Enhanced Attack Report

Smooth Operator

Generated on 2025-03-06

Disclaimer

This report has been generated automatically and should be used for informational purposes only. To generate this report, Large Language Models (LLMs) were used to analyze the provided data. This technology is not perfect and may generate incorrect or misleading results. The results should be reviewed by a human expert before taking any action based on the information provided.

Definitions

Pre-Conditions: Conditions that must be true to execute the attack steps in the milestone.

Post-Conditions: Traces that an attacker leaves behind after executing the attack steps in the milestone.

Attack Steps: Steps that an attacker would take to achieve the goal of the milestone.

MITRE Technique: Techniques from the MITRE ATT&CK; framework that are relevant to the milestone.

STIX

Malware Name: Smooth Operator
Malware Description: â€¢ Smooth Operator malware targets the macOS operating system. â€¢ Smooth Operator was distributed to victims as part of the 3CX supply chain attack. â€¢ The infected software package was signed by 3CX and notarized by Apple. â€¢ HTTPS is used as a C2 channel, with an additional custom encoding algorithm used to obfuscate exfiltrated data. â€¢ Smooth Operator randomises the C2 server it communicates with. The 3CX website is included in the list of C2 Servers it can beacon to. â€¢ Malicious code inserted into a dynamic library (dylib) packaged with the 3CX software, downloads and runs a second stage payload.

Quick Overview

Milestone 1
1. Supply Chain Compromise: Compromise Software Dependencies and Development Tools as used by the malware (T1195)

Milestone 2
1. Compromise Client Software Binary as used by the malware (T1574)

Milestone 3
1. Deobfuscate/Decode Files or Information as used by the malware (T1140)
2. Indicator Removal: File Deletion as used by the malware (T1070.010)
3. Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware (T1497.003)

Milestone 4
1. Automated Collection as used by the malware (T1005)

Milestone 5
1. Application Layer Protocol: Web Protocols as used by the malware (T1071.004)
2. Fallback Channels as used by the malware (T1008)

Milestone 6
1. Automated Exfiltration as used by the malware (T1020)

Milestone 1

Pre-Conditions:

- The necessary permissions and access rights are available.
- A way to cover the tracks of the compromise (e.g. a way to erase logs or other evidence).
- The necessary resources and tools are available to the malware.
- A connection to the software dependencies and development tools (e.g. network access).
- The software dependencies and development tools are vulnerable to compromise.
- The malware is using software dependencies and development tools.
- A way to analyze and understand the compromised software dependencies and development tools.
- The necessary permissions and access rights to compromise the software dependencies and development tools.
- The software dependencies and development tools are accessible.
- A secure environment to perform the compromise.
- The environment allows for the compromise of software dependencies and development tools.
- A way to execute the compromise (e.g. a command-line interface or a script).
- A way to monitor and control the compromise (e.g. a logging mechanism).
- The software dependencies and development tools are not properly secured.
- The malware has the necessary privileges to access the software dependencies and development tools.

Post-Conditions:

- Email attachments and files downloaded via links in emails often contain executable code.
- OneHub web services used to distribute remote access tools.
- Ransomware is deployed on victims' systems.
- Highly sensitive information is accessed by attackers.
- Files uploaded to victims' systems by malware, including additional files and tools.
- Sensitive data is stolen from government and commercial networks.
- Base64-encoded and custom hex byte swapping encoded data in C2 communications.
- Network connections to C2 infrastructure and other malicious servers.
- Registry keys added for persistence on victims' systems.
- C2 infrastructure is used to receive exfiltrated data.
- Small Sieve sample NSIS installer and index.exe backdoor on victims' systems.
- Users who browse the internet, use email, and execute code with administrator privileges are targeted by attackers.
- ScreenConnect application used to manage systems remotely and move laterally.
- Compromised websites used to relay information to C2.
- Victims' systems are compromised, enabling attackers to move laterally across the network and gain additional accesses.
- Logs of spearphishing attempts and successful infections.
- Proxy networks used to obfuscate C2 location.
- Malware is installed on victims' systems, including variants of PowGoop, Small Sieve, Canopy, Mori, and POWERSTATS.
- Enumeration scripts and web logs monitored by attackers.

Attack Step 1.1

==
Name: Supply Chain Compromise: Compromise Software Dependencies and Development Tools as used by the malware
Description: MuddyWater Supply Chain Compromise: Trojanised 3CX Software MuddyWater has demonstrated a sophisticated approach to supply chain compromise by targeting software dependencies and development tools. Here's how they perform this action: Target Selection: MuddyWater identifies popular software dependencies or development tools used by organizations they wish to target. Compromise: They gain access to the source code repositories, build systems, or distribution channels of these software packages. Trojanization: MuddyWater injects malicious code into the legitimate software components. This could involve: Backdoors: Adding hidden functionality that allows attackers to remotely control infected systems. Data Exfiltration: Modifying the software to steal sensitive data from compromised systems. Lateral Movement: Enabling the malware to spread to other systems within the target network. Distribution: The compromised software is then distributed through legitimate channels, such as official software repositories or company websites. Infection: When users download and install the compromised software, they unknowingly introduce the malware into their systems. Example: 3CX Software MuddyWater has been observed trojanizing 3CX software components. These components were signed and notarized, making them appear legitimate to users. This allowed MuddyWater to bypass security measures and gain access to a wide range of organizations that rely on 3CX for their communication needs. Consequences: Supply chain compromises can have devastating consequences for organizations. They can lead to: Data Breaches: Sensitive data can be stolen and used for malicious purposes. System Compromise: Attackers can gain control of critical systems and disrupt operations. Financial Losses: Organizations can incur significant costs to recover from a breach and mitigate damage. Reputational Damage: A supply chain compromise can damage an organization's reputation and erode customer trust.

MITRE Technique

ID: T1195
Name: Supply chain compromise
Description: Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise.
More info: https://attack.mitre.org/techniques/T1195/

Indicators

- Smooth Operator sample (MD5: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (SHA-1: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (SHA-256: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (MD5: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')
- Smooth Operator sample (SHA-1: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')
- Smooth Operator sample (SHA-256: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')
- Smooth Operator sample (MD5: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Smooth Operator sample (SHA-1: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Smooth Operator sample (SHA-256: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Filename: UpdateAgent

Milestone 2

Pre-Conditions:

- Resources: The malware requires system resources, such as memory and CPU, to execute and compromise the client software binary.
- Environment: A Windows-based environment with a vulnerable client software binary.
- Connectivity: The malware requires network connectivity to communicate with its command and control server.
- Execution: The malware requires the ability to execute malicious code on the system.
- The malware is present in the environment.
- The environment allows for the execution of malicious code.
- The client software binary is vulnerable to exploitation.
- Tools: The malware, which includes tools such as PowGoop, Small Sieve, Canopy, Mori, and POWERSTATS.
- Knowledge: The malware requires knowledge of the client software binary's vulnerabilities and weaknesses.
- The client software binary is accessible to the malware.
- Privileges: The malware requires elevated privileges to access and compromise the client software binary.

Post-Conditions:

- The victim's system has a malicious PowerShell script executed.
- A log entry in the Windows Event Viewer for the use of the compromised account for accessing sensitive information.
- The victim's system has a Registry Run key 'KCU\Software\Microsoft\Windows\CurrentVersion\Run\SystemTextEncoding' for persistence.
- The victim's system has a Word Template, 'Normal.dotm', for persistence.
- The backdoor is added as a Run key in the registry for persistence after reboot.
- The victim's system has a malicious link clicked.
- The victim's system has a malicious Python script executed.
- A log entry in the Windows Event Viewer for the addition of the backdoor as a Run key in the registry.
- A log entry in the Windows Event Viewer for the execution of the malicious JavaScript file.
- A network connection to the compromised account's RDP server.
- A network connection to the compromised account's email server.
- The victim's system has a scheduled task for persistence.
- A network connection to the file sharing service.
- A network connection to the compromised account's FTP server.
- A scheduled task in the Task Scheduler.
- The victim's system has a malicious VBS file executed.
- A log entry in the Windows Event Viewer for the downloading of the malicious file.
- The victim's system has a compromised account used for executing malicious code.
- A file 'Normal.dotm' in the Word Templates directory.
- A file 'index.exe' in the AppData/Roaming directory.
- The victim's system has a persistence registry key with the argument 'Platypus'.
- The victim's system has a malicious Office document executed.
- The victim's system has a compromised account used for spearphishing emails.

- A Registry Run key 'KCU\Software\Microsoft\Windows\CurrentVersion\Run\SystemTextEncoding' in the registry.
- The victim's system has a malicious JavaScript file executed.
- A log entry in the Windows Event Viewer for the execution of the malicious Python script.
- The victim's system has a malicious file 'index.exe' installed.
- The victim's system has a backdoor installed in the AppData/Roaming directory.
- A network connection to the C2 server.
- The victim's system has a malicious file downloaded.

Attack Step 2.1

==

Name: Compromise Client Software Binary as used by the malware
Description: Based on the provided context, here's how the "Compromise Client Software Binary" action is likely performed by the MuddyWater malware: MuddyWater likely compromises the 3CX client software binary through the following steps: Infection: MuddyWater gains initial access to a target system, possibly through phishing emails, exploiting vulnerabilities, or other means. Payload Delivery: Once inside the system, MuddyWater delivers its malicious payload, which could be a backdoor, a downloader, or a script. Modification: MuddyWater modifies the legitimate 3CX client software binary. This could involve: Injecting malicious code: Adding its own code into the existing binary, allowing it to execute when the 3CX client is launched. Replacing legitimate code: Overwriting parts of the original binary with its own malicious instructions. Appending malicious code: Adding malicious code as a separate section to the binary, which is executed after the legitimate code. Persistence: MuddyWater ensures its modified 3CX binary remains on the system and executes whenever the 3CX client is used. This could involve: Modifying registry entries: Adding entries that automatically launch the modified 3CX binary. Creating scheduled tasks: Setting up tasks to run the modified binary at specific times. Using other persistence mechanisms: Employing techniques like creating hidden files or folders to store and execute the modified binary. Command and Control: The compromised 3CX client now acts as a conduit for MuddyWater to communicate with its command and control (C2) server. This allows the attackers to: Steal data: Exfiltrate sensitive information from the compromised system. Execute commands: Run additional malicious code on the system. Spread laterally: Use the compromised system to attack other systems on the network. Let me know if you have any other questions.

MITRE Technique

ID: T1574
Name: Hijack execution flow
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.
More info: https://attack.mitre.org/techniques/T1574/

Indicators

- Smooth Operator sample (MD5: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (SHA-1: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (SHA-256: d5101c3b86d973a848ab7ed79cd11e5a Filename: '3CXDesktopApp-18.12.416.dmg')
- Smooth Operator sample (MD5: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')

- Smooth Operator sample (SHA-1: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')
- Smooth Operator sample (SHA-256: 660ea9b8205fbd2da59fefd26ae5115c Filename: 'libffmpeg.dylib')
- Smooth Operator sample (MD5: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Smooth Operator sample (SHA-1: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Smooth Operator sample (SHA-256: 5faf36ca90f6406a78124f538a03387a Filename: 'UpdateAgent')
- Filename: UpdateAgent

Milestone 3

Pre-Conditions:

- Connectivity: The malware communicates with the actor's C2 server over either IPv4 or IPv6, depending on the value of an unidentified flag.
- Environment: The malware is executed in a Windows environment.
- The deobfuscated/decoded files or information is obtained.
- Tools: The malware uses PowerShell and JavaScript files to maintain persistent access to the victim's systems.
- Resources: The malware uses the Registry Keys, HKLM\\Software\\NFC\\IPA and HKLM\\Software\\NFC\\(Default), to read and/or write data.
- Environment: The malware is executed in a Windows environment, likely on a 64-bit system, as it uses PowerShell scripts and registry keys, which are commonly used in Windows.
- Connectivity: The malware communicates using HTTP over either IPv4 or IPv6, which implies that the system has internet connectivity.
- Resources: The malware uses registry keys, registry keys, and PowerShell scripts, which suggests that the system has sufficient resources (e.g., memory, CPU, and CPU) to execute these operations.

Post-Conditions:

- The system's local proxy settings are disabled.
- Ransomware is deployed on the victim's system.
- Logs of credential dumping with Mimikatz and procdump64.exe.
- The victim's OS version and machine name are collected.
- Logs of process discovery.
- Logs of file and directory discovery.
- Modified registry keys and system settings.
- The victim's username is collected.
- Logs of lateral movement across the network.
- Registry keys modified by malware.
- Network connections to command and control servers.
- Modified system files and configuration.
- Ransomware files and logs.
- Logs of system compromise and infection.
- A list of running processes on the system is obtained.
- Logs of system exploitation and compromise.
- Logs of credential dumping with LaZagne.
- The system's administrator privileges are exploited to move laterally across the network.
- Logs of system network configuration discovery.
- The ProgramData folder is checked for folders or files with specific keywords.
- Network connections to C2 servers.
- Logs of password stealing from email.
- Logs of system owner/user discovery.
- The victim's IP address and domain name are collected.
- The system is infected with malware, including Small Sieve, POWERSTATS, and other tools.
- Files with malicious macros or executable code.
- The victim's email is accessed and passwords are stolen.
- System network connections discovery may be performed.

- mshta.exe may be used to execute PowerShell one-liners.
- Files with keywords "Kasper," "Panda," or "ESET" may be checked in the ProgramData folder.
- Process discovery may be performed.
- File and directory discovery may be performed.
- makecab.exe may be used to compress stolen data to be uploaded.
- The victim's sensitive networks may be identified and targeted by the Iranian government-sponsored APT group.
- System information discovery may be performed.
- The victim's system may be used to collect sensitive information, including credentials, system information, and network connections.
- procdump64.exe and LaZagne may be used to dump credentials from LSASS memory and LSA secrets.
- The victim's system may be used to dump credentials from LSASS memory and LSA secrets.
- The victim's system may be used to capture screenshots and compress stolen data to be uploaded.
- The victim's system may be used to avoid detection by using custom string and traffic obfuscation schemes.
- PowerShell backdoor may be used to check for Skype connectivity and security tools.
- rundll32.exe may be used to execute a .dll.
- The victim's system may be compromised with malware, including variants of PowGoop, Small Sieve, Canopy, Mori, and POWERSTATS.
- The victim's system may be used as a foothold to maintain and expand access to the victim's infrastructure.
- Telegram Bot API may be used to communicate with the C2 server.
- The victim's system may be used to enumerate domain users and check for security software.
- The victim's system may be used to disable system's local proxy settings.
- Registry keys (HKLM\Software\NFC\IPA and HKLM\Software\NFC(Default)) may be accessed and modified.
- cmd.exe net user/domain may be used to enumerate domain users.
- HTTP communications may be used for C2 communications.
- System network configuration and owner/user discovery may be performed.
- The victim's system may be used to check for Skype connections and security tools on the target machine.
- Malware is used to steal passwords saved in victim email, leaving behind logs.
- The attackers can check for Skype connections on the target machine.
- The attackers can move laterally across the network, gain additional accesses, and access highly sensitive information.
- cmd.exe net user/domain commands are executed to enumerate domain users, leaving behind Windows logs.
- The attackers can check if the ProgramData folder has folders or files with the keywords "Kasper," "Panda," or "ESET."
- The attackers can collect the victim's username.
- The attackers can compress stolen data to be uploaded using the native Windows cabinet creation tool, makecab.exe.
- ZIP files containing malicious macros or PDF files that drop malicious files are downloaded.
- Malware is used to perform credential dumping with LaZagne, leaving behind logs.
- The attackers can collect the victim's OS version and machine name.
- Malware is used to perform credential dumping with Mimikatz and procdump64.exe, leaving behind logs.
- Malware is used to collect the victim's OS version and machine name, which is stored in logs.
- The attackers can capture screenshots of the victim's machine.
- The victim's system is compromised with malware, allowing the attackers to gain access to sensitive information.
- HTTP communications are used for C2 communications, leaving behind network logs.

- The victim's system is infected with various malware variants, including PowGoop, Small Sieve, Canopy, Mori, and POWERSTATS.
- Executable code is executed from email attachments or files downloaded via links in emails.
- The attackers can collect the victim's IP address and domain name.
- Malware is used to disable the system's local proxy settings, leaving behind logs.
- The attackers can check running processes against a hard-coded list of security tools often used by malware researchers.
- PowerShell backdoors are used to check for Skype connections on the target machine, leaving behind PowerShell logs.
- The attackers can obtain a list of running processes on the system.
- The attackers can enumerate domain users using cmd.exe net user/domain.

Attack Step 3.1

==
Name: Deobfuscate/Decode Files or Information as used by the malware
Description: Let's break down how MuddyWater uses deobfuscation and decode techniques. MuddyWater's Deobfuscation Tactics MuddyWater employs a multi-layered approach to obfuscate its malicious activities, making it harder for security tools to detect and analyze. Here's a breakdown: Custom Algorithm: MuddyWater utilizes a unique algorithm to scramble data both before it's sent to its command-and-control (C2) server and after it receives instructions. This custom algorithm is a key part of their evasion strategy. File-Based Obfuscation: When MuddyWater writes data to files on the infected system, it applies the same obfuscation algorithm. This hides the true nature of the data from casual inspection. Response Decoding: MuddyWater also decodes the responses it receives from its C2 server. This ensures that even if a security tool intercepts the communication, the decrypted instructions remain hidden. Why This Matters Evasion: Obfuscation makes it significantly harder for security software to recognize and block MuddyWater's malicious actions. Stealth: By hiding its data, MuddyWater can operate more stealthily on infected systems, making it harder to detect its presence. Analysis Challenges: Decoding MuddyWater's data requires specialized tools and expertise, making it more difficult for security researchers to understand its full capabilities and develop effective countermeasures. Mitigating the Threat Advanced Threat Detection: Implement security solutions that can detect and analyze suspicious file activity, network traffic patterns, and code execution, even when obfuscated. Behavioral Analysis: Use security tools that focus on identifying malicious behavior rather than relying solely on signature-based detection. Threat Intelligence: Stay informed about the latest tactics used by MuddyWater and other advanced persistent threats (APTs). Regular Security Updates: Keep your systems and software patched to minimize vulnerabilities that attackers can exploit. Let me know if you have any other questions about MuddyWater or malware analysis techniques.

MITRE Technique

ID: T1140
Name: Deobfuscate/decode files or information
Description: Adversaries may use Obfuscated Files or Information 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.
More info: https://attack.mitre.org/techniques/T1140/

Indicators

- The file '3CXDesktopApp-18.12.416.dmg' has a MD5 hash of 'd5101c3b86d973a848ab7ed79cd11e5a'.

- The file '3CXDesktopApp-18.12.416.dmg' has a SHA-1 hash of '3dc840d32ce86cebf657b17cef62814646ba8e98'.
- The file '3CXDesktopApp-18.12.416.dmg' has a SHA-256 hash of 'ebbbcc33815b9f20b0cf832d7401dd893fbc467c800728b5891336706da0dbcecc'.
- The file 'libffmpeg.dylib' has a MD5 hash of '660ea9b8205fbd2da59fefd26ae5115c'.
- The file 'libffmpeg.dylib' has a SHA-1 hash of '769383fc65dc1386dd141c960c9970114547da0c2'.
- The file 'libffmpeg.dylib' has a SHA-256 hash of 'a64fa9f1c76457ecc58402142a8728c34ccba378c17318b3340083eeeb7acc67'.
- The file 'UpdateAgent' has a MD5 hash of '5faf36ca90f6406a78124f538a03387a'.
- The file 'UpdateAgent' has a SHA-1 hash of '9e9a5f8d86356796162cee881c843cde9eaedefb3'.
- The file 'UpdateAgent' has a SHA-256 hash of '6c121f2bf2befa6592c2c22b29218157ec9e63f385e7a1d7425857d603ddeff8c59'.
- The file ' .main_storage' has a name of '.main_storage'.

Attack Step 3.2

==
Name: Indicator Removal: File Deletion as used by the malware
Description: Let's break down the provided text and address the request for a specific action and its description. Understanding the Text The text describes the tactics, techniques, and procedures (TTPs) used by the MuddyWater threat actor. It highlights various malicious activities, including: Initial Access: How MuddyWater gains access to target systems (not explicitly detailed in this excerpt). Persistence: Techniques used to maintain access to compromised systems (e.g., PowerShell backdoors, Registry Run keys). Credential Access: Methods for stealing user credentials (e.g., Mimikatz, LaZagne). Discovery: How MuddyWater gathers information about the target environment (e.g., system information, running processes, network connections). Defense Evasion: Tactics used to avoid detection by security tools (e.g., string obfuscation, execution guardrails). Indicator Removal: File Deletion The text mentions: "Indicator Removal: File Deletion as used by the malware the malware’s second stage deletes itself from disk immediately on execution." Action Description: The action is Indicator Removal: File Deletion. This means the malware actively attempts to remove itself from the victim's system to hinder detection and analysis. How it's Performed: The malware's second stage (likely a payload) deletes itself from disk immediately after execution. This prevents security tools from finding remnants of the malware on the system, making it harder to trace back the attack. Let me know if you have any other questions or need further clarification on MuddyWater's TTPs.

MITRE Technique

ID: T1070.010
Name: Indicator removal: relocate malware
Description: Once a payload is delivered, adversaries may reproduce copies of the same malware on the victim system to remove evidence of their presence and/or avoid defenses. Copying malware payloads to new locations may also be combined with File Deletion to cleanup older artifacts.
More info: https://attack.mitre.org/techniques/T1070/010

Indicators

- The file '3CXDesktopApp-18.12.416.dmg' was deleted.
- The file 'libffmpeg.dylib' was deleted.
- The file 'UpdateAgent' was deleted.
- The file '.main_storage' was deleted.

Attack Step 3.3

==
Name: Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware
Description: You're asking about a specific evasion technique used by the MuddyWater APT group, which is called Time-Based Evasion. Here's how it works in the context of the provided information: The Malware Sleeps: The malware, specifically PowGoop, doesn't immediately start communicating with its command-and-control (C2) server after infecting a system. Instead, it enters a dormant state, effectively "sleeping" for a significant period. Delayed Beaconing: After a week (or potentially longer), the malware wakes up and establishes a connection with its C2 server. This delayed beaconing is the core of the time-based evasion technique. Why This Works: Analysis Delays: Security researchers and analysts often rely on sandboxes and virtualized environments to analyze malware. These environments typically have a limited lifespan. If malware sleeps for a week, it might evade detection in a sandbox before it has a chance to exhibit its malicious behavior. Reduced Visibility: By delaying its activity, the malware reduces its chances of being noticed by security tools that monitor for suspicious network traffic or process activity. Mitigations: Long-Term Monitoring: Security teams need to implement monitoring solutions that can track activity over extended periods, not just within the typical timeframe of a sandbox analysis. Behavioral Analysis: Focus on analyzing the malware's behavior rather than just its immediate actions. Look for patterns and anomalies that might indicate delayed or scheduled activity. Threat Intelligence: Stay informed about known TTPs used by threat actors like MuddyWater. This knowledge can help you anticipate and detect time-based evasion techniques. Let me know if you have any other questions.

MITRE Technique

ID: T1497.003
Name: Virtualization/sandbox evasion: time based evasion
Description: Adversaries may employ various time-based methods to detect and avoid virtualization and analysis environments. This may include enumerating time-based properties, such as uptime or the system clock, as well as the use of timers or other triggers to avoid a virtual machine environment (VME) or sandbox, specifically those that are automated or only operate for a limited amount of time.
More info: https://attack.mitre.org/techniques/T1497/003

Indicators

No indicators found.

Milestone 4

Pre-Conditions:

Post-Conditions:

- Publicly reported vulnerabilities are exploited.
- HTTP requests are made to C2 servers over IPv4 or IPv6.
- PowerShell scripts are run to maintain persistent access to victim systems.
- Custom hex byte swapping encoding scheme is used to obfuscate tasking traffic.
- Compromised websites are used to relay information to C2.
- Registry keys are modified (HKLM\Software\NFC\IPA and HKLM\Software\NFC\(Default)).
- Malicious files are dropped onto victim networks.
- Persistent access is maintained to victim systems through the POWERSTATS backdoor.
- Logs are created on C2 servers and victim systems.
- ScreenConnect is used to manage systems remotely and move laterally.
- Malicious tools and strategies are used to gain access to sensitive data on victim systems.
- Small Sieve beacons and tasking are performed using the Telegram API over HTTPS.
- Base64 encoding is used to obfuscate tasking traffic.
- Proxy networks are used to obfuscate C2 locations.
- JSON is used to serialize C2 commands and/or their results.
- Malware is uploaded to Virus Total for analysis.
- Exfiltrated data is received through C2 channels.
- Web services are used to distribute remote access tools.
- C2 servers are established and used to communicate with victim systems.
- Ransomware is deployed on victim systems.
- Sensitive data is stolen and shared with the Iranian government.
- Files are created on victim systems to store malicious tools and data.
- System calls are made to execute malicious code on victim systems.
- Malicious macros are executed on victim systems.

Attack Step 4.1

==
Name: Automated Collection as used by the malware
Description: Let's break down how MuddyWater's malware performs automated data collection and exfiltration. Automated Data Collection: Initial Access: MuddyWater often gains access to victim systems through phishing campaigns, exploiting vulnerabilities, or using compromised credentials. Payload Delivery: Once inside, they deploy various malware tools, including backdoors like POWERSTATS and Small Sieve. Data Reconnaissance: These backdoors act as persistent agents, allowing MuddyWater to: Enumrate System Information: Gather details about the operating system, running processes, user accounts, and network configurations. Read Files: Access sensitive files like documents, configuration files, and credentials stored on the victim's machine. Capture Keystrokes: Monitor user activity and steal passwords, credit card information, or other sensitive data. Data Encoding and Obfuscation: MuddyWater uses techniques like Base64 encoding and a custom hex byte swapping scheme (Small Sieve) to disguise the collected data, making it harder to detect. Exfiltration: C2 Communication: The stolen data is transmitted back to MuddyWater's command-and-control (C2) servers. This communication often occurs through: Telegram Bot API: MuddyWater leverages the

Telegram messaging platform for C2 communication, hiding its malicious activity within seemingly legitimate chat messages. Other Web Services: They may also use compromised websites or legitimate web services like OneHub to relay stolen data. Data Storage: Once received, the exfiltrated data is likely stored on MuddyWater's servers for later analysis, exploitation, or distribution to other malicious actors. Mitigations: Multi-Factor Authentication: Enforce MFA for all critical accounts to prevent unauthorized access. Endpoint Security: Implement robust endpoint security solutions with up-to-date antivirus and anti-malware protection. Network Monitoring: Monitor network traffic for suspicious activity, including unusual communication patterns or attempts to exfiltrate data. User Awareness Training: Educate users about phishing attacks and the importance of reporting suspicious emails or links. Vulnerability Management: Regularly patch systems and applications to address known vulnerabilities. Data Loss Prevention (DLP): Implement DLP solutions to detect and prevent the unauthorized transfer of sensitive data. Let me know if you have any other questions.

MITRE Technique

ID: T1005
Name: Data from local system
Description: Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration.
More info: https://attack.mitre.org/techniques/T1005/

Indicators

- The file '3CXDesktopApp-18.12.416.dmg' has a MD5 hash of 'd5101c3b86d973a848ab7ed79cd11e5a'.
- The file '3CXDesktopApp-18.12.416.dmg' has a SHA-1 hash of '3dc840d32ce86cebf657b17cef62814646ba8e98'.
- The file '3CXDesktopApp-18.12.416.dmg' has a SHA-256 hash of 'ebbbcc33815b9f20b0cf832d7401dd893fbc467c800728b5891336706da0dbcecc'.
- The file 'libffmpeg.dylib' has a MD5 hash of '660ea9b8205fbd2da59fefd26ae5115c'.
- The file 'libffmpeg.dylib' has a SHA-1 hash of '769383fc65dc1386dd141c960c9970114547da0c2'.
- The file 'libffmpeg.dylib' has a SHA-256 hash of 'a64fa9f1c76457ecc58402142a8728c34ccba378c17318b3340083eeeb7acc67'.
- The file 'UpdateAgent' has a MD5 hash of '5faf36ca90f6406a78124f538a03387a'.
- The file 'UpdateAgent' has a SHA-1 hash of '9e9a5f8d86356796162cee881c843cde9eaedefb3'.
- The file 'UpdateAgent' has a SHA-256 hash of '6c121f2bf2e6a6592c2c22b29218157ec9e63f385e7a1d7425857d603ddeff8c5'.
- The domain 'msstorageazure.com' was observed.
- The domain 'officestoragebox.com' was observed.
- The domain 'visualstudiofactory.com' was observed.
- The domain 'azuredeploystore.com' was observed.
- The domain 'msstorageboxes.com' was observed.
- The domain 'officeaddons.com' was observed.
- The domain 'sourceslabs.com' was observed.
- The domain 'zacharryblogs.com' was observed.
- The domain 'pbxcloudeservices.com' was observed.
- The domain 'pbxphonenetwork.com' was observed.
- The domain 'akamaitechcloudservices.com' was observed.
- The domain 'azureonlinestorage.com' was observed.
- The domain 'msedgepackageinfo.com' was observed.
- The domain 'glcloudservice.com' was observed.
- The domain 'pbxsources.com' was observed.
- The domain 'sbmsa.wiki' was observed.

- The URL 'https://msstorageazure.com/analysis' was observed.
- The URL 'https://officestoragebox.com/api/biosync' was observed.
- The URL 'https://visualstudiofactory.com/groupcore' was observed.
- The URL 'https://azuredeploystore.com/cloud/images' was observed.
- The URL 'https://msstorageboxes.com/xbox' was observed.
- The URL 'https://officeaddons.com/quality' was observed.
- The URL 'https://sourceslabs.com/status' was observed.
- The URL 'https://zacharryblogs.com/xmlquery' was observed.
- The URL 'https://pbxcloudeservices.com/network' was observed.
- The URL 'https://pbxphonenetwork.com/phone' was observed.
- The URL 'https://akamaitechcloudservices.com/v2/fileapi' was observed.
- The URL 'https://azureonlinestorage.com/google/storage' was observed.
- The URL 'https://msedgepackageinfo.com/ms-webview' was observed.
- The URL 'https://glcloudservice.com/v1/status' was observed.
- The URL 'https://pbxsources.com/queue' was observed.
- The URL 'https://sbmsa.wiki/blog/_insert' was observed.
- The file 'UpdateAgent' was observed.
- The file '.main_storage' was observed.

Milestone 5

Pre-Conditions:

- The malware has access to a network connection.
- Tools: A web browser or a web client library/framework (e.g., Python's requests library).
- Connectivity: A stable internet connection with access to the web server or proxy server.
- The malware has the necessary credentials to authenticate with the web server or proxy server.
- Resources: A web server or proxy server with the necessary protocols and ports open.
- Environment: A network connection with access to the internet.
- The malware has the necessary ports open to establish a connection to the web server.
- The malware has the necessary encryption keys to secure the web communication.
- The target system has a compatible operating system and architecture.
- The malware has established a connection to the primary C2 server.
- The malware has access to the necessary resources, including storage and processing power.
- The malware has access to the necessary tools and resources, including encryption and obfuscation tools.
- The malware has been successfully deployed on the target system.
- The malware has access to the necessary tools and resources to establish a fallback channel.
- The target system has a stable internet connection.
- The fallback channel is configured and available for use.
- The fallback channel is secured with encryption and obfuscation to prevent detection.
- The target system has the necessary software and libraries to support the fallback channel.

Post-Conditions:

- The malware checks running processes against a hard-coded list of security tools.
- The C2 commands and/or their results are serialized in JSON.
- PowerShell script logs.
- Screenshot capture logs.
- The traffic to the Telegram Bot API is protected by TLS.
- The ScreenConnect application is used to manage systems remotely and move laterally.
- The Registry Keys HKLM\Software\NFC\IPA and HKLM\Software\NFC(Default) are read and/or written.
- The malware captures screenshots of the victim’s machine.
- The stolen data is compressed using the native Windows cabinet creation tool, makecab.exe.
- The C2 infrastructure is used to receive exfiltrated data.
- The updated token will be stored in the encoded MicrosoftWindowsOutlookDataPlus.txt file.
- Hex byte shuffling algorithm logs.
- The backdoor will reconnect to the Telegram Bot API using the provided token.
- C2 infrastructure logs.
- The MuddyWater group uses a custom hex byte swapping encoding scheme to obfuscate tasking traffic.
- HTTP logs over IPv4 or IPv6.
- The original connection to Telegram is terminated.
- The POWERSTATS backdoor runs PowerShell scripts to maintain persistent access to the victim systems.
- HTTP protocol logs for C2 communications.
- Custom hex byte swapping encoding scheme logs.
- TLS encryption logs.

- The Small Sieve obfuscates its tasking and response using a hex byte shuffling algorithm.
- Logs of the Telegram Bot API connection and disconnection.
- ScreenConnect application logs.
- System process logs (malware, ScreenConnect, etc.).
- The malware checks for Skype connectivity on the target machine.
- File system modifications logs (encoded MicrosoftWindowsOutlookDataPlus.txt, etc.).
- The C2 communications use HTTP over either IPv4 or IPv6.

Attack Step 5.1

==
Name: Application Layer Protocol: Web Protocols as used by the malware
Description: The malware, Small Sieve, uses the Telegram Bot API over HTTPS for its command and control (C2) communication. This means that the malicious instructions sent from the attackers to the infected system, and the data sent back from the infected system, travel over the internet using standard web protocols (HTTP) but with an encrypted layer (HTTPS) to mask the communication. Let me know if you have any other actions you'd like me to explain!

MITRE Technique

ID: T1071.004
Name: Application layer protocol: dns
Description: Adversaries may communicate using the Domain Name System (DNS) application layer protocol to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.
More info: https://attack.mitre.org/techniques/T1071/004

Indicators

- Domain Name: msstorageazure.com
- Domain Name: officestoragebox.com
- Domain Name: visualstudiofactory.com
- Domain Name: azuredeploystore.com
- Domain Name: msstorageboxes.com
- Domain Name: officeaddons.com
- Domain Name: sourceslabs.com
- Domain Name: zacharryblogs.com
- Domain Name: pbxcloudeservices.com
- Domain Name: pbxphonenetwork.com
- Domain Name: akamaitechcloudservices.com
- Domain Name: azureonlinestorage.com
- Domain Name: msedgepackageinfo.com
- Domain Name: glcloudservice.com
- Domain Name: pbxsources.com
- Domain Name: sbmsa.wiki
- URL: https://msstorageazure.com/analysis
- URL: https://officestoragebox.com/api/biosync
- URL: https://visualstudiofactory.com/groupcore
- URL: https://azuredeploystore.com/cloud/images
- URL: https://msstorageboxes.com/xbox
- URL: https://officeaddons.com/quality

- URL: https://sourceslabs.com/status
- URL: https://zacharryblogs.com/xmlquery
- URL: https://pbxcloudeservices.com/network
- URL: https://pbxphonenetwork.com/phone
- URL: https://akamaitechcloudservices.com/v2/fileapi
- URL: https://azureonlinestorage.com/google/storage
- URL: https://msedgepackageinfo.com/ms-webview
- URL: https://glcloudservice.com/v1/status
- URL: https://pbxsources.com/queue
- URL: https://sbmsa.wiki/blog/_insert

Attack Step 5.2

==
Name: Fallback Channels as used by the malware
Description: The malware implements Fallback Channels as a resilience mechanism. It maintains a list of multiple C2 servers and, for each beacon transmission, randomly selects a new server from this list. This ensures that if one C2 server becomes unavailable or is disrupted, the malware can continue to communicate with the adversary through an alternative server. Let me know if you have any other actions you'd like me to describe!

MITRE Technique

ID: T1008
Name: Fallback channels
Description: Adversaries may use fallback or alternate communication channels if the primary channel is compromised or inaccessible in order to maintain reliable command and control and to avoid data transfer thresholds.
More info: https://attack.mitre.org/techniques/T1008/

Indicators

- The domain name "msstorageazure.com" was observed.
- The domain name "officestoragebox.com" was observed.
- The domain name "visualstudiofactory.com" was observed.
- The domain name "azuredeploystore.com" was observed.
- The domain name "msstorageboxes.com" was observed.
- The domain name "officeaddons.com" was observed.
- The domain name "sourceslabs.com" was observed.
- The domain name "zacharryblogs.com" was observed.
- The domain name "pbxcloudeservices.com" was observed.
- The domain name "pbxphonenetwork.com" was observed.
- The domain name "akamaitechcloudservices.com" was observed.
- The domain name "azureonlinestorage.com" was observed.
- The domain name "msedgepackageinfo.com" was observed.
- The domain name "glcloudservice.com" was observed.
- The domain name "pbxsources.com" was observed.
- The URL "https://msstorageazure.com/analysis" was observed.
- The URL "https://officestoragebox.com/api/biosync" was observed.
- The URL "https://visualstudiofactory.com/groupcore" was observed.
- The URL "https://azuredeploystore.com/cloud/images" was observed.
- The URL "https://msstorageboxes.com/xbox" was observed.

- The URL "https://officeaddons.com/quality" was observed.
- The URL "https://sourceslabs.com/status" was observed.
- The URL "https://zacharryblogs.com/xmlquery" was observed.
- The URL "https://pbxcloudeservices.com/network" was observed.
- The URL "https://pbxphonenetwork.com/phone" was observed.
- The URL "https://akamaitechcloudservices.com/v2/fileapi" was observed.
- The URL "https://azureonlinestorage.com/google/storage" was observed.
- The URL "https://msedgepackageinfo.com/ms-webview" was observed.
- The URL "https://glcloudservice.com/v1/status" was observed.
- The URL "https://pbxsources.com/queue" was observed.
- The URL "https://sbmsa.wiki/blog/_insert" was observed.

Milestone 6

Pre-Conditions:

- A secure protocol such as HTTPS for communication between the malware and the C2 server.
- The malware has been successfully installed on the victim's machine.
- The malware has established a connection to the Command and Control (C2) server.
- The C2 server is operational and configured to receive exfiltrated data.
- The malware has the capability to encode or obfuscate the exfiltrated data.
- The malware has the necessary credentials or permissions to access sensitive data.
- A Virus Total account or repository for malware aggregation and analysis.
- The malware has access to the victim's network and can transmit data over the network.
- A Telegram API account or token for Small Sieve malware to communicate with the C2 server.
- A network connection between the victim's machine and the C2 server.
- A ScreenConnect application for remote access and lateral movement.
- A PowerShell environment for the POWERSTATS backdoor to run scripts and maintain persistent access.

Post-Conditions:

- The actors have used remote access software to manage systems remotely and move laterally.
- Telegram API logs used to communicate with Small Sieve beacons.
- The actors have used malware to exfiltrate data from victim systems.
- The actors have used proxy networks to obfuscate their C2 location.
- Logs of multi-stage channel communications.
- Files used to distribute malware, including PowGoop, Small Sieve, Canopy/Starwhale, Mori, and POWERSTATS.
- Obfuscated PowerShell scripts used to hide C2 functions.
- Logs of side-loading DLLs used to trick legitimate programs into running malware.
- Logs of proxy network connections.
- The actors have used various techniques to maintain persistent access to victim systems.
- PowerShell scripts used to maintain persistent access to victim systems.
- The actors have used web services to distribute remote access tools.
- Logs of C2 communications used to receive exfiltrated data.
- Logs of remote access software usage.
- Encoded data stored on victim systems.
- The actors have used exfiltration over C2 channels to receive exfiltrated data.
- Exfiltrated data stored on C2 infrastructure.
- Hex byte swapping encoding scheme used to obfuscate tasking traffic.
- Logs of spearphishing emails used to distribute Canopy/Starwhale malware.
- The actors have used multi-stage channels to communicate with victim systems.
- The actors have conducted cyber operations against these networks.
- Registry keys modified by malware on victim systems.
- Obfuscated Base64 function used to obfuscate tasking traffic.
- Logs of HTTPS connections used to communicate with Small Sieve beacons.
- Files used to establish connections back to malicious infrastructure.
- Files created by malware on victim systems.
- Encoded data stored on Telegram API servers.
- The Iranian Government-Sponsored Actors have gained access to global government and commercial networks.

- Network connections to web services used to distribute remote access tools.
- The actors have used data encoding to obfuscate C2 communications.

Attack Step 6.1

==
Name: Automated Exfiltration as used by the malware
Description: Let's break down how PowGoop and Small Sieve achieve automated exfiltration. PowGoop's Role: Data Collection: While the provided text doesn't explicitly detail what data PowGoop collects, it's common for malware like this to gather sensitive information such as: System information (OS version, hardware details) User credentials Network configurations Files from specific directories Configuration: The config.txt script likely contains instructions for PowGoop on what data to collect, where to send it, and how to handle communication with the attacker's command-and-control (C2) server. Small Sieve's Role: Exfiltration Channel: Small Sieve, written in Python, uses the Telegram Bot API for communication. This is a clever tactic because Telegram is often overlooked as a potential C2 channel. Automated Transmission: The text mentions that Small Sieve "communicates with the MuddyWater C2 server using an encrypted channel." This implies that Small Sieve is designed to automatically send the data it receives from PowGoop (or other sources) to the attacker's server without requiring manual intervention. How Automated Exfiltration Works: Data Gathering: PowGoop runs on the compromised system and collects the specified data. Data Encoding/Obfuscation: The collected data might be encoded or obfuscated to make it harder to detect by security tools. Transmission to Small Sieve: PowGoop likely sends the data to Small Sieve, which acts as a intermediary. Telegram Exfiltration: Small Sieve uses the Telegram Bot API to securely transmit the data to the attacker's server. Key Points: Persistence: PowGoop's use of DLL side-loading suggests it's designed to persist on the system, allowing it to continue collecting data even after reboots. Evasion: The use of Telegram as a C2 channel and string obfuscation are techniques used by attackers to evade detection by security tools. Command and Control: The attacker can use the config.txt script to modify PowGoop's behavior, including the types of data it collects and how it's transmitted. Let me know if you have any other questions.

MITRE Technique

ID: T1020
Name: Automated exfiltration
Description: Adversaries may exfiltrate data, such as sensitive documents, through the use of automated processing after being gathered during Collection.
More info: https://attack.mitre.org/techniques/T1020/

Indicators

- A file named 'UpdateAgent' was observed.
- A file named '.main_storage' was observed.
- The following URLs were accessed: