Nov 27 2023

NETMANAGEIT

Intelligence Report

Israel-Hamas War Spotlight: Shaking the Rust Off SysJoker

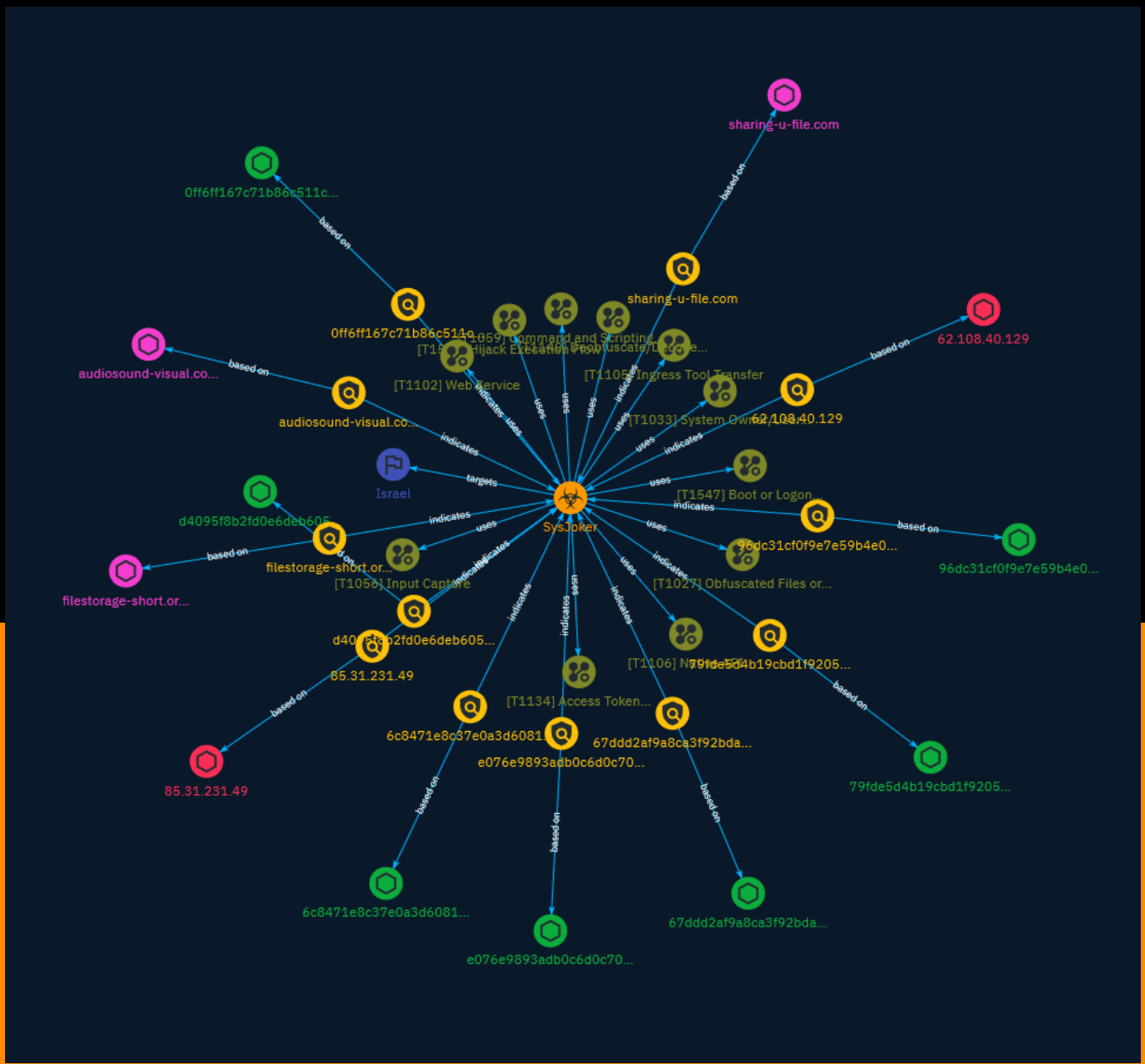

Table of contents

Overview

- Description 4
- Confidence 4
- Content 5

Entities

- Attack-Pattern 6
- Indicator 14
- Country 19
- Malware 20

Observables

- Domain-Name 21
- StixFile 22
- IPv4-Addr 23

External References

- External References 24

Overview

Description

A new variant of the SysJoker Windows malware, developed in the Rust language, has been linked to targeted attacks against Israel by a Hamas-affiliated threat actor.

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
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

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. 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. 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 use assembly to directly or indirectly invoke syscalls in an attempt to subvert defensive sensors and detection signatures such as user mode API-hooks.(Citation: Redops Syscalls) Adversaries may also attempt to tamper with sensors and defensive tools associated with API monitoring, such as unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001).

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

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

Ingress Tool Transfer

ID

T1105

Description

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)). On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil]

(https://attack.mitre.org/software/S0160), and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as `IEX(New-Object Net.WebClient).downloadString()` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`. (Citation: t1105_lolbas) Adversaries may also abuse installers and package managers, such as `yum` or `winget`, to download tools to victim hosts. Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt Dec 2016) In some cases, adversaries may be able to leverage services that sync between a web-based and an on-premises client, such as Dropbox or OneDrive, to transfer files onto victim systems. For example, by compromising a cloud account and logging into the service's web portal, an adversary may be able to trigger an automatic syncing process that transfers the file onto the victim's machine.(Citation: Dropbox Malware Sync)

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

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)

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)

Name

Web Service

ID

T1102

Description

Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites and social media acting as a mechanism for C2 may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to a compromise. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. Web service providers commonly use SSL/TLS

encryption, giving adversaries an added level of protection. Use of Web services may also protect back-end C2 infrastructure from discovery through malware binary analysis while also enabling operational resiliency (since this infrastructure may be dynamically changed).

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)

Indicator

Name
62.108.40.129
Description
CC=DE ASN=AS30962 comtrance service GmbH
Pattern Type
stix
Pattern
[ipv4-addr:value = '62.108.40.129']
Name
85.31.231.49
Description
CC=DE ASN=AS47583 Hostinger International Limited
Pattern Type
stix

Pattern

[ipv4-addr:value = '85.31.231.49']

Name

0ff6ff167c71b86c511c36cba8f75d1d5209710907a807667f97ce323df9c4ba

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0ff6ff167c71b86c511c36cba8f75d1d5209710907a807667f97ce323df9c4ba']

Name

79fde5d4b19cbd1f920535215c558b6ff63973b7af7d6bd488e256821711e0b1

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '79fde5d4b19cbd1f920535215c558b6ff63973b7af7d6bd488e256821711e0b1']

Name

6c8471e8c37e0a3d608184147f89d81d62f9442541a04d15d9ead0b3e0862d95

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6c8471e8c37e0a3d608184147f89d81d62f9442541a04d15d9ead0b3e0862d95']

Name

sharing-u-file.com

Pattern Type

stix

Pattern

[domain-name:value = 'sharing-u-file.com']

Name

e076e9893adb0c6d0c70cd7019a266d5fd02b429c01cfe51329b2318e9239836

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e076e9893adb0c6d0c70cd7019a266d5fd02b429c01cfe51329b2318e9239836']

Name

audiosound-visual.com

Pattern Type

stix

Pattern

[domain-name:value = 'audiosound-visual.com']

Name

96dc31cf0f9e7e59b4e00627f9c7f7a8cac3b8f4338b27d713b0aaf6abacfe6f

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '96dc31cf0f9e7e59b4e00627f9c7f7a8cac3b8f4338b27d713b0aaf6abacfe6f']

Name

d4095f8b2fd0e6deb605baa1530c32336298afd026afc0f41030fa43371e3e72

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd4095f8b2fd0e6deb605baa1530c32336298afd026afc0f41030fa43371e3e72']

Name

filestorage-short.org

Pattern Type

stix

Pattern

[domain-name:value = 'filestorage-short.org']

Name

67ddd2af9a8ca3f92bda17bd990e0f3c4ab1d9bea47333fe31205eede8ecc706

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '67ddd2af9a8ca3f92bda17bd990e0f3c4ab1d9bea47333fe31205eede8ecc706']

Country

Name
Israel

Malware

Name
SysJoker

Domain-Name

Value
audiosound-visual.com
sharing-u-file.com
filestorage-short.org

StixFile

Value
79fde5d4b19cbd1f920535215c558b6ff63973b7af7d6bd488e256821711e0b1
0ff6ff167c71b86c511c36cba8f75d1d5209710907a807667f97ce323df9c4ba
96dc31cf0f9e7e59b4e00627f9c7f7a8cac3b8f4338b27d713b0aaf6abacfe6f
67ddd2af9a8ca3f92bda17bd990e0f3c4ab1d9bea47333fe31205eede8ecc706
d4095f8b2fd0e6deb605baa1530c32336298afd026afc0f41030fa43371e3e72
6c8471e8c37e0a3d608184147f89d81d62f9442541a04d15d9ead0b3e0862d95
e076e9893adb0c6d0c70cd7019a266d5fd02b429c01cfe51329b2318e9239836

IPv4-Addr

Value
85.31.231.49
62.108.40.129

External References

- https://otx.alienvault.com/pulse/6564bb8418af8424b8befa1b
- https://research.checkpoint.com/2023/israel-hamas-war-spotlight-shaking-the-rust-off-sysjoker/
- https://intezer.com/blog/research/wildcard-evolution-of-sysjoker-cyber-threat/