Aug 26 2023

NETMANAGEIT

Intelligence Report

Malware-as-a-Service: Redline Stealer Variants Demonstrate a Low-Barrier-to-Entry Threat

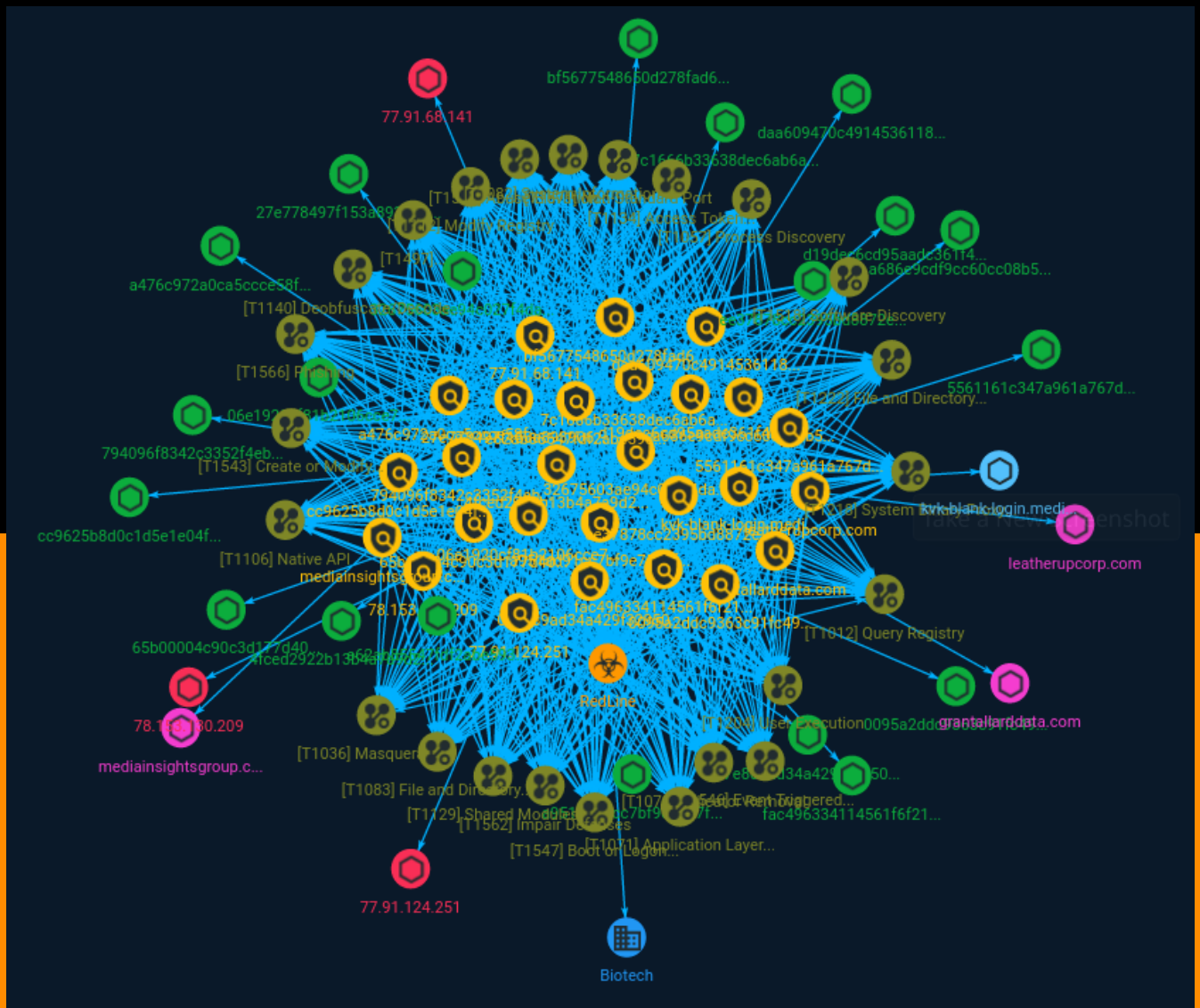

Table of contents

Overview

Entities

Observables

External References

Overview

Description

Analysts observed a malware family targeting financial information to be used for immediate gain as well as reconnaissance functions to perform initial information gathering and establish persistence. RedLine stealer is almost always accompanied by other malware; either preceded by a loader to install it or succeeded by further malware.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

