Aug 26 2023

NETMANAGEIT

Intelligence Report

Tunnel Warfare: Exposing DNS Tunneling Campaigns using Generative Models – CoinLoader Case Study

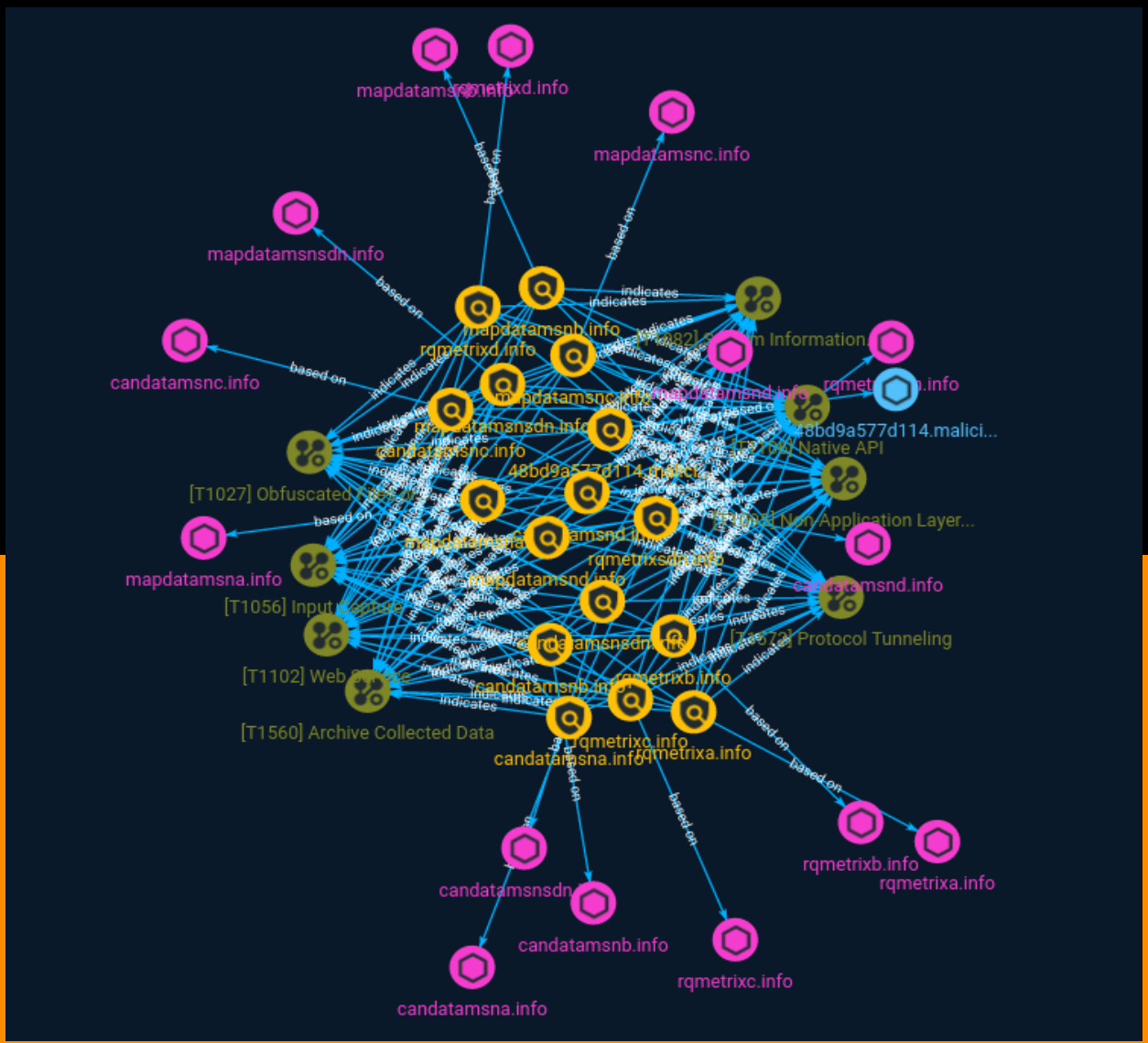

Table of contents

Overview

Entities

Observables

External References

Overview

Description

Generative AI has been around for nearly a decade, strictly speaking, but the recent boom in this technology has inspired renewed interest in its possible applications to challenges facing the information security community. Finding these challenges entails searching through a very large haystack consisting of brand-new binaries, documents, domains and other artifacts that flood the web every day.

Confidence

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

15 / 100

Indicator

Name
48bd9a577d114.maliciousdomain.xyz
Pattern Type
stix
Pattern
[hostname:value = '48bd9a577d114.maliciousdomain.xyz']
Name
rqmetrixc.info
Pattern Type
stix
Pattern
[domain-name:value = 'rqmetrixc.info']
Name
candatamsnb.info

Pattern Type
stix
Pattern
[domain-name:value = 'candatamsnb.info']
Name
rqmetrixb.info
Pattern Type
stix
Pattern
[domain-name:value = 'rqmetrixb.info']
Name
mapdatamsnsdn.info
Pattern Type
stix
Pattern
[domain-name:value = 'mapdatamsnsdn.info']
Name
mapdatamsnc.info

Pattern Type
stix
Pattern
[domain-name:value = 'mapdatamsnc.info']
Name
candatamsna.info
Pattern Type
stix
Pattern
[domain-name:value = 'candatamsna.info']
Name
candatamsnd.info
Pattern Type
stix
Pattern
[domain-name:value = 'candatamsnd.info']
Name
candatamsnsdn.info

Pattern Type
stix
Pattern
[domain-name:value = 'candatamsnsdn.info']
Name
mapdatamsna.info
Pattern Type
stix
Pattern
[domain-name:value = 'mapdatamsna.info']
Name
mapdatamsnb.info
Pattern Type
stix
Pattern
[domain-name:value = 'mapdatamsnb.info']
Name
rqmetrixa.info

Pattern Type
stix
Pattern
[domain-name:value = 'rqmetrixa.info']
Name
candatamsnc.info
Pattern Type
stix
Pattern
[domain-name:value = 'candatamsnc.info']
Name
rqmetrixsdn.info
Pattern Type
stix
Pattern
[domain-name:value = 'rqmetrixsdn.info']
Name
mapdatamsnd.info

Pattern Type
stix
Pattern
[domain-name:value = 'mapdatamsnd.info']
Name
rqmetrixd.info
Pattern Type
stix
Pattern
[domain-name:value = 'rqmetrixd.info']

Attack-Pattern

Name

Protocol Tunneling

ID

T1572

Description

Adversaries may tunnel network communications to and from a victim system within a separate protocol to avoid detection/network filtering and/or enable access to otherwise unreachable systems. Tunneling involves explicitly encapsulating a protocol within another. This behavior may conceal malicious traffic by blending in with existing traffic and/or provide an outer layer of encryption (similar to a VPN). Tunneling could also enable routing of network packets that would otherwise not reach their intended destination, such as SMB, RDP, or other traffic that would be filtered by network appliances or not routed over the Internet. There are various means to encapsulate a protocol within another protocol. For example, adversaries may perform SSH tunneling (also known as SSH port forwarding), which involves forwarding arbitrary data over an encrypted SSH tunnel. (Citation: SSH Tunneling) [Protocol Tunneling](https://attack.mitre.org/techniques/T1572) may also be abused by adversaries during [Dynamic Resolution](https://attack.mitre.org/techniques/T1568). Known as DNS over HTTPS (DoH), queries to resolve C2 infrastructure may be encapsulated within encrypted HTTPS packets.(Citation: BleepingComp Godlua JUL19) Adversaries may also leverage [Protocol Tunneling](https://attack.mitre.org/techniques/T1572) in conjunction with [Proxy](https://attack.mitre.org/techniques/T1090) and/or [Protocol Impersonation](https://attack.mitre.org/techniques/T1001/003) to further conceal C2 communications and infrastructure.

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. 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

Archive Collected Data

ID

T1560

Description

An adversary may compress and/or encrypt data that is collected prior to exfiltration. Compressing the data can help to obfuscate the collected data and minimize the amount of data sent over the network. Encryption can be used to hide information that is being exfiltrated from detection or make exfiltration less conspicuous upon inspection by a defender. Both compression and encryption are done prior to exfiltration, and can be performed using a utility, 3rd party library, or custom method.

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

Non-Application Layer Protocol

ID

T1095

Description

Adversaries may use an OSI non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive.(Citation: Wikipedia OSI) Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP

communication between hosts is one example.(Citation: Cisco Synful Knock Evolution) Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts.(Citation: Microsoft ICMP) However, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.

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

System Information Discovery

ID

T1082

Description

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](https://attack.mitre.org/software/S0096) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather detailed system information (e.g. `show version`).(Citation: US-CERT-TA18-106A) [System Information Discovery](https://attack.mitre.org/techniques/T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment.(Citation: OSX.FairyTale)(Citation: 20 macOS Common Tools and Techniques) Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virutal Machine API)

Domain-Name

Value
candatamsnd.info
mapdatamsnsdn.info
rqmetrixa.info
candatamsna.info
mapdatamsnb.info
rqmetrixb.info
candatamsnsdn.info
mapdatamsnd.info
candatamsnb.info
rqmetrixsdn.info
rqmetrixd.info
candatamsnc.info
mapdatamsnc.info

rqmetrixc.info

mapdatamsna.info

Hostname

Value
48bd9a577d114.maliciousdomain.xyz

External References

- https://otx.alienvault.com/pulse/64e8688ae20e56da413ac148
- https://research.checkpoint.com/2023/tunnel-warfare-exposing-dns-tunneling-campaigns-using-generative-models-coinloader-case-study/