May 10 2024

NETMANAGEIT

Intelligence Report

macOS Cuckoo Stealer | Ensuring Detection and Defense as New Samples Rapidly Emerge

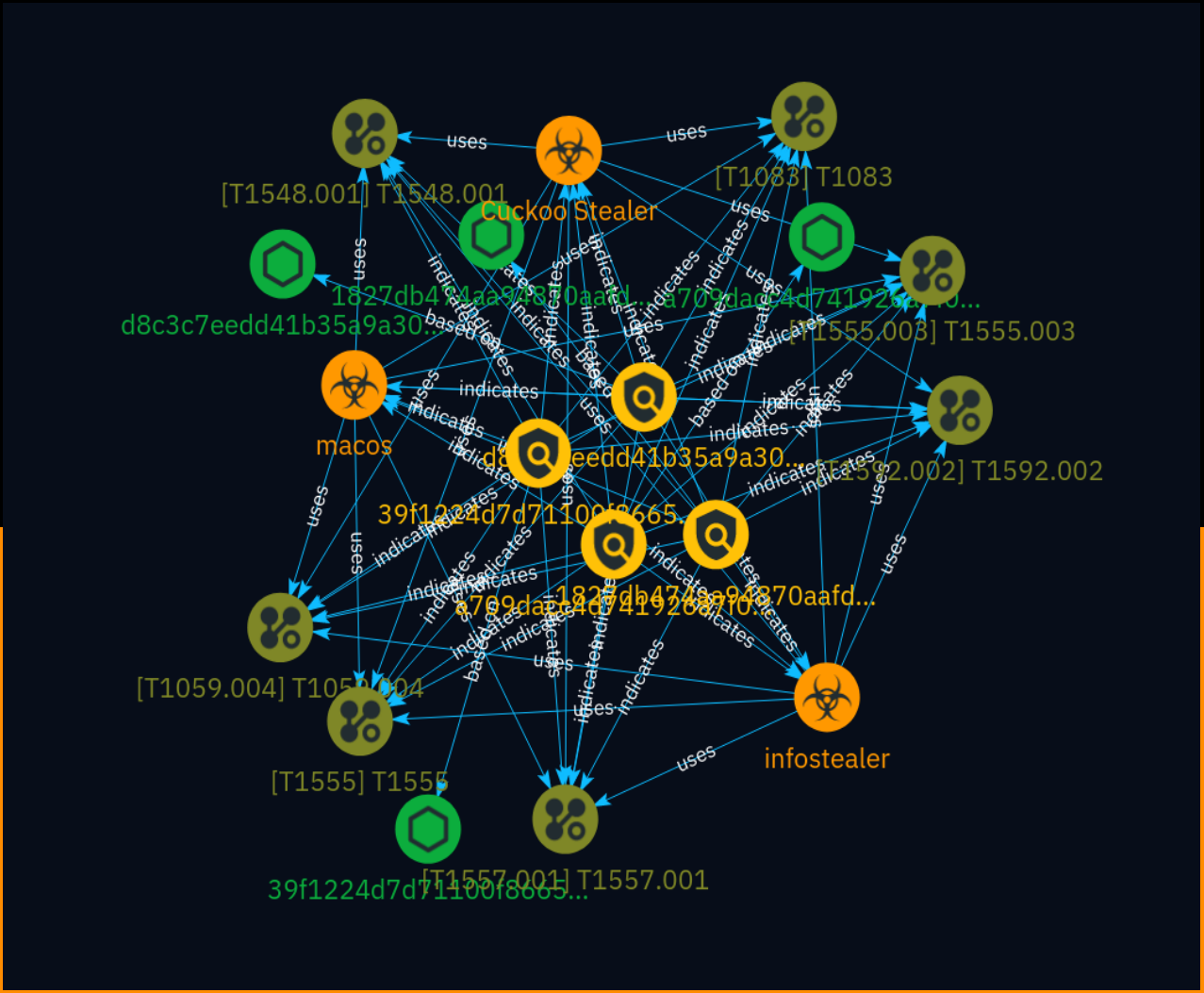

Table of contents

Overview

Entities

Observables

External References

Overview

Description

This analysis discusses the emergence of a new macOS malware family called 'Cuckoo Stealer', which acts as an infostealer and spyware. It describes Cuckoo Stealer's main features, logic, and provides indicators of compromise to assist threat hunters and defenders. The malware employs techniques like obfuscation, scraping admin passwords, and installing persistence mechanisms. Although attempts were made to conceal its behavior, analysis reveals similarities with other recent infostealers targeting macOS devices. SentinelOne's Singularity XDR platform detects and prevents the execution of Cuckoo Stealer, protecting customers from this emerging threat.

Confidence

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

100 / 100

Content

N/A

Indicator

Name
d8c3c7eedd41b35a9a30a99727b9e0b47e652b8f601b58e2c20e2a7d30ce14a8
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'd8c3c7eedd41b35a9a30a99727b9e0b47e652b8f601b58e2c20e2a7d30ce14a8']
Name
a709dacc4d741926a7f04cad40a22adfc12dd7406f016dd668dd98725686a2dc
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a709dacc4d741926a7f04cad40a22adfc12dd7406f016dd668dd98725686a2dc']
Name

39f1224d7d71100f86651012c87c181a545b0a1606edc49131730f8c5b56bdb7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '39f1224d7d71100f86651012c87c181a545b0a1606edc49131730f8c5b56bdb7']

Name

1827db474aa94870aafdd63bdc25d61799c2f405ef94e88432e8e212dfa51ac7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '1827db474aa94870aafdd63bdc25d61799c2f405ef94e88432e8e212dfa51ac7']

Malware

Name
Cuckoo Stealer

Name
infostealer

Name
macos

Attack-Pattern

Name
T1592.002
ID
T1592.002
Description
Adversaries may gather information about the victim's host software that can be used during targeting. Information about installed software may include a variety of details such as types and versions on specific hosts, as well as the presence of additional components that might be indicative of added defensive protections (ex: antivirus, SIEMs, etc.). Adversaries may gather this information in various ways, such as direct collection actions via [Active Scanning](https://attack.mitre.org/techniques/T1595) (ex: listening ports, server banners, user agent strings) or [Phishing for Information](https://attack.mitre.org/techniques/T1598). Adversaries may also compromise sites then include malicious content designed to collect host information from visitors.(Citation: ATT ScanBox) Information about the installed software may also be exposed to adversaries via online or other accessible data sets (ex: job postings, network maps, assessment reports, resumes, or purchase invoices). Gathering this information may reveal opportunities for other forms of reconnaissance (ex: [Search Open Websites/Domains](https://attack.mitre.org/techniques/T1593) or [Search Open Technical Databases](https://attack.mitre.org/techniques/T1596)), establishing operational resources (ex: [Develop Capabilities](https://attack.mitre.org/techniques/T1587) or [Obtain Capabilities](https://attack.mitre.org/techniques/T1588)), and/or for initial access (ex: [Supply Chain Compromise](https://attack.mitre.org/techniques/T1195) or [External Remote Services](https://attack.mitre.org/techniques/T1133)).

Name

T1059.004

ID

T1059.004

Description

Adversaries may abuse Unix shell commands and scripts for execution. Unix shells are the primary command prompt on Linux and macOS systems, though many variations of the Unix shell exist (e.g. sh, bash, zsh, etc.) depending on the specific OS or distribution. (Citation: DieNet Bash)(Citation: Apple ZShell) Unix shells can control every aspect of a system, with certain commands requiring elevated privileges. Unix shells also support scripts that enable sequential execution of commands as well as other typical programming operations such as conditionals and loops. Common uses of shell scripts include long or repetitive tasks, or the need to run the same set of commands on multiple systems. Adversaries may abuse Unix shells to execute various commands or payloads. Interactive shells may be accessed through command and control channels or during lateral movement such as with [SSH](https://attack.mitre.org/techniques/T1021/004). Adversaries may also leverage shell scripts to deliver and execute multiple commands on victims or as part of payloads used for persistence.

Name

T1083

ID

T1083

Description

Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the

adversary fully infects the target and/or attempts specific actions. Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the [Native API](https://attack.mitre.org/techniques/T1106). Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather file and directory information (e.g. `dir`, `show flash`, and/or `nvram`).(Citation: US-CERT-TA18-106A) Some files and directories may require elevated or specific user permissions to access.

Name

T1557.001

ID

T1557.001

Description

By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary controlled system. This activity may be used to collect or relay authentication materials. Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) are Microsoft Windows components that serve as alternate methods of host identification. LLMNR is based upon the Domain Name System (DNS) format and allows hosts on the same local link to perform name resolution for other hosts. NBT-NS identifies systems on a local network by their NetBIOS name. (Citation: Wikipedia LLMNR)(Citation: TechNet NetBIOS) Adversaries can spoof an authoritative source for name resolution on a victim network by responding to LLMNR (UDP 5355)/NBT-NS (UDP 137) traffic as if they know the identity of the requested host, effectively poisoning the service so that the victims will communicate with the adversary controlled system. If the requested host belongs to a resource that requires identification/authentication, the username and NTLMv2 hash will then be sent to the adversary controlled system. The adversary can then collect the hash information sent over the wire through tools that monitor the ports for traffic or through [Network Sniffing](https://attack.mitre.org/techniques/T1040) and crack the hashes offline through [Brute Force](https://attack.mitre.org/techniques/T1110) to obtain the plaintext passwords. In some cases where an adversary has access to a system that is in the authentication path between systems or when automated scans that use credentials attempt to authenticate to an adversary controlled system, the NTLMv1/v2 hashes can be intercepted and relayed to access and execute code against a target system. The relay step can happen in conjunction with poisoning but may also be independent of it.(Citation: byt3bl33d3r NTLM

