Apr 15 2024

NETMANAGEIT

Intelligence Report
Don't Take the Bait: The XWorm Tax Scam

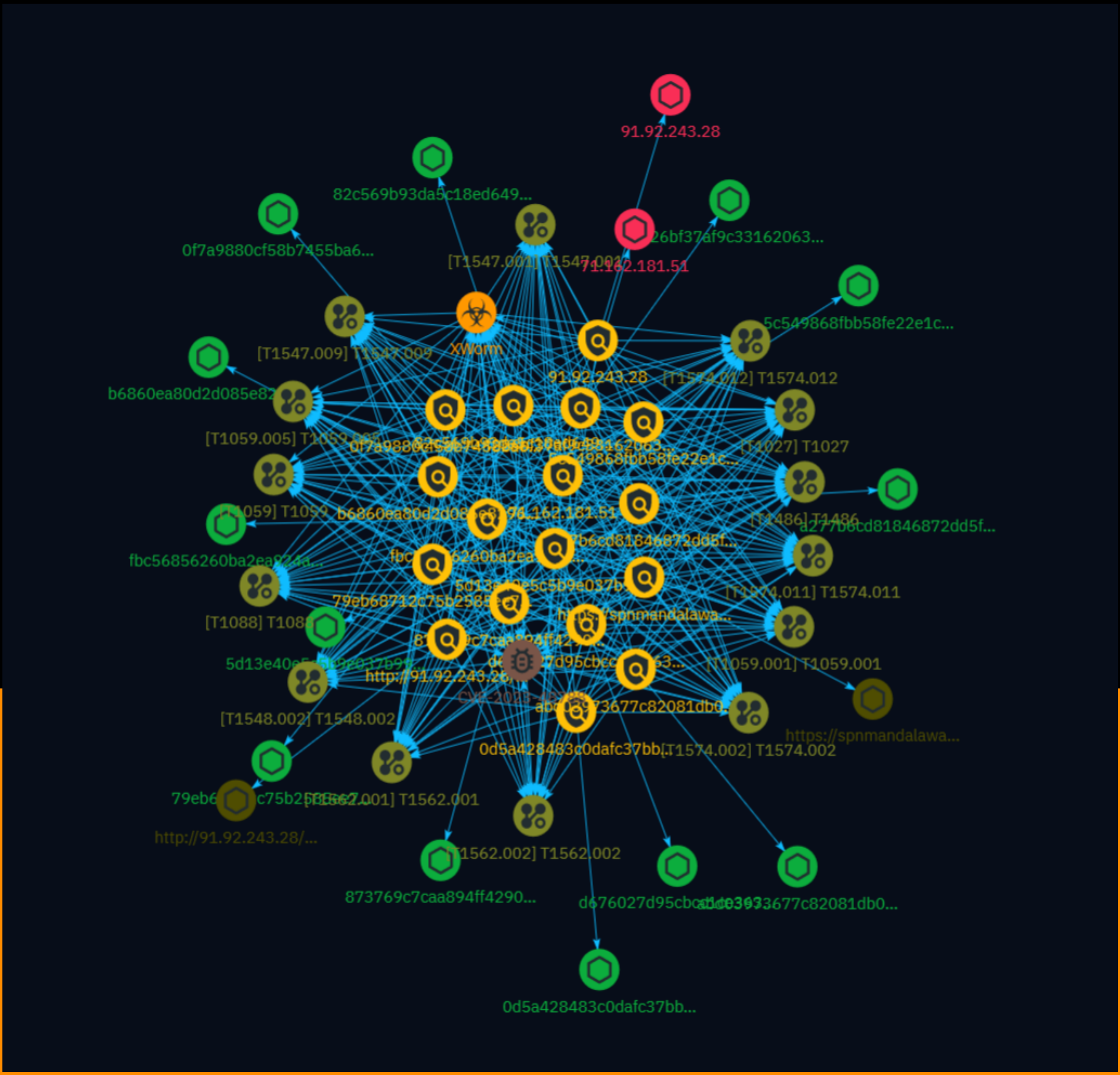

Table of contents

Overview

- Description 4
- Confidence 4
- Content 5

Entities

- Indicator 6
- Attack-Pattern 14
- Vulnerability 27
- Malware 28

Observables

- IPv4-Addr 29
- StixFile 30
- Url 31

External References

● External References 32

Overview

Description

This report provides insights into a recent tax-themed threat campaign that delivers the XWorm Remote Access Trojan as the final payload. The initial infection vector is a phishing email with a malicious JavaScript attachment that executes a multi-stage process involving process termination, disabling security features, establishing persistence, and ultimately deploying the XWorm RAT. The tactics employed by the attackers aim to evade detection and maintain long-term access to compromised systems, highlighting the significance of implementing robust security measures.

Confidence

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

100 / 100

Content

N/A

Indicator

Name
71.162.181.51
Description
- **Zip Code:** N/A - **ISP:** Verizon Fios Business - **ASN:** 701 - **Organization:** Verizon Fios Business - **Is Crawler:** False - **Timezone:** America/New_York - **Mobile:** False - **Host:** static-71-162-181-51.phlapa.fios.verizon.net - **Proxy:** True - **VPN:** True - **TOR:** False - **Active VPN:** False - **Active TOR:** False - **Recent Abuse:** False - **Bot Status:** False - **Connection Type:** Premium required. - **Abuse Velocity:** Premium required. - **Country Code:** US - **Region:** Pennsylvania - **City:** Brookhaven - **Latitude:** 39.86669922 - **Longitude:** -75.38619995
Pattern Type
stix
Pattern
[ipv4-addr:value = '71.162.181.51']
Name
fbc56856260ba2ea924a94a301616540addd202d8840539613aece22780c473c
Pattern Type

stix
Pattern
[file:hashes.'SHA-256' = 'fbc56856260ba2ea924a94a301616540addd202d8840539613aece22780c473c']
Name
f226bf37af9c33162063db3eb018fed7f088f86d0a20ca54c013fda96c7f2e05
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'f226bf37af9c33162063db3eb018fed7f088f86d0a20ca54c013fda96c7f2e05']
Name
d676027d95cbcc1ce363268fe213fad71beeec1be359950518739b61507763fd
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'd676027d95cbcc1ce363268fe213fad71beeec1be359950518739b61507763fd']
Name

b6860ea80d2d085e8296692579b886d0218cff29b1f6702f40e5543c86dafe3c
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'b6860ea80d2d085e8296692579b886d0218cff29b1f6702f40e5543c86dafe3c']
Name
a277b6cd81846872dd5f6d92f815b506cac7c38d0327f9e8303d1e4328134f98
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'a277b6cd81846872dd5f6d92f815b506cac7c38d0327f9e8303d1e4328134f98']
Name
873769c7caa894ff42904def7ad206473fa726b6598522b1f466416c8bbfd6cd
Pattern Type
stix
Pattern

[file:hashes.'SHA-256' = '873769c7caa894ff42904def7ad206473fa726b6598522b1f466416c8bbfd6cd']

Name

82c569b93da5c18ed649ebd4c2c79437db4611a6a1373e805a3cb001c64130b7

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '82c569b93da5c18ed649ebd4c2c79437db4611a6a1373e805a3cb001c64130b7']

Name

5d13e40e5c5b9e037b99fb4f81769d0e8505be2dae8ffdf8013f236c62dad220

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '5d13e40e5c5b9e037b99fb4f81769d0e8505be2dae8ffdf8013f236c62dad220']

Name

79eb68712c75b2585ee74cb782cb3369b2f2c6fc09c9abbdbeb096fd0ccbd401

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '79eb68712c75b2585ee74cb782cb3369b2f2c6fc09c9abbdbeb096fd0ccbd401']

Name

5c549868fbb58fe22e1c37b99699783fab996cec6c0e83c897f722a754f89def

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '5c549868fbb58fe22e1c37b99699783fab996cec6c0e83c897f722a754f89def']

Name

0d5a428483c0dafc37bb01ea323958d4eb303edc80d307a3e4898c6e22d6c12f

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0d5a428483c0dafc37bb01ea323958d4eb303edc80d307a3e4898c6e22d6c12f']

Name

0f7a9880cf58b7455ba6ed2de808ed38eeef9ad34b28ff95e63d63b4f2a51e51

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '0f7a9880cf58b7455ba6ed2de808ed38eeef9ad34b28ff95e63d63b4f2a51e51']

