Jul 11 2024

NETMANAGEIT

Intelligence Report

VayGren and Mr.Burns: Strong Ties in Finance

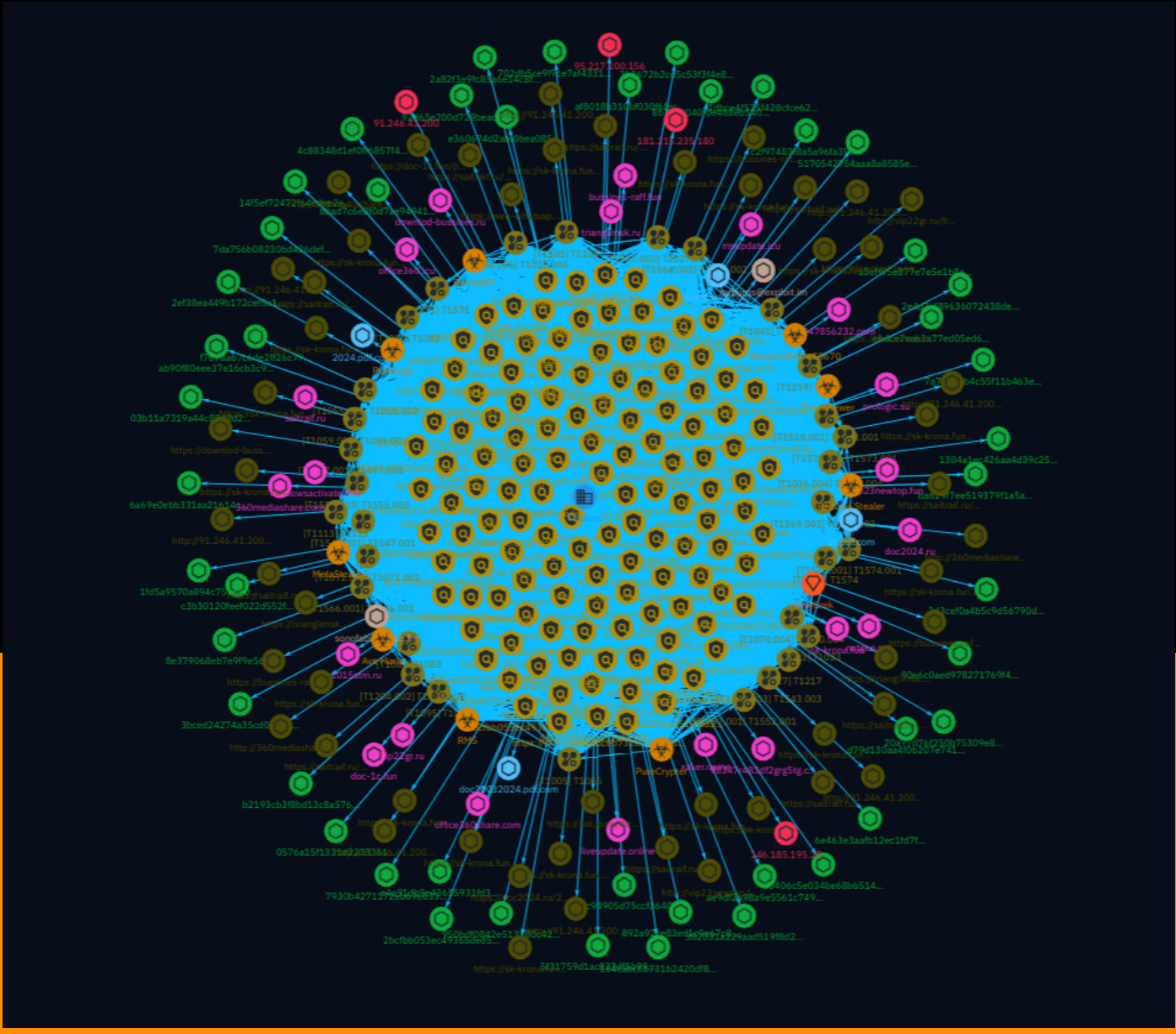

Table of contents

Overview

Entities

Observables

External References

Overview

Description

F.A.C.C.T experts analyzed the tools and connections of cybercriminals attacking Russian accountants. An analysis of the infection chain of the VasyGrek attacker, his forum activity and connection with the malware developer Mr.Burns is presented. The history of Mr.Burns, starting in 2010, is given, as well as a description of the current version of the BurnsRAT malware, sold on forums and used in attacks on Russian companies.

