Sep 11 2023

NETMANAGEIT

Intelligence Report

HijackLoader

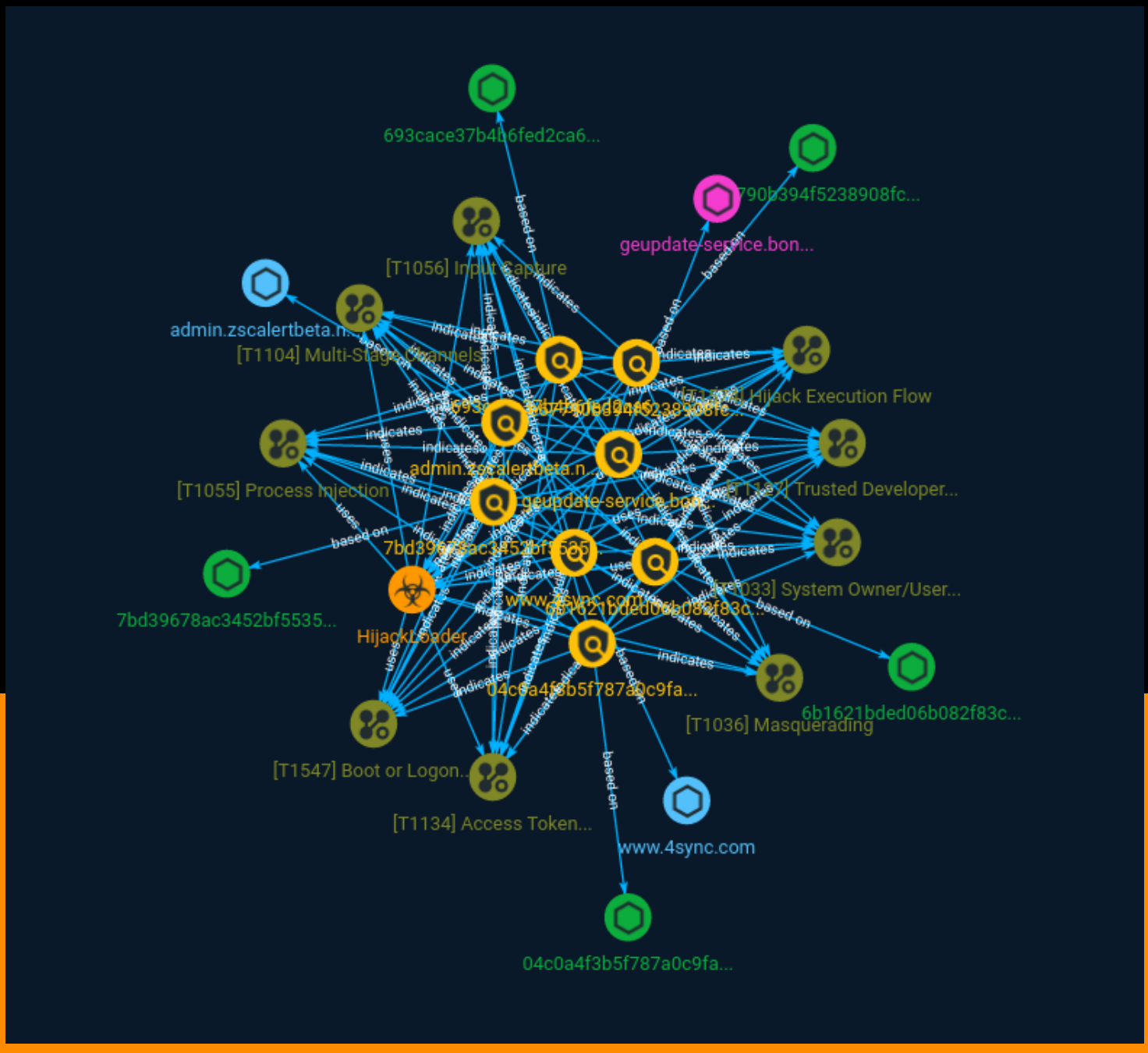

Table of contents

Overview

- Description 4
- Confidence 4

Entities

- Indicator 5
- Malware 9
- Attack-Pattern 10

Observables

- Domain-Name 16
- StixFile 17
- Hostname 18

External References

- External References 19

Overview

Description

HijackLoader is a new malware loader, which has grown in popularity over the past few months.