Name

https://spnmandalawangi.banten.polri.go.id/Tax_docs_2023.htm

Description

- **Unsafe:** False - **Server:** Apache - **Domain Rank:** 20817 - **DNS Valid:** True - **Parking:** False - **Spamming:** False - **Malware:** False - **Phishing:** False - **Suspicious:** False - **Adult:** False - **Category:** Politics, society and law - **Domain Age:** {'human': '24 years ago', 'timestamp': 957979775, 'iso': '2000-05-10T13:29:35-04:00'} - **IPQS: Domain:** spnmandalawangi.banten.polri.go.id - **IPQS: IP Address:** 120.29.231.201

Pattern Type

stix

Pattern

[url:value = 'https://spnmandalawangi.banten.polri.go.id/Tax_docs_2023.htm']

Name

http://91.92.243.28/poom/atom.xml

Description
- **Unsafe:** False - **Server:** N/A - **Domain Rank:** 0 - **DNS Valid:** False - **Parking:** False - **Spamming:** False - **Malware:** False - **Phishing:** False - **Suspicious:** True - **Adult:** False - **Category:** N/A - **Domain Age:** {'human': 'N/A', 'timestamp': None, 'iso': None} - **IPQS: Domain:** 91.92.243.28 - **IPQS: IP Address:** N/A

Pattern Type
stix

Pattern
[url:value = 'http://91.92.243.28/poom/atom.xml']

Name
91.92.243.28

Description
- **Zip Code:** N/A - **ISP:** LIMENET - **ASN:** 394711 - **Organization:** LIMENET - **Is Crawler:** False - **Timezone:** Europe/Amsterdam - **Mobile:** False - **Host:** 91.92.243.28 - **Proxy:** True - **VPN:** False - **TOR:** False - **Active VPN:** False - **Active TOR:** False - **Recent Abuse:** False - **Bot Status:** False - **Connection Type:** Premium required. - **Abuse Velocity:** Premium required. - **Country Code:** NL - **Region:** Noord-Holland - **City:** Amsterdam - **Latitude:** 52.34999847 - **Longitude:** 4.92000008

Pattern Type
stix

Pattern

[ipv4-addr:value = '91.92.243.28']
Name
abd03973677c82081db0b0af02ca302bb5a854f96c52acd2113ec110b373daa6
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'abd03973677c82081db0b0af02ca302bb5a854f96c52acd2113ec110b373daa6']

Attack-Pattern

Name
T1088
ID
T1088
Name
T1574.012
ID
T1574.012
Description
Adversaries may leverage the COR_PROFILER environment variable to hijack the execution flow of programs that load the .NET CLR. The COR_PROFILER is a .NET Framework feature which allows developers to specify an unmanaged (or external of .NET) profiling DLL to be loaded into each .NET process that loads the Common Language Runtime (CLR). These profilers are designed to monitor, troubleshoot, and debug managed code executed by the .NET CLR.(Citation: Microsoft Profiling Mar 2017)(Citation: Microsoft COR_PROFILER Feb 2013) The COR_PROFILER environment variable can be set at various scopes (system, user, or process) resulting in different levels of influence. System and user-wide environment variable scopes are specified in the Registry, where a [Component Object Model](https://attack.mitre.org/techniques/T1559/001) (COM) object can be registered as a profiler DLL. A process scope COR_PROFILER can also be created in-memory without modifying the

Registry. Starting with .NET Framework 4, the profiling DLL does not need to be registered as long as the location of the DLL is specified in the COR_PROFILER_PATH environment variable.(Citation: Microsoft COR_PROFILER Feb 2013) Adversaries may abuse COR_PROFILER to establish persistence that executes a malicious DLL in the context of all .NET processes every time the CLR is invoked. The COR_PROFILER can also be used to elevate privileges (ex: [Bypass User Account Control](https://attack.mitre.org/techniques/T1548/002)) if the victim .NET process executes at a higher permission level, as well as to hook and [Impair Defenses](https://attack.mitre.org/techniques/T1562) provided by .NET processes.(Citation: RedCanary Mockingbird May 2020)(Citation: Red Canary COR_PROFILER May 2020)(Citation: Almond COR_PROFILER Apr 2019)(Citation: GitHub OmerYa Invisi-Shell)(Citation: subTee .NET Profilers May 2017)