Confidence

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

100 / 100

Content

N/A

Attack-Pattern

Name
T1036.005
ID
T1036.005
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. Adversaries may also use the same icon of the file they are trying to mimic.
Name
T1070.004
ID
T1070.004

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](https://attack.mitre.org/techniques/T1105)) 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. There are tools available from the host operating system to perform cleanup, but adversaries may use other tools as well.(Citation: Microsoft SDelete July 2016) Examples of built-in [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) functions include `del` on Windows and `rm` or `unlink` on Linux and macOS.

Name

T1518.001

ID

T1518.001

Description

Adversaries may attempt to get a listing of security software, configurations, defensive tools, and sensors that are installed on a system or in a cloud environment. This may include things such as cloud monitoring agents and anti-virus. Adversaries may use the information from [Security Software Discovery](https://attack.mitre.org/techniques/T1518/001) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Example commands that can be used to obtain security software information are [netsh](https://attack.mitre.org/software/S0108), `reg query` with [Reg](https://attack.mitre.org/software/S0075), `dir` with [cmd](https://attack.mitre.org/software/S0106), and [Tasklist](https://attack.mitre.org/software/S0057), but other indicators of discovery behavior may be more specific to the type of software or security system the adversary is looking for. It is becoming more common to see macOS malware perform checks for LittleSnitch and KnockKnock software. Adversaries may also utilize the [Cloud API](https://attack.mitre.org/techniques/T1059/009) to discover cloud-native security software installed on compute infrastructure, such as the AWS CloudWatch agent, Azure VM Agent, and Google Cloud

Monitor agent. These agents may collect metrics and logs from the VM, which may be centrally aggregated in a cloud-based monitoring platform.

Name

T1571

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

T1555.003

ID

T1555.003

Description

Adversaries may acquire credentials from web browsers by reading files specific to the target browser.(Citation: Talos Olympic Destroyer 2018) Web browsers commonly save credentials such as website usernames and passwords so that they do not need to be entered manually in the future. Web browsers typically store the credentials in an encrypted format within a credential store; however, methods exist to extract plaintext credentials from web browsers. For example, on Windows systems, encrypted credentials

may be obtained from Google Chrome by reading a database file, `AppData\Local\Google\Chrome\User Data\Default\Login Data` and executing a SQL query: `SELECT action_url, username_value, password_value FROM logins;`. The plaintext password can then be obtained by passing the encrypted credentials to the Windows API function `CryptUnprotectData`, which uses the victim's cached logon credentials as the decryption key.(Citation: Microsoft CryptUnprotectData April 2018) Adversaries have executed similar procedures for common web browsers such as FireFox, Safari, Edge, etc. (Citation: Proofpoint Vega Credential Stealer May 2018)(Citation: FireEye HawkEye Malware July 2017) Windows stores Internet Explorer and Microsoft Edge credentials in Credential Lockers managed by the [Windows Credential Manager](https://attack.mitre.org/techniques/T1555/004). Adversaries may also acquire credentials by searching web browser process memory for patterns that commonly match credentials.(Citation: GitHub Mimikittenz July 2016) After acquiring credentials from web browsers, adversaries may attempt to recycle the credentials across different systems and/or accounts in order to expand access. This can result in significantly furthering an adversary's objective in cases where credentials gained from web browsers overlap with privileged accounts (e.g. domain administrator).

Name

T1217

ID

T1217

Description

Adversaries may enumerate information about browsers to learn more about compromised environments. Data saved by browsers (such as bookmarks, accounts, and browsing history) may reveal a variety of personal information about users (e.g., banking sites, relationships/interests, social media, etc.) as well as details about internal network resources such as servers, tools/dashboards, or other related infrastructure.(Citation: Kaspersky Autofill) Browser information may also highlight additional targets after an adversary has access to valid credentials, especially [Credentials In Files](https://attack.mitre.org/techniques/T1552/001) associated with logins cached by a browser. Specific storage locations vary based on platform and/or application, but browser information is typically stored in local files and databases (e.g., `%APPDATA%/Google/Chrome`).(Citation: Chrome Roaming Profiles)

Name

T1071.001

ID

T1071.001

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. Protocols such as HTTP/S(Citation: CrowdStrike Putter Panda) and WebSocket(Citation: Brazking-Websockets) that carry web traffic may be very common in environments. HTTP/S packets have many fields and headers in which data can be concealed. An adversary may abuse these protocols to communicate with systems under their control within a victim network while also mimicking normal, expected traffic.

Name

T1552.001

ID

T1552.001

Description

Adversaries may search local file systems and remote file shares for files containing insecurely stored credentials. These can be files created by users to store their own credentials, shared credential stores for a group of individuals, configuration files containing passwords for a system or service, or source code/binary files containing embedded passwords. It is possible to extract passwords from backups or saved virtual machines through [OS Credential Dumping](https://attack.mitre.org/techniques/T1003). (Citation: CG 2014) Passwords may also be obtained from Group Policy Preferences stored on the Windows Domain Controller.(Citation: SRD GPP) In cloud and/or containerized

environments, authenticated user and service account credentials are often stored in local configuration and credential files.(Citation: Unit 42 Hildegard Malware) They may also be found as parameters to deployment commands in container logs.(Citation: Unit 42 Unsecured Docker Daemons) In some cases, these files can be copied and reused on another machine or the contents can be read and then used to authenticate without needing to copy any files.(Citation: Specter Ops - Cloud Credential Storage)

Name

T1204.002

ID

T1204.002

Description

An adversary may rely upon a user opening a malicious file in order to gain execution. Users may be subjected to social engineering to get them to open a file that will lead to code execution. This user action will typically be observed as follow-on behavior from [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001). Adversaries may use several types of files that require a user to execute them, including .doc, .pdf, .xls, .rtf, .scr, .exe, .lnk, .pif, and .cpl. Adversaries may employ various forms of [Masquerading](https://attack.mitre.org/techniques/T1036) and [Obfuscated Files or Information](https://attack.mitre.org/techniques/T1027) to increase the likelihood that a user will open and successfully execute a malicious file. These methods may include using a familiar naming convention and/or password protecting the file and supplying instructions to a user on how to open it.(Citation: Password Protected Word Docs) While [Malicious File](https://attack.mitre.org/techniques/T1204/002) 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).

Name

T1566.001

ID

T1566.001

Description

Adversaries may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon [User Execution](https://attack.mitre.org/techniques/T1204) to gain execution. (Citation: Unit 42 DarkHydrus July 2018) Spearphishing may also involve social engineering techniques, such as posing as a trusted source. There are many options for the attachment such as Microsoft Office documents, executables, PDFs, or archived files. Upon opening the attachment (and potentially clicking past protections), the adversary's payload exploits a vulnerability or directly executes on the user's system. The text of the spearphishing email usually tries to give a plausible reason why the file should be opened, and may explain how to bypass system protections in order to do so. The email may also contain instructions on how to decrypt an attachment, such as a zip file password, in order to evade email boundary defenses. Adversaries frequently manipulate file extensions and icons in order to make attached executables appear to be document files, or files exploiting one application appear to be a file for a different one.

Name

T1059.003

ID

T1059.003

Description

Adversaries may abuse the Windows command shell for execution. The Windows command shell ([cmd](https://attack.mitre.org/software/S0106)) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via [Remote Services](https://attack.mitre.org/techniques/T1021) such as [SSH](https://attack.mitre.org/

techniques/T1021/004).(Citation: SSH in Windows) Batch files (ex: .bat or .cmd) also provide the shell with a list of sequential commands to run, as well as normal scripting operations such as conditionals and loops. Common uses of batch files include long or repetitive tasks, or the need to run the same set of commands on multiple systems. Adversaries may leverage [cmd](https://attack.mitre.org/software/S0106) to execute various commands and payloads. Common uses include [cmd](https://attack.mitre.org/software/S0106) to execute a single command, or abusing [cmd](https://attack.mitre.org/software/S0106) interactively with input and output forwarded over a command and control channel.

Name

T1005

ID

T1005

Description

Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration. Adversaries may do this using a [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), such as [cmd](https://attack.mitre.org/software/S0106) as well as a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008), which have functionality to interact with the file system to gather information.(Citation: show_run_config_cmd_cisco) Adversaries may also use [Automated Collection](https://attack.mitre.org/techniques/T1119) on the local system.

Name

T1574

ID

T1574

Description

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.

