Sep 11 2023

NETMANAGEIT

Intelligence Report

The Case of LummaC2 v4.0

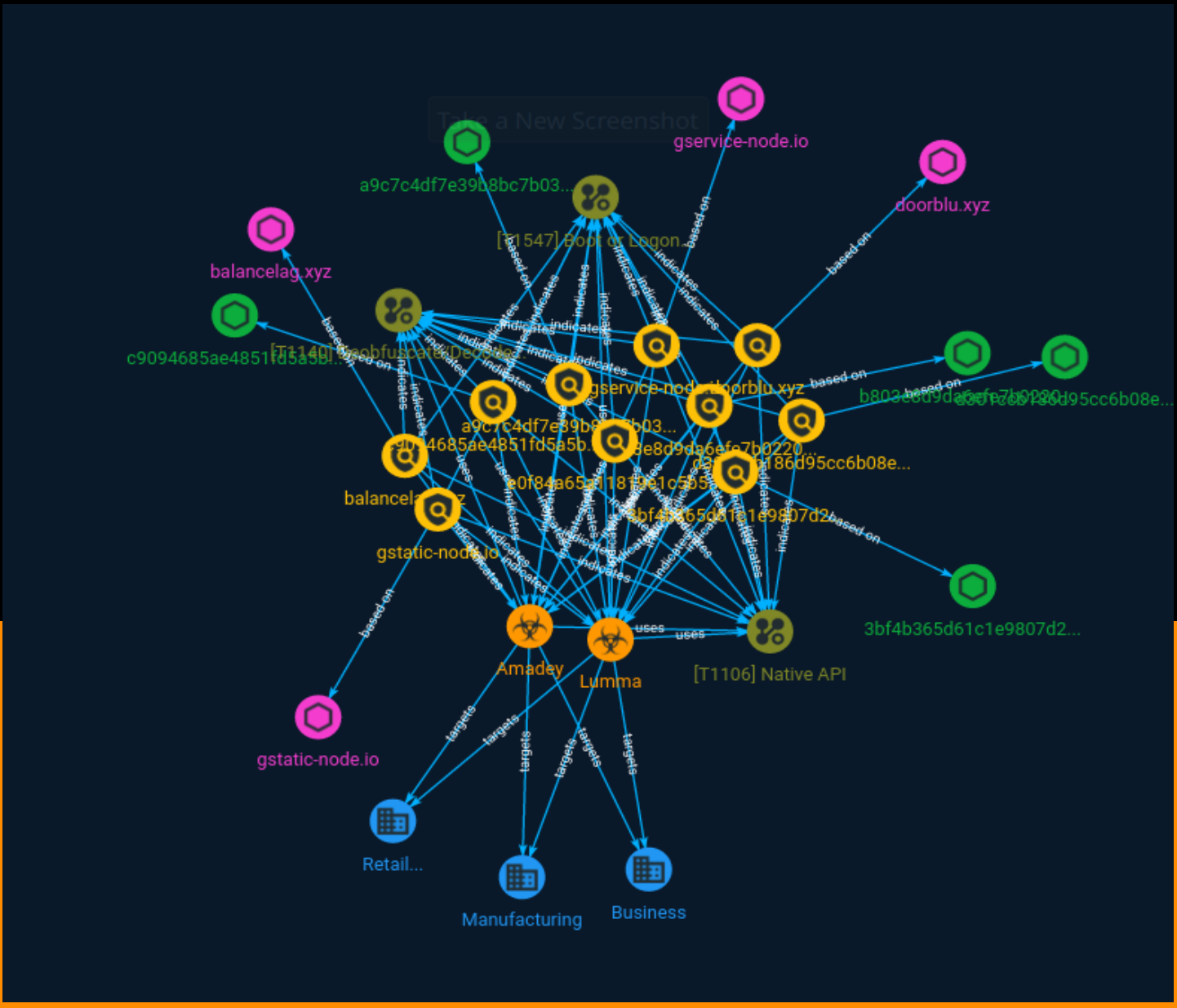

Table of contents

Overview

- Description 4
- Confidence 4

Entities

- Sector 5
- Indicator 6
- Malware 11
- Attack-Pattern 12

Observables

- Domain-Name 15
- StixFile 16

External References

- External References 17

Overview

Description

Since the beginning of August 2023, the eSentire Threat Response Unit (TRU) has observed 5 cases of Lumma Stealer infections across manufacturing, retail, and business industries.

Confidence

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

15 / 100

Sector

Name
Business
Name
Retail (distribution)
Description
Distribution and sale of goods directly to the consumer.
Name
Manufacturing
Description
Private entities transforming and selling goods, products and equipment which are not included in other activity sectors.

Indicator

Name
a9c7c4df7e39b8bc7b03b7d4053066d26154bc76c3e261fff84f2339d44a474e
Description
InnoSetupInstaller
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a9c7c4df7e39b8bc7b03b7d4053066d26154bc76c3e261fff84f2339d44a474e']
Name
balancelag.xyz
Pattern Type
stix
Pattern

[domain-name:value = 'balancelag.xyz']

Name

b803e8d9da6efe7b0220f654c7afb784d21dc222afd4a4e41396e74f861cbf30

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'b803e8d9da6efe7b0220f654c7afb784d21dc222afd4a4e41396e74f861cbf30']

Name

d301ccb186d95cc6b08e1c6287c9884edbe2c06625e119cd6305604f03fe9871

Description

Delphi SHA256 of a4e04604c178fe4fd19996b4d48b8885

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd301ccb186d95cc6b08e1c6287c9884edbe2c06625e119cd6305604f03fe9871']

Name

gservice-node.io

Pattern Type

stix

Pattern

[domain-name:value = 'gservice-node.io']

Name

e0f84a65a11819e1c5b5fcacc9cffc11adbefa91

Description

LummaC2 Detection

Pattern Type

yara

Pattern

rule LummaC2 { meta: author = "RussianPanda" description = "LummaC2 Detection" strings: $p1="lid=%s&j=%s&ver" $p2= {89 ca 83 e2 03 8a 54 14 08 32 54 0d 04} condition: all of them and filesize <= 500KB }

Name

3bf4b365d61c1e9807d20e71375627450b8fea1635cb6ddb85f2956e8f6b3ec3

Description

Zeppelin_10

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '3bf4b365d61c1e9807d20e71375627450b8fea1635cb6ddb85f2956e8f6b3ec3']

Name

c9094685ae4851fd5a5b886b73c7b07efd9b47ea0bdae3f823d035cf1b3b9e48

Description

Zeppelin_10

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'c9094685ae4851fd5a5b886b73c7b07efd9b47ea0bdae3f823d035cf1b3b9e48']

Name

doorblu.xyz

Pattern Type

stix

Pattern

[domain-name:value = 'doorblu.xyz']

Name
gstatic-node.io
Description
Win32/Lumma
Pattern Type
stix
Pattern
[domain-name:value = 'gstatic-node.io']

Malware

Name
Lumma
Name
Amadey
Description
[Amadey](https://attack.mitre.org/software/S1025) is a Trojan bot that has been used since at least October 2018.(Citation: Korean FSI TA505 2020)(Citation: BlackBerry Amadey 2020)

Attack-Pattern

Name
Boot or Logon Autostart Execution
ID
T1547
Description
Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon.(Citation: Microsoft Run Key)(Citation: MSDN Authentication Packages)(Citation: Microsoft TimeProvider)(Citation: Cylance Reg Persistence Sept 2013)(Citation: Linux Kernel Programming) These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel. Since some boot or logon autostart programs run with higher privileges, an adversary may leverage these to elevate privileges.
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. 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 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. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001)).

Name

Deobfuscate/Decode Files or Information

ID

T1140

Description

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they

intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [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)

Domain-Name

Value
balancelag.xyz
gservice-node.io
doorblu.xyz
gstatic-node.io

StixFile

Value
d301ccb186d95cc6b08e1c6287c9884edbe2c06625e119cd6305604f03fe9871
b803e8d9da6efe7b0220f654c7afb784d21dc222afd4a4e41396e74f861cbf30
a9c7c4df7e39b8bc7b03b7d4053066d26154bc76c3e261fff84f2339d44a474e
3bf4b365d61c1e9807d20e71375627450b8fea1635cb6ddb85f2956e8f6b3ec3
c9094685ae4851fd5a5b886b73c7b07efd9b47ea0bdae3f823d035cf1b3b9e48

External References

- https://otx.alienvault.com/pulse/64ff2b229877dd74bb34cb82
- https://www.esentire.com/blog/the-case-of-lummac2-v4-0