Name

T1574.002

ID

T1574.002

Description

Adversaries may execute their own malicious payloads by side-loading DLLs. Similar to [DLL Search Order Hijacking](https://attack.mitre.org/techniques/T1574/001), side-loading involves hijacking which DLL a program loads. But rather than just planting the DLL within the search order of a program then waiting for the victim application to be invoked, adversaries may directly side-load their payloads by planting then invoking a legitimate application that executes their payload(s). Side-loading takes advantage of the DLL search order used by the loader by positioning both the victim application and malicious payload(s) alongside each other. Adversaries likely use side-loading as a means of masking actions they perform under a legitimate, trusted, and potentially elevated system or software process. Benign executables used to side-load payloads may not be flagged during delivery and/or execution. Adversary payloads may also be encrypted/packed or otherwise obfuscated until loaded into the memory of the trusted process.(Citation: FireEye DLL Side-Loading)

Name

T1059.005

ID
T1059.005

Description
Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as [Component Object Model](https://attack.mitre.org/techniques/T1559/001) and the [Native API](https://attack.mitre.org/techniques/T1106) through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.(Citation: VB .NET Mar 2020)(Citation: VB Microsoft) Derivative languages based on VB have also been created, such as Visual Basic for Applications (VBA) and VBScript. VBA is an event-driven programming language built into Microsoft Office, as well as several third-party applications.(Citation: Microsoft VBA)(Citation: Wikipedia VBA) VBA enables documents to contain macros used to automate the execution of tasks and other functionality on the host. VBScript is a default scripting language on Windows hosts and can also be used in place of [JavaScript](https://attack.mitre.org/techniques/T1059/007) on HTML Application (HTA) webpages served to Internet Explorer (though most modern browsers do not come with VBScript support).(Citation: Microsoft VBScript) Adversaries may use VB payloads to execute malicious commands. Common malicious usage includes automating execution of behaviors with VBScript or embedding VBA content into [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001) payloads (which may also involve [Mark-of-the-Web Bypass](https://attack.mitre.org/techniques/T1553/005) to enable execution).(Citation: Default VBS macros Blocking)

Name
T1486

ID
T1486

Description
Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to

render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.(Citation: US-CERT Ransomware 2016)(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017)(Citation: US-CERT SamSam 2018) In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted (and often renamed and/or tagged with specific file markers). Adversaries may need to first employ other behaviors, such as [File and Directory Permissions Modification](https://attack.mitre.org/techniques/T1222) or [System Shutdown/Reboot](https://attack.mitre.org/techniques/T1529), in order to unlock and/or gain access to manipulate these files.(Citation: CarbonBlack Conti July 2020) In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.(Citation: US-CERT NotPetya 2017) To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like [Valid Accounts](https://attack.mitre.org/techniques/T1078), [OS Credential Dumping](https://attack.mitre.org/techniques/T1003), and [SMB/Windows Admin Shares](https://attack.mitre.org/techniques/T1021/002).(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017) Encryption malware may also leverage [Internal Defacement](https://attack.mitre.org/techniques/T1491/001), such as changing victim wallpapers, or otherwise intimidate victims by sending ransom notes or other messages to connected printers (known as "print bombing").(Citation: NHS Digital Egregor Nov 2020) In cloud environments, storage objects within compromised accounts may also be encrypted.(Citation: Rhino S3 Ransomware Part 1)

Name

T1562.001

ID

T1562.001

Description

Adversaries may modify and/or disable security tools to avoid possible detection of their malware/tools and activities. This may take many forms, such as killing security software processes or services, modifying / deleting Registry keys or configuration files so that tools do not operate properly, or other methods to interfere with security tools scanning or reporting information. Adversaries may also disable updates to prevent the latest security patches from reaching tools on victim systems.(Citation: SCADAfence_ransomware)

