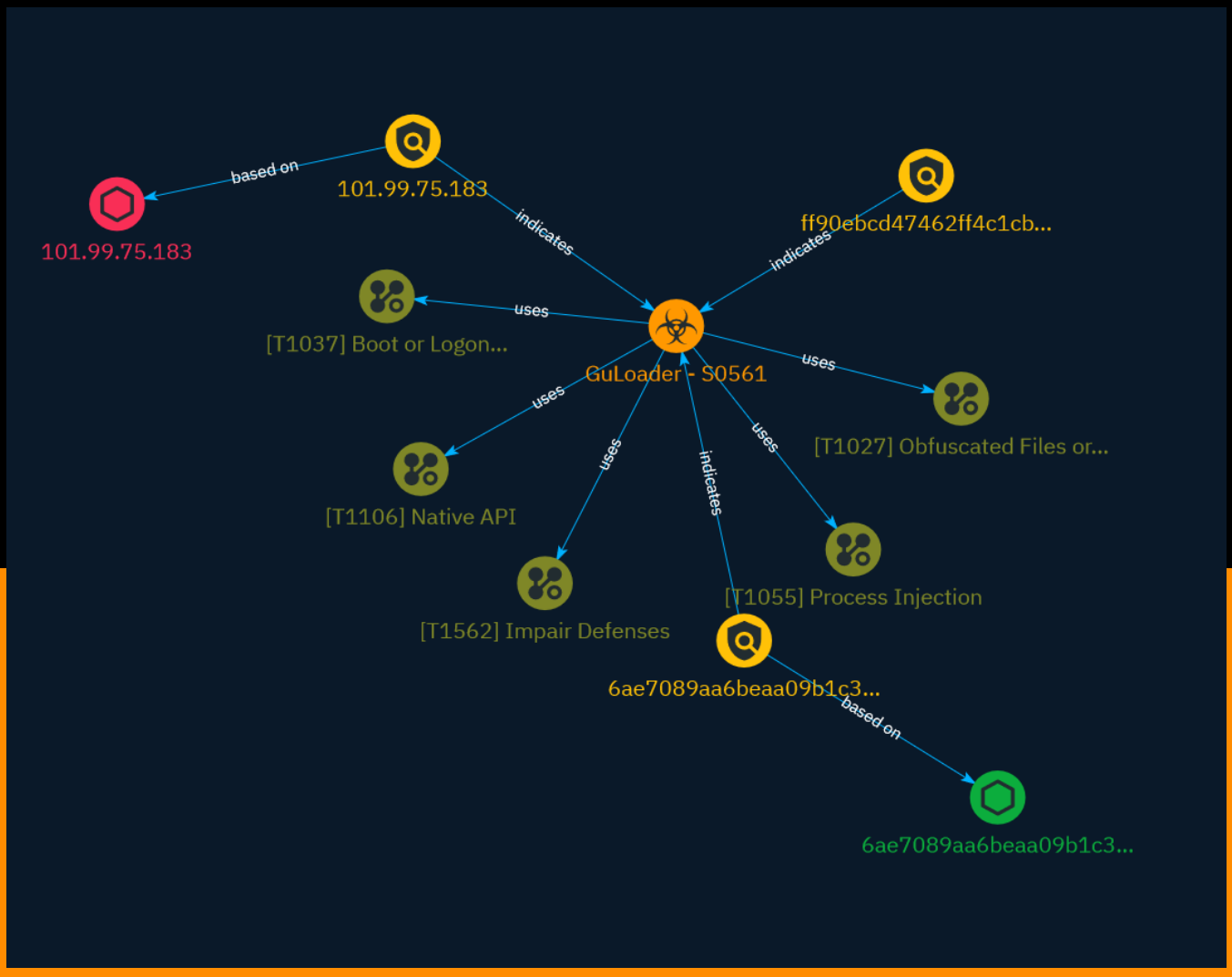

Table of contents

Overview

Entities

Observables

External References

Overview

Description

GULOADER is a well-known shellcode downloader that has a number of anti-analysis tricks designed to make it difficult to spot when it is being used by security researchers and researchers.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Content

