Enhanced Attack Report

Goofy Guineapig

Generated on 2025-03-11

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: Goofy Guineapig
Malware Description: The Goofy Guineapig loader is a UPX packed, trojanised NSIS Firefox installer. Once extracted, it masquerades as a Google update component. Goofy Guineapig maintains persistence as a Windows service. Goofy Guineapig provides a framework into which additional plugins may be loaded. The backdoor supports multiple communications methods, including HTTP, HTTPS and KCP. The configuration is embedded in the binary, and the configuration for the binary analysed results in command and control communications occurring over HTTPS. Many defence evasion techniques are implemented throughout execution.

Quick Overview

Milestone 1
1. Create or Modify System Process: Windows Service as used by the malware (T1543.003)

Milestone 2
1. Masquerading: Match Legitimate Name or Location as used by the malware (T1036.005)
2. Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware (T1497.003)
3. Virtualization/Sandbox Evasion: System Checks as used by the malware (T1497.001)
4. Virtualization/Sandbox Evasion: User Activity Based Checks as used by the malware (T1497.002)
5. Obfuscated Files or Information: Software Packing as used by the malware (T1027.002)
6. Deobfuscate/Decode Files or Information as used by the malware (T1140)
7. Hide Artifacts: Hidden Window as used by the malware (T1564.003)
8. Indicator Removal on Host: File Deletion as used by the malware (T1070.004)
9. Hijack Execution Flow: DLL Side-Loading as used by the malware (T1574.002)
10. Process Injection: Process Hollowing as used by the malware (T1055.012)
11. Signed Binary Proxy Execution: Rundll32 as used by the malware (T1218.011)

Milestone 3
1. System Information Discovery as used by the malware (T1082)

Milestone 4
1. Application Layer Protocol: Web Protocols as used by the malware (T1071.001)
2. Fallback Channels as used by the malware (T1008)
3. Non-Standard Port as used by the malware (T1571)

Milestone 1

Pre-Conditions:

- Administrative privileges.
- The malware has successfully executed its initial code.
- The malware requires access to system functions for creating and modifying services.
- Access to the Windows Service Control Manager API.
- Windows operating system.

Post-Conditions:

- Suspicious files created or modified
- Execution of malicious code
- New Windows service installed
- Network traffic to command and control servers
- Unusual process activity logs
- Data theft
- Remote access by attackers
- Altered system event logs
- Modified registry entries
- YARA rule match in security logs
- Hidden files or folders
- System instability
- Compromised system security

Attack Step 1.1

==========

Name: Create or Modify System Process: Windows Service as used by the malware
Description: A persistent infection is established by Goofy Guineapig through the creation and modification of a Windows service. The malware utilizes standard Windows API calls to generate a new service on the infected machine, employing a plausible name to evade detection. Service configuration parameters, including startup type, display name, and description, are meticulously set by the malware, potentially mimicking legitimate service attributes. The malware's executable code is deployed as the service binary, enabling continuous background execution even after user logoff or system reboot. A persistence loop is established within the malware's code, running as a service, to monitor for specific triggers such as scheduled times or network events. Upon activation of these triggers, the malicious payload is executed, encompassing activities like communication with command-and-control servers, data exfiltration, or initiation of further attacks. The effectiveness of this approach stems from the deep integration of Windows services within the operating system, designed for continuous operation. Additionally, services often possess elevated privileges, granting malware capabilities otherwise inaccessible.

MITRE Technique

ID: T1543.003
Name: Create or modify system process: windows service

Description: Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions. Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry.
More info: https://attack.mitre.org/techniques/T1543/003

Indicators

- A file named "tmp.bat" is located in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".

Milestone 2

Pre-Conditions:

- Access to system files and processes.
- The malware has access to information about legitimate Google update components.
- Ability to modify file metadata.
- The malware is capable of modifying its own name or location.
- The Goofy Guineapig malware is running.
- Access to the CPU timestamp counter.
- The system time register is accessible.
- The malware is executing.
- A Windows operating system.
- The system has logical processors.
- The system has a physical memory size.
- The malware is running.
- Access to Windows APIs for retrieving system information (e.g., physical memory size, disk size, number of logical processors).
- Operating System with access to system information (e.g., Windows)
- Access to process information on the system.
- The malware is running on a system.
- Processes are running on the system.
- The malware executable exists.
- A legitimate NSIS installer is accessible.
- UPX packing tool is available.
- The ability to analyze and interpret binary code.
- Understanding of stack-based string obfuscation methods.
- The obfuscated strings within the malware exist.
- The malware binary is present.
- A system capable of executing the malware binary.
- Knowledge of XOR encryption and decryption techniques.
- A process with a specific name is running.
- The malware is running.
- A window exists with relevant strings.
- A command execution capability exists within the environment.
- Access to the ProgramData directory is available.
- Access to the initial download location is available.
- The malware has successfully downloaded and extracted its files.
- The malware has determined the initial download location.
- A file system is present.
- The malware has identified a legitimate-looking directory.
- A legitimate executable is present on the system.
- The malicious DLL exists.
- Network connectivity is available for downloading the malicious DLL.
- The Goofy Guineapig loader is active.
- File system access is available to write and execute files.
- A Windows operating system is present.
- The malware has the capability to read and modify memory.
- A target system running Windows.
- The malware has downloaded a payload executable.

- A running instance of the dllhost.exe process exists.
- url.dll is available.
- exe is available.
- The legitimate executable is loaded into memory.

Post-Conditions:

- Unusual process activity and resource usage
- YARA rule matches indicating the presence of Goofy Guineapig strings
- Modified system registry entries
- Data theft
- Remote access by attackers
- New files created in various directories (e.g., temporary folders, user profiles)
- Modified system configuration files
- Network connections to suspicious IP addresses
- Evidence of data exfiltration (e.g., unusual file transfers)
- Execution of malicious plugins
- System instability
- Compromised system security
- Altered system logs (event logs, application logs)
- Presence of Goofy Guineapig malware code fragments in memory or on disk
- Presence of malware artifacts in memory dumps
- Modified system registry entries
- System compromise
- Data exfiltration
- Log entries indicating process creation and execution of malicious code
- Malware persistence
- Unusual network traffic to suspicious IP addresses
- Evidence of data transfer via encrypted channels
- Potential for further malicious activity
- New files created in various directories (e.g., %TEMP%, AppData)
- Network communication with C2 server
- Altered system configurations
- Modified system settings
- Network connections to command and control servers
- System compromise
- Data exfiltration
- New registry entries
- File downloads and uploads
- Unusual process creation and termination events
- Malware infection
- Modified system files
- Process hijacking
- Encrypted data remnants
- Persistence mechanisms (e.g., scheduled tasks, boot-sector infections)
- Resource depletion
- Log file entries indicating suspicious activity
- Network traffic to command and control servers
- Modified system registry entries
- System compromise
- Data exfiltration
- Altered log files

- Potential for further malware execution
- New files created in various directories
- Persistence on the system
- YARA rule matches in security logs
- Process hollowing artifacts
- Network traffic to command and control servers
- Altered file timestamps
- System compromise
- Data exfiltration
- New registry entries
- Potential for further malware execution
- Unusual process creation and termination events
- Modified system files
- Log files containing suspicious activity
- Persistence established
- Hidden or encrypted files
- Network communication with C2 server
- YARA rule files
- Hidden files and folders
- Network traffic to malicious servers
- Modified system registry entries
- Leftover shellcode fragments
- Data theft
- Remote access by attackers
- Installation of additional malware
- Altered system files
- Unusual process activity
- Firewall rules changes
- System instability
- Obfuscated code remnants in memory
- Log entries indicating suspicious activity
- Compromised system security
- DNS requests to unknown domains
- New user accounts created
- Persistence established on the system
- Hidden files and folders
- System compromise
- Data exfiltration
- New registry entries
- Unusual process activity logs
- Named pipe created with hashed computer name
- Remote access granted to attacker
- Modified system files
- Modified DLLhost.exe process
- Potential for further malware infections
- Network connections to malicious servers
- YARA rule matches in security logs
- Altered system configurations
- Data exfiltration and/or manipulation.
- Altered log files indicating unusual activity.
- File fragments containing Goofy Guineapig strings.
- System instability and performance degradation.
- Hidden files and directories in ProgramData.

- New processes running with suspicious names or behavior.
- Deleted original download location files.
- Modified system registry entries.
- Compromised system with persistent malware infection.
- Increased vulnerability to further attacks.
- YARA rule matches in security logs.
- Network connections to malicious command-and-control servers.
- Network connections to command-and-control servers.
- Log files containing suspicious activity.
- UPX packed NSIS installer file.
- Modified Firefox installation package.
- System instability and performance degradation.
- Presence of Goofy Guineapig loader shellcode remnants.
- Malicious DLL file in the ProgramData directory.
- Data exfiltration and/or data manipulation.
- Compromised system with persistent malware infection.
- Increased vulnerability to further attacks.
- Modified registry entries related to the malicious process.
- Altered system files and configurations.
- Execution of malicious code
- Modified system registry entries
- Network traffic to command-and-control servers
- System compromise
- Data exfiltration
- Firewall logs showing unauthorized connections
- Modified system configuration files
- Malware persistence
- Log entries indicating unusual activity
- Altered process lists
- Process injection
- Shadow copies containing malware artifacts
- New files created with suspicious names and content
- Network traffic to command and control servers
- Remote code execution
- Modified system registry entries
- Altered process logs
- Hidden or encrypted data files
- Data exfiltration
- Modified firewall rules
- Unusual system resource usage patterns
- New files created with malicious content
- Potential for further malware infections
- Persistence on the system
- Compromised system integrity
- Suspicious DNS queries

Attack Step 2.1

==========

Name: Masquerading: Match Legitimate Name or Location as used by the malware
Description: The malware Goofy Guineapig employs masquerading techniques to evade detection by assuming the guise of legitimate software installations. It specifically presents itself as Firefox Installer

and Google Updater, leveraging user trust in these widely recognized applications. By mimicking their graphical elements and nomenclature, Goofy Guineapig induces users to execute the malicious file under the assumption that it is legitimate software, thereby circumventing security measures that may otherwise flag suspicious files based on their actual name or origin.

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

- The file "tmp.bat" is located in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".

Attack Step 2.2

==========
Name: Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware
Description: The Goofy Guineapig malware implements a Time-Based Evasion technique to circumvent detection within virtualized or sandboxed environments. This technique involves an initial system time register reading followed by a delay exceeding 100 milliseconds. Subsequently, the system time register is read again. A comparison of the timestamps is then performed. If the difference exceeds an anticipated range, indicative of artificially accelerated time progression characteristic of sandbox environments, malware execution is terminated.

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

- The malware utilizes time-based evasion techniques.

Attack Step 2.3

==========
Name: Virtualization/Sandbox Evasion: System Checks as used by the malware

Description: System checks are conducted by the Goofy Guineapig malware to ascertain the presence of a virtualized or sandboxed environment. These checks encompass an analysis of the system's hard drive size, physical memory size, and logical processor count. Deviations in these parameters from established norms for real systems may indicate a sandboxed or virtualized environment, prompting the termination of Goofy Guineapig's execution to preclude the exposure of malicious activities within a controlled testing setting.

MITRE Technique

ID: T1497.001
Name: Virtualization/sandbox evasion: system checks
Description: Adversaries may employ various system checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
More info: https://attack.mitre.org/techniques/T1497/001

Indicators

- The system checks for virtualization software presence.

Attack Step 2.4

==========
Name: Virtualization/Sandbox Evasion: User Activity Based Checks as used by the malware
Description: Process name monitoring is employed by Goofy Guineapig malware to identify potentially hostile environments. The malware scans active processes for keywords indicative of debugging or reverse engineering activities. These keywords include "dbg," "debug," and "ida." Upon detection of these keywords, the malware terminates its own execution, thereby evading analysis in sandboxed environments or during debugging sessions. This behavior is predicated on the common practice within the software development and security research communities of utilizing debugging tools and naming processes accordingly during malware analysis.

MITRE Technique

ID: T1497.002
Name: Virtualization/sandbox evasion: user activity based checks
Description: Adversaries may employ various user activity checks to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.
More info: https://attack.mitre.org/techniques/T1497/002

Indicators

- The process name is "tmp.bat".
- The file path includes "C:\ProgramData\GoogleUpdate\GoogleUpdate\tmp.bat".

Attack Step 2.5

==========
Name: Obfuscated Files or Information: Software Packing as used by the malware
Description: Goofy Guineapig malware is obfuscated through software packing utilizing UPX (Ultimate Packer for X86). The malicious code undergoes compression by UPX to minimize its size and impede analysis. Furthermore, it is integrated within a legitimate Windows Installer (.exe) constructed using the Nullsoft Scriptable Install System (NSIS), thereby enhancing its perceived legitimacy. Upon execution of the installer, UPX decrypts and decompresses the Goofy Guineapig code in memory prior to its execution. This process renders static analysis of the packed file challenging as the actual malicious instructions are not readily discernible within the original file.

MITRE Technique

ID: T1027.002
Name: Obfuscated files or information: software packing
Description: Adversaries may perform software packing or virtual machine software protection to conceal their code. Software packing is a method of compressing or encrypting an executable. Packing an executable changes the file signature in an attempt to avoid signature-based detection. Most decompression techniques decompress the executable code in memory. Virtual machine software protection translates an executable's original code into a special format that only a special virtual machine can run. A virtual machine is then called to run this code.
More info: https://attack.mitre.org/techniques/T1027/002

Indicators

- The file "tmp.bat" is located in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".
- The file "config.dat" has a SHA-256 hash of "3a1af09a0250c602569d458e79db90a45e305b76d8423b81eeeca14c69847b81c".
- The file "GoogUpdate" is located in the directory "C:\ProgramData\GoogleUpdate".

Attack Step 2.6

==========
Name: Deobfuscate/Decode Files or Information as used by the malware
Description: The malware Goofy Guineapig employs two primary techniques for deobfuscation and decoding: Single Byte XOR and Subtraction. Stack-based strings within the binary are encrypted via the XOR operation utilizing a single byte key. This process renders the characters unreadable without the application of the inverse XOR operation. Subtraction is similarly employed to obfuscate certain code segments or data. A predetermined numerical value is subtracted from each character's ASCII representation, rendering the information unintelligible without the reversal of the subtraction process. These encrypted strings are embedded within the malware's own code and structure. Upon execution reaching a point requiring a decoded string, the program applies the corresponding XOR or subtraction algorithm utilizing the predetermined key. This reveals the original, meaningful text necessary for functions such as command execution, data transmission, or communication with command-and-control servers. These techniques are commonly utilized in malware due to their relative simplicity and effectiveness in concealing malicious intent. Analysis and comprehension of these obfuscation methods are crucial for security researchers and analysts to effectively detect, reverse engineer, and mitigate the threat posed by Goofy Guineapig.

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 "tmp.bat" is located at "C:\ProgramData\GoogleUpdate\GoogleUpdate".
- The malware uses the User Agent string "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.71 Safari/537.36".

Attack Step 2.7

==========
Name: Hide Artifacts: Hidden Window as used by the malware
Description: Process hollowing is employed by Goofy Guineapig malware against instances of dllhost.exe. This technique involves the identification of a running dllhost.exe process followed by the emptying of its memory contents. Subsequently, malicious shellcode or payload is injected into the vacated memory space. The process name remains unchanged, and initial behavior appears consistent with legitimate dllhost.exe functionality. This obfuscation hinders detection by security tools and analysts reliant on process name examination or code signature analysis. The hijacking of a system process grants the malware increased stability and persistence within the target machine.

MITRE Technique

ID: T1564.003
Name: Hide artifacts: hidden window
Description: Adversaries may use hidden windows to conceal malicious activity from the plain sight of users. In some cases, windows that would typically be displayed when an application carries out an operation can be hidden. This may be utilized by system administrators to avoid disrupting user work environments when carrying out administrative tasks.
More info: https://attack.mitre.org/techniques/T1564/003

Indicators

- The process name is "tmp.bat".
- A file named "config.dat" exists.

Attack Step 2.8

==========
Name: Indicator Removal on Host: File Deletion as used by the malware
Description: The Goofy Guineapig malware utilizes "Indicator Removal on Host: File Deletion" to obfuscate its presence. Upon initial execution from a temporary download location, the malware relocates downloaded files to directories commonly associated with system software or applications. Subsequently, the original files are deleted from the initial download directory, effectively eliminating traces of the malware's entry point. This tactic serves to conceal malware activity and evade detection

by security tools and analysts.

MITRE Technique

ID: T1070.004
Name: Indicator removal: file deletion
Description: Adversaries may delete files left behind by the actions of their intrusion activity. Malware, tools, or other non-native files dropped or created on a system by an adversary (ex: Ingress Tool Transfer) may leave traces to indicate to what was done within a network and how. Removal of these files can occur during an intrusion, or as part of a post-intrusion process to minimize the adversary's footprint.
More info: https://attack.mitre.org/techniques/T1070/004

Indicators

- A file named "tmp.bat" was located in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".
- The file "config.dat" was deleted from the system.

Attack Step 2.9

==========
Name: Hijack Execution Flow: DLL Side-Loading as used by the malware
Description: The "Hijack Execution Flow: DLL Side-Loading" technique employed by Goofy Guineapig involves the exploitation of legitimate software processes for malicious payload delivery. A malicious DLL is introduced alongside a legitimate installer package, leveraging the established practice of executable loading of additional DLLs for functionality. This technique exploits the trust associated with signed executables, such as GoogleUpdate.exe, to execute the malicious DLL without raising suspicion from security software. Upon execution, the malicious DLL verifies its own format and triggers its entry point, initiating the core functionality defined by the attacker. A specific function, "plugin_run," is likely responsible for executing the intended malicious actions.

MITRE Technique

ID: T1574.002
Name: Hijack execution flow: dll side-loading
Description: Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to DLL Search Order Hijacking, side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s).
More info: https://attack.mitre.org/techniques/T1574/002

Indicators

- The file "tmp.bat" is located in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".
- The malware utilizes a DLL side-loading technique.

Attack Step 2.10

==========
Name: Process Injection: Process Hollowing as used by the malware

Description: Process injection via process hollowing is executed by Goofy Guineapig malware against the dllhost.exe binary. Malicious code is retrieved from a Command and Control (C2) server. The malicious code replaces the existing memory space of the identified dllhost.exe process, effectively "hollowing it out." Execution of the injected malicious code occurs within the hollowed dllhost.exe process, mimicking legitimate behavior and circumventing security measures that target newly spawned processes.

MITRE Technique

ID: T1055.012
Name: Process injection: process hollowing
Description: Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process.
More info: https://attack.mitre.org/techniques/T1055/012

Indicators

- The process name is "tmp.bat".
- The file path includes "C:\ProgramData\GoogleUpdate\GoogleUpdate\tmp.bat".

Attack Step 2.11

==========
Name: Signed Binary Proxy Execution: Rundll32 as used by the malware
Description: Rundll32 is utilized by Goofy Guineapig malware for persistence through a multi-stage process. A legitimate executable is deployed concurrently with a malicious DLL. A rundll32 command is constructed to leverage the URLMON_OpenURL function within url.dll, a system DLL responsible for URL protocol execution. The crafted command, when executed, triggers the loading and execution of the malicious DLL by exploiting the specified function. The malicious DLL, operating with elevated privileges if applicable, facilitates communication with a command-and-control server, data exfiltration, additional malware installation, and arbitrary command execution. This technique is favored due to rundll32's inherent low suspicion as a common Windows utility, enabling function hijacking through URLMON_OpenURL for reduced detection, and facilitating DLL injection into legitimate processes for heightened permissions and increased removal complexity.

MITRE Technique

ID: T1218.011
Name: System binary proxy execution: rundll32
Description: Adversaries may abuse rundll32.exe to proxy execution of malicious code. Using rundll32.exe, vice executing directly (i.e. Shared Modules), may avoid triggering security tools that may not monitor execution of the rundll32.exe process because of allowlists or false positives from normal operations. Rundll32.exe is commonly associated with executing DLL payloads (ex: rundll32.exe {DLLname, DLLfunction}).
More info: https://attack.mitre.org/techniques/T1218/011

Indicators

- The file "tmp.bat" is located at "C:\ProgramData\GoogleUpdate\GoogleUpdate".

Milestone 3

Pre-Conditions:

- The victim machine has an operating system.
- The malware is running on a victim machine.
- Network connectivity is required for communication with the Command and Control (C2) server.
- Access to the Windows Management Instrumentation (WMI) is available.
- Access to relevant Windows APIs is available.
- A Windows operating system is present.
- HTTPS protocol is used for communication with the C2 server.

Post-Conditions:

- WMI queries indicating information gathering about the system.
- Data exfiltration from infected machine.
- Potential for further malicious activity execution.
- Modified system files with malicious code injections.
- Obfuscated strings and encoded data within system files.
- Unusual network traffic to command-and-control servers.
- YARA rule matches identifying Goofy Guineapig components.
- Traces of communication protocols like HTTP, HTTPS, and KCP.
- Compromised system with persistent backdoor access.
- System instability and performance degradation.
- Log entries indicating suspicious process activity and file modifications.
- New files created in various directories (e.g., malware binaries, configuration files).
- Modified Windows registry entries.

Attack Step 3.1

==========

Name: System Information Discovery as used by the malware
Description: System Information Discovery is conducted by the Goofy Guineapig malware through the collection of various details pertaining to the infected machine. These data points are subsequently embedded within an obfuscated "Authorization" string transmitted in each communication (C2 packet) over HTTPS. Data acquisition is achieved through the utilization of COM and WMI, enabling the retrieval of information such as the operating system caption and antivirus product display name. Additional system details, including adapter information, host and hostname, and computer name, are obtained through direct invocation of relevant Windows APIs. The collected information undergoes a process of obfuscation to mitigate detection by security measures. This "Authorization" string, containing the disguised system data, is incorporated into each HTTP header directed towards the Command and Control (C2) server. Consequently, the obfuscated "Authorization" string accompanies every C2 communication, effectively transmitting sensitive system information to the attackers without immediately raising suspicion.

MITRE Technique

ID: T1082
Name: System information discovery

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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.
More info: https://attack.mitre.org/techniques/T1082/

Indicators

- The system's operating system is Windows NT 6.1.
- The system architecture is WOW64.

Milestone 4

Pre-Conditions:

- A functioning internet connection.
- The malware is running.
- The infected machine has an internet connection.
- HTTPS protocol support.
- The ability to send and receive HTTP GET and POST requests.
- Static[.]tcplog[.]com domain resolvability.
- A network connection is available.
- The KCP protocol is accessible.
- The malware is running.
- The embedded configuration string contains "UDP" or "udp".
- An embedded configuration string exists within the malware binary.

Post-Conditions:

- Modified system registry entries
- System compromise
- Data exfiltration
- Network traffic to C2 server (static[.]tcplog[.]com)
- New files created in various directories
- Presence of Goofy Guineapig malware code fragments
- Potential for further malicious activity
- Unusual process activity and memory usage patterns
- Persistence on the system
- Logs containing suspicious activity
- Altered system configurations
- Remote access by attacker
- Unusual process activity and resource usage
- Modified system registry entries
- New files created with malicious code
- Data theft
- Network traffic to suspicious IP addresses
- Altered log files
- Unique identifier based on concatenated host information and MD5 hash
- System instability
- Evidence of data exfiltration
- Compromised system security
- Hardcoded configuration strings within malware binaries
- Presence of YARA rule matches in system logs
- New files created in various directories (e.g., DLLs, configuration files)
- Modified system registry entries
- Evidence of process creation and execution with suspicious names
- Presence of malicious code within legitimate executables
- Data theft
- Remote access by attackers
- Malware persistence
- Unusual disk activity patterns

- Modified system event logs
- System instability
- Compromised system security
- Altered network traffic logs (unusual connections, non-standard ports)
- Leftover communication data in temporary files or memory dumps

Attack Step 4.1

==========
Name: Application Layer Protocol: Web Protocols as used by the malware
Description: HTTPS is employed by the malware for establishing communication with its command and control server. Obfuscated data pertaining to the infected machine is incorporated into HTTP headers, specifically within the "Authorization" field of requests transmitted to the C2 server.

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

- The URL HTTPS://static.tcplog.com is accessed.
- A User Agent string indicating Chrome/54.0.2840.71 Safari/537.36 is used.

Attack Step 4.2

==========
Name: Fallback Channels as used by the malware
Description: The malware's communication channels are configurable and include UDP, KCP, and direct socket communications. UDP is employed for connectionless communication suitable for applications prioritizing low latency. KCP, a protocol designed for real-time communication, offers speed, security, and efficiency. Direct socket connections provide flexibility but may be more complex to implement. The specific channel utilized is determined by an embedded configuration string within the malware, enabling attackers to dynamically select the most appropriate method based on environmental factors and operational objectives.

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 file tmp.bat is located in the directory C:\ProgramData\GoogleUpdate\GoogleUpdate.

Attack Step 4.3

==========
Name: Non-Standard Port as used by the malware
Description: The utilization of a non-standard HTTPS port (4443) for communication is employed by the malware Goofy Guineapig. This practice deviates from the standard HTTPS port (443) conventionally utilized by legitimate connections. The implementation of this non-standard port constitutes an attempt to circumvent detection and analysis by security tools that predominantly focus on traffic traversing the standard HTTPS port.

MITRE Technique

ID: T1571
Name: Non-standard port
Description: Adversaries may communicate using a protocol and port pairing that are typically not associated. For example, HTTPS over port 8088 or port 587 as opposed to the traditional port 443. Adversaries may make changes to the standard port used by a protocol to bypass filtering or muddle analysis/parsing of network data.
More info: https://attack.mitre.org/techniques/T1571/

Indicators

No indicators found.