Mar 27 2024

NETMANAGEIT

Intelligence Report

Agent Tesla's New Ride: The Rise of a Novel Loader

ab9cd59d789e6c7841b9...
3a1fe17d53a198f64051...
frevillon.acsitec@pr...
[T1140] T1140
[T1027] T1027
Agent Tesla - S0331
[T1573] T1573
https://github.com/T...
agent tesla
http://artemis-rat.c...
[T1566] T1566
[T1003] T1003
[T1071] T1071
[T1134] T1134
f74e1a37a218dc6fcfab...
[T1055] T1055
e3cb3a5608f9a8ba...
merve@temikan.com.tr
[T1568] T1568
infostealer
http://artemis-rat.c...
a02388b5c352f13334f3...
a3645f81079b19ff6038...
indicates
uses
based-on

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A new and sophisticated loader has been observed delivering the Agent Tesla infostealer malware using advanced techniques like polymorphism, anti-analysis, and proxy communications to evade detection. The loader is delivered via phishing emails and executes the infostealer payload entirely in memory. Agent Tesla then captures sensitive information and exfiltrates it using compromised email accounts. This novel loader marks an evolution in the tactics used to deploy Agent Tesla and will likely facilitate the distribution of other malware families beyond just Agent Tesla.

Confidence

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

100 / 100

Content

N/A

Indicator

Name

https://github.com/TheSpeedX/PROXY-List/blob/master/http.txt

Pattern Type

stix

Pattern

[url:value = 'https://github.com/TheSpeedX/PROXY-List/blob/master/http.txt']

Name

merve@temikan.com.tr

Pattern Type

stix

Pattern

[email-addr:value = 'merve@temikan.com.tr']

Name

frevillon.acsitec@proton.me

Pattern Type

stix

Pattern

[email-addr:value = 'frevillon.acsitec@proton.me']

Name

http://artemis-rat.com/get/65f0e7dd5b705f429be16c65

Pattern Type

stix

Pattern

[url:value = 'http://artemis-rat.com/get/65f0e7dd5b705f429be16c65']

Name

http://artemis-rat.com/get/65eb0afe3a680a9851f23712

Pattern Type

stix

Pattern

[url:value = 'http://artemis-rat.com/get/65eb0afe3a680a9851f23712']

Name

f74e1a37a218dc6fcfabeb1435537f709d742505505a11e4757fc7417e5eb962

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'f74e1a37a218dc6fcfabeb1435537f709d742505505a11e4757fc7417e5eb962']

Name

e3cb3a5608f9a8baf9c1da86324474739d6c33f8369cc3bb2fd8c79e919089c4

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'e3cb3a5608f9a8baf9c1da86324474739d6c33f8369cc3bb2fd8c79e919089c4']

Name

ab9cd59d789e6c7841b9d28689743e700d492b5fae1606f184889cc7e6acadcc

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'ab9cd59d789e6c7841b9d28689743e700d492b5fae1606f184889cc7e6acadcc']

Name
a3645f81079b19ff60386cb244696ea56f5418ae556fba4fd0afe77cfcb29211
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a3645f81079b19ff60386cb244696ea56f5418ae556fba4fd0afe77cfcb29211']
Name
a02388b5c352f13334f30244e9eedac3384bc2bf475d8bc667b0ce497769cc6a
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a02388b5c352f13334f30244e9eedac3384bc2bf475d8bc667b0ce497769cc6a']
Name
3a1fe17d53a198f64051a449c388f54002e57995b529635758248dc4da7f5080
Pattern Type
stix
Pattern

```
[file:hashes.'SHA-256' =
'3a1fe17d53a198f64051a449c388f54002e57995b529635758248dc4da7f5080']
```

Malware

Name
Agent Tesla - S0331

Name
infostealer

Name
agent tesla

Description
[Agent Tesla](https://attack.mitre.org/software/S0331) is a spyware Trojan written for the .NET framework that has been observed since at least 2014.(Citation: Fortinet Agent Tesla April 2018)(Citation: Bitdefender Agent Tesla April 2020)(Citation: Malwarebytes Agent Tesla April 2020)

Attack-Pattern

Name
T1134

ID
T1134

Description
Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. These token can then be applied to an existing process (i.e. [Token Impersonation/Theft](https://attack.mitre.org/techniques/T1134/001)) or used to spawn a new process (i.e. [Create Process with Token](https://attack.mitre.org/techniques/T1134/002)). An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can then use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.(Citation: Pentestlab Token Manipulation) Any standard user can use the \`runas\` command, and the Windows API functions, to create impersonation tokens; it does not require access to an administrator account. There are also other mechanisms, such as Active Directory fields, that can be used to modify access tokens.

Name
T1573
ID
T1573
Description
Adversaries may employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if secret keys are encoded and/or generated within malware samples/configuration files.
Name
T1568
ID
T1568
Description
Adversaries may dynamically establish connections to command and control infrastructure to evade common detections and remediations. This may be achieved by using malware that shares a common algorithm with the infrastructure the adversary uses to receive the malware's communications. These calculations can be used to dynamically adjust parameters such as the domain name, IP address, or port number the malware uses for command and control. Adversaries may use dynamic resolution for the purpose of [Fallback Channels](https://attack.mitre.org/techniques/T1008). When contact is lost with the primary command and control server malware may employ dynamic resolution as a means to reestablishing command and control.(Citation: Talos CCleanup 2017)(Citation: FireEye POSHSPY April 2017)(Citation: ESET Sednit 2017 Activity)

Name

T1027

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

T1566

ID

T1566

Description

Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a trusted source, as well as evasive techniques such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages (e.g., [Email Hiding Rules](https://attack.mitre.org/techniques/T1564/008)).(Citation: Microsoft OAuth Spam 2022)(Citation: Palo Alto Unit 42 VBA Infostealer 2014) Another way to accomplish this is by forging or spoofing(Citation: Proofpoint-spoof) the identity of the sender which can be used to fool both the human recipient as well as automated security tools.(Citation: cyberproof-double-bounce) Victims may also receive phishing messages that instruct them to call a phone number where they are directed to visit a malicious URL, download malware,(Citation: sygnia Luna Month)(Citation: CISA Remote Monitoring and Management Software) or install adversary-accessible remote management tools onto their computer (i.e., [User Execution](https://attack.mitre.org/techniques/T1204)).(Citation: Unit42 Luna Moth)

Name

T1055

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

T1140

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)

Name

T1071

ID

T1071

Description

Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

Name

T1003

ID

T1003

Description

Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform [Lateral Movement](https://attack.mitre.org/tactics/TA0008) and access restricted information. Several of the tools mentioned in associated sub-techniques may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.

Url

Value
https://github.com/TheSpeedX/PROXY-List/blob/master/http.txt
http://artemis-rat.com/get/65f0e7dd5b705f429be16c65
http://artemis-rat.com/get/65eb0afe3a680a9851f23712

Email-Addr

Value
merve@temikan.com.tr
frevillon.acsitec@proton.me

StixFile

Value
f74e1a37a218dc6fcfabeb1435537f709d742505505a11e4757fc7417e5eb962
e3cb3a5608f9a8baf9c1da86324474739d6c33f8369cc3bb2fd8c79e919089c4
ab9cd59d789e6c7841b9d28689743e700d492b5fae1606f184889cc7e6acadcc
a3645f81079b19ff60386cb244696ea56f5418ae556fba4fd0afe77cfcb29211
3a1fe17d53a198f64051a449c388f54002e57995b529635758248dc4da7f5080
a02388b5c352f13334f30244e9eedac3384bc2bf475d8bc667b0ce497769cc6a

External References

- https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/agent-teslas-new-ride-the-rise-of-a-novel-loader/
- https://otx.alienvault.com/pulse/6603ebc9f7d8b30ccfe70c54