N/A

Attack-Pattern

Name
Process Injection
ID
T1055
Description
Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.
Name
Impair Defenses
ID
T1562

Description

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

Name

Native API

ID

T1106

Description

Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. (Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. Native API functions (such as `NtCreateProcess`) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API

functions exist for various system operations.(Citation: Microsoft Win32)(Citation: LIBC) (Citation: GLIBC) Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS Cocoa)(Citation: macOS Foundation) Adversaries may use assembly to directly or indirectly invoke syscalls in an attempt to subvert defensive sensors and detection signatures such as user mode API-hooks.(Citation: Redops Syscalls) Adversaries may also attempt to tamper with sensors and defensive tools associated with API monitoring, such as unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001).

Name

Obfuscated Files or Information

ID

T1027

Description

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control

mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

Name

Boot or Logon Initialization Scripts

ID

T1037

Description

Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely. Adversaries may use these scripts to maintain persistence on a single system. Depending on the access configuration of the logon scripts, either local credentials or an administrator account may be necessary. An adversary may also be able to escalate their privileges since some boot or logon initialization scripts run with higher privileges.

Indicator

Name
6ae7089aa6beaa09b1c3aa3ecf28a884d8ca84f780aab39902223721493b1f99
Description
Nullsoft_NSIS
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '6ae7089aa6beaa09b1c3aa3ecf28a884d8ca84f780aab39902223721493b1f99']
Name
101.99.75.183
Description
ISP: Shinjiru Technology Sdn Bhd **OS:** None ---------- Hostnames: - server1.kamon.la ---------- Domains: - kamon.la ---------- Services: **80:** ``` HTTP/1.1 404 Not Found Date: Thu, 23 Nov 2023 21:33:19 GMT Server: Apache/2.4.41 (Ubuntu) Connection: close Content-Length: 0 Content-Type: text/html; charset=UTF-8 ``` ----------

Pattern Type

stix

Pattern

[ipv4-addr:value = '101.99.75.183']

Name

ff90ebcd47462ff4c1cbd466bdf955febe477316

Pattern Type

yara

Pattern

rule Windows_Trojan_Guloader { meta: author = "Elastic Security" creation_date = "2023-10-30" last_modified = "2023-11-02" reference_sample = "6ae7089aa6beaa09b1c3aa3ecf28a884d8ca84f780aab39902223721493b1f99" severity = 100 arch = "x86" threat_name = "Windows.Trojan.Guloader" license = "Elastic License v2" os = "windows" strings: $djb2_str_compare = { 83 C0 08 83 3C 04 00 0F 84 [4] 39 14 04 75 } $check_exception = { 8B 45 ?? 8B 00 38 EC 8B 58 ?? 84 FD 81 38 05 00 00 C0 } $parse_mem = { 18 00 10 00 00 83 C0 18 50 83 E8 04 81 00 00 10 00 00 50 } $hw_bp = { 39 48 0C 0F 85 [4] 39 48 10 0F 85 [4] 39 48 14 0F 85 [7] 39 48 18 } $scan_protection = { 39 ?? 14 8B [5] 0F 84 } condition: 2 of them }

Malware

Name
GuLoader - S0561

StixFile

Value
6ae7089aa6beaa09b1c3aa3ecf28a884d8ca84f780aab39902223721493b1f99

IPv4-Addr

Value
101.99.75.183

External References

- https://otx.alienvault.com/pulse/65720200ae1af0d2096610bf
- https://www.elastic.co/security-labs/getting-gooey-with-guloader-downloader?ultron=esl:_threat_research%2Bvulnerability_updates&blade=twitter&hulk=social&utm_content=120128088158