Enhanced Attack Report

Small Sieve

Generated on 2025-03-20

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: Small Sieve
Malware Description: Small Sieve is a simple â€“ possibly disposable â€“ Python backdoor which is distributed using an NSIS installer that performs persistence. It provides basic functionality required to maintain and expand a foothold in victim infrastructure using custom string and traffic obfuscation schemes together with the Telegram Bot API to avoid detection.

Quick Overview

Milestone 1
1. Command and Scripting Interpreter: Python as used by the malware (T1059.006)

Milestone 2
1. Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder as used by the malware (T1547.001)

Milestone 3
1. Obfuscated Files or Information as used by the malware (T1027.013)
2. Execution Guardrails as used by the malware (T1480.001)
3. Masquerading: Match Legitimate Name or Location as used by the malware (T1036.005)

Milestone 4
1. Application Layer Protocol: Web Protocols as used by the malware (T1071.001)
2. Data Encoding: Non-Standard Encoding as used by the malware (T1132.002)

Milestone 1

Pre-Conditions:

- Access to a Python interpreter.
- MuddyWater actors have developed tools in Python.
- Connectivity to command and control servers (implied for tool deployment and communication).

Post-Conditions:

- PowerShell scripts used for execution and communication.
- Network traffic to Telegram servers using bot API.
- Files downloaded from external URLs containing malicious code.
- Compromised system with persistent malware infection.
- Potential for further lateral movement and escalation within the compromised network.
- Indicators of compromise (IOCs) matching known MuddyWater tactics and techniques.
- Modified system proxy settings.
- Altered registry entries related to malware persistence.
- Increased risk of ransomware attacks or other malicious activities.
- Logs indicating suspicious activity, including command executions and data transfers.
- Data exfiltration to external servers controlled by MuddyWater actors.
- Small Sieve payload files on infected systems.

Attack Step 1.1

==

Name: Command and Scripting Interpreter: Python as used by the malware
Description: The malicious code of MuddyWater malware is encapsulated within a PyInstaller executable, thereby obscuring its Python scripting language foundation. This encapsulation technique hinders detection by conventional antivirus software.

MITRE Technique

ID: T1059.006
Name: Command and scripting interpreter: python
Description: Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the python.exe interpreter) or via scripts (.py) that can be written and distributed to different systems. Python code can also be compiled into binary executables.
More info: https://attack.mitre.org/techniques/T1059/006

Indicators

No indicators found.

Milestone 2

Pre-Conditions:

- The victim's operating system has a registry.
- A Windows operating system is present.
- The malware has knowledge of the appropriate registry key path (HKCU\Software\Microsoft\Windows\CurrentVersion\Run).

Post-Conditions:

- Small Sieve Python script files.
- Compromised system with persistent malware infection.
- Altered system text encoding.
- Embedded Windows Script Files (.wsf) installed in the startup folder.
- Data exfiltration and potential loss of sensitive information.
- System instability and performance degradation.
- Modified system proxy settings.
- LaZagne logs and credential dumps.
- Network connections to command-and-control servers.
- Registry run keys added for persistence.
- Reshuffled hexadecimal strings in .wsf files.
- Increased vulnerability to further attacks.

Attack Step 2.1

==
Name: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder as used by the malware
Description: Persistence on the victim's system is established by the "Small Sieve" malware through the addition of itself to the HKCU\Software\Microsoft\Windows\CurrentVersion\Run key within the Windows Registry. A new entry, designated as either "SystemTextEncoding" or potentially "OutlookMicrosift", is created within this key. Upon system boot or user logon, any programs listed in the Run key are automatically executed by Windows, resulting in the malware's execution on each instance.

MITRE Technique

ID: T1547.001
Name: Boot or logon autostart execution: registry run keys / startup folder
Description: Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level.
More info: https://attack.mitre.org/techniques/T1547/001

Indicators

- Registry value name: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\OutlookMicrosift
- Path: %AppData%\OutlookMicrosift\index.exe

Milestone 3

Pre-Conditions:

- An active internet connection for communication with Telegram and potential C2 servers.
- The malware Small Sieve is present on the victim's system.
- An obfuscated Base64 function is implemented within Small Sieve.
- A custom hex byte swapping encoding scheme is implemented within Small Sieve.
- The victim's system has an active internet connection.
- A system running a vulnerable application or process.
- The word "Platypus" is present in the command line arguments.
- Network connectivity to receive and execute commands from the attacker.
- The ability to execute code on the target system.
- The ability to modify file names and registry key names.
- A target system with a running instance of Small Sieve.
- Knowledge of legitimate Microsoft and Outlook naming conventions.
- The malware Small Sieve is present in the system.
- The attacker desires to avoid detection during casual inspection.

Post-Conditions:

- Network connections to unknown servers
- Encrypted data transfers
- Obfuscated code fragments
- Lateral movement within the network
- New executables and files in unexpected locations
- Registry modifications
- Compromised system integrity
- Increased vulnerability to further attacks
- Modified system settings
- Loss of sensitive information
- Altered log files with suspicious activity
- Data exfiltration
- Modified system processes
- Compromised system credentials
- Hidden configuration files
- Unusual network connections to Telegram servers
- Obfuscated command and control traffic logs
- LaZagne output files containing stolen credentials
- New executables in user directories
- Network traffic anomalies
- Files downloaded from malicious URLs
- System instability
- Malware persistence
- Presence of Small Sieve malware code
- Modified system registry entries for persistence
- Altered log files
- Modified system proxy settings
- Data exfiltration
- Compromised system credentials

- Network reconnaissance
- Modified system registry keys
- Unusual network traffic to command and control servers
- Suspicious process activity logs
- Modified Outlook data files (.txt)
- System instability
- Malware persistence
- Altered log files
- Obfuscated strings in memory
- Telegram API logs
- New files with malicious code
- Data exfiltration

Attack Step 3.1

==
Name: Obfuscated Files or Information as used by the malware
Description: MuddyWater malware implements a dual obfuscation strategy encompassing custom hexadecimal byte swapping and an obfuscated Base64 function. The malware's code and data undergo rearrangement via a non-standard hexadecimal pattern within the byte structure. Subsequently, an obfuscated variant of the Base64 encoding scheme is applied to further conceal the meaning and organization of the encoded information. This combined methodology presents significant challenges for security tools and analysts in deciphering the malware's code, commands, and communication with command-and-control (C2) servers.

MITRE Technique

ID: T1027.013
Name: Obfuscated files or information: encrypted/encoded file
Description: Adversaries may encrypt or encode files to obfuscate strings, bytes, and other specific patterns to impede detection. Encrypting and/or encoding file content aims to conceal malicious artifacts within a file used in an intrusion. Many other techniques, such as Software Packing, Steganography, and Embedded Payloads, share this same broad objective. Encrypting and/or encoding files could lead to a lapse in detection of static signatures, only for this malicious content to be revealed (i.e., Deobfuscate/Decode Files or Information) at the time of execution/use.
More info: https://attack.mitre.org/techniques/T1027/013

Indicators

- Small Sieve sample (Filename: gram_app.exe)
- Small Sieve sample (Filename: index.exe)
- Path: %LocalAppData%\MicrosoftWindowsOutlookDataPlus.txt
- Registry value name: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\OutlookMicrosift
- Path: %AppData%\OutlookMicrosift\index.exe

Attack Step 3.2

==
Name: Execution Guardrails as used by the malware
Description: The execution of MuddyWater's Small Sieve payload is contingent upon the provision of the specific command-line argument "Platypus." This parameter requirement constitutes an inherent safeguard, as the mere identification and activation of the payload file will not result in successful

infection without the stipulated trigger phrase.

MITRE Technique

ID: T1480.001
Name: Execution guardrails: environmental keying
Description: Adversaries may environmentally key payloads or other features of malware to evade defenses and constraint execution to a specific target environment. Environmental keying uses cryptography to constrain execution or actions based on adversary supplied environment specific conditions that are expected to be present on the target. Environmental keying is an implementation of Execution Guardrails that utilizes cryptographic techniques for deriving encryption/decryption keys from specific types of values in a given computing environment.
More info: https://attack.mitre.org/techniques/T1480/001

Indicators

- Small Sieve sample (Filename: gram_app.exe)
- Small Sieve sample (Filename: index.exe)
- Path: %AppData%\OutlookMicrosift\index.exe

Attack Step 3.3

==
Name: Masquerading: Match Legitimate Name or Location as used by the malware
Description: The malware, designated as Small Sieve, employs filename variations incorporating "Microsoft" (represented as "Microsift") and "Outlook." This nomenclature is strategically implemented to facilitate camouflage with legitimate Microsoft software applications and potentially circumvent detection during cursory file inspections.

MITRE Technique

ID: T1036.005
Name: Masquerading: match legitimate name or location
Description: Adversaries may match or approximate the name or location of legitimate files or resources when naming/placing them. This is done for the sake of evading defenses and observation. This may be done by placing an executable in a commonly trusted directory (ex: under System32) or giving it the name of a legitimate, trusted program (ex: svchost.exe). In containerized environments, this may also be done by creating a resource in a namespace that matches the naming convention of a container pod or cluster. Alternatively, a file or container image name given may be a close approximation to legitimate programs/images or something innocuous.
More info: https://attack.mitre.org/techniques/T1036/005

Indicators

- Path: %LocalAppData%\MicrosoftWindowsOutlookDataPlus.txt
- Registry value name: HKCU\Software\Microsoft\Windows\CurrentVersion\Run\OutlookMicrosift
- Path: %AppData%\OutlookMicrosift\index.exe

Milestone 4

Pre-Conditions:

- The malware Small Sieve is present in the system.
- Access to the Telegram Bot API.
- A connection to the internet is established.
- The malware Small Sieve is present in the system.
- An active internet connection for communication with the Telegram Bot API.
- The target system has an active internet connection.

Post-Conditions:

- Obfuscated log files containing communication with C2 servers.
- Network traffic to Telegram API endpoints.
- Modified system settings, including proxy configurations.
- Altered file timestamps and access logs.
- Compromised system with persistent malware infection.
- Presence of suspicious registry entries related to Small Sieve.
- Data exfiltration to external servers controlled by attackers.
- Small Sieve executable files on compromised systems.
- System instability and performance degradation.
- Increased risk of further attacks and lateral movement within the network.
- Telegram bot account activity logs.
- Files with names mimicking legitimate software (e.g., "Microsift", "Outlook").
- Hex byte swapped data in network traffic.
- Compromised systems with persistent malware infections.
- Data exfiltration from compromised systems.
- Increased risk of further attacks and exploitation.
- Modified system proxy settings.
- Files containing stolen data transferred to external servers.
- Unusual network connections to known malicious IP addresses.
- Obfuscated communication logs between infected systems and C2 servers via Telegram API.
- Altered system registry entries related to malware execution and persistence.
- New user accounts or processes created by the malware.
- Potential disruption of critical infrastructure or services.
- Small Sieve payload files on infected systems.

Attack Step 4.1

==
Name: Application Layer Protocol: Web Protocols as used by the malware
Description: Application layer communication for MuddyWater's "Small Sieve" malware is conducted via web protocols, namely HTTP and HTTPS. The Telegram Bot Application Programming Interface (API) is employed by the malware to establish a connection with its command-and-control server. All communication between "Small Sieve" and the C2 server is executed over HTTPS, ensuring data encryption during internet transmission. Commands directed towards the malware are prefixed with "/com[Bot ID]" for identification as legitimate requests from the authorized Telegram Bot. Data transmitted by "Small Sieve" upon beaconing to the C2 server includes the configured bot ID, the

username of the currently logged-in user, and the host's IP address.

MITRE Technique

ID: T1071.001
Name: Application layer protocol: web protocols
Description: Adversaries may communicate using application layer protocols associated with web traffic 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/T1071/001

Indicators

No indicators found.

Attack Step 4.2

==
Name: Data Encoding: Non-Standard Encoding as used by the malware
Description: MuddyWater's Small Sieve malware implements a custom hex byte swapping encoding scheme for obfuscation of tasking traffic. Data exchanged between the malware and its command-and-control server undergoes scrambling via an algorithm characterized by byte order manipulation within each data segment. This process renders the raw data unintelligible to security analysts without application of the corresponding decryption algorithm, thereby hindering comprehension of communication content.

MITRE Technique

ID: T1132.002
Name: Data encoding: non-standard encoding
Description: Adversaries may encode data with a non-standard data encoding system to make the content of command and control traffic more difficult to detect. Command and control (C2) information can be encoded using a non-standard data encoding system that diverges from existing protocol specifications. Non-standard data encoding schemes may be based on or related to standard data encoding schemes, such as a modified Base64 encoding for the message body of an HTTP request.
More info: https://attack.mitre.org/techniques/T1132/002

Indicators

No indicators found.