Confidence

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

15 / 100

Indicator

Name
geupdate-service.bond
Pattern Type
stix
Pattern
[domain-name:value = 'geupdate-service.bond']
Name
www.4sync.com
Pattern Type
stix
Pattern
[hostname:value = 'www.4sync.com']
Name
693cace37b4b6fed2ca67906c7a4b1c11273110561a207a222aa4e62fb4a184a

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '693cace37b4b6fed2ca67906c7a4b1c11273110561a207a222aa4e62fb4a184a']

Name

6b1621bded06b082f83c731319c9deb2fdf751a4cec1d1b2b00ab9e75f4c29ca

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6b1621bded06b082f83c731319c9deb2fdf751a4cec1d1b2b00ab9e75f4c29ca']

Name

7bd39678ac3452bf55359b44c5192b79412ce61a82cd72eef88f91aba5792ee6

Description

Delphi

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7bd39678ac3452bf55359b44c5192b79412ce61a82cd72eef88f91aba5792ee6']

Name

e67790b394f5238908fcc326a9db940b200d9b50cbb45f0bfa94038db50beeae

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e67790b394f5238908fcc326a9db940b200d9b50cbb45f0bfa94038db50beeae']

Name

04c0a4f3b5f787a0c9fa8f6d8ef19e01097185dd1f2ba40ae4bbbeca9c3a1c72

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '04c0a4f3b5f787a0c9fa8f6d8ef19e01097185dd1f2ba40ae4bbbeca9c3a1c72']

Name

admin.zscalertbeta.net

Pattern Type

stix
Pattern
[hostname:value = 'admin.zscalertbeta.net']

Malware

Name
HijackLoader

Attack-Pattern

Name
Trusted Developer Utilities Proxy Execution
ID
T1127
Description
Adversaries may take advantage of trusted developer utilities to proxy execution of malicious payloads. There are many utilities used for software development related tasks that can be used to execute code in various forms to assist in development, debugging, and reverse engineering.(Citation: engima0x3 DNX Bypass)(Citation: engima0x3 RCSI Bypass)(Citation: Exploit Monday WinDbg)(Citation: LOLBAS Tracker) These utilities may often be signed with legitimate certificates that allow them to execute on a system and proxy execution of malicious code through a trusted process that effectively bypasses application control solutions.
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

Input Capture

ID

T1056

Description

Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. [Credential API Hooking](https://attack.mitre.org/techniques/T1056/004)) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. [Web Portal Capture](https://attack.mitre.org/techniques/T1056/003)).

Name

Masquerading

ID

T1036

Description

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusable system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site)

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

Hijack Execution Flow

ID

T1574

Description

Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution. There are many ways an adversary may hijack the flow of execution, including by manipulating how the operating system locates programs to be executed. How the operating system locates libraries to be used by a program can also be intercepted. Locations where the operating system looks for programs/resources, such as file directories and in the case of Windows the Registry, could also be poisoned to include malicious payloads.

Name

Access Token Manipulation

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

Multi-Stage Channels

ID

T1104

Description

Adversaries may create multiple stages for command and control that are employed under different conditions or for certain functions. Use of multiple stages may obfuscate the command and control channel to make detection more difficult. Remote access tools will call back to the first-stage command and control server for instructions. The first stage may have automated capabilities to collect basic host information, update tools, and upload additional files. A second remote access tool (RAT) could be uploaded at that point to redirect the host to the second-stage command and control server. The second stage will likely be more fully featured and allow the adversary to interact with the system through a reverse shell and additional RAT features. The different stages will likely be hosted separately with no overlapping infrastructure. The loader may also have backup first-stage callbacks or [Fallback Channels](https://attack.mitre.org/techniques/T1008) in case the original first-stage communication path is discovered and blocked.

Name

System Owner/User Discovery

ID

T1033

Description

Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping](https://attack.mitre.org/techniques/T1003). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery](https://attack.mitre.org/techniques/T1033) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Various utilities and commands may acquire this information, including `whoami`. In macOS and Linux, the currently logged in user can be identified with `w` and `who`. On macOS the `dscl . list /Users | grep -v '_'` command can also be used to enumerate user accounts. Environment variables, such as `%USERNAME%` and `$USER`, may also be used to access this information. On network devices, [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `show users` and `show ssh` can be used to display users currently logged into the device.(Citation: show_ssh_users_cmd_cisco)(Citation: US-CERT TA18-106A Network Infrastructure Devices 2018)

Domain-Name

Value
geupdate-service.bond

StixFile

Value
6b1621bded06b082f83c731319c9deb2fdf751a4cec1d1b2b00ab9e75f4c29ca
7bd39678ac3452bf55359b44c5192b79412ce61a82cd72eef88f91aba5792ee6
e67790b394f5238908fcc326a9db940b200d9b50cbb45f0bfa94038db50beeae
04c0a4f3b5f787a0c9fa8f6d8ef19e01097185dd1f2ba40ae4bbbeca9c3a1c72
693cace37b4b6fed2ca67906c7a4b1c11273110561a207a222aa4e62fb4a184a

Hostname

Value
www.4sync.com
admin.zscalertbeta.net

External References

- https://otx.alienvault.com/pulse/64ff2d650b8a5d85ec0f4f63
- https://www.zscaler.com/blogs/security-research/technical-analysis-hijackloader