Enhanced Attack Report

Goofy Guineapig

Generated on 2025-03-21

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:

- The malware has access to necessary system functions.
- The malware has successfully executed its initial code.
- Windows operating system environment.
- Access to system APIs for creating and modifying services.
- The ability to write to system files.

Post-Conditions:

- UPX packed executable file.
- Modified Firefox installer file.
- Files downloaded or uploaded by the malware.
- Changes in system performance and resource usage.
- Unusual process names and behavior.
- Compromised system with persistent backdoor.
- Log files containing malware activity.
- Modified system registry entries.
- Potential for further malware infections.
- Data exfiltration to C2 server.
- Windows service created by the malware.

Attack Step 1.1

==
Name: Create or Modify System Process: Windows Service as used by the malware
Description: A new Windows service is created by the malware to achieve persistence. The malware utilizes functions from the Windows API to register a unique service on the target system. Specific parameters are configured for the service, including a display name and description designed to appear innocuous. The executable path is set to point to either the malicious Goofy Guineapig DLL or a dropper that loads it. The startup type is configured as automatic, ensuring the service initiates upon each system boot. Following registration, the newly created service is initiated, causing the execution of Goofy Guineapig's code continuously in the background.

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:

- The malware has access to a legitimate GoogleUpdate.exe executable.
- A computer system running Windows.
- The malware has access to legitimate Firefox installation files.
- The ability to execute files on the system.
- The Goofy Guineapig malware is executing.
- The system has a functioning time register.
- The malware is running.
- The system has physical memory.
- The system has a disk drive.
- A Windows operating system.
- The system has multiple logical processors.
- The malware is running.
- Access to the system's hardware information via Windows APIs.
- Access to system information, including running processes and their names.
- A running instance of the Goofy Guineapig malware.
- The system has processes running.
- The malware is running.
- Access to system time.
- The malware code exists.
- A suitable environment for compiling and executing code exists.
- UPX packing tool is available.
- Stack-based strings within the malware binary exist.
- The Goofy Guineapig malware binary is present.
- Capability to perform single byte XOR and subtraction operations.
- The Goofy Guineapig malware is present and active.
- System resources are available for the malware to manipulate windows.
- A target window exists.
- A window needs to be hidden.
- The malware is running.
- The malware has successfully downloaded and extracted its files to a temporary location.
- The malware has knowledge of the directory containing the extracted Firefox files.
- The malware has the ability to read and write files.
- The malware has established persistence mechanisms.
- A file system is available.
- A legitimate executable is installed.
- The Goofy Guineapig loader is present.
- A malicious DLL exists.
- A system with a running operating system.
- The ability to execute code.
- A running instance of Windows operating system.
- The malware binary is present and executable.
- The malware has downloaded the necessary payload data from the C2 server.
- A target process named dllhost.exe is running.
- Network connectivity to the C2 server.
- rundll32.exe is available on the system.
- exe`.

- A Windows operating system is present.
- url.dll is available on the system.
- Access to the file system is available to the malware.

Post-Conditions:

- Child processes are spawned based on command IDs.
- Trojanized Firefox installation package files.
- UPX packed NSIS installer file.
- Process activity logs showing Goopdate.dll loading and execution.
- File system modifications indicating dropped files and modified locations.
- Legitimate GoogleUpdate.exe executable.
- Logs related to service creation and execution.
- A Windows service is running persistently.
- C2 communications are established using HTTPS and RC4 encryption.
- Increased risk of further malware infections.
- Modified system files or registry entries.
- Log entries indicating suspicious activity and process executions.
- Potential data exfiltration and theft.
- Compromised system with persistent backdoor access.
- Network connections to C2 server using HTTPS protocol.
- Modified "dllhost.exe" process.
- UPX packed executable file ("Goofy Guineapig loader").
- Network communication
- Temporary files
- New process instances
- Data alteration
- Modified files
- Log entries
- Network traffic logs
- System modifications
- Registry changes
- Process creation and execution
- Hidden processes running
- Compromised system
- DNS queries for malicious domains
- Lateral movement
- Persistence established
- Network traffic to command and control server
- Altered log files
- New files created in various directories
- Data exfiltration
- Modified system registry entries
- Unusual system resource usage patterns
- Modified Firefox installer
- Compromised system
- Persistence on the system
- Network connections to C2 server
- Potential for further malware infections
- UPX packed files
- "Goopdate.dll" DLL file
- Log entries related to process execution and network activity