Relaying)(Citation: Secure Ideas SMB Relay) Additionally, adversaries may encapsulate the NTLMv1/v2 hashes into various protocols, such as LDAP, SMB, MSSQL and HTTP, to expand and use multiple services with the valid NTLM response. Several tools may be used to poison name services within local networks such as NBNSpoof, Metasploit, and [Responder](https://attack.mitre.org/software/S0174).(Citation: GitHub NBNSpoof)(Citation: Rapid7 LLMNR Spoofer)(Citation: GitHub Responder)

Name

T1548.001

ID

T1548.001

Description

An adversary may abuse configurations where an application has the setuid or setgid bits set in order to get code running in a different (and possibly more privileged) user's context. On Linux or macOS, when the setuid or setgid bits are set for an application binary, the application will run with the privileges of the owning user or group respectively.(Citation: setuid man page) Normally an application is run in the current user's context, regardless of which user or group owns the application. However, there are instances where programs need to be executed in an elevated context to function properly, but the user running them may not have the specific required privileges. Instead of creating an entry in the sudoers file, which must be done by root, any user can specify the setuid or setgid flag to be set for their own applications (i.e. [Linux and Mac File and Directory Permissions Modification](https://attack.mitre.org/techniques/T1222/002)). The `chmod` command can set these bits with bitmasking, `chmod 4777 [file]` or via shorthand naming, `chmod u+s [file]`. This will enable the setuid bit. To enable the setgid bit, `chmod 2775` and `chmod g+s` can be used. Adversaries can use this mechanism on their own malware to make sure they're able to execute in elevated contexts in the future.(Citation: OSX Keydnap malware) This abuse is often part of a "shell escape" or other actions to bypass an execution environment with restricted permissions. Alternatively, adversaries may choose to find and target vulnerable binaries with the setuid or setgid bits already enabled (i.e. [File and Directory Discovery](https://attack.mitre.org/techniques/T1083)). The setuid and setguid bits are indicated with an "s" instead of an "x" when viewing a file's attributes via `ls -l`. The `find` command can also be used to search for such files. For example, `find / -perm +4000 2>/dev/null` can be used to find files with setuid set and

`find / -perm +2000 2>/dev/null` may be used for setgid. Binaries that have these bits set may then be abused by adversaries.(Citation: GTFOBins Suid)

Name

T1555.003

ID

T1555.003

Description

Adversaries may acquire credentials from web browsers by reading files specific to the target browser.(Citation: Talos Olympic Destroyer 2018) Web browsers commonly save credentials such as website usernames and passwords so that they do not need to be entered manually in the future. Web browsers typically store the credentials in an encrypted format within a credential store; however, methods exist to extract plaintext credentials from web browsers. For example, on Windows systems, encrypted credentials may be obtained from Google Chrome by reading a database file, `AppData\Local\Google\Chrome\User Data\Default\Login Data` and executing a SQL query: `SELECT action_url, username_value, password_value FROM logins;`. The plaintext password can then be obtained by passing the encrypted credentials to the Windows API function `CryptUnprotectData`, which uses the victim's cached logon credentials as the decryption key.(Citation: Microsoft CryptUnprotectData April 2018) Adversaries have executed similar procedures for common web browsers such as FireFox, Safari, Edge, etc. (Citation: Proofpoint Vega Credential Stealer May 2018)(Citation: FireEye HawkEye Malware July 2017) Windows stores Internet Explorer and Microsoft Edge credentials in Credential Lockers managed by the [Windows Credential Manager](https://attack.mitre.org/techniques/T1555/004). Adversaries may also acquire credentials by searching web browser process memory for patterns that commonly match credentials.(Citation: GitHub Mimikittenz July 2016) After acquiring credentials from web browsers, adversaries may attempt to recycle the credentials across different systems and/or accounts in order to expand access. This can result in significantly furthering an adversary's objective in cases where credentials gained from web browsers overlap with privileged accounts (e.g. domain administrator).

Name

T1555

ID

T1555

Description

Adversaries may search for common password storage locations to obtain user credentials.(Citation: F-Secure The Dukes) Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications and services that store passwords to make them easier for users to manage and maintain, such as password managers and cloud secrets vaults. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

StixFile

Value
d8c3c7eedd41b35a9a30a99727b9e0b47e652b8f601b58e2c20e2a7d30ce14a8
a709dacc4d741926a7f04cad40a22adfc12dd7406f016dd668dd98725686a2dc
39f1224d7d71100f86651012c87c181a545b0a1606edc49131730f8c5b56bdb7
1827db474aa94870aafdd63bdc25d61799c2f405ef94e88432e8e212dfa51ac7

External References

- https://www.sentinelone.com/blog/macos-cuckoo-stealer-ensuring-detection-and-defense-as-new-samples-rapidly-emerge/
- https://otx.alienvault.com/pulse/663ddb61f90a0b40f00447de