15 / 100

Sector

Name
Biotech

Indicator

Name
07e889ad34a429f3295011d92258f5d43a6e015eeb072695fc81535f82b460c1
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '07e889ad34a429f3295011d92258f5d43a6e015eeb072695fc81535f82b460c1']
Name
a951ad91cc7bf9e7507f9ac1c2ff3c2fb80303e5343b87fee1b205233693e6ba
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a951ad91cc7bf9e7507f9ac1c2ff3c2fb80303e5343b87fee1b205233693e6ba']
Name

7c1666b33638dec6ab6a915dd701a1b6025f2b05d32bad9034f5da4622821b65

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '7c1666b33638dec6ab6a915dd701a1b6025f2b05d32bad9034f5da4622821b65']

Name

fac496334114561f6f21874bdc003325cba7821c4a294d0ad3a5c23f94a29300

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'fac496334114561f6f21874bdc003325cba7821c4a294d0ad3a5c23f94a29300']

Name

794096f8342c3352f4eb5642acb38241b688608f2026501e1430a70e759fb551

Description

ALF:HeraklezEval:HackTool:Win32/DefenderControl

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '794096f8342c3352f4eb5642acb38241b688608f2026501e1430a70e759fb551']

Name

32675603ae94c027f4da61496f5e80994a933ae69f51b86c1ce0a8d38672c114

Description

Win.Trojan.Redline-9938775-1

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '32675603ae94c027f4da61496f5e80994a933ae69f51b86c1ce0a8d38672c114']

Name

65b00004c90c3d177d400cc52e13c20b489903db211fb91b8216e5fb23d86859

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '65b00004c90c3d177d400cc52e13c20b489903db211fb91b8216e5fb23d86859']

Name

4fced2922b13b4a7a9d22ac8c3f78b805ec44e9942e21c8368b45dd092dc1543

Description

RedLine

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '4fced2922b13b4a7a9d22ac8c3f78b805ec44e9942e21c8368b45dd092dc1543']

Name

ee37878cc2395bd8872e1d5531b374ddd3da459aaa0e63f74b4c34aa7c7d63dc

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'ee37878cc2395bd8872e1d5531b374ddd3da459aaa0e63f74b4c34aa7c7d63dc']

Name

06e1920cf81b2106cce759969b30d5ab5e93218c4abfe682e7be2ac11b47726f

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '06e1920cf81b2106cce759969b30d5ab5e93218c4abfe682e7be2ac11b47726f']

Name

d19dec6cd95aadc361f4c3811b989775a7bf8630c33e455844ae48d8ffcf8a39

Description

DotNET_ConfuserEx

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd19dec6cd95aadc361f4c3811b989775a7bf8630c33e455844ae48d8ffcf8a39']

Name

grantallarddata.com

Pattern Type

stix

Pattern

[domain-name:value = 'grantallarddata.com']

Name

27e778497f153a8939069c654af632f5bf322e6cc4da39555c818f6e67411782

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '27e778497f153a8939069c654af632f5bf322e6cc4da39555c818f6e67411782']

Name

a686e9cdf9cc60cc08b5da9e50dd5124f4295f81e5f91222ae77184e190b29f6

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'a686e9cdf9cc60cc08b5da9e50dd5124f4295f81e5f91222ae77184e190b29f6']

Name

78.153.130.209

Description

RedLine CC=AT ASN=AS210644 AEZA GROUP Ltd

Pattern Type

stix

Pattern

[ipv4-addr:value = '78.153.130.209']

Name

e62ab85547fdf2abe5936b39003db1e5ed3c9b42f35420fac7ea32b3387a0849

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'e62ab85547fdf2abe5936b39003db1e5ed3c9b42f35420fac7ea32b3387a0849']

Name

kvk-blank-login.mediainsightsgroup.com

Pattern Type

stix

Pattern

[hostname:value = 'kvk-blank-login.mediainsightsgroup.com']

Name

cc9625b8d0c1d5e1e04f293737eec2403a7aa8b496abfbfe4421c16de28bcd25

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'cc9625b8d0c1d5e1e04f293737eec2403a7aa8b496abfbfe4421c16de28bcd25']

Name

daa609470c4914536118a028e1ebc237fd0b623c776263538cdeaafd57da6068

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'daa609470c4914536118a028e1ebc237fd0b623c776263538cdeaafd57da6068']

Name

77.91.124.251

Description

RedLine CC=FI ASN=AS203727 Daniil Yevchenko

Pattern Type

stix

Pattern

[ipv4-addr:value = '77.91.124.251']

Name

77.91.68.141

Description

CC=FI ASN=AS203727 Daniil Yevchenko

Pattern Type

stix

Pattern

[ipv4-addr:value = '77.91.68.141']

Name

leatherupcorp.com

Pattern Type

stix

Pattern

[domain-name:value = 'leatherupcorp.com']

Name

mediainsightsgroup.com

Pattern Type

stix

Pattern

[domain-name:value = 'mediainsightsgroup.com']

Name

bf5677548650d278fad6f14ad8b20e4ad4e6a87cf4fe83a47aa5b367f30a3690

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'bf5677548650d278fad6f14ad8b20e4ad4e6a87cf4fe83a47aa5b367f30a3690']

Name

0095a2ddc9363c91fc497296555de15fbbc6aeec81e731e0683fc2fca0fd3b06

Description

Win.Trojan.Redline-9938775-1

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0095a2ddc9363c91fc497296555de15fbbc6aeec81e731e0683fc2fca0fd3b06']

Name
a476c972a0ca5ccce58f67b0a51dbde50c915eab506fb1d843e7897f7e785f5e
Description
RedLine
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a476c972a0ca5ccce58f67b0a51dbde50c915eab506fb1d843e7897f7e785f5e']
Name
5561161c347a961a767d0e6994cc89bc831b538e29c508893f9af6bb4678655c
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '5561161c347a961a767d0e6994cc89bc831b538e29c508893f9af6bb4678655c']

Attack-Pattern

Name
File and Directory Permissions Modification
ID
T1222
Description
Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.(Citation: Hybrid Analysis Icacls1 June 2018)(Citation: Hybrid Analysis Icacls2 May 2018) File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.). Modifications may include changing specific access rights, which may require taking ownership of a file or directory and/or elevated permissions depending on the file or directory's existing permissions. This may enable malicious activity such as modifying, replacing, or deleting specific files or directories. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Accessibility Features](https://attack.mitre.org/techniques/T1546/008), [Boot or Logon Initialization Scripts](https://attack.mitre.org/techniques/T1037), [Unix Shell Configuration Modification](https://attack.mitre.org/techniques/T1546/004), or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](https://attack.mitre.org/techniques/T1574). Adversaries may also change permissions of symbolic links. For example, malware (particularly ransomware) may modify symbolic links and associated settings to enable access to files from local shortcuts with remote paths.(Citation: new_rust_based_ransomware)(Citation: bad_luck_blackcat)(Citation: falconoverwatch_blackcat_attack)(Citation: blackmatter_blackcat)(Citation: fsutil_behavior)

Name

Shared Modules

ID

T1129

Description

Adversaries may execute malicious payloads via loading shared modules. The Windows module loader can be instructed to load DLLs from arbitrary local paths and arbitrary Universal Naming Convention (UNC) network paths. This functionality resides in NTDLL.dll and is part of the Windows [Native API](https://attack.mitre.org/techniques/T1106) which is called from functions like `CreateProcess`, `LoadLibrary`, etc. of the Win32 API.(Citation: Wikipedia Windows Library Files) The module loader can load DLLs: * via specification of the (fully-qualified or relative) DLL pathname in the IMPORT directory; * via EXPORT forwarded to another DLL, specified with (fully-qualified or relative) pathname (but without extension); * via an NTFS junction or symlink program.exe.local with the fully-qualified or relative pathname of a directory containing the DLLs specified in the IMPORT directory or forwarded EXPORTs; * via `<file name="filename.extension" loadFrom="fully-qualified or relative pathname">` in an embedded or external "application manifest". The file name refers to an entry in the IMPORT directory or a forwarded EXPORT. Adversaries may use this functionality as a way to execute arbitrary payloads on a victim system. For example, malware may execute share modules to load additional components or features.

Name

Abuse Elevation Control Mechanism

ID

T1548

Description

Adversaries may circumvent mechanisms designed to control elevate privileges to gain higher-level permissions. Most modern systems contain native elevation control

mechanisms that are intended to limit privileges that a user can perform on a machine. Authorization has to be granted to specific users in order to perform tasks that can be considered of higher risk. An adversary can perform several methods to take advantage of built-in control mechanisms in order to escalate privileges on a system.

Name

Event Triggered Execution

ID

T1546

Description

Adversaries may establish persistence and/or elevate privileges using system mechanisms that trigger execution based on specific events. Various operating systems have means to monitor and subscribe to events such as logons or other user activity such as running specific applications/binaries. Cloud environments may also support various functions and services that monitor and can be invoked in response to specific cloud events. (Citation: Backdooring an AWS account)(Citation: Varonis Power Automate Data Exfiltration) (Citation: Microsoft DART Case Report 001) Adversaries may abuse these mechanisms as a means of maintaining persistent access to a victim via repeatedly executing malicious code. After gaining access to a victim system, adversaries may create/modify event triggers to point to malicious content that will be executed whenever the event trigger is invoked. (Citation: FireEye WMI 2015)(Citation: Malware Persistence on OS X)(Citation: amnesia malware) Since the execution can be proxied by an account with higher permissions, such as SYSTEM or service accounts, an adversary may be able to abuse these triggered execution mechanisms to escalate their privileges.

Name

Create or Modify System Process

ID

T1543

Description

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](https://attack.mitre.org/techniques/T1543/004) and [Launch Agent](https://attack.mitre.org/techniques/T1543/001) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

Name

Process Discovery

ID

T1057

Description

Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software/applications running on systems within the network. Adversaries may use the information from [Process Discovery](https://attack.mitre.org/techniques/T1057) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. In Windows environments, adversaries could obtain details on running processes using the [Tasklist](https://attack.mitre.org/software/S0057) utility via [cmd](https://attack.mitre.org/software/S0106) or `Get-Process` via [PowerShell](https://attack.mitre.org/techniques/T1059/001). Information about processes can also be extracted from the output of [Native API](https://attack.mitre.org/techniques/T1106) calls such as `CreateToolhelp32Snapshot`. In Mac and Linux, this is accomplished with the `ps` command. Adversaries may also opt to enumerate processes via /proc. On network devices, [Network Device CLI](https://attack.mitre.org/techniques/

T1059/008) commands such as `show processes` can be used to display current running processes.(Citation: US-CERT-TA18-106A)(Citation: show_processes_cisco_cmd)

Name

Boot or Logon Autostart Execution

ID

T1547

Description

Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon.(Citation: Microsoft Run Key)(Citation: MSDN Authentication Packages)(Citation: Microsoft TimeProvider)(Citation: Cylance Reg Persistence Sept 2013)(Citation: Linux Kernel Programming) These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel. Since some boot or logon autostart programs run with higher privileges, an adversary may leverage these to elevate privileges.

Name

Virtualization/Sandbox Evasion

ID

T1497

Description

Adversaries may employ various means 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](https://attack.mitre.org/techniques/T1497) during automated discovery to shape follow-on behaviors.(Citation: Deloitte Environment Awareness) Adversaries may use several methods to accomplish [Virtualization/Sandbox Evasion](https://attack.mitre.org/techniques/T1497) such as checking for security monitoring tools (e.g., Sysinternals, Wireshark, etc.) or other system artifacts associated with analysis or virtualization. Adversaries may also check for legitimate user activity to help determine if it is in an analysis environment. Additional methods include use of sleep timers or loops within malware code to avoid operating within a temporary sandbox. (Citation: Unit 42 Pirpi July 2015)

Name

Query Registry

ID

T1012

Description

Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. The Registry contains a significant amount of information about the operating system, configuration, software, and security.(Citation: Wikipedia Windows Registry) Information can easily be queried using the [Reg](https://attack.mitre.org/software/S0075) utility, though other means to access the Registry exist. Some of the information may help adversaries to further their operation within a network. Adversaries may use the information from [Query Registry](https://attack.mitre.org/techniques/T1012) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Name

Masquerading

ID

T1036

Description

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](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site)

Name

Non-Standard Port

ID

T1571

Description

Adversaries may communicate using a protocol and port pairing that are typically not associated. For example, HTTPS over port 8088(Citation: Symantec Elfin Mar 2019) or port 587(Citation: Fortinet Agent Tesla April 2018) 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. Adversaries may also make changes to victim systems to abuse non-standard ports. For example, Registry keys and other configuration settings can be used to modify protocol and port pairings.(Citation: change_rdp_port_conti)

Name

Indicator Removal

ID

T1070

Description

Adversaries may delete or modify artifacts generated within systems 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.

Name

Phishing

ID

T1566

Description

Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a trusted source, as well as evasive techniques such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages (e.g., [Email Hiding Rules](https://attack.mitre.org/techniques/T1564/008)).(Citation: Microsoft OAuth Spam 2022)(Citation: Palo Alto Unit 42 VBA Infostealer 2014) Another way to accomplish this is by forging or spoofing(Citation: Proofpoint-spoof) the identity of the

sender which can be used to fool both the human recipient as well as automated security tools.(Citation: cyberproof-double-bounce) Victims may also receive phishing messages that instruct them to call a phone number where they are directed to visit a malicious URL, download malware,(Citation: sygnia Luna Month)(Citation: CISA Remote Monitoring and Management Software) or install adversary-accessible remote management tools onto their computer (i.e., [User Execution](https://attack.mitre.org/techniques/T1204)).(Citation: Unit42 Luna Moth)

Name

Software Discovery

ID

T1518

Description

Adversaries may attempt to get a listing of software and software versions that are installed on a system or in a cloud environment. Adversaries may use the information from [Software Discovery](https://attack.mitre.org/techniques/T1518) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Adversaries may attempt to enumerate software for a variety of reasons, such as figuring out what security measures are present or if the compromised system has a version of software that is vulnerable to [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068).

Name

Impair Defenses

ID

T1562

Description

Adversaries may maliciously modify components of a victim environment in order 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 may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

Name

Modify Registry

ID

T1112

Description

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](https://attack.mitre.org/software/S0075) may be used for local or remote Registry modification. (Citation: Microsoft Reg) 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](https://attack.mitre.org/software/S0075) or other utilities using the Win32 API. (Citation: Microsoft Reghide NOV 2006) Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence. (Citation: TrendMicro POWELIKS AUG 2014) (Citation: SpectorOps Hiding Reg Jul 2017) The Registry of a remote system may be modified to aid in execution of files as part of lateral movement. It requires the remote Registry service to be running on the target system. (Citation: Microsoft Remote) Often [Valid Accounts](https://attack.mitre.org/techniques/T1078) are required, along with access to the remote system's [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002) for RPC communication.

Name

User Execution

ID

T1204

Description

An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of [Phishing](https://attack.mitre.org/techniques/T1566). While [User Execution](https://attack.mitre.org/techniques/T1204) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](https://attack.mitre.org/techniques/T1534). Adversaries may also deceive users into performing actions such as enabling [Remote Access Software](https://attack.mitre.org/techniques/T1219), allowing direct control of the system to the adversary, or downloading and executing malware for [User Execution](https://attack.mitre.org/techniques/T1204). For example, tech support scams can be facilitated through [Phishing](https://attack.mitre.org/techniques/T1566), vishing, or various forms of user interaction. Adversaries can use a combination of these methods, such as spoofing and promoting toll-free numbers or call centers that are used to direct victims to malicious websites, to deliver and execute payloads containing malware or [Remote Access Software](https://attack.mitre.org/techniques/T1219).(Citation: Telephone Attack Delivery)

Name

Native API

ID

T1106

Description

Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. (Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Native API functions (such as `NtCreateProcess`) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations. (Citation: Microsoft Win32)(Citation: LIBC)(Citation: GLIBC) Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS Cocoa)(Citation: macOS Foundation) Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001)).

Name

Access Token Manipulation

ID

T1134

Description

Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process

also takes on the security context associated with the new token. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. These token can then be applied to an existing process (i.e. [Token Impersonation/Theft](https://attack.mitre.org/techniques/T1134/001)) or used to spawn a new process (i.e. [Create Process with Token](https://attack.mitre.org/techniques/T1134/002)). An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can then use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.(Citation: Pentestlab Token Manipulation) Any standard user can use the `runas` command, and the Windows API functions, to create impersonation tokens; it does not require access to an administrator account. There are also other mechanisms, such as Active Directory fields, that can be used to modify access tokens.

Name

Application Layer Protocol

ID

T1071

Description

Adversaries may communicate using OSI application layer protocols 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. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

Name

Deobfuscate/Decode Files or Information

ID

T1140

Description

Adversaries may use [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) 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. One such example is the use of [certutil](https://attack.mitre.org/software/S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016)

Name

System Binary Proxy Execution

ID

T1218

Description

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.(Citation: LOLBAS Project) 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. Similarly, on Linux systems adversaries may abuse trusted binaries such as `split` to proxy execution of malicious commands.(Citation: split man page)(Citation: GTFO split)

Name

File and Directory Discovery

ID

T1083

Description

Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the [Native API](https://attack.mitre.org/techniques/T1106). Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather file and directory information (e.g. `dir`, `show flash`, and/or `nvram`).(Citation: US-CERT-TA18-106A)

Name

System Information Discovery

ID

T1082

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](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](https://attack.mitre.org/software/S0096) can be used to gather

detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather detailed system information (e.g. `show version`).(Citation: US-CERT-TA18-106A) [System Information Discovery](https://attack.mitre.org/techniques/T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment.(Citation: OSX.FairyTale)(Citation: 20 macOS Common Tools and Techniques) Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virutal Machine API)

Malware

Name
RedLine

Domain-Name

Value
leatherupcorp.com
grantallarddata.com
mediainsightsgroup.com

StixFile

Value
cc9625b8d0c1d5e1e04f293737eec2403a7aa8b496abfbfe4421c16de28bcd25
a686e9cdf9cc60cc08b5da9e50dd5124f4295f81e5f91222ae77184e190b29f6
07e889ad34a429f3295011d92258f5d43a6e015eeb072695fc81535f82b460c1
bf5677548650d278fad6f14ad8b20e4ad4e6a87cf4fe83a47aa5b367f30a3690
06e1920cf81b2106cce759969b30d5ab5e93218c4abfe682e7be2ac11b47726f
e62ab85547fdf2abe5936b39003db1e5ed3c9b42f35420fac7ea32b3387a0849
65b00004c90c3d177d400cc52e13c20b489903db211fb91b8216e5fb23d86859
5561161c347a961a767d0e6994cc89bc831b538e29c508893f9af6bb4678655c
794096f8342c3352f4eb5642acb38241b688608f2026501e1430a70e759fb551
daa609470c4914536118a028e1ebc237fd0b623c776263538cdeaafd57da6068
a951ad91cc7bf9e7507f9ac1c2ff3c2fb80303e5343b87fee1b205233693e6ba
ee37878cc2395bd8872e1d5531b374ddd3da459aaa0e63f74b4c34aa7c7d63dc
7c1666b33638dec6ab6a915dd701a1b6025f2b05d32bad9034f5da4622821b65

d19dec6cd95aadc361f4c3811b989775a7bf8630c33e455844ae48d8ffcf8a39

0095a2ddc9363c91fc497296555de15fbbc6aeec81e731e0683fc2fca0fd3b06

fac496334114561f6f21874bdc003325cba7821c4a294d0ad3a5c23f94a29300

32675603ae94c027f4da61496f5e80994a933ae69f51b86c1ce0a8d38672c114

27e778497f153a8939069c654af632f5bf322e6cc4da39555c818f6e67411782

4fced2922b13b4a7a9d22ac8c3f78b805ec44e9942e21c8368b45dd092dc1543

a476c972a0ca5ccce58f67b0a51dbde50c915eab506fb1d843e7897f7e785f5e

Hostname

Value
kvk-blank-login.mediainsightsgroup.com

IPv4-Addr

Value
77.91.124.251
78.153.130.209
77.91.68.141

External References

- https://otx.alienvault.com/pulse/64e86f59c7e4868f4fd05414
- https://blog.eclecticiq.com/redline-stealer-variants-demonstrate-a-low-barrier-to-entry-threat