- Encrypted data files
- Data exfiltration
- Modified registry settings
- Unusual system resource usage patterns
- New processes running with suspicious names.
- UPX packed executable file.
- System instability and performance degradation.
- Potential for further malicious activity.
- Deleted or modified original files.
- Modified system registry entries.
- Compromised system with malware installed.
- Obfuscated configuration strings.
- XOR encrypted binary embedded in shellcode.
- Altered system files.
- Network connections to hardcoded C2 server IPs and domains.
- Data exfiltration to C2 server.
- RC4 encrypted C2 communications.
- Logs containing unusual activity and requests.
- Modified system files.
- Registry entries related to persistence mechanism.
- New files containing malicious code.
- Log entries indicating suspicious activity.
- Traces of deleted tasking requests (0x29 and 0x64).
- Potential data exfiltration to C2 server.
- Modified dllhost.exe process.
- exe and url.dll.
- Deleted initial download location files.
- System instability and performance degradation.
- Modified Firefox installer file.
- Compromised system with persistent backdoor.
- Altered system processes and running services.
- Registry modifications related to persistence.
- Log entries indicating suspicious activity.
- Potential for further malware infections.
- Hidden malicious files in ProgramData directory.
- Data exfiltration to C2 server.
- System instability and performance degradation.
- Altered registry entries related to malware execution and persistence.
- Malicious DLL (Goopdate.dll) in the system.
- UPX packed NSIS installer file.
- Compromised system with persistent backdoor access.
- New service created for persistence.
- Logs indicating process injection and hollowing techniques.
- Potential for further malware infections.
- Data exfiltration to C2 server.
- Modified legitimate executable files (e.g., GoogleUpdate.exe).
- Modified dllhost.exe process memory.
- Compromised system with persistent backdoor access.
- Altered registry entries for persistence.
- Files downloaded from the C2 server.
- Network connections to the C2 server (HTTPS).
- Traces of deleted tasking data.
- Data exfiltration to C2 server.

- Logs of system activity, including process injections and file modifications.
- Potential for further malware infections and attacks.
- UPX packed executable file ("Goofy Guineapig loader").
- UPX packed executable file.
- System instability and performance degradation.
- Modified dllhost.exe binary.
- Compromised system with persistent backdoor.
- Potential for further malware infections.
- New Windows service.
- Data exfiltration to C2 server.
- Files downloaded from C2 server.
- Log files containing suspicious activity.
- Altered system registry entries.

Attack Step 2.1

==

Name: Masquerading: Match Legitimate Name or Location as used by the malware
Description: The malware Goofy Guineapig is presented as both a Firefox installer and a Google Update component through the utilization of packaging techniques and file naming conventions that mimic legitimate software. Malicious code is encapsulated within an NSIS installer file, commonly associated with Firefox installations, to exploit user trust in established software installers. File names are likely designed to resemble those of legitimate Firefox or Google Update components, further deceiving users and security tools during initial scans. This approach aims to circumvent detection by appearing as innocuous updates or software installations, thereby inducing users to execute the malicious code unknowingly.

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: Time Based Evasion is employed by the Goofy Guineapig malware to circumvent detection within sandboxed environments. The malware initiates a process wherein the current system time is retrieved using a designated system call or API. This initial timestamp is subsequently recorded. A deliberate delay, lasting at least 100 milliseconds but potentially varying, is then introduced,

suspending execution. This delay aims to mimic typical program behavior and disrupt sandbox timers. Following the delay, the system time is read again. A comparison is performed between the second timestamp and the first. If a difference exceeding 100 milliseconds is observed, it indicates normal system clock functionality and suggests a non-sandboxed environment. Conversely, if the time difference falls significantly short of 100 milliseconds, indicative of a manipulated clock, malware execution is terminated to avoid in-depth analysis within a potentially artificial environment.

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 malware Goofy Guineapig to evade detection within virtualized or sandboxed environments. The total disk size is queried by the malware to ascertain its value. Execution is terminated if the determined size falls below a predetermined threshold, indicative of a potential virtual environment. Similarly, the amount of physical RAM available is measured. If the measured RAM quantity falls below a suspicious threshold, execution is aborted, suggesting a limited memory allocation characteristic of virtual machines. The malware also counts the number of logical processors (CPU cores) present on the system. An unusually low processor count, often observed in virtual machines compared to physical machines, triggers termination. This multi-layered approach presents challenges for analysts attempting to execute Goofy Guineapig within controlled sandbox environments, as its detection mechanisms are specifically designed to identify such scenarios.

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: User Activity Based Checks are employed by the "Goofy Guineapig" malware to evade detection within sandboxed environments or during reverse engineering procedures. Process monitoring capabilities are utilized to identify running processes on the infected system. The malware analyzes process names and strings within those names for signatures indicative of debugging and analysis tools, such as "dbg", "debug", and "ida". Upon detection of any matching processes, malware execution is terminated, precluding further analysis in a controlled setting.

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: The malware's core malicious code is compressed using UPX (Ultimate Packer for eXecutibles), resulting in a reduced executable file size that impedes traditional analysis methods. Furthermore, the UPX-packed malware is integrated within a legitimate NSIS (Nullsoft Scriptable Install System) installer designed for Firefox, thereby creating an appearance of legitimacy and potentially deceiving users into executing the malware unknowingly.

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 stack-based string obfuscation coupled with simple ciphers for code deobfuscation. Strings integral to the malware's operation are not directly embedded within the binary code but rather stored on the program's call stack, rendering them less readily apparent through conventional string analysis techniques. Further obfuscation of these stack-based strings is achieved through the application of XOR and subtraction ciphers. Each character within these strings undergoes encryption via a single byte XOR operation utilizing a fixed key consistent throughout the entire string. A similar approach involving subtraction, rather than bitwise operations, is also employed for character obfuscation. The deobfuscation process appears to be contingent upon the successful completion of a specific request identified as request ID 0x15, suggesting a potential sequence of events or actions within the malware that trigger the deobfuscation phase subsequent to this initial request. Recovery of the original strings necessitates reverse engineering efforts aimed at elucidating the malware's mechanisms for accessing and processing strings from the call stack. Identification of the XOR/subtraction keys may be accomplished through code pattern analysis or techniques such as frequency analysis to infer the key(s). Subsequent decryption of the obfuscated strings can then be performed by applying the reverse XOR or subtraction operations utilizing the identified keys. Upon successful recovery, these original strings can be subjected to analysis to determine their semantic meaning and functional roles within the malware's overall capabilities.

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 the malware to achieve "Hide Artifacts" functionality. The dllhost.exe process is targeted for this technique. The malware injects its malicious code into the memory space of the selected dllhost.exe process. Subsequently, the original contents of the dllhost.exe process's memory are removed or overwritten. The malware's code effectively replaces the

legitimate functionality of dllhost.exe. Due to the execution of malicious code within the existing dllhost.exe process, privileges and system context are inherited. This results in: Stealth is achieved as malicious activity appears to originate from the legitimate dllhost.exe process, hindering detection by security tools. Traditional signature-based detection methods are less effective because they rely on identifying known malware code, which is now replaced within the legitimate process. The hollowed dllhost.exe process continues to execute the malicious code, enabling the Goofy Guineapig malware to establish persistence, communicate with its command and control (C2) server, and perform further malicious actions.

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: Indicator Removal on Host: File Deletion is executed by Goofy Guineapig malware through a series of actions. Initial execution occurs within the directory of the downloaded file. Subsequently, the malware relocates its files, and potentially other malicious payloads, to a seemingly legitimate directory on the system. Following relocation, the original copies of the malware files are deleted from the initial download directory. This tactic aims to obfuscate the malware's presence and hinder detection by security analysts or antivirus software.

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 "Goofy Guineapig" malware exploits DLL side-loading to hijack execution flow. A legitimate executable is installed alongside a malicious DLL file, typically through a trojanised Firefox installer. Upon initiation of the legitimate executable, it searches for and loads plugins or extensions based on predefined criteria. The malicious DLL, disguised as a legitimate plugin or extension, is inadvertently included during this process. Consequently, the malicious DLL's code is executed within the context of the trusted legitimate program. This execution under the privileges and identity of the legitimate program grants access to system resources, enabling malicious actions and circumventing security measures designed to protect against known malware.

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 employed by the malware designated as "Goofy Guineapig" targeting the "dllhost.exe" binary. The selection of "dllhost.exe", a Windows process responsible for hosting dynamic-link libraries (DLLs), is a key element in this technique. Process hollowing involves the allocation of sufficient memory to completely overwrite the contents of the "dllhost.exe" process. Subsequently, the original code of "dllhost.exe" is replaced with malicious payload downloaded from a Command and Control (C2) server. The execution flow is initiated by the hollowed-out "dllhost.exe", now running with the injected malicious code. This allows for the execution of commands under the guise of a legitimate process, thereby increasing the difficulty of detection. Process hollowing is considered a powerful technique due to its ability to camouflage malicious code within a legitimate process, evade immediate suspicion from security software, and leverage existing privileges and resources of "dllhost.exe". Furthermore, traditional antivirus signatures are less effective against code injected into existing 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: Signed Binary Proxy Execution is employed by the Goofy Guineapig malware utilizing rundll32.exe and url.dll. A legitimate binary on the system, potentially possessing elevated privileges, is identified by the malware. Rundll32.exe is leveraged to load the url.dll library. Within url.dll, malicious DLL code is injected indirectly through the exploitation of url.dll's functionality for loading and executing payloads. This technique circumvents security measures due to the legitimacy of rundll32.exe and url.dll as Windows components, potentially evading detection by security software. The malware exploits the inherent trust users place in built-in Windows processes.

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:

- COM functionality is enabled.
- Access to the relevant Windows APIs is available.
- The infected machine has a network connection.
- A Windows operating system is present.
- The malware is running.
- Network connectivity is established.

Post-Conditions:

- Event log entries related to system changes and process creation.
- System instability and performance degradation.
- Compromised system with potential data exfiltration.
- Modified system registry entries.
- Modified system files or DLLs.
- Process activity logs indicating execution of suspicious processes.
- Potential for further malicious activity execution.
- New files created in various directories (e.g., malware executable, configuration files).
- Network traffic logs showing communication with Command and Control (C2) server using HTTPS.

Attack Step 3.1

==

Name: System Information Discovery as used by the malware
Description: System Information Discovery is conducted by the malware through the collection of various data points pertaining to the infected machine. These data points are subsequently obfuscated and transmitted to a Command and Control (C2) server. Data points collected encompass: * Operating system caption, potentially retrieved via COM and WMI access. * Antivirus product display name, also likely obtained through COM and WMI. * Adapters information, which may be acquired using Windows APIs. * Host and host name identification. Obfuscation techniques are employed to encode the collected information, rendering its true meaning obscure upon casual inspection. Transmission of the obfuscated system information occurs within each communication packet as part of the HTTP header, facilitating discreet data transfer to the C2 server.

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:

- An internet connection to reach static.tcplog.com.
- The infected machine has a network connection.
- A system with a network connection.
- The Goofy Guineapig malware binary contains the hardcoded configuration string.
- The malware is running.
- The malware binary is loaded into memory.
- The capability to communicate using UDP and/or KCP protocols.
- The embedded configuration string is accessible.
- A system with a network connection.

Post-Conditions:

- Logon session enumeration data.
- Modified system processes and services.
- Obfuscated strings containing sensitive information within malware code.
- Unusual HTTP headers in network traffic.
- Potential for further malware infections and lateral movement.
- Compromised system with persistent backdoor access.
- Data exfiltration from infected machine.
- New files created or modified on the system (e.g., malware binaries, configuration files).
- Network traffic logs showing communication with C2 server using HTTPS and/or UDP.
- Evidence of deleted or modified task scheduling entries.
- Modified system registry settings.
- Encrypted communication logs
- Performance degradation
- Network connections to suspicious IP addresses
- Traces of data transfer in system event logs
- System instability
- Modified system registry entries
- Modified or deleted existing log files
- Unusual process activity logs
- Potential for further malware infections
- Presence of malicious code in memory
- Indicator of Compromise (IOC) files on the system.
- New files created in various directories
- Data exfiltration
- Compromised system security
- Log file entries indicating suspicious activity
- Traces of data transfer via HTTP(S) requests
- Unusual process activity and memory usage
- System instability
- Compromised system functionality
- New files created in non-standard locations
- Modified or deleted existing log files
- Potential for further malware infections
- Presence of malicious code in system processes

- Data exfiltration
- Network connections to static[.]tcplog[.]com on port 4443
- Modified system registry entries

Attack Step 4.1

==

Name: Application Layer Protocol: Web Protocols as used by the malware

Description: Command and Control (C2) communications executed by the malware are conducted via HTTPS. Secure connections are established utilizing the HTTP protocol over a Transport Layer Security (TLS) layer, thereby ensuring encrypted data transmission between the compromised machine and the attacker's server. The specific encryption algorithm employed is not disclosed.

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: Goofy Guineapig malware implements fallback communication channels through several mechanisms. UDP communication is facilitated via the User Datagram Protocol, enabling faster but less reliable data transfer compared to TCP. The KCP (Keepalive Connection Protocol) is utilized for communication, providing a fast and reliable protocol suitable for real-time applications and unpredictable network conditions. Direct socket communications are also supported, offering flexibility in establishing connections independent of predefined protocols. The specific communication method employed (UDP, KCP, or direct socket) is determined by an embedded configuration string within the malware, allowing attackers to dynamically select the most appropriate channel based on factors such as network conditions and target environment.

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: Communication between the Goofy Guineapig malware and its Command and Control (C2) server is established over an HTTPS port deviating from the standard 443. Specifically, a non-standard port, 4443, is utilized for this purpose.

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.