Enhanced Attack Report

Goofy Guineapig

Generated on 2025-04-10

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 m1
a1. Create or Modify System Process: Windows Service as used by the malware (T1543.003)

Milestone m2
a2. Masquerading: Match Legitimate Name or Location as used by the malware (T1036.005)
a3. Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware (T1497.003)
a4. Virtualization/Sandbox Evasion: System Checks as used by the malware (T1497.001)
a5. Virtualization/Sandbox Evasion: User Activity Based Checks as used by the malware (T1497.002)
a6. Obfuscated Files or Information: Software Packing as used by the malware (T1027.002)
a7. Deobfuscate/Decode Files or Information as used by the malware (T1140)
a8. Hide Artifacts: Hidden Window as used by the malware (T1564.003)
a9. Indicator Removal on Host: File Deletion as used by the malware (T1070.004)
a10. Hijack Execution Flow: DLL Side-Loading as used by the malware (T1574.002)
a11. Process Injection: Process Hollowing as used by the malware (T1055.012)
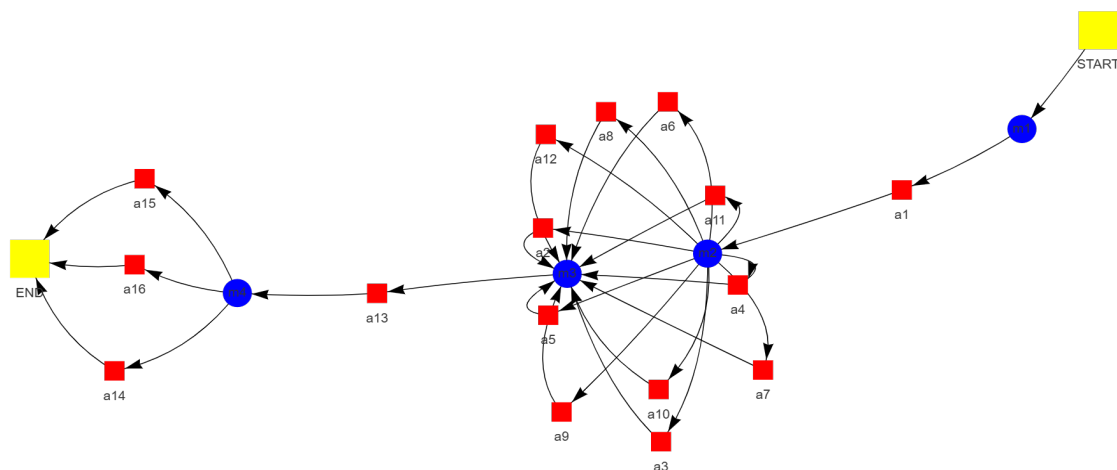a12. Signed Binary Proxy Execution: Rundll32 as used by the malware (T1218.011)

Milestone m3
a13. System Information Discovery as used by the malware (T1082)

Milestone m4
a14. Application Layer Protocol: Web Protocols as used by the malware (T1071.001)
a15. Fallback Channels as used by the malware (T1008)
a16. Non-Standard Port as used by the malware (T1571)

Attack Graph

Milestone *m1*

Attack Step *a1*

```
=====================================================
```
Name: Create or Modify System Process: Windows Service as used by the malware

Description: Windows service creation or modification is employed by the malware as a persistence mechanism. This technique involves the installation of a Windows service that executes malicious code at predetermined intervals or upon specific triggers. The service functionality may encompass the execution of the core payload, the download and execution of additional modules or updates, and communication with command-and-control (C&C;) servers for instructions. Services are system processes that operate in the background and persist even during user logoff. This persistence is achieved through automatic startup during system boot-up. Services often execute with elevated privileges, granting access to sensitive system resources and data. The background nature of services can render them less detectable by conventional security tools.

Pre-Conditions:

- A Windows operating system is present.
- The malware has the ability to create or modify service configuration files.
- The malware has successfully infected the target system.

Post-Conditions:

- Decrypted DLL file with MZ header and PE header bytes.
- Compromised system with persistent malware infection.
- Network connections to C2 server using HTTPS.
- Modified Windows service.
- Downloaded files in ProgramData directory.
- Logs of system process modifications and service creation.
- Files related to Firefox installation and Google updater.
- Data exfiltration to C2 server.
- Process hollowing activity involving dllhost.exe.
- Obfuscated "Authorization" strings in HTTP headers.
- Potential for further malicious activity execution.

