

NETMANAGEIT

Intelligence Report

Unmasking Cronus: How Fake PayPal Documents Execute Fileless Ransomware via PowerShell

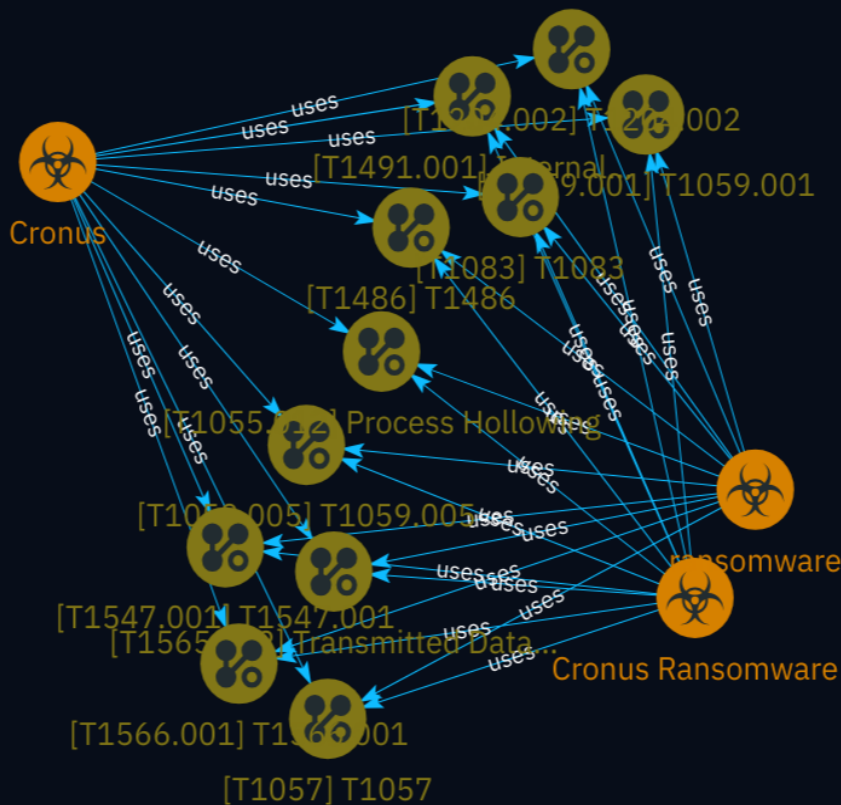


Table of contents

Overview

● Description	3
● Confidence	3
● Content	4

Entities

● Attack-Pattern	5
● Malware	14
● uses	15

External References

● External References	19
-----------------------	----

Overview

Description

The analysis reveals a sophisticated campaign employing fake PayPal receipts as lures to distribute a new variant of the Cronus ransomware. The infection chain begins with a malicious Word document containing an obfuscated VBA macro that downloads a PowerShell loader from a remote server. This loader employs reflective DLL loading to execute the fileless ransomware payload directly in memory, evading disk-based detection. The ransomware exhibits various malicious behaviors, such as enumerating and encrypting specific file types, terminating processes, establishing persistence, and manipulating clipboard data.

Confidence

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

100 / 100

Content

N/A

Attack-Pattern

Name

T1057

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. Administrator or otherwise elevated access may provide better process details. 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

Process Hollowing

ID

T1055.012

Description

Adversaries may inject malicious code into suspended and hollowed processes in order to evade process-based defenses. Process hollowing is a method of executing arbitrary code in the address space of a separate live process. Process hollowing is commonly performed by creating a process in a suspended state then unmapping/hollowing its memory, which can then be replaced with malicious code. A victim process can be created with native Windows API calls such as `CreateProcess`, which includes a flag to suspend the processes primary thread. At this point the process can be unmapped using APIs calls such as `ZwUnmapViewOfSection` or `NtUnmapViewOfSection` before being written to, realigned to the injected code, and resumed via `VirtualAllocEx`, `WriteProcessMemory`, `SetThreadContext`, then `ResumeThread` respectively.(Citation: Leitch Hollowing)(Citation: Elastic Process Injection July 2017) This is very similar to [Thread Local Storage](https://attack.mitre.org/techniques/T1055/005) but creates a new process rather than targeting an existing process. This behavior will likely not result in elevated privileges since the injected process was spawned from (and thus inherits the security context) of the injecting process. However, execution via process hollowing may also evade detection from security products since the execution is masked under a legitimate process.

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

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

Internal Defacement

ID

T1491.001

Description

An adversary may deface systems internal to an organization in an attempt to intimidate or mislead users, thus discrediting the integrity of the systems. This may take the form of modifications to internal websites, or directly to user systems with the replacement of the desktop wallpaper.(Citation: Novetta Blockbuster) Disturbing or offensive images may be used as a part of [Internal Defacement](<https://attack.mitre.org/techniques/T1491/001>) in order to cause user discomfort, or to pressure compliance with accompanying messages. Since internally defacing systems exposes an adversary's presence, it often takes place after other intrusion goals have been accomplished.(Citation: Novetta Blockbuster Destructive Malware)

Name

Transmitted Data Manipulation

ID

T1565.002

Description

Adversaries may alter data en route to storage or other systems in order to manipulate external outcomes or hide activity, thus threatening the integrity of the data.(Citation: FireEye APT38 Oct 2018)(Citation: DOJ Lazarus Sony 2018) By manipulating transmitted data, adversaries may attempt to affect a business process, organizational understanding, and decision making. Manipulation may be possible over a network connection or between system processes where there is an opportunity deploy a tool that will intercept and change information. The type of modification and the impact it will have depends on the target transmission mechanism as well as the goals and objectives of the adversary. For complex systems, an adversary would likely need special expertise and possibly access to specialized software related to the system that would typically be gained through a prolonged information gathering campaign in order to have the desired impact.

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

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

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

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.

Malware

Name

ransomware

Name

Cronus Ransomware

Name

Cronus

uses

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name

Name
Name

External References

-
- <https://www.seqrte.com/blog/unmasking-cronus-how-fake-paypal-documents-deliver-fileless-ransomware-via-powershell/>
-
- <https://otx.alienvault.com/pulse/66b3311df1f0791c05cb53b7>