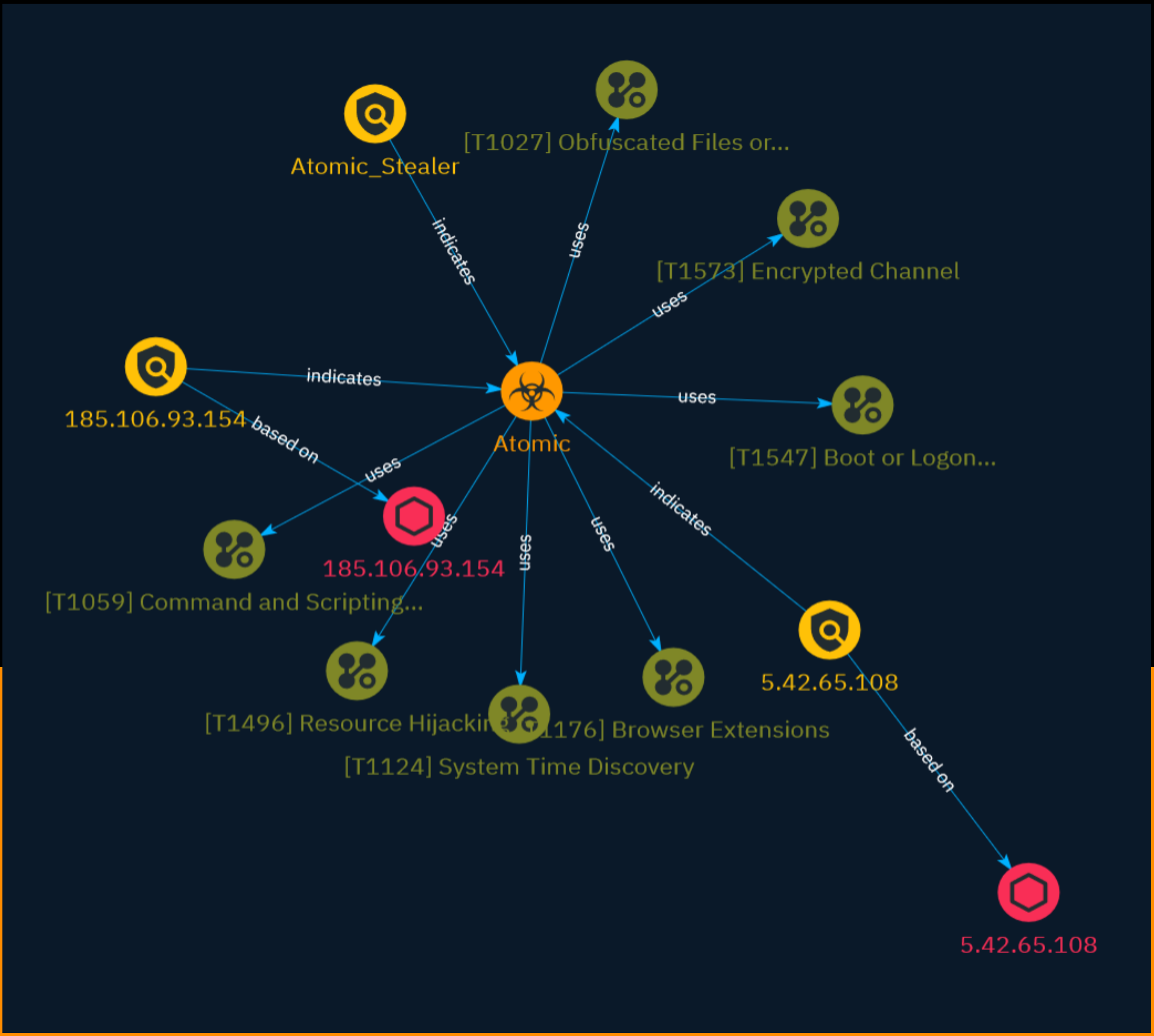
Jan 19 2024

NETMANAGEIT

# Intelligence Report

# From Russia With Code: Disarming Atomic Stealer


Atomic_Stealer
[T1027] Obfuscated Files or...
indicates
uses
[T1573] Encrypted Channel
uses
185.106.93.154
indicates
Atomic
uses
[T1547] Boot or Logon...
based on
uses
185.106.93.154
uses
uses
uses
indicates
[T1059] Command and Scripting...
5.42.65.108
[T1496] Resource Hijacking
[T1176] Browser Extensions
[T1124] System Time Discovery
based on
5.42.65.108

# Table of contents

## Overview

## Entities

## Observables

## External References

# Overview

## Description

The latest version of Atomic Stealer is circulating on the Internet and has been described as “one of the most sophisticated tools in the world” by Jérôme Segura, a security researcher. For 3000$ per month, the user gets the access to the panel. The user provides Telegram Bot ID and build ID to the seller and the user receives the build. The stealer allegedly has the following functionalities and features: Login Keychain dump Extract system information FileGrabber (from Desktop, Documents) MacOS Password retrieval Convenient web panel MetaMask brute-forcer Crypto-checker (tool to check the information on crypto assets) Telegram logs

## Confidence

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

15 / 100

# Content

N/A

# 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** |
| System Time Discovery |
| **ID** |
| T1124 |

### Description

An adversary may gather the system time and/or time zone from a local or remote system. The system time is set and stored by the Windows Time Service within a domain to maintain time synchronization between systems and services in an enterprise network. (Citation: MSDN System Time)(Citation: Technet Windows Time Service) System time information may be gathered in a number of ways, such as with [Net](https://attack.mitre.org/software/S0039) on Windows by performing `net time \\hostname` to gather the system time on a remote system. The victim's time zone may also be inferred from the current system time or gathered by using `w32tm /tz`.(Citation: Technet Windows Time Service) On network devices, [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `show clock detail` can be used to see the current time configuration.(Citation: show_clock_detail_cisco_cmd) This information could be useful for performing other techniques, such as executing a file with a [Scheduled Task/Job](https://attack.mitre.org/techniques/T1053)(Citation: RSA EU12 They're Inside), or to discover locality information based on time zone to assist in victim targeting (i.e. [System Location Discovery](https://attack.mitre.org/techniques/T1614)). Adversaries may also use knowledge of system time as part of a time bomb, or delaying execution until a specified date/time.(Citation: AnyRun TimeBomb)

### Name

Encrypted Channel

### 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

Browser Extensions

**ID**

T1176

**Description**

Adversaries may abuse Internet browser extensions to establish persistent access to victim systems. Browser extensions or plugins are small programs that can add functionality and customize aspects of Internet browsers. They can be installed directly or through a browser's app store and generally have access and permissions to everything that the browser can access.(Citation: Wikipedia Browser Extension)(Citation: Chrome Extensions Definition) Malicious extensions can be installed into a browser through malicious app store downloads masquerading as legitimate extensions, through social engineering, or by an adversary that has already compromised a system. Security can be limited on browser app stores so it may not be difficult for malicious extensions to defeat automated scanners.(Citation: Malicious Chrome Extension Numbers) Depending on the browser, adversaries may also manipulate an extension's update url to install updates from an adversary controlled server or manipulate the mobile configuration file to silently install additional extensions. Previous to macOS 11, adversaries could silently install browser extensions via the command line using the `profiles` tool to install malicious `.mobileconfig` files. In macOS 11+, the use of the `profiles` tool can no longer install configuration profiles, however `.mobileconfig` files can be planted and installed with user interaction.(Citation: xorrior chrome extensions macOS) Once the extension is installed, it can browse to websites in the background, steal all information that a user enters into a browser (including credentials), and be used as an installer for a RAT for persistence. (Citation: Chrome Extension Crypto Miner)(Citation: ICEBRG Chrome Extensions)(Citation: Banker Google Chrome Extension Steals Creds)(Citation: Catch All Chrome Extension) There have also been instances of botnets using a persistent backdoor through malicious Chrome extensions.(Citation: Stantinko Botnet) There have also been similar examples of extensions being used for command & control.(Citation: Chrome Extension C2 Malware)

**Name**

Resource Hijacking

**ID**

T1496

## Description

Adversaries may leverage the resources of co-opted systems to complete resource-intensive tasks, which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive.(Citation: Kaspersky Lazarus Under The Hood Blog 2017) Servers and cloud-based systems are common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining.(Citation: CloudSploit - Unused AWS Regions) Containerized environments may also be targeted due to the ease of deployment via exposed APIs and the potential for scaling mining activities by deploying or compromising multiple containers within an environment or cluster.(Citation: Unit 42 Hildegard Malware)(Citation: Trend Micro Exposed Docker APIs) Additionally, some cryptocurrency mining malware identify then kill off processes for competing malware to ensure it’s not competing for resources.(Citation: Trend Micro War of Crypto Miners) Adversaries may also use malware that leverages a system's network bandwidth as part of a botnet in order to facilitate [Network Denial of Service](https://attack.mitre.org/techniques/T1498) campaigns and/or to seed malicious torrents.(Citation: GoBotKR) Alternatively, they may engage in proxyjacking by selling use of the victims' network bandwidth and IP address to proxyware services.(Citation: Sysdig Proxyjacking)

## 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**

Command and Scripting Interpreter

**ID**

T1059

**Description**

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python](https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

# Indicator

**Name**

Atomic_Stealer

**Description**

Atomic_Stealer Detects Atomic Stealer targering MacOS

**Pattern Type**

yara

**Pattern**

rule Atomic_Stealer { meta: author = "RussianPanda" description = "Detects Atomic Stealer targering MacOS" date = "1/13/2024" reference1 = "https://www.malwarebytes.com/blog/threat-intelligence/2024/01/atomic-stealer-rings-in-the-new-year-with-updated-version/amp" reference2 = "https://www.bleepingcomputer.com/news/security/macos-info-stealers-quickly-evolve-to-evade-xprotect-detection/" hash = "dd8aa38c7f06cb1c12a4d2c0927b6107" strings: $s1 = {8B 09 83 C1 (01|02|04|05|03) 39 C8 0F 85 38 00 00 00 48 8B 85} $s2 = {C7 40 04 00 00 00 00 C6 40 08 00 C6 40 09 00} $t1 = {80 75 D?} $t2 = {0F 57 05 ?? 1B 01 00} $t3 = {8A 06 34 DE 88 07 8A 46 01 34 DF 88 47 01} $c1 = {28 ?? 40 39} $c2 = {64 65 73 6B 77 61 6C 6C 65 74 73} condition: ( uint32(0) == 0xfeedface or // Mach-O MH_MAGIC uint32(0) == 0xcefaedfe or // Mach-O MH_CIGAM uint32(0) == 0xfeedfacf or // Mach-O MH_MAGIC_64 uint32(0) == 0xcffaedfe or // Mach-O MH_CIGAM_64 uint32(0) == 0xcafebabe or // Mach-O FAT_MAGIC uint32(0) == 0xbebafeca // Mach-O FAT_CIGAM ) and all of ($s*) and #s1 > 60 and #s2 > 100 or (all of ($t*) and #t1 > 10 and #t2 > 5) or (#c1 > 200 and $c2) }

## Name

5.42.65.108

## Description

**ISP:** Partner LLC **OS:** None ------------------------- Hostnames: ------------------------- Domains: ------------------------- Services: **22:** ``` SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.4 Key type: ecdsa-sha2-nistp256 Key: AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDPyvPV/K0mUFW5fXUIgMmor K+ki8kXGPXfsvV40AWtJ5d6O6AipNOXB/H4IcYO/Ez7bUmXrKCsV6WY+kGFvmKM= Fingerprint: d8:45:11:47:8b:b2:ef:f3:ad:65:6d:92:8f:96:89:6f Kex Algorithms: curve25519-sha256 curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521 sntrup761x25519-sha512@openssh.com diffie-hellman-group-exchange-sha256 diffie-hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256 Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519 Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression Algorithms: none zlib@openssh.com ``` ------------------ **80:** ``` HTTP/1.1 404 Not Found Content-Type: text/plain; charset=utf-8 X-Content-Type-Options: nosniff Date: Mon, 08 Jan 2024 12:45:05 GMT Content-Length: 19 ``` ------------------

## Pattern Type

stix

## Pattern

[ipv4-addr:value = '5.42.65.108']

## Name

185.106.93.154

| Description |
|---|
| CC=TR ASN=AS210644 Aeza International Ltd |

| Pattern Type |
|---|
| stix |

| Pattern |
|---|
| [ipv4-addr:value = '185.106.93.154'] |

# Malware

| Name |
|---|
| Atomic |

# IPv4-Addr

| Value |
|---|
| 185.106.93.154 |
| 5.42.65.108 |

# External References

- https://russianpanda.com/2024/01/15/Atomic-Stealer-AMOS/
- https://github.com/RussianPanda95/Yara-Rules/blob/main/AtomicStealer/Atomic_Stealer.yar
- https://otx.alienvault.com/pulse/65a90c366fd0b2428c0905ec