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 *m2*

Attack Step *a2*

==
Name: Masquerading: Match Legitimate Name or Location as used by the malware
Description: The "Goofy Guineapig" malware employs tactics to evade detection through masquerading as legitimate software entities. This tactic involves packaging the malware within a legitimate Firefox installer file and mimicking a Google Update process. The malware leverages existing NSIS (Nullsoft Scriptable Install System) installer files commonly associated with legitimate software distribution. File naming conventions and visual branding elements are likely employed to resemble genuine Firefox or Google Update installers, capitalizing on user familiarity and trust. This masquerade strategy aims to circumvent traditional antivirus detection mechanisms that often rely on signature-based analysis. By mimicking known software, the malware avoids identification as a threat. Additionally, this tactic exploits user trust in established brands, increasing the likelihood of unwitting execution by users who perceive the download or installation as legitimate.

Pre-Conditions:

- The operating system allows for file name and location manipulation.
- The malware must have the ability to change its own file name or location.
- The malware has access to legitimate file names or locations associated with Firefox and Google updates.

Post-Conditions:

- System compromised
- Network traffic to C2 server using HTTPS
- Data exfiltration possible
- Persistence established
- Log entries related to service creation and process modifications
- Decrypted DLL file
- Registry changes indicating persistence mechanism
- Traces of RC4 decryption algorithm usage
- Modified system configuration files
- Modified system processes
- Obfuscated strings in HTTP headers
- New files created in ProgramData directory

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 *a3*

```
===================================================
```
Name: Virtualization/Sandbox Evasion: Time Based Evasion as used by the malware
Description: Time-based evasion is implemented within "Goofy Guineapig" malware through repeated checks of the system's time register at predetermined intervals. A defined delay threshold is established between these checks. Should the elapsed time exceed this threshold, exceeding 100 milliseconds, execution is halted. This behavior is designed to detect potential sandbox environments where system timers may be artificially manipulated for accelerated analysis. The discrepancy in timekeeping between a real environment and a sandbox can trigger the malware's termination mechanism, preventing in-depth examination within a controlled setting. While effective against conventional sandboxes, this technique may prove less reliable against advanced analysis platforms capable of maintaining realistic system timings.

Pre-Conditions:

- The malware is running.
- The system time register is accessible.

Post-Conditions:

- Data Exfiltration Potential
- Deleted Batch Script File
- System Resource Consumption
- Process Activity Logs
- Malicious File Execution
- Altered System Time Records
- Modified Registry Entries
- System Information Disclosure
- Obfuscated "Authorization" Strings in HTTP Headers
- Command and Control Communication Established
- Downloaded Malicious Files (DLL) in ProgramData Directory
- Memory Dumps with Decrypted DLL Code
- HTTPS Network Traffic Logs

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

- The malware utilizes a time-based evasion technique.

Attack Step *a4*

```
==================================================
```
Name: Virtualization/Sandbox Evasion: System Checks as used by the malware
Description: System checks are employed by malware such as "Goofy Guineapig" to evade detection within virtualized environments or sandboxes. These checks involve the analysis of hardware characteristics on the infected system. Discrepancies between measured values and predefined thresholds indicative of typical real-world configurations may lead to the termination of the malware's execution, thereby avoiding analysis and potential containment. Information regarding disk size, physical memory capacity, and the number of logical processors is gathered through system calls or libraries. The comparison of these values against established ranges facilitates the determination of whether the operating environment is likely a sandbox.

Pre-Conditions:

- The system has physical memory.
- The malware is running.
- The system has a disk drive.

Post-Conditions:

- System performance degradation
- Network connections to command and control servers
- Data exfiltration
- Altered log files
- Malware persistence
- New files created with malicious content
- System compromise
- Unusual process activity logs
- Modified system registry entries

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 *a5*

==
Name: Virtualization/Sandbox Evasion: User Activity Based Checks as used by the malware
Description: User Activity-Based Checks for Virtualization/Sandbox Evasion are employed by malware samples such as Goofy Guineapig to circumvent detection by security researchers and analysis tools. These tools often operate within controlled environments known as sandboxes, which simulate real systems while incorporating monitoring and logging capabilities. Process monitoring is a key component of this evasion technique. Goofy Guineapig actively scans the list of running processes on an infected machine for specific process names or strings indicative of debugging or analysis environments. These include terms such as "dbg," "debug," and "ida," which are commonly associated with debuggers like gdb, WinDbg, and IDA Pro, respectively. Upon detection of these suspicious process names, Goofy Guineapig is programmed to terminate its own execution. This behavior effectively prevents the malware from running within a sandboxed environment or during active debugging sessions. The efficacy of this technique stems from the fact that sandboxes often emulate user behavior, potentially leading to the initiation of debugging tools or the utilization of software such as IDA Pro for analysis purposes. However, false positives can arise if legitimate processes on a system coincidentally contain these strings in their names. To mitigate this risk, Goofy Guineapig likely incorporates additional checks beyond simple string matching. Despite its effectiveness, this evasion technique is not infallible. Sophisticated sandboxes and security analysts may employ advanced monitoring and analysis techniques to circumvent such checks.

Pre-Conditions:

- Processes are running on the system.
- The malware is running on a system.

Post-Conditions:

- Disk writes indicating data transfer
- Registry modifications
- Data exfiltration
- Network connections to command and control servers
- Shadow copies of compromised files
- New files created (e.g., malware binaries, configuration files)
- Altered system event logs
- Persistence on the system
- Execution of malicious payloads
- Modified system files
- System compromise
- Process hollowing
- Unusual process activity logs

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 *a6*

==
Name: Obfuscated Files or Information: Software Packing as used by the malware
Description: The malware designated as "Goofy Guineapig" employs UPX software packing to obfuscate its core functionality. This process involves compressing the malware executable using the UPX (Ultimate Packer for eXecutables) tool, thereby reducing its size and hindering direct analysis. Furthermore, the packed malware is embedded within a legitimate Nullsoft Scriptable Install System (NSIS) installer package, serving as an additional layer of camouflage. This packaging strategy presents challenges for security researchers due to the requirement for unpacking prior to code analysis. Additionally, the utilization of a legitimate installer may facilitate evasion of initial security checks that primarily target suspicious file types or anomalous executable behavior.

Pre-Conditions:

- The malware author has knowledge of UPX and how to utilize it.
- The malware contains code to perform software packing.
- The malware executable exists.
- UPX packer is available.

Post-Conditions:

- Data Exfiltration Potential
- Persistence Mechanism Established
- Modified Firefox Installation Files
- MD5 Hash of Concatenated System Information
- Obfuscated DLL File ("Goopdate.dll")
- Modified System Registry Entries
- UPX Packed Installer File
- System Information Compromise
- Log Entries in Windows Event Logs
- Modified GoogleUpdate.exe File
- XOR-Encoded Strings within Binary Code
- Command and Control Communication Established
- Malicious Files Downloaded and Executed
- New Files in ProgramData Directory
- Network Traffic to C2 Server

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 *a7*

```
=====================================================
```
Name: Deobfuscate/Decode Files or Information as used by the malware
Description: The malware designated as "Goofy Guineapig" is characterized by sophisticated evasion techniques aimed at obscuring its malicious intent. String obfuscation is achieved through the utilization of stack-based strings, wherein string data is assembled on the program's call stack during execution. This practice hinders detection during static analysis procedures. Further complicating matters, these stack-based strings are subjected to XOR obfuscation using a key value of 0x59. This process renders the string data unreadable without the application of the corresponding XOR operation. Subtraction is also employed as an obfuscation technique within the binary, although specific implementation details remain undisclosed. Deobfuscation processes typically involve static analysis to identify unusual patterns indicative of obfuscation techniques. Pattern recognition algorithms are utilized to detect common obfuscation signatures, such as XOR operations and string encoding schemes. Dynamic analysis, conducted in a controlled environment (sandbox), allows for observation of malware behavior and identification of string manipulation methods. String decryption involves reverse engineering the XOR or subtraction algorithm employed by "Goofy Guineapig," often requiring the identification of the key value or patterns within the code revealing the decryption logic. The MITRE ATT&CK; framework categorizes "Goofy Guineapig's" tactics as follows: T1027.002 Obfuscated Files or Information, which describes the use of UPX packing and stack-based string obfuscation to conceal malware code; and T1140 Deobfuscate/Decode Files or Information, which captures the act of reverse engineering the XOR or subtraction techniques employed by "Goofy Guineapig."

Pre-Conditions:

- Access to the obfuscated files or information within the malware binary.
- The malware binary is present.
- A system capable of executing the malware binary.

Post-Conditions:

- New files created (e.g., shellcode, backdoor)
- Indicators of compromise (IOCs) matching known Goofy Guineapig characteristics
- Registry modifications
- Data exfiltration
- Deleted or modified log files

- Altered system configurations
- Network traffic to C2 server
- Potential for further malware installation
- Evidence of obfuscation techniques used in malware code
- Modified system files
- System compromise
- Remote control of infected machine
- Unusual process activity logs

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 *a8*

```
==================================================
```
Name: Hide Artifacts: Hidden Window as used by the malware
Description: Process hollowing is employed by Goofy Guineapig to obfuscate its presence within the system. The malware targets the legitimate dllhost.exe process for exploitation. The execution of dllhost.exe is suspended, and its original code and data are subsequently purged from memory. Goofy Guineapig's malicious payload is then injected into the vacated memory space of dllhost.exe. Following injection, the modified dllhost.exe is resumed, effectively concealing the malware's presence within a legitimate process. This technique minimizes the generation of artifacts that could alert security tools. The utilization of an established and recognized system process, dllhost.exe, reduces the likelihood of suspicion. Furthermore, the avoidance of new process creation limits the number of entries appearing in task lists or process monitoring utilities.

Pre-Conditions:

- The ability to modify process windows.
- Access to the system's memory and processes.
- A running instance of the Goofy Guineapig malware.
- A process named "dllhost.exe" exists on the system.
- The malware is running.

Post-Conditions:

- Data Exfiltration Potential
- Modified System Registry
- Altered Process Memory

- Hidden Named Pipe
- Process Hollowing Performed
- Command and Control Established
- Modified DLL File with MZ Header and PE Header Bytes
- Obfuscated HTTP Headers with "Authorization" String
- System Information Compromise
- RC4 Decryption Key (2UFdRF06kYvIXWOW)
- Log Entries of Suspicious Activity
- Malicious Files Downloaded and Executed
- New Files in ProgramData Directory
- Network Traffic to C2 Server

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 hides its window.

Attack Step *a9*

==
Name: Indicator Removal on Host: File Deletion as used by the malware
Description: String decoding techniques are employed by the malware to obfuscate functionality. Specific byte sequences are XORed with a key (0x59) to reveal crucial instructions. Process hollowing is utilized by Goofy Guineapig, a technique involving the injection of malicious code into legitimate processes such as "dllhost.exe." This allows the malware to bypass security measures and operate under the guise of a trusted application. Indicator removal on host, specifically file deletion, is observed as part of the malware's behavior. Files are initially run in their download location before being moved to seemingly legitimate directories and subsequently deleted from the original location. This action aligns with Defense Evasion (T1070) tactics. A hardcoded configuration string identifying communication protocols (HTTP/S or UDP/KCP) is utilized by the malware for Command and Control (T1020) purposes. The malware exhibits behavior consistent with Virtualization/Sandbox Evasion (T1497). Processes such as "dbg," "debug," or "ida" indicative of a debugging environment are checked. Execution is halted upon detection, suggesting an attempt to evade analysis in sandboxes or virtual machines.

Pre-Conditions:

- The initial download location contains files related to the malware.
- The malware has knowledge of a legitimate-looking directory.
- A file system is present.
- The malware is running.

Post-Conditions:

- Network traffic to static.tcplog.com
- Data exfiltration
- New files in ProgramData directory
- Obfuscated strings in memory
- System instability
- Malware persistence
- Registry entries
- Log files with suspicious activity
- Indicator removal attempts in logs
- Modified system files
- Deleted batch script remnants
- Compromised system

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 deleted from the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".

Attack Step *a10*

==
Name: Hijack Execution Flow: DLL Side-Loading as used by the malware
Description: A comprehensive analysis of "Goofy Guineapig" has been conducted, elucidating its operational tactics and techniques. Key findings include the utilization of a Trojanized Firefox installer to facilitate initial access, exploiting user trust and circumventing security measures. The malware leverages DLL side-loading, hijacking the execution flow by forcing legitimate processes such as GoogleUpdate.exe to load malicious DLLs. Communication obfuscation techniques are employed, including an embedded configuration string with hardcoded URLs encoded using a XOR cipher and the utilization of UDP and KCP protocols for evading network monitoring. Mutex creation based on the system's MD5 hash is implemented to ensure only one instance of the malware operates concurrently on a given machine, hindering detection through multiple processes. Alignment with the MITRE ATT&CK;® framework has been established, specifically highlighting T1574.002: Hijack Execution Flow: DLL Side-Loading. String analysis techniques are utilized to identify potential malicious code within the sample, further demonstrating the value of the ATT&CK; framework. Security implications necessitate increased user vigilance regarding software downloads from untrusted sources and suspicious links. Robust endpoint protection solutions capable of detecting and blocking known malware, as well as analyzing unknown files for malicious behavior, are crucial. Network monitoring for unusual traffic patterns, such as communication with hidden C2 servers or the use of obscure protocols like KCP, is essential. Software patching remains paramount in mitigating vulnerabilities exploitable by

attackers. Further research could delve into the specific functions implemented by the 'plugin_run' function and their role within the malware campaign. Detailed analysis of the C2 communication protocol and exchanged data between infected systems and the command-and-control server is warranted. Identifying indicators of compromise (IOCs) for detecting ongoing infections or future campaigns employing similar techniques is crucial.

Pre-Conditions:

- Network connectivity may be required for the malware to download the malicious DLL.
- A vulnerable application with a dependency on dynamic loading of libraries.
- The ability to manipulate the environment variables or registry settings used by the vulnerable application.
- A vulnerable application is running.
- The malware has successfully infiltrated the system.
- The malicious DLL exists in a location accessible to the vulnerable application.

Post-Conditions:

- Network traffic to static.tcplog.com
- Altered system performance metrics
- Data exfiltration
- New files in ProgramData directory
- Presence of obfuscated strings in code
- Modified system configuration settings
- Hollowed process with altered memory content
- System instability
- Potential for further malware infections
- Named pipe created with hashed computer name
- Log entries indicating suspicious activity
- Modified system registry entries
- Remote control of infected machine
- Compromised system

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 process attempts to load the DLL "config.dat".

Attack Step *a11*

==

Name: Process Injection: Process Hollowing as used by the malware
Description: Process injection via Process Hollowing is employed by the Goofy Guineapig malware to execute malicious code within the memory space of the dllhost.exe process. The malware first identifies dllhost.exe as its target process. Subsequently, a new memory region is allocated within the target process's address space, sufficient to accommodate the entirety of the malicious payload. The original executable code residing in the dllhost.exe process memory is then overwritten with the attacker's malicious code, effectively hollowing out the legitimate process. Control flow within the hijacked dllhost.exe process is redirected to the entry point of the injected malicious code, enabling the execution of the attacker's instructions under the guise of a legitimate system process. The injected malicious code may engage in various activities, including establishing communication with command and control (C2) servers for receiving further instructions or exfiltrating stolen data, as well as compromising the system by installing additional malware, modifying system settings, acquiring elevated privileges, or conducting reconnaissance operations. To evade detection by security solutions, the malware implements anti-debugging checks to detect if the process is being debugged or analyzed in a sandbox environment, analyzes system properties and running processes to determine if it is operating within a controlled testing environment, and validates system time for anomalies that might indicate analysis. This specific action aligns with the MITRE ATT&CK;® technique T1055.012 - Process Injection: Process Hollowing.

Pre-Conditions:

- The malware has the capability to inject code into a running process.
- A running instance of the dllhost.exe process is available.
- The malware has successfully downloaded the dllhost.exe binary.

Post-Conditions:

- Registry modifications
- Data exfiltration
- Deleted batch script remnants
- New files in ProgramData directory
- Persistence on the system
- Log entries indicating suspicious activity
- Modified dllhost.exe process
- Obfuscated strings in HTTP headers
- Network connections to C2 server
- System compromise
- Altered system time
- Potential for further malicious activity

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 *a12*

```
=====================================================
```
Name: Signed Binary Proxy Execution: Rundll32 as used by the malware
Description: A detailed analysis of the "Goofy Guineapig" malware family is presented herein, outlining its tactics, techniques, and procedures (TTPs) in accordance with the MITRE ATT&CK; framework. The malware, designated as "Goofy Guineapig," is characterized as a sophisticated loader designed for the delivery of malicious payloads and the establishment of persistent control over compromised systems. Technical analysis reveals the utilization of string encoding techniques within the malware's code, likely intended to impede analysis by security tools. Process hollowing is employed, whereby malicious code is injected into legitimate processes such as dllhost.exe. This technique effectively conceals the malware's activities by leveraging the privileges of the compromised process. System binary proxy execution is also utilized, with rundll32.exe and url.dll being leveraged to execute a legitimate binary that loads the malicious DLL, thereby establishing persistence. This method exploits existing system tools for code execution, rendering detection more challenging. Persistence mechanisms are implemented through both process hollowing and rundll32 execution, ensuring malware survival across system reboots. A unique mutex creation method based on multiple MD5 hashes of the computer name is also employed, further complicating detection efforts. Command and Control (C2) communication protocols include both HTTP(S) and UDP via KCP, enhancing resilience to network monitoring and analysis. A hardcoded configuration string embedded within the malware specifies C2 endpoints and protocol preferences. This string undergoes XOR encoding with a key for additional obfuscation. Sandbox evasion capabilities are incorporated into Goofy Guineapig, including checks for virtualized environments or debugging tools, thereby preventing analysis in controlled settings. The malware likely monitors system events and processes associated with sandboxes. Active enumeration of active user sessions on the infected machine is conducted by the malware, potentially gathering information about logged-in users and their activities. Potential impacts associated with Goofy Guineapig infection include data theft, remote access control, system disruption, and the propagation of further malware. Mitigation strategies encompass the implementation of robust Endpoint Detection and Response (EDR) solutions capable of detecting suspicious process behavior, network communication patterns, and attempted file modifications. Security Information and Event Management (SIEM) systems should be utilized to correlate security alerts and identify potential threats based on anomalous activity. Regular patching of operating systems and software applications is crucial to address known vulnerabilities exploited by malware such as Goofy Guineapig. User education programs aimed at recognizing phishing attempts, suspicious emails, and malicious websites are essential. Goofy Guineapig presents a significant threat, employing sophisticated techniques to evade detection and establish persistent control over compromised systems. Proactive security measures are paramount in mitigating the risks posed by this malware family.

Pre-Conditions:

- The url.dll file is present.
- exe` tool is available.
- Connectivity to the C2 server for retrieving instructions and potentially additional files.

Post-Conditions:

- Network traffic to C2 server (HTTPS/UDP)
- Altered system performance metrics
- Data exfiltration

- Log entries indicating process creation and execution
- Residual DLL remnants in memory
- Presence of obfuscated code within memory
- System compromise
- Potential for further malware execution
- New files created in ProgramData directory
- Modified system registry entries

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 in the directory "C:\ProgramData\GoogleUpdate\GoogleUpdate".

Milestone *m3*

Attack Step *a13*

==
Name: System Information Discovery as used by the malware
Description: System Information Discovery by Goofy Guineapig Malware: A Technical Analysis Goofy Guineapig malware employs obfuscation techniques to conceal sensitive system data within HTTP communication with its Command and Control (C2) server. This is achieved through the manipulation of an "Authorization" header, rendering the transmitted information difficult to decipher without specialized analysis. Data extraction methodologies utilized by Goofy Guineapig include: * **COM & WMI Access:** The malware leverages COM interfaces to query the Windows Management Instrumentation (WMI) service, enabling the retrieval of system-specific details such as operating system version, antivirus software display name, and other relevant information. * **Windows APIs:** Goofy Guineapig exploits standard Windows Application Programming Interfaces (APIs) to access and gather additional system data points, including network adapter configurations, hostname, and computer name. The implications of this data collection are significant: * **Attribution & Tracking:** The precise details gathered about infected machines facilitate the linkage of compromised systems to specific campaigns or malicious actors. * **Targeted Exploitation:** Information regarding installed antivirus software and system configurations enables attackers to tailor their exploitation strategies, potentially circumventing existing security measures. * **Custom Payload Delivery:** System-specific data, such as operating system version and hardware specifications, allows for the creation of customized malware payloads optimized for maximum impact and compatibility with the target environment. Defensive Strategies: * **Network Monitoring:** Continuous analysis of network traffic for anomalous HTTP requests featuring obfuscated headers can serve as a preliminary defense mechanism against Goofy Guineapig activity. * **Endpoint Security:** Robust endpoint security solutions, encompassing intrusion detection systems (IDS) and advanced antivirus software, are essential for detecting and mitigating malicious activities associated with Goofy Guineapig. * **WMI Access Controls:** Implementing restrictive access controls to the WMI service through group policies or other security mechanisms can hinder the malware's ability to extract sensitive system information.

Pre-Conditions:

- A Windows operating system is present.
- The malware is running on a victim machine.
- The victim machine has an operating system.
- Network connectivity is available.
- Access to relevant Windows APIs is permitted.
- Access to the Windows Management Instrumentation (WMI) is permitted.

Post-Conditions:

- Malware persistence on the system.
- Hidden processes running in the background.
- Altered system logs with suspicious activity.
- System performance degradation.
- New files created in various directories (e.g., temporary folders, user profiles).
- Obfuscated code remnants within existing or newly created files.
- Modified system registry entries.
- Compromised system with potential data theft or manipulation.

- Unusual network traffic to command and control servers.
- Increased risk of further attacks.

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 *m4*

Attack Step *a14*

==
Name: Application Layer Protocol: Web Protocols as used by the malware
Description: HTTPS is employed as the underlying protocol for all communication between the infected machine and the Command and Control (C2) server. This implementation results in data transmission being encrypted via SSL/TLS, thereby increasing the difficulty of interception and analysis during transit. Furthermore, HTTPS typically utilizes port 443, a standard port commonly associated with web traffic, which may facilitate evasion of detection by firewalls or security software primarily configured to monitor non-standard ports.

Pre-Conditions:

- HTTPS protocol support.
- An active internet connection.
- A C2 server accessible via the internet.
- RC4 encryption capabilities.
- The infected machine has an active internet connection.
- The malware is running.

Post-Conditions:

- Network traffic to C2 server (static.tcplog.com)
- Data exfiltration
- Compromised system
- Modified system event logs
- System instability
- Potential for further malware infections
- Obfuscated strings within malware binary
- New files created in various directories
- Indicator of Compromise (IOCs) present in network traffic analysis
- Unusual process activity logs
- Modified system registry entries

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 sent.

Attack Step *a15*

===
Name: Fallback Channels as used by the malware
Description: Fallback channels are alternative communication methods employed by attackers when primary communication channels are compromised or disrupted. The malware known as Goofy Guineapig utilizes embedded configuration strings to dictate its communication protocols. These configurations may specify the use of UDP and the KCP protocol, or direct socket communications. Goofy Guineapig dynamically selects the most suitable communication method based on the configuration string. This dynamic selection enables seamless switching between methods if one is blocked or detected. The implementation of fallback channels enhances the malware's evasion capabilities, contributing to persistent infections and complicating threat analysis due to the diverse range of potential communication channels. To mitigate the risks associated with Goofy Guineapig and similar malware, organizations are advised to implement comprehensive security measures. These include network monitoring and intrusion detection systems (IDS) to identify anomalous network traffic patterns, multi-layered security controls encompassing firewalls, antivirus software, intrusion prevention systems (IPS), and endpoint security solutions, and regular software updates to address known vulnerabilities.

Pre-Conditions:

- An embedded configuration string exists within the malware.
- The malware is running.
- The malware has the capability to utilize UDP and the KCP protocol.
- A network connection is available.

Post-Conditions:

- Registry modifications
- Data exfiltration
- Unusual process activity (e.g., dllhost.exe)
- Leftover command and control communication data
- Altered system configurations
- New files created
- Persistence on the system
- System compromise
- Log entries indicating suspicious behavior
- Network traffic to static.tcplog.com:4443
- Modified system files
- Potential for further malicious activity

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/

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

Attack Step *a16*

==
Name: Non-Standard Port as used by the malware
Description: The "Non-Standard Port" action is executed by Goofy Guineapig through the establishment of an HTTPS connection utilizing port 4443 instead of the standard HTTPS port (443). This deviation from the conventional port configuration serves to circumvent security measures primarily focused on port 443, thereby enhancing stealth and evasion capabilities.

Pre-Conditions:

- Network connectivity to the C2 server.
- A system infected with the Goofy Guineapig malware.
- The malware is running.
- The ability to monitor network traffic.

Post-Conditions:

- Modified or created user accounts.
- New files created or modified (e.g., malicious DLL, configuration files).
- Potential for further malicious activity on the infected system.
- Process activity logs indicating execution of the malicious DLL and other suspicious processes.
- Compromised system with persistent malware infection.
- Encrypted data files containing stolen information.
- Modified system registry entries.
- Network traffic logs showing communication with the C2 server over HTTPS port 4443 and potentially UDP/KCP.
- Event log entries related to system changes and process creation.
- Data exfiltration to C2 server.

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.