Name

T1105

ID

T1105

Description

Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](https://attack.mitre.org/software/S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](https://attack.mitre.org/techniques/T1570)). On Windows, adversaries may use various utilities to download tools, such as `copy`, `finger`, [certutil](https://attack.mitre.org/software/S0160), and [PowerShell](https://attack.mitre.org/techniques/T1059/001) commands such as `IEX(New-Object Net.WebClient).downloadString()` and `Invoke-WebRequest`. On Linux and macOS systems, a variety of utilities also exist, such as `curl`, `scp`, `sftp`, `tftp`, `rsync`, `finger`, and `wget`. (Citation: t1105_lolbas) Adversaries may also abuse installers and package managers, such as `yum` or `winget`, to download tools to victim hosts. Adversaries have also abused file application features, such as the Windows `search-ms` protocol handler, to deliver malicious files to victims through remote file searches invoked by [User Execution](https://attack.mitre.org/techniques/T1204) (typically after interacting with [Phishing](https://attack.mitre.org/techniques/T1566) lures).(Citation: T1105: Trellix_search-ms) Files can also be transferred using various [Web Service](https://attack.mitre.org/techniques/T1102)s as well as native or otherwise present tools on the victim system.(Citation: PTSecurity Cobalt

Dec 2016) In some cases, adversaries may be able to leverage services that sync between a web-based and an on-premises client, such as Dropbox or OneDrive, to transfer files onto victim systems. For example, by compromising a cloud account and logging into the service's web portal, an adversary may be able to trigger an automatic syncing process that transfers the file onto the victim's machine.(Citation: Dropbox Malware Sync)

Name

T1095

ID

T1095

Description

Adversaries may use an OSI non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive.(Citation: Wikipedia OSI) Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP communication between hosts is one example.(Citation: Cisco Synful Knock Evolution) Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts.(Citation: Microsoft ICMP) However, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.

Name

T1219

ID

T1219

Description

An adversary may use legitimate desktop support and remote access software to establish an interactive command and control channel to target systems within networks. These services, such as `VNC`, `Team Viewer`, `AnyDesk`, `ScreenConnect`, `LogMein`, `AmmyyAdmin`, and other remote monitoring and management (RMM) tools, are commonly used as legitimate technical support software and may be allowed by application control within a target environment.(Citation: Symantec Living off the Land)(Citation: CrowdStrike 2015 Global Threat Report)(Citation: CrySyS Blog TeamSpy) Remote access software may be installed and used post-compromise as an alternate communications channel for redundant access or as a way to establish an interactive remote desktop session with the target system. They may also be used as a component of malware to establish a reverse connection or back-connect to a service or adversary-controlled system. Adversaries may similarly abuse response features included in EDR and other defensive tools that enable remote access. Installation of many remote access software may also include persistence (e.g., the software's installation routine creates a [Windows Service](https://attack.mitre.org/techniques/T1543/003)). Remote access modules/features may also exist as part of otherwise existing software (e.g., Google Chrome’s Remote Desktop).(Citation: Google Chrome Remote Desktop)(Citation: Chrome Remote Desktop)

Name

T1055.002

ID

T1055.002

Description

Adversaries may inject portable executables (PE) into processes in order to evade process-based defenses as well as possibly elevate privileges. PE injection is a method of executing arbitrary code in the address space of a separate live process. PE injection is commonly performed by copying code (perhaps without a file on disk) into the virtual address space of the target process before invoking it via a new thread. The write can be performed with native Windows API calls such as `VirtualAllocEx` and `WriteProcessMemory`, then invoked with `CreateRemoteThread` or additional code (ex: shellcode). The displacement of the injected code does introduce the additional requirement for functionality to remap memory references. (Citation: Elastic Process Injection July 2017) 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 PE injection may also evade detection from security products since the execution is masked under a legitimate process.

Name

T1547.001

ID

T1547.001

Description

Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. (Citation: Microsoft Run Key) These programs will be executed under the context of the user and will have the account's associated permissions level. The following run keys are created by default on Windows systems: * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce` Run keys may exist under multiple hives.(Citation: Microsoft Wow6432Node 2018)(Citation: Malwarebytes Wow6432Node 2016) The `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnceEx` is also available but is not created by default on Windows Vista and newer. Registry run key entries can reference programs directly or list them as a dependency.(Citation: Microsoft Run Key) For example, it is possible to load a DLL at logon using a "Depend" key with RunOnceEx: `reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\0001\Depend /v 1 /d "C:\temp\evil[.]dll"` (Citation: Oddvar Moe RunOnceEx Mar 2018) Placing a program within a startup folder will also cause that program to execute when a user logs in. There is a startup folder location for individual user accounts as well as a system-wide startup folder that will be checked regardless of which user account logs in. The startup folder path for the current user is `C:\Users\[Username]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`. The startup folder path for all users is `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp`. The following Registry keys can be used to set startup folder items for persistence: * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders` *

`HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders` * `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders` * `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders` The following Registry keys can control automatic startup of services during boot: * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce` * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices` Using policy settings to specify startup programs creates corresponding values in either of two Registry keys: * `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run` * `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run` Programs listed in the load value of the registry key `HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows` run automatically for the currently logged-on user. By default, the multistring `BootExecute` value of the registry key `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager` is set to `autocheck autochk *`. This value causes Windows, at startup, to check the file-system integrity of the hard disks if the system has been shut down abnormally. Adversaries can add other programs or processes to this registry value which will automatically launch at boot. Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use [Masquerading](https://attack.mitre.org/techniques/T1036) to make the Registry entries look as if they are associated with legitimate programs.

Name

T1566.002

ID

T1566.002

Description

Adversaries may send spearphishing emails with a malicious link in an attempt to gain access to victim systems. Spearphishing with a link is a specific variant of spearphishing. It

is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attaching malicious files to the email itself, to avoid defenses that may inspect email attachments. Spearphishing may also involve social engineering techniques, such as posing as a trusted source. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this case, the malicious emails contain links. Generally, the links will be accompanied by social engineering text and require the user to actively click or copy and paste a URL into a browser, leveraging [User Execution](https://attack.mitre.org/techniques/T1204). The visited website may compromise the web browser using an exploit, or the user will be prompted to download applications, documents, zip files, or even executables depending on the pretext for the email in the first place. Adversaries may also include links that are intended to interact directly with an email reader, including embedded images intended to exploit the end system directly. Additionally, adversaries may use seemingly benign links that abuse special characters to mimic legitimate websites (known as an "IDN homograph attack").(Citation: CISA IDN ST05-016) URLs may also be obfuscated by taking advantage of quirks in the URL schema, such as the acceptance of integer- or hexadecimal-based hostname formats and the automatic discarding of text before an “@” symbol: for example, `hxxp://google.com@1157586937`.(Citation: Mandiant URL Obfuscation 2023) Adversaries may also utilize links to perform consent phishing, typically with OAuth 2.0 request URLs that when accepted by the user provide permissions/access for malicious applications, allowing adversaries to [Steal Application Access Token](https://attack.mitre.org/techniques/T1528)s.(Citation: Trend Micro Pawn Storm OAuth 2017) These stolen access tokens allow the adversary to perform various actions on behalf of the user via API calls. (Citation: Microsoft OAuth 2.0 Consent Phishing 2021) Adversaries may also utilize spearphishing links to [Steal Application Access Token](https://attack.mitre.org/techniques/T1528)s that grant immediate access to the victim environment. For example, a user may be lured through “consent phishing” into granting adversaries permissions/access via a malicious OAuth 2.0 request URL .(Citation: Trend Micro Pawn Storm OAuth 2017)(Citation: Microsoft OAuth 2.0 Consent Phishing 2021) Similarly, malicious links may also target device-based authorization, such as OAuth 2.0 device authorization grant flow which is typically used to authenticate devices without UIs/browsers. Known as “device code phishing,” an adversary may send a link that directs the victim to a malicious authorization page where the user is tricked into entering a code/credentials that produces a device token.(Citation: SecureWorks Device Code Phishing 2021)(Citation: Netskope Device Code Phishing 2021) (Citation: Optiv Device Code Phishing 2021)

Name

T1033

ID

T1033

Description

Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping](https://attack.mitre.org/techniques/T1003). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery](https://attack.mitre.org/techniques/T1033) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Various utilities and commands may acquire this information, including `whoami`. In macOS and Linux, the currently logged in user can be identified with `w` and `who`. On macOS the `dscl . list /Users | grep -v '_'` command can also be used to enumerate user accounts. Environment variables, such as `%USERNAME%` and `$USER`, may also be used to access this information. On network devices, [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `show users` and `show ssh` can be used to display users currently logged into the device.(Citation: show_ssh_users_cmd_cisco)(Citation: US-CERT TA18-106A Network Infrastructure Devices 2018)

Name

T1497.001

ID

T1497.001

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](https://attack.mitre.org/techniques/T1497) during automated discovery to shape follow-on behaviors.(Citation: Deloitte Environment Awareness) Specific checks will vary based on the target and/or adversary, but may involve behaviors such as [Windows Management Instrumentation](https://attack.mitre.org/techniques/T1047), [PowerShell](https://attack.mitre.org/techniques/T1059/001), [System Information Discovery](https://attack.mitre.org/techniques/T1082), and [Query Registry](https://attack.mitre.org/techniques/T1012) to obtain system information and search for VME artifacts. Adversaries may search for VME artifacts in memory, processes, file system, hardware, and/or the Registry. Adversaries may use scripting to automate these checks into one script and then have the program exit if it determines the system to be a virtual environment. Checks could include generic system properties such as host/domain name and samples of network traffic. Adversaries may also check the network adapters addresses, CPU core count, and available memory/drive size. Once executed, malware may also use [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) to check if it was saved in a folder or file with unexpected or even analysis-related naming artifacts such as `malware`, `sample`, or `hash`. Other common checks may enumerate services running that are unique to these applications, installed programs on the system, manufacturer/product fields for strings relating to virtual machine applications, and VME-specific hardware/processor instructions.(Citation: McAfee Virtual Jan 2017) In applications like VMWare, adversaries can also use a special I/O port to send commands and receive output. Hardware checks, such as the presence of the fan, temperature, and audio devices, could also be used to gather evidence that can be indicative a virtual environment. Adversaries may also query for specific readings from these devices.(Citation: Unit 42 OilRig Sept 2018)

Name

T1036.004

ID

T1036.004

Description

Adversaries may attempt to manipulate the name of a task or service to make it appear legitimate or benign. Tasks/services executed by the Task Scheduler or systemd will typically be given a name and/or description.(Citation: TechNet Schtasks)(Citation: Systemd Service Units) Windows services will have a service name as well as a display name. Many benign tasks and services exist that have commonly associated names. Adversaries may give tasks or services names that are similar or identical to those of

legitimate ones. Tasks or services contain other fields, such as a description, that adversaries may attempt to make appear legitimate.(Citation: Palo Alto Shamoon Nov 2016) (Citation: Fysbis Dr Web Analysis)

Name

T1573.001

ID

T1573.001

Description

Adversaries may employ a known symmetric encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Symmetric encryption algorithms use the same key for plaintext encryption and ciphertext decryption. Common symmetric encryption algorithms include AES, DES, 3DES, Blowfish, and RC4.

Name

T1543.003

ID

T1543.003

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.(Citation: TechNet Services) Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry. Adversaries may install a new service or modify an existing service to execute at startup in order to persist on a system. Service configurations can be set or modified using system utilities (such as sc.exe), by directly modifying the Registry, or by interacting directly with

the Windows API. Adversaries may also use services to install and execute malicious drivers. For example, after dropping a driver file (ex: `.sys`) to disk, the payload can be loaded and registered via [Native API](https://attack.mitre.org/techniques/T1106) functions such as `CreateServiceW()` (or manually via functions such as `ZwLoadDriver()` and `ZwSetValueKey()`), by creating the required service Registry values (i.e. [Modify Registry](https://attack.mitre.org/techniques/T1112)), or by using command-line utilities such as `PnPUtil.exe`.(Citation: Symantec W.32 Stuxnet Dossier)(Citation: Crowdstrike DriveSlayer February 2022)(Citation: Unit42 AcidBox June 2020) Adversaries may leverage these drivers as [Rootkit](https://attack.mitre.org/techniques/T1014)s to hide the presence of malicious activity on a system. Adversaries may also load a signed yet vulnerable driver onto a compromised machine (known as "Bring Your Own Vulnerable Driver" (BYOVD)) as part of [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068).(Citation: ESET InvisiMole June 2020)(Citation: Unit42 AcidBox June 2020) Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges. Adversaries may also directly start services through [Service Execution](https://attack.mitre.org/techniques/T1569/002). To make detection analysis more challenging, malicious services may also incorporate [Masquerade Task or Service](https://attack.mitre.org/techniques/T1036/004) (ex: using a service and/or payload name related to a legitimate OS or benign software component). Adversaries may also create ‘hidden’ services (i.e., [Hide Artifacts](https://attack.mitre.org/techniques/T1564)), for example by using the `sc sdset` command to set service permissions via the Service Descriptor Definition Language (SDDL). This may hide a Windows service from the view of standard service enumeration methods such as `Get-Service`, `sc query`, and `services.exe`.(Citation: SANS 1)(Citation: SANS 2)

Name

T1027.002

ID

T1027.002

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.(Citation: ESET FinFisher Jan 2018) Utilities used to perform software packing are called packers. Example packers are MPRESS and UPX. A more comprehensive list of known packers is available, but adversaries may create their own packing techniques that do not leave the same artifacts as well-known packers to evade defenses.(Citation: Awesome Executable Packing)

Name

T1569.002

ID

T1569.002

Description

Adversaries may abuse the Windows service control manager to execute malicious commands or payloads. The Windows service control manager (`services.exe`) is an interface to manage and manipulate services.(Citation: Microsoft Service Control Manager) The service control manager is accessible to users via GUI components as well as system utilities such as `sc.exe` and [Net](https://attack.mitre.org/software/S0039). [PsExec](https://attack.mitre.org/software/S0029) can also be used to execute commands or payloads via a temporary Windows service created through the service control manager API.(Citation: Russinovich Sysinternals) Tools such as [PsExec](https://attack.mitre.org/software/S0029) and `sc.exe` can accept remote servers as arguments and may be used to conduct remote execution. Adversaries may leverage these mechanisms to execute malicious content. This can be done by either executing a new or modified service. This technique is the execution used in conjunction with [Windows Service](https://attack.mitre.org/techniques/T1543/003) during service persistence or privilege escalation.

Name

T1083

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) Some files and directories may require elevated or specific user permissions to access.

Name

T1574.001

ID

T1574.001

Description

Adversaries may execute their own malicious payloads by hijacking the search order used to load DLLs. Windows systems use a common method to look for required DLLs to load into a program. (Citation: Microsoft Dynamic Link Library Search Order)(Citation: FireEye Hijacking July 2010) Hijacking DLL loads may be for the purpose of establishing persistence as well as elevating privileges and/or evading restrictions on file execution. There are many ways an adversary can hijack DLL loads. Adversaries may plant trojan dynamic-link library files (DLLs) in a directory that will be searched before the location of a legitimate library that will be requested by a program, causing Windows to load their malicious library when it is called for by the victim program. Adversaries may also perform DLL preloading, also called binary planting attacks, (Citation: OWASP Binary Planting) by placing a malicious DLL with the same name as an ambiguously specified DLL in a location that Windows searches before the legitimate DLL. Often this location is the current working directory of the program.(Citation: FireEye fxsst June 2011) Remote DLL preloading attacks occur when a program sets its current directory to a remote location such as a Web share before loading a DLL. (Citation: Microsoft Security Advisory 2269637) Phantom

DLL hijacking is a specific type of DLL search order hijacking where adversaries target references to non-existent DLL files.(Citation: Adversaries Hijack DLLs) They may be able to load their own malicious DLL by planting it with the correct name in the location of the missing module. Adversaries may also directly modify the search order via DLL redirection, which after being enabled (in the Registry and creation of a redirection file) may cause a program to load a different DLL.(Citation: Microsoft Dynamic-Link Library Redirection) (Citation: Microsoft Manifests)(Citation: FireEye DLL Search Order Hijacking) If a search order-vulnerable program is configured to run at a higher privilege level, then the adversary-controlled DLL that is loaded will also be executed at the higher level. In this case, the technique could be used for privilege escalation from user to administrator or SYSTEM or from administrator to SYSTEM, depending on the program. Programs that fall victim to path hijacking may appear to behave normally because malicious DLLs may be configured to also load the legitimate DLLs they were meant to replace.

Name

T1113

ID

T1113

Description

Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations. Taking a screenshot is also typically possible through native utilities or API calls, such as `CopyFromScreen`, `xwd`, or `screencapture`.(Citation: CopyFromScreen .NET)(Citation: Antiquated Mac Malware)

Name

T1082

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)

Name

T1041

ID

T1041

Description

Adversaries may steal data by exfiltrating it over an existing command and control channel. Stolen data is encoded into the normal communications channel using the same protocol as command and control communications.

Domain-Name

Value
downlod-bussines.ru
natgeo.pro
xaker.name
trianglimsk.ru
vip22gr.ru
doc2024.ru
office360.icu
prologic.su
vip23newtop.fun
office360share.com
liveupdate.online
saitraif.ru
mts2015stm.ru

98347r483df2grg5tg.com
360mediashare.com
sk-krona.fun
doc-1c.fun
windowsactivate.link
bussines-raff.fun
msupdate.icu
047856232.com

Email-Addr

Value
mrburns@exploit.im
sonofabitch@ua.fm

StixFile

Value
bbad7c6e8f0d7ae94941257e7ece4d2b144aad56e25760c8876b808f3e8420e6
3bced24274a35cd08a3698e32623a14a319fbb60f4f9a950d41834710393c32f
90e6c0aed978271769f4fface9a27edbb8d72cd463cfd57b443710aa703a1f98
ab90f80eee37e16cb3c94f524e2fde3fe13669386512ea36b4ad6ac4d9fbf773
2e4d3cf89636072438deb7e690ea376e8433c5dc59d8befedc0f5b79ea9a6b7d
03b11a7319a44c8848d239b8ce49ebb43ebe90dfb9927771a2258bbe3d0e655e
7930b4271172eb69e63349282bfe62a111a6e0a8bc8b23ae8729ab6be006ecf5
05406c5e034be68b6514fc3ae1b31f603ec7d1865963fe0716ed48605af0fd98
2bcfbb053ec4936bded589848b8429cd37b0a7bf5bf85e5e3ace494f4512bfa9
ae9df2b98a9e5561c749cc96a4e24f9d5bb0451889a3924fd7ed73436466495f
1304a1ec426aa4d39c255aef059bc5b2cb9fef096cd6d136c63ddf8a3b936b96
7a79bb8b4c55f11b463efee0c8cbfaf24c85daac04b67f4f4c25f6851dda57df
af8018b310bf030f6feca0f6f23d3e65f8926114d7cd493573badae24f5da0d1

892a92ce83ed1c9e67c8f7ab0120d1f28e1dfd3a93146da3fde6e9226e22222b

950bdf0842e513180c42ab3809e57c0779456c51a53e41ce8e833ed36880230e

ba629f7ee519379f1a5a8a4683ee9a48d1b0996268bfaf1162e4bf0f2b792b77

8b7e5a040f0e468eb540211a3ac73dadd6628177dc09eaff06bfbce10c6eeab9

a5eff95e877e7e5e1b8a57e3169cb6f545ae353ed1908840dabb9554ff001500

92d65e200d729beac212563a7559fbdc657a4832d462e02dab4d937b5571983c

6a69e0ebb331aa21614ccc0c4028b5cde242f0710300fe7b441b2017c71a8e16

0576a15f1331d220336163510cc71deb37d1ae0b57ff6ad661c5e547086b57e2

2a82f3e9fc83a6e14c8ff13ed5d450580235981958a7bd262c7ea597e1c94078

c2f97483f8a5a96fa39e8bd3d3458093ac527a8c8efd662e838d95a9bc2354fb

4c88348d1ef0ff6857f48761ac82d8455661849b34e4f4a6bc07a765818361a3

5170542754aaa8a8585e4d7c12f77deb7fc0cb24ec6626d53e3fa9997e303e77

2ef38ea449b172cef5e1015bc4b5e37de8ece7d4be087b6bdded5a992493e7aa

164cabc6b731b2420df8a0fa8e4a2590e45cc027d9cf72ccc74252383ec0f65b

c3b30120feef022d552f85b780d4c988ee82bc07e6b5948db5d32e59d44fa704

14f5ef72472f64edee2e852d1c677ad4f61b780c3ac93649835c4cc30f5c5b2f

e360674d2abf0bea085d01bc3595e19efb3ac061ab8090a32d0c579c621c46f6

6e463e3aafb12ec1fd7ff347038b3df15a93b3b2c506c9d670498b0937d6dce7

d79d130aa4f0b207e741909c45be613a1e3720cb82a0578012cc508c28da6bad

1fd5a9570a894c751610c1b49b2f2f00c0c618d365be14a4980f1266a3772c90

1dbce4f525f428cfce626726209ca973f2fdb93cd905a94a1bc538f75e0a16ca

20a77d76f250b75309e8ccaf1470d9729dc99b95168085ff30b1e46be6ce2138

382031a229aad519f8d243923e504e8dedf0106f4ce274ab9640ce55542b962d

f7878a67c6de2ff26c79ab890e4a60b76c67a7583c6a24bd96cd93a5f4a0e0aa

5f31759d1ac833df5b990b436dabb88cf3e85ba7495440a62364723bc8490907

bf9fc94905d75ccf3640d35899d533e50c7ba8bdce396443ae2d0507657a9e81

7da756b08230bd426defeaea35588b899057228ac19f3a21625582038e405c76

e4a91db9e43655931fd3926ec00dbe8a063fbe0d3f0af7d902fd3b9d8281fb3d

ebdce7eae3a77ed05ed6279c46a8be8c560085f82ce0f9e4de0ad8c700c16fc4

3b8672b2cd5c53f3f4e823ed3873d930b5786a05cc7f2d49b07cb5bda21d933e

3d3cef0a4b5c9d56790dbb8c8ac838d42caac2171f5435495682a51c45160bc3

8e379068eb7e9f9e5635531526dacdc03bf505e67775dd186edba27b33a93805

b2193cb3f8bd13c8a5769d5ce499a36b9c44e2eb2800bcdf22320525beaf9586

External References

- https://www.facct.ru/blog/vasygrek-and-mr-burns/
- https://otx.alienvault.com/pulse/668e59166cfc4e877eedb26c