Adversaries may also tamper with artifacts deployed and utilized by security tools. Security tools may make dynamic changes to system components in order to maintain visibility into specific events. For example, security products may load their own modules and/or modify those loaded by processes to facilitate data collection. Similar to [Indicator Blocking](https://attack.mitre.org/techniques/T1562/006), adversaries may unhook or otherwise modify these features added by tools (especially those that exist in userland or are otherwise potentially accessible to adversaries) to avoid detection.(Citation: OutFlank System Calls)(Citation: MDSec System Calls) Adversaries may also focus on specific applications such as Sysmon. For example, the "Start" and "Enable" values in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Microsoft-Windows-Sysmon-Operational` may be modified to tamper with and potentially disable Sysmon logging.(Citation: disable_win_evt_logging) On network devices, adversaries may attempt to skip digital signature verification checks by altering startup configuration files and effectively disabling firmware verification that typically occurs at boot.(Citation: Fortinet Zero-Day and Custom Malware Used by Suspected Chinese Actor in Espionage Operation)(Citation: Analysis of FG-IR-22-369) In cloud environments, tools disabled by adversaries may include cloud monitoring agents that report back to services such as AWS CloudWatch or Google Cloud Monitor. Furthermore, although defensive tools may have anti-tampering mechanisms, adversaries may abuse tools such as legitimate rootkit removal kits to impair and/or disable these tools.(Citation: chasing_avaddon_ransomware)(Citation: dharma_ransomware)(Citation: demystifying_ryuk)(Citation: doppelpaymer_crowdstrike) For example, adversaries have used tools such as GMER to find and shut down hidden processes and antivirus software on infected systems.(Citation: demystifying_ryuk) Additionally, adversaries may exploit legitimate drivers from anti-virus software to gain access to kernel space (i.e. [Exploitation for Privilege Escalation](https://attack.mitre.org/techniques/T1068)), which may lead to bypassing anti-tampering features.(Citation: avoslocker_ransomware)

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

T1059.001

ID

T1059.001

Description

Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system.(Citation: TechNet PowerShell) Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the `Start-Process` cmdlet which can be used to run an executable and the `Invoke-Command` cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems). PowerShell may also be used to download and run executables from the Internet, which can be executed from disk or in memory without touching disk. A number of PowerShell-based offensive testing tools are available, including [Empire](https://attack.mitre.org/software/S0363), [PowerSploit](https://attack.mitre.org/software/S0194), [PoshC2](https://attack.mitre.org/software/S0378), and PSAttack.(Citation: Github PSAttack) PowerShell commands/scripts can also be executed without directly invoking the `powershell.exe` binary through interfaces to PowerShell's underlying `System.Management.Automation` assembly DLL exposed through the .NET framework and Windows Common Language Interface (CLI).(Citation: Sixdub PowerPick Jan 2016)(Citation: SilentBreak Offensive PS Dec 2015)(Citation: Microsoft PSfromCsharp APR 2014)

Name

T1059

ID

T1059

Description

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](https://attack.mitre.org/techniques/T1059/004) while Windows installations include the [Windows Command Shell](https://attack.mitre.org/techniques/T1059/003) and [PowerShell](https://attack.mitre.org/techniques/T1059/001). There are also cross-platform interpreters such as [Python](https://attack.mitre.org/techniques/T1059/006), as well as those commonly associated with client applications such as [JavaScript](https://attack.mitre.org/techniques/T1059/007) and [Visual Basic](https://attack.mitre.org/techniques/T1059/005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](https://attack.mitre.org/tactics/TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](https://attack.mitre.org/techniques/T1021) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

Name

T1027

ID

T1027

Description

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 in order 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](https://attack.mitre.org/techniques/T1140) for [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) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

Name

T1562.002

ID

T1562.002

Description

Adversaries may disable Windows event logging to limit data that can be leveraged for detections and audits. Windows event logs record user and system activity such as login attempts, process creation, and much more.(Citation: Windows Log Events) This data is used by security tools and analysts to generate detections. The EventLog service maintains event logs from various system components and applications.(Citation: EventLog_Core_Technologies) By default, the service automatically starts when a system powers on. An audit policy, maintained by the Local Security Policy (secpol.msc), defines which system events the EventLog service logs. Security audit policy settings can be changed by running secpol.msc, then navigating to `Security Settings\Local Policies\Audit Policy` for basic audit policy settings or `Security Settings\Advanced Audit Policy Configuration` for advanced audit policy settings.(Citation: Audit_Policy_Microsoft)(Citation: Advanced_sec_audit_policy_settings) `auditpol.exe` may also be used to set audit policies.(Citation: auditpol) Adversaries may target system-wide logging or just that of a particular application. For example, the Windows EventLog service may be disabled using the `Set-Service -Name EventLog -Status Stopped` or `sc config eventlog start=disabled` commands (followed by manually stopping the service using `Stop-Service -Name EventLog`).(Citation: Disable_Win_Event_Logging)(Citation: disable_win_evt_logging) Additionally, the service may be disabled by modifying the “Start” value in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog` then restarting the system for the change to take effect.(Citation: disable_win_evt_logging) There are several ways to disable the EventLog service via registry key modification. First, without Administrator privileges, adversaries may modify the "Start" value in the key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Security`, then reboot the system to disable the Security EventLog.(Citation: winser19_file_overwrite_bug_twitter) Second, with Administrator privilege, adversaries may modify the same values in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-System` and `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Application` to disable the entire EventLog.(Citation: disable_win_evt_logging) Additionally, adversaries may use `auditpol` and its sub-commands in a command prompt to disable auditing or clear the audit policy. To enable or disable a specified setting or audit category, adversaries may use the `/success` or `/failure` parameters. For example, `auditpol /set /category:”Account Logon” /success:disable /failure:disable` turns off auditing for the Account Logon category.(Citation: auditpol.exe_STRONTIC)(Citation: T1562.002_redcanaryco) To clear the audit policy, adversaries may run the following lines: `auditpol /clear /y` or `auditpol /remove /allusers`.(Citation: T1562.002_redcanaryco) By disabling Windows event logging, adversaries can operate while leaving less evidence of a compromise behind.

Name
T1547.009
ID
T1547.009
Description

Adversaries may create or modify shortcuts that can execute a program during system boot or user login. Shortcuts or symbolic links are used to reference other files or programs that will be opened or executed when the shortcut is clicked or executed by a system startup process. Adversaries may abuse shortcuts in the startup folder to execute their tools and achieve persistence.(Citation: Shortcut for Persistence) Although often used as payloads in an infection chain (e.g. [Spearphishing Attachment](https://attack.mitre.org/techniques/T1566/001)), adversaries may also create a new shortcut as a means of indirection, while also abusing [Masquerading](https://attack.mitre.org/techniques/T1036) to make the malicious shortcut appear as a legitimate program. Adversaries can also edit the target path or entirely replace an existing shortcut so their malware will be executed instead of the intended legitimate program. Shortcuts can also be abused to establish persistence by implementing other methods. For example, LNK browser extensions may be modified (e.g. [Browser Extensions](https://attack.mitre.org/techniques/T1176)) to persistently launch malware.

Name

T1574.011

ID

T1574.011

Description

Adversaries may execute their own malicious payloads by hijacking the Registry entries used by services. Adversaries may use flaws in the permissions for Registry keys related to services to redirect from the originally specified executable to one that they control, in order to launch their own code when a service starts. Windows stores local service configuration information in the Registry under `HKLM\SYSTEM\CurrentControlSet\Services`. The information stored under a service's Registry keys can be manipulated to modify a service's execution parameters through tools such as the service controller, sc.exe, [PowerShell](https://attack.mitre.org/techniques/T1059/001), or [Reg](https://attack.mitre.org/software/S0075). Access to Registry keys is controlled through access control lists and user permissions. (Citation: Registry Key Security)(Citation: malware_hides_service) If the permissions for users and groups are not properly set and allow access to the Registry keys for a service, adversaries may change the service's binPath/ImagePath to point to a different executable under their control. When the service starts or is restarted, then the adversary-controlled program will execute, allowing the adversary to establish persistence and/or privilege escalation to the account context the service is set to execute under (local/domain account, SYSTEM, LocalService, or NetworkService). Adversaries may also alter other Registry keys in the service’s Registry tree. For example, the `FailureCommand` key may be changed so that the service is executed in an elevated context anytime the service fails or is intentionally corrupted.(Citation: Kansa Service related collectors)(Citation: Tweet Registry Perms Weakness) The `Performance` key contains the name of a driver service's performance DLL and the names of several exported functions in the DLL.(Citation: microsoft_services_registry_tree) If the `Performance` key is not already present and if an adversary-controlled user has the `Create Subkey` permission, adversaries may create the `Performance` key in the service’s Registry tree to point to a malicious DLL.(Citation: insecure_reg_perms) Adversaries may also add the `Parameters` key, which stores driver-specific data, or other custom subkeys for their malicious services to establish persistence or enable other malicious activities.(Citation: microsoft_services_registry_tree)(Citation: troj_zegost) Additionally, If adversaries launch their malicious services using svchost.exe, the service’s file may be identified using `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\servicename\Parameters\ServiceDll`.(Citation: malware_hides_service)

Name
T1548.002

ID
T1548.002

Description
Adversaries may bypass UAC mechanisms to elevate process privileges on system. Windows User Account Control (UAC) allows a program to elevate its privileges (tracked as integrity levels ranging from low to high) to perform a task under administrator-level permissions, possibly by prompting the user for confirmation. The impact to the user ranges from denying the operation under high enforcement to allowing the user to perform the action if they are in the local administrators group and click through the prompt or allowing them to enter an administrator password to complete the action. (Citation: TechNet How UAC Works) If the UAC protection level of a computer is set to anything but the highest level, certain Windows programs can elevate privileges or execute some elevated [Component Object Model](https://attack.mitre.org/techniques/T1559/001) objects without prompting the user through the UAC notification box.(Citation: TechNet Inside UAC)(Citation: MSDN COM Elevation) An example of this is use of [Rundll32](https://attack.mitre.org/techniques/T1218/011) to load a specifically crafted DLL which loads an

auto-elevated [Component Object Model](https://attack.mitre.org/techniques/T1559/001) object and performs a file operation in a protected directory which would typically require elevated access. Malicious software may also be injected into a trusted process to gain elevated privileges without prompting a user.(Citation: Davidson Windows) Many methods have been discovered to bypass UAC. The Github readme page for UACME contains an extensive list of methods(Citation: Github UACMe) that have been discovered and implemented, but may not be a comprehensive list of bypasses. Additional bypass methods are regularly discovered and some used in the wild, such as: * `eventvwr.exe` can auto-elevate and execute a specified binary or script.(Citation: enigma0x3 Fileless UAC Bypass)(Citation: Fortinet Fareit) Another bypass is possible through some lateral movement techniques if credentials for an account with administrator privileges are known, since UAC is a single system security mechanism, and the privilege or integrity of a process running on one system will be unknown on remote systems and default to high integrity.(Citation: SANS UAC Bypass)

Vulnerability

Name
CVE-2023-48788
Description
Fortinet FortiClient EMS contains a SQL injection vulnerability that allows an unauthenticated attacker to execute commands as SYSTEM via specifically crafted requests.

Malware

Name
XWorm

IPv4-Addr

Value
71.162.181.51
91.92.243.28

StixFile

Value
fbc56856260ba2ea924a94a301616540addd202d8840539613aece22780c473c
f226bf37af9c33162063db3eb018fed7f088f86d0a20ca54c013fda96c7f2e05
d676027d95cbcc1ce363268fe213fad71beeec1be359950518739b61507763fd
b6860ea80d2d085e8296692579b886d0218cff29b1f6702f40e5543c86dafe3c
a277b6cd81846872dd5f6d92f815b506cac7c38d0327f9e8303d1e4328134f98
873769c7caa894ff42904def7ad206473fa726b6598522b1f466416c8bbfd6cd
82c569b93da5c18ed649ebd4c2c79437db4611a6a1373e805a3cb001c64130b7
79eb68712c75b2585ee74cb782cb3369b2f2c6fc09c9abbdbeb096fd0ccbd401
5d13e40e5c5b9e037b99fb4f81769d0e8505be2dae8ffdf8013f236c62dad220
5c549868fbb58fe22e1c37b99699783fab996cec6c0e83c897f722a754f89def
0d5a428483c0dafc37bb01ea323958d4eb303edc80d307a3e4898c6e22d6c12f
0f7a9880cf58b7455ba6ed2de808ed38eeef9ad34b28ff95e63d63b4f2a51e51
abd03973677c82081db0b0af02ca302bb5a854f96c52acd2113ec110b373daa6

Url

Value
https://spnmandalawangi.banten.polri.go.id/Tax_docs_2023.htm
http://91.92.243.28/poom/atom.xml

External References

- https://github.com/esThreatIntelligence/iocs/blob/main/XWorm/iocs.txt
- https://www.esentire.com/blog/dont-take-the-bait-the-xworm-tax-scam
- https://otx.alienvault.com/pulse/661cedb5e64e8a248f876bf1