Enhanced Attack Report

Small Sieve

Generated on 2025-03-12

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:

- A Python interpreter is available on the victim system.
- A system running a compatible operating system with a Python interpreter installed.
- The malware must possess Python code to be executed.

Post-Conditions:

- Downloaded malicious executables (e.g., index.exe)
- Credential dumps (e.g., .txt, .csv)
- Obfuscated Python scripts (.py)
- Modified registry keys
- Suspicious network connections to command and control servers
- Malware persistence
- Network reconnaissance
- Obfuscated VBS scripts (.vbs)
- System instability
- Log files containing unusual activity
- Modified system files
- Data exfiltration
- Phishing email artifacts
- Telegram bot API interactions logs
- Compromised user accounts

Attack Step 1.1

==========

Name: Command and Scripting Interpreter: Python as used by the malware
Description: Python is utilized by MuddyWater within its malicious activities, as evidenced by the deployment of a PyInstaller-packed Python script. This script is obfuscated through the packaging process facilitated by PyInstaller, resulting in an executable file format that resembles conventional 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:

- Network connectivity may be required for initial deployment or communication with a command and control server.
- A Windows operating system is present.
- The target system is running a supported operating system.
- The malware has access to the registry.

Post-Conditions:

- Log entries indicating suspicious activity
- Modified or deleted files containing sensitive information
- Network traffic to command and control servers
- Malware persistence
- New files in user directories (malware binaries)
- Registry modifications (persistence keys)
- Execution of malicious code
- System instability
- Altered system settings
- Data exfiltration
- Compromised user accounts

Attack Step 2.1

==========

Name: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder as used by the malware
Description: A registry run key is utilized to initiate the execution of the "Small Sieve" malware. This involves the insertion of malicious code into a designated location within the Windows registry (HKCU\Software \Microsoft\ Windows\CurrentVersion \Run). Consequently, upon system boot or user logon, "Small Sieve" is automatically launched and commences its malicious operations.

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:

- The malware has access to program strings and Telegram credentials.
- The malware possesses the capability to implement custom hex byte swapping encoding schemes.
- The malware requires a method to encode information.
- The malware has the ability to utilize obfuscated Base64 functions.
- The word "Platypus" is passed to the Small Sieve payload on the command line.
- The ability to execute commands in a terminal or command prompt.
- Access to the command line interface.
- A system running the Small Sieve malware.
- Internet connectivity for retrieving information about legitimate names and locations.
- Access to a system capable of running the malware.
- Storage space on the system to store the malware code and data.
- The malware is designed to use legitimate names or locations.

Post-Conditions:

- Deleted or altered files containing sensitive information.
- Compromised system security and stability.
- Presence of malware signatures and indicators of compromise (IOCs).
- Spread of malware and ransomware infections.
- New user accounts created with elevated privileges.
- Unusual network traffic patterns and connections to suspicious IP addresses.
- Loss of sensitive data and intellectual property.
- Financial losses due to fraudulent activities.
- Log files containing malicious activity and unauthorized access attempts.
- Backdoors and remote access tools installed on compromised systems.
- Damage to reputation and trust.
- Modified system files and registry entries.
- Increased risk of data breaches and identity theft.
- Data exfiltration logs
- Compromised email communications
- Altered registry entries
- Backdoor executables
- Malware infection
- New user accounts created
- Modified system files
- Hidden files and folders
- Data theft
- Unusual network traffic logs
- System instability
- Suspicious process activity in event logs
- Disruption of services
- Loss of sensitive information
- Network intrusion
- Compromised user accounts

Attack Step 3.1

==========
Name: Obfuscated Files or Information as used by the malware
Description: The Small Sieve framework implemented by MuddyWater employs a multi-layered obfuscation technique to safeguard program strings and updated Telegram credentials. Hexadecimal byte swapping, executed according to a custom scheme, is utilized to rearrange the order of bytes within the strings, rendering them unintelligible. Subsequently, an obfuscated Base64 encoding function is applied to the already altered data, compounding the level of obscurity. This dual-layered obfuscation significantly impedes the ability of analysts to discern the original meaning of the strings, effectively concealing sensitive information such as program commands and communication details.

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 document presents an analysis of Tactics, Techniques, and Procedures (TTPs) employed by the threat actor MuddyWater. A unique payload designated as "Small Sieve" is identified, characterized by a specific execution requirement: the inclusion of the word "Platypus" as a command-line argument for functionality. This execution constraint aligns with the TTP category Execution Guardrails (T1480). Furthermore, the actor's demonstrated capability to disable local proxy settings (T1562.001) is highlighted as a technique employed to evade detection and analysis. The information presented is classified under the Traffic Light Protocol (TLP) rating of WHITE, indicating public availability. A product identifier, AA22-055A, is associated with a specific malware sample or toolset potentially linked to MuddyWater, facilitating identification and analysis.

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: Deceptive filenames are employed by malware authors to obfuscate their malicious intent. This involves utilizing filename variations that closely resemble legitimate software names, such as "Microsoft" (Microsift) and "Outlook," thereby increasing the likelihood of user acceptance during file browsing processes.

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 Telegram API must be accessible.
- Small Sieve malware has access to the necessary system resources to establish a network connection and communicate with the Telegram Bot API.
- An active internet connection is required for communication via HTTPS.
- An active internet connection for communication with Telegram API and C2 servers.
- Access to the necessary files and resources within the compromised system.
- The target machine has an active internet connection.
- The malware Small Sieve is present in the system.

Post-Conditions:

- Backdoor programs installed
- Altered registry settings
- Data exfiltration logs
- Reputational damage
- Unusual log entries
- Compromised email communications
- Data breaches and theft
- New user accounts created
- Financial losses
- Network traffic to malicious servers
- Hidden files and folders
- System compromise and control
- Network disruption and instability
- Legal repercussions
- Deleted or modified evidence
- Modified system files
- Encrypted data transfers
- Modified system registry entries
- Network traffic to command and control servers
- Malware persistence
- New user accounts created
- Suspicious files with unusual names or extensions
- Backdoors installed on the system
- Unusual DNS queries
- Network reconnaissance
- Data theft
- Altered log files with deleted or modified entries
- System instability
- Command and control infrastructure established
- Leftover malware code fragments
- Modified system configurations
- Compromised user accounts

Attack Step 4.1

==========
Name: Application Layer Protocol: Web Protocols as used by the malware
Description: Communication between the malware and its command-and-control server is facilitated via the Telegram API over HTTPS. This interaction falls under the classification of "Command and Control: Application Layer Protocol: Web Protocols." HTTPS encryption is employed to secure the transmission of messages, thereby hindering interception and analysis.

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 utilizes a custom hex byte swapping encoding scheme to obfuscate tasking traffic. This technique, classified as Data Encoding: Non-Standard Encoding, involves modifying the standard representation of data through a specific pattern of hexadecimal byte swapping. This process renders the traffic more difficult to interpret by security tools and analysts engaged in tracking malicious activity.

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.