Sep 20 2023

NETMANAGEIT

Intelligence Report

New ShroudedSnooper actor targets telecommunications firms in the Middle East with novel Implants

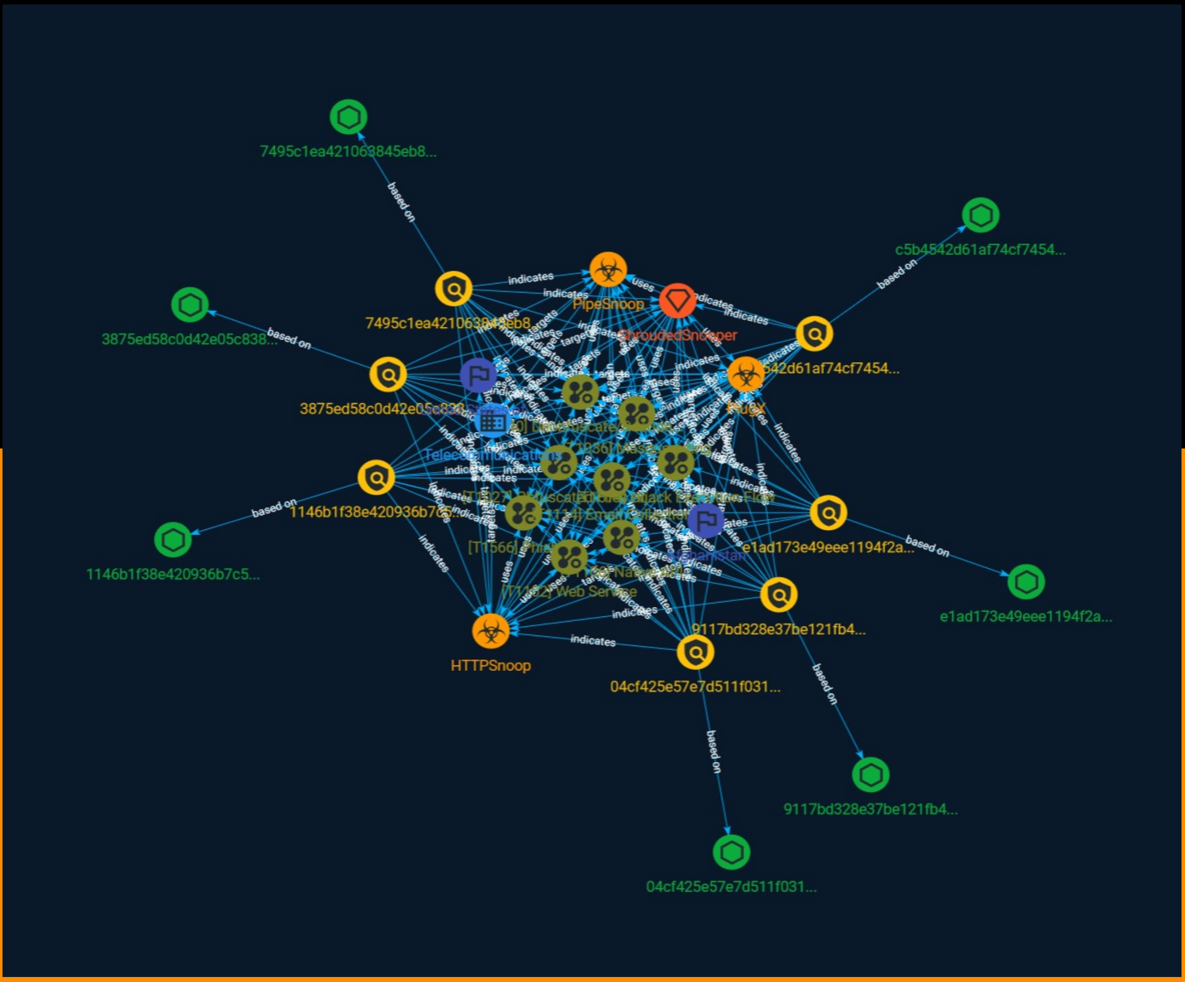

Table of contents

Overview

Entities

Observables

External References

Overview

Description

Researchers recently discovered a new malware family we're calling "HTTPSnoop" being deployed against telecommunications providers in the Middle East. HTTPSnoop is a simple, yet effective, backdoor that consists of novel techniques to interface with Windows HTTP kernel drivers and devices to listen to incoming requests for specific HTTP(S) URLs and execute that content on the infected endpoint.

Confidence

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

15 / 100

Attack-Pattern

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
Phishing
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

Email Collection

ID

T1114

Description

Adversaries may target user email to collect sensitive information. Emails may contain sensitive data, including trade secrets or personal information, that can prove valuable to adversaries. Adversaries can collect or forward email from mail servers or clients.

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. 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 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. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: 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

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)

Sector

Name
Telecommunications
Description
Private and public entities involved in the production, transport and dissemination of information and communication signals.

Indicator

Name

7495c1ea421063845eb8f4599a1c17c105f700ca0671ca874c5aa5aef3764c1c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'7495c1ea421063845eb8f4599a1c17c105f700ca0671ca874c5aa5aef3764c1c']

Name

9117bd328e37be121fb497596a2d0619a0eaca44752a1854523b8af46a5b0ceb

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'9117bd328e37be121fb497596a2d0619a0eaca44752a1854523b8af46a5b0ceb']

Name

1146b1f38e420936b7c5f6b22212f3aa93515f3738c861f499ed1047865549cb

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'1146b1f38e420936b7c5f6b22212f3aa93515f3738c861f499ed1047865549cb']

Name

c5b4542d61af74cf7454d7f1c8d96218d709de38f94ccfa7c16b15f726dc08c0

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'c5b4542d61af74cf7454d7f1c8d96218d709de38f94ccfa7c16b15f726dc08c0']

Name

04cf425e57e7d511f03189749c8c0a95483eeeb4c423e9ee1a6a766d2fe0094c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '04cf425e57e7d511f03189749c8c0a95483eeeb4c423e9ee1a6a766d2fe0094c']

Name

3875ed58c0d42e05c83843b32ed33d6ba5e94e18ffe8fb1bf34fd7dedf3f82a7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '3875ed58c0d42e05c83843b32ed33d6ba5e94e18ffe8fb1bf34fd7dedf3f82a7']

Name

e1ad173e49eee1194f2a55afa681cef7c3b8f6c26572f474dec7a42e9f0cdc9d

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e1ad173e49eee1194f2a55afa681cef7c3b8f6c26572f474dec7a42e9f0cdc9d']

Intrusion-Set

Name
ShroudedSnooper

Country

Name
Afghanistan

Name
United States of America

Malware

Name
PlugX

Description
[PlugX](https://attack.mitre.org/software/S0013) is a remote access tool (RAT) with modular plugins that has been used by multiple threat groups.(Citation: Lastline PlugX Analysis)(Citation: FireEye Clandestine Fox Part 2)(Citation: New DragonOK)(Citation: Dell TG-3390)

Name
HTTPSnoop

Name
PipeSnoop

StixFile

Value
04cf425e57e7d511f03189749c8c0a95483eeeb4c423e9ee1a6a766d2fe0094c
7495c1ea421063845eb8f4599a1c17c105f700ca0671ca874c5aa5aef3764c1c
3875ed58c0d42e05c83843b32ed33d6ba5e94e18ffe8fb1bf34fd7dedf3f82a7
9117bd328e37be121fb497596a2d0619a0eaca44752a1854523b8af46a5b0ceb
1146b1f38e420936b7c5f6b22212f3aa93515f3738c861f499ed1047865549cb
e1ad173e49eee1194f2a55afa681cef7c3b8f6c26572f474dec7a42e9f0cdc9d
c5b4542d61af74cf7454d7f1c8d96218d709de38f94ccfa7c16b15f726dc08c0

External References

- https://otx.alienvault.com/pulse/6509d00a36ac14cefa70d5e2
- https://blog.talosintelligence.com/introducing-shrouded-snooper/