2024
DUBLIN

2 - 4 October, 2024 / Dublin, Ireland

CRACKEDCANTIL: A MALWARE SYMPHONY DELIVERED BY CRACKED SOFTWARE; PERFORMED BY LOADERS, INFOSTEALERS, RANSOMWARE, ET AL.

Lena Yu

World Cyber Health, Japan

lena.yu@worldcyberhealth.org

ABSTRACT

The digital landscape teems with diverse malware families, each engineered with distinct capabilities – ranging from data theft, deployment of additional malicious payloads, to destruction of data and more.

Yet, beneath their varied functionalities, these pieces of malware may unite in a complex and orchestrated performance, functioning in concert to unleash potent malware infections. This intricate interplay, which I have termed the ‘malware symphony’, mirrors the harmonious collaboration of instruments in an orchestra, where each contributes its unique timbre to the overall composition.

This paper introduces the concept of the malware symphony by analysing the CrackedCantil campaign. It explores how each malware component contributes to a harmonious yet malicious concert designed to compromise and exploit systems with unparalleled sophistication.

INTRODUCTION

Malware continues to evolve, becoming increasingly elusive, destructive, efficient and widespread [1]. There exists a vast array of malware families, each distinguished by its specific traits. These diverse malware families can collaborate seamlessly to orchestrate a potent infection.

This paper will explore a concept coined the ‘malware symphony’. It illustrates how various malware can collaborate, akin to instruments in an orchestra, thereby enhancing the overall impact of the infection.

Previously, infections involving multiple malware were typically referred to as ‘loader campaigns’. However, this term was general and could include infections involving multiple malware but lacking coordination.

The inadequacy of existing terms to fully capture the orchestrated nature of these threats demonstrates the need for a new definition.

By introducing the unique concept of the malware symphony, this paper aims to enhance the classification and analysis of complex multi-malware infections.

DECOMPOSING THE MALWARE SYMPHONY

There are numerous cases of multiple malware infections, but not all of them are coordinated and they can result in the conflicts shown in Table 1.

Conflict	Description
Ransomware encrypts files before other malware can perform	This makes the infection obvious to the victim, who will then take measures to remediate the infection. The system may go down, which means that other malware does not get a chance to perform. Even if infostealers successfully exfiltrate encrypted data, the attacker may not have the decryption key, rendering the stolen data useless. Some resources may be inaccessible to other malware.
More than one ransomware attempting to encrypt files	Complicates the encryption/decryption process. Race conditions may occur if multiple ransomware attempt to encrypt the same files at the same time. Spikes in computational resource usage can alert the system.
Malware attempt to kill each other	Malware developed by competing parties may attempt to kill each other, as seen in the case of botnet malware Mirai [2]. Some malware disguises itself as legitimate processes and antivirus programs, while other malware attempts to kill these, mistaking them for legitimate processes or antivirus programs [3].
Malware competing for resources	Malware such as coinminers utilize a lot of computational resources, which can cause other malware and crucial system processes to slow down.
Other interferences	Malware blocking certain connections/resources which are required by other malware. Multiple malware attempting to access the same resources at the same time could lead to race conditions, errors, glitches and more.

Table 1: Examples of conflicts between multiple malware.

Coordination is key in deciding whether to categorize a multi-malware infection as a malware symphony. Firstly, it should not have any of the obvious conflicts shown in Table 1. Secondly, a typical malware symphony can be deconstructed into ‘movements’ (akin to the movements of a symphony), as shown in Table 2.

The description, actions, and MITRE techniques typically observed in a malware symphony are also shown in Table 2. Note, however, that this classification is not absolute, and discrepancies may exist between different malware symphonies.

For this paper, any malware that is not a loader, infostealer, or ransomware will be referred to as ‘otherware’. Additionally, some malware can fall under multiple categories – for example, Amadey can act both as a loader and as an infostealer. For simplicity, we assume each malware falls into one category.

Order	Symphony movement	General description	Action	Common MITRE techniques
1	Overture of the Loaders	Starts and coordinates the malware symphony	System checks before starting the malware symphony	T1518: Software Discovery
				T1082: System Information Discovery
				T1012: Query Registry
				T1497: Virtualization/Sandbox Evasion
				T1016: System Network Configuration Discovery
			Communicate with C2	T1071: Application Layer Protocol
				T1571: Non-Standard Port
			Make C2 traffic hard to analyse	T1132: Data Encoding
				T1573: Encrypted Channel
			Ensure smooth entry of other malware	T1562: Impair Defenses
				T1588: Obtain Capabilities
			Time the execution of other malware	T1547: Boot or Logon Autostart Execution
				T1053: Scheduled Task/Job
				T1569: System Services
2	Ensemble of the Infostealers	A variety of infostealers can be involved, with a diverse range of stolen data and exfiltration techniques	Communicate with C2	T1071: Application Layer Protocol
				T1571: Non-Standard Port
			Make C2 traffic hard to analyse	T1132: Data Encoding
				T1573: Encrypted Channel
			Check environment values	T1518: Software Discovery
				T1012: Query Registry
				T1082: System Information Discovery
			Allow easy re-entry of itself	T1547: Boot or Logon Autostart Execution
				T1053: Scheduled Task/Job
			Collect the data	T1552: Unsecured Credentials
				T1555: Credentials from Password Stores
				T1115: Clipboard Data
				T1113: Screen Capture
			Exfiltrate the data	T1567: Exfiltration Over Web Service
				T1041: Exfiltration Over C2 Channel
				T1048: Exfiltration Over Alternative Protocol
2	Chorale of the ‘Otherware’	Any malware that doesn’t fall into the category of a loader, infostealer, ransomware – typically, malware that hijacks device resources	Communicate with C2	T1071: Application Layer Protocol
				T1571: Non-Standard Port
			Hijack resources	T1496: Resource Hijacking
3	Finale of the Ransomware	Encryption activities happen last, and solo, to prevent double encryption	Give other malware time to perform	T1547: Boot or Logon Autostart Execution
				T1053: Scheduled Task/Job
			Prevent double encryption	T1057: Process Discovery
				T1083: File and Directory Discovery
			Encrypt the files	T1486: Data Encrypted for Impact

Table 2: The typical composition of a malware symphony.

A prime exemplar of such orchestrated cyber malevolence is CrackedCantil – a malware symphony that originates from cracked software hosted on reputable platforms like *Google Groups*. The CrackedCantil campaign stands out for its collaborative use of numerous distinct families of malware.

'CrackedCantil' is a moniker derived from the symphony's roots in cracked software and its analogy to the venomous Cantil viper [4]. A malware symphony that originates from cracked software, and is orchestrated by PrivateLoader will be referred to as the CrackedCantil symphony.

For ease of classification, the naming convention *'Symphony no. <ID>, <Name of malware symphony>'* will be used for malware symphonies, where *<ID>* represents the unique number identifying the specific case of the campaign, and *<Name of malware symphony>* denotes its name.

This naming convention was introduced to uniquely identify specific cases of the malware symphony. Each symphony may be subtly different, even if it belongs to the same campaign and shares a similar overall composition, as shown in Table 3. The malware in Table 3 only includes infamous malware – there may be other malware not listed that are also involved.

Title	Category	Malware
Symphony No. 1, CrackedCantil [5]	Loaders	PrivateLoader
		Smoke
	Infostealers	Lumma
		RedLine
		RisePro
		Amadey
		Stealc
	Otherware	Socks5Systemz
		Coinminers
	Ransomware	STOP
Symphony No. 2, CrackedCantil [6]	Loaders	PrivateLoader
		Smoke
		Glupteba
	Infostealers	Lumma
		Stealc
		Risepro
		Redline
	Otherware	XMRig
	Ransomware	STOP
Symphony No. 3, CrackedCantil [7]	Loaders	PrivateLoader
	Infostealers	Lumma
		Redline
		Amadey
		RisePro
		Stealc
	Otherware	Kelihos
		Socks5Systemz
		Coinminers
	Ransomware	STOP

Table 3: The various CrackedCantil symphonies.

The malware symphony examined in this paper will be *Symphony No. 1, CrackedCantil*. The sandbox environment utilized in the analysis is shown in Table 4.

Sandbox service	ANY.RUN
Operating system	Windows 11
Device name	DESKTOP-BFTPUHP
User	admin
MAC address	52:54:00:4a:ad:11
Screen resolution	1280x720
Language	en-US
Residential proxy IP	174.161.239.28
Residential proxy location	United States

Table 4: Sandbox environment used for the analysis of Symphony No. 1, CrackedCantil.

STAGING THE MALWARE SYMPHONY

There are various ways a malware symphony can be staged. One of the most popular methods for delivery is through cracked software, as this is the primary vector of infection for loaders like PrivateLoader.

The usage and distribution of cracked software is illegal in many places as it falls under software piracy laws [8]. However, many are still willing to take the risk to save money despite the possible legal repercussions and malware infections. Attackers take advantage of the fact that victims are not legally protected, and are less likely to seek help when infected due to having willingly downloaded cracked software.

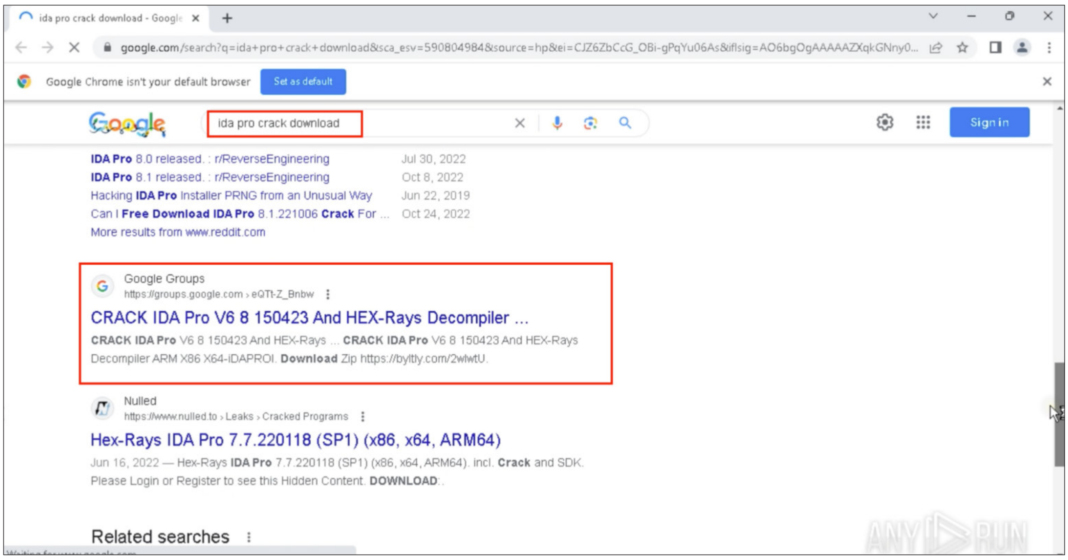
In addition, specific versions of cracked software can be used to distribute malware that is compatible with the system. For example, someone that searches for and downloads a ‘Cracked Photoshop for Windows 10’ would likely be using *Windows 10* for the OS, thus the attacker would embed malware targeted at *Windows 10* within this version of cracked software. This eliminates the need for browser User-Agent checks to redirect the victim to a compatible version download, which greatly increases the range of platforms on which this cracked software can be hosted.

Take for instance the Roaming Mantis smishing campaign. When the victim clicks on the malicious link in the instant message, they are redirected to an attacker-hosted site that performs User-Agent checks. If the site identifies the User-Agent as an *Android* device, it downloads an APK that is *Android* malware disguised as a legitimate application. If identified as an *iOS* device, it redirects the victim to an *iCloud* phishing site. To perform these User-Agent checks, the site must be hosted by the attacker due to custom scripts [9].

Knowing the operating system and architecture of the victim device significantly increases the success rate of the malware symphony. Loaders can drop malware that is suitable for the victim's environment, but the loaders themselves are usually constrained by the underlying operating system and architecture.

Staged on Google Groups

The CrackedCantil campaign was rampant between the end of 2023 and the start of 2024. Searching for cracked versions of paid popular software (e.g. ‘IDA PRO Crack Download’, ‘Photoshop Free Download’) on search engines during this period would likely have shown a *Google Groups* result with the cracked software as one of the top search results, as shown in Figure 1.


ida pro crack download - Google
google.com/search?q=ida+pro+crack+download&sca_esv=590804984&source=hp&ei=CJZ6ZbCcG_OBi-gPqYu06As&iflsig=AO6bgOgAAAAAZXqkGNny0...
Google Chrome isn't your default browser
Set as default
Google
ida pro crack download
Sign in
ANY.RUN

IDA Pro 8.0 released. : r/ReverseEngineering — Jul 30, 2022
IDA Pro 8.1 released. : r/ReverseEngineering — Oct 8, 2022
Hacking IDA Pro Installer PRNG from an Unusual Way — Jun 22, 2019
Can I Free Download IDA Pro 8.1.221006 Crack For ... — Oct 24, 2022
More results from www.reddit.com

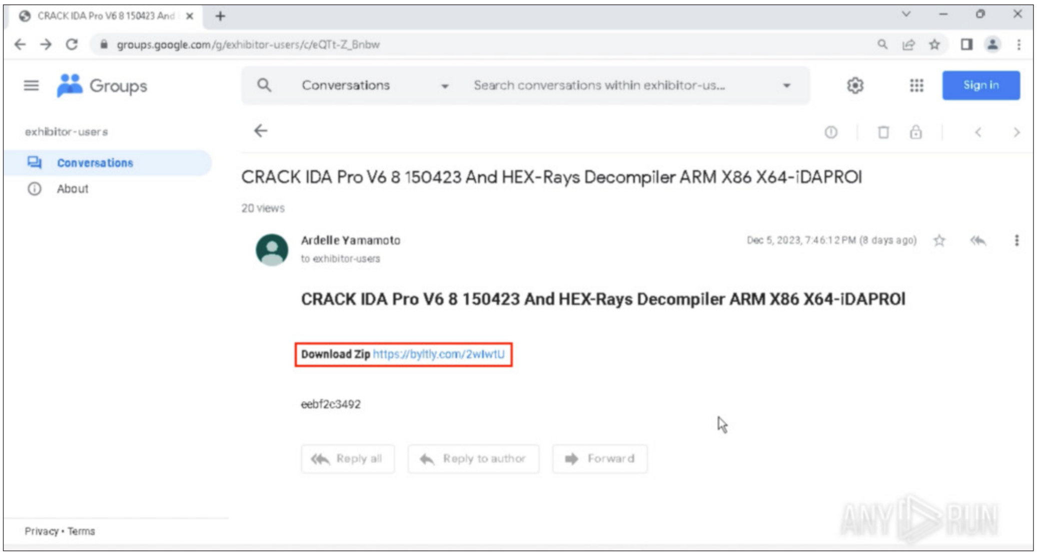
Google Groups
https://groups.google.com › eQTt-Z_Bnbw
CRACK IDA Pro V6 8 150423 And HEX-Rays Decompiler ...
CRACK IDA Pro V6 8 150423 And HEX-Rays ... CRACK IDA Pro V6 8 150423 And HEX-Rays Decompiler ARM X86 X64-iDAPROI. Download Zip https://byltly.com/2wlwtU.

Nulled
https://www.nulled.to › Leaks › Cracked Programs
Hex-Rays IDA Pro 7.7.220118 (SP1) (x86, x64, ARM64)
Jun 16, 2022 — Hex-Rays IDA Pro 7.7.220118 (SP1) (x86, x64, ARM64). incl. Crack and SDK. Please Login or Register to see this Hidden Content. DOWNLOAD:.

Related searches

Figure 1: Google Search results for ‘ida pro crack download’ in December 2023.

Once the victim visits the *Google Groups* result, they are greeted with a shortened download link inside a *Google Groups* conversation, as shown in Figure 2.


CRACK IDA Pro V6 8 150423 And
groups.google.com/g/exhibitor-users/c/eQTt-Z_Bnbw
Groups
Conversations
Search conversations within exhibitor-us...
Sign in
exhibitor-users
Conversations
About
Reply all
Reply to author
Forward
Privacy • Terms
ANY RUN

CRACK IDA Pro V6 8 150423 And HEX-Rays Decompiler ARM X86 X64-iDAPROI

20 views

Ardelle Yamamoto
to exhibitor-users

Dec 5, 2023, 7:46:12 PM (8 days ago)

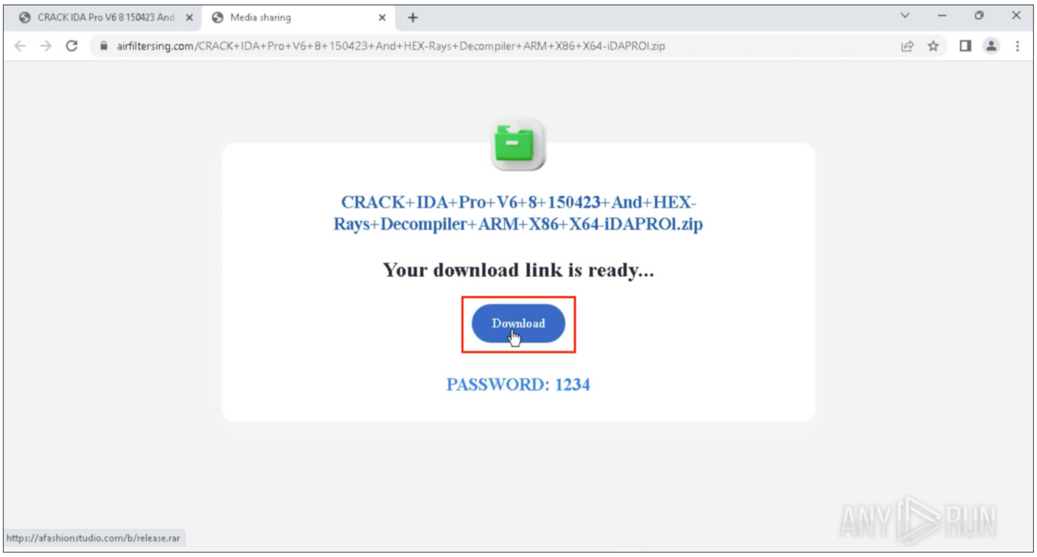
CRACK IDA Pro V6 8 150423 And HEX-Rays Decompiler ARM X86 X64-iDAPROI

Download Zip https://byltly.com/2wlwtU

eebf2c3492

Figure 2: Google Groups conversation with the shortened link.

The names of the archive files downloaded from the shortened links included ‘release.rar’ and ‘release_vX.rar’, with ‘X’ denoting the version number. These password-protected archives typically ranged from 7 to 9 MB, and the password was displayed on the download page, as shown in Figure 3.


CRACK IDA Pro V6 8 150423 And
Media sharing
airfiltersing.com/CRACK+IDA+Pro+V6+8+150423+And+HEX-Rays+Decompiler+ARM+X86+X64-iDAPROI.zip
Download
https://afashionstudio.com/b/release.rar
ANY RUN

CRACK+IDA+Pro+V6+8+150423+And+HEX-Rays+Decompiler+ARM+X86+X64-iDAPROI.zip

Your download link is ready...

PASSWORD: 1234

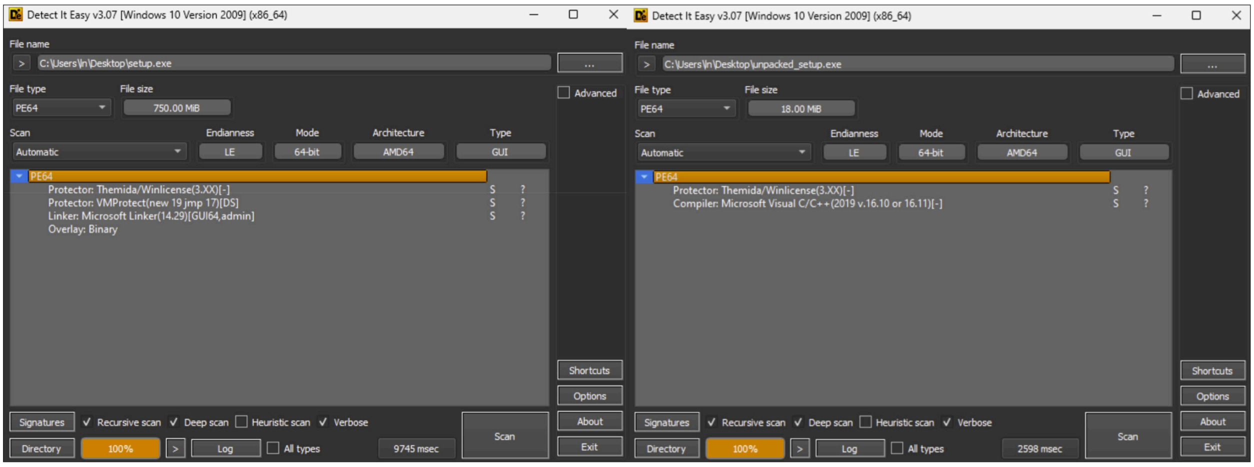
Figure 3: The download site with the password.

Delivered by cracked software

In *Symphony No. 1, CrackedCantil*, the executable was inside a password-protected archive, where the password was shown on the download page [4]. The executable was revealed to be massive, at 750 MB, which hinders analysis, as some antivirus programs cannot scan password-protected and very large files. Additionally, analysis will be more tedious if the analyst does not have the password to the sample, and online malware analysis services like *VirusTotal* have an upload limit of 650 MB [10].

To deter reverse engineering, the executable is protected by both Themida/Winlicense(3.XX) and VMprotect. The executable disguises itself as a ‘Logitech Plugin Installer Utility’. This same file name and description were seen across various CrackedCantil campaigns on *Google Groups*, regardless of the cracked software it purported to be. This suggests the executable was recycled across various campaigns, as it contained nothing specific to the cracked software.

Upon removing the layers of protectors with available tools [11], the payload was revealed to be only 18 MB, as shown in Figure 4.



Detect It Easy v3.07 [Windows 10 Version 2009] (x86_64)

File name: C:\Users\ln\Desktop\setup.exe

☐ Advanced

File type: PE64 File size: 750.00 MiB

Scan: Automatic Endianness: LE Mode: 64-bit Architecture: AMD64 Type: GUI

```
PE64
    Protector: Themida/Winlicense(3.XX)[-]                S   ?
    Protector: VMProtect(new 19 jmp 17)[DS]               S   ?
    Linker: Microsoft Linker(14.29)[GUI64,admin]          S   ?
    Overlay: Binary
```

✓ Recursive scan ✓ Deep scan ☐ Heuristic scan ✓ Verbose

100% ☐ All types 9745 msec

Detect It Easy v3.07 [Windows 10 Version 2009] (x86_64)

File name: C:\Users\ln\Desktop\unpacked_setup.exe

☐ Advanced

File type: PE64 File size: 18.00 MiB

Scan: Automatic Endianness: LE Mode: 64-bit Architecture: AMD64 Type: GUI

```
PE64
    Protector: Themida/Winlicense(3.XX)[-]                       S   ?
    Compiler: Microsoft Visual C/C++(2019 v.16.10 or 16.11)[-]   S   ?
```

✓ Recursive scan ✓ Deep scan ☐ Heuristic scan ✓ Verbose

100% ☐ All types 2598 msec

Figure 4: Packed CrackedCantil executable on the left; unpacked CrackedCantil executable on the right.

The core of the CrackedCantil symphony lies in the PrivateLoader malware, which will be explored in detail in the following sections.

THE OVERTURE OF THE LOADERS

The malware symphony starts with the Overture of the Loaders. The loaders are like the maestros of a symphony, who are responsible for cueing the start, setting the tempo, and preparing the smooth entry of other performers.

These loaders typically depend heavily on communication with the C2, so before the symphony starts, they will perform a communication check with the C2 to ensure it can be reached. If it cannot be reached, the malware symphony usually ends abruptly.

Once the loader confirms the connection with the C2, it prepares a malware symphony tailor-made for the victim's environment by collecting environment information and sending it to the C2, which then responds with suitable malware payloads.

The loaders are responsible for coordinating the entire symphony and ensuring the collaborative harmony among the various malware – making sure there are no obvious conflicts such as those shown in Table 1.

In the case of the CrackedCantil symphony, PrivateLoader serves as the primary maestro, spawning Smoke Loader which then serves as the secondary maestro.

It is important to note that, in most cases, the malware spawned from the loaders are not aware of the presence of other malware. This means the loader is responsible for orchestrating the symphony in a way that ensures the malware do not conflict with one another.

PrivateLoader

PrivateLoader, serving as the primary maestro of the CrackedCantil symphony, is central to this analysis and will be examined in detail.

In *Symphony No. 1, CrackedCantil*, PrivateLoader begins by checking the system environment (T1082: System Information Discovery and T1012: Query Registry) to ensure it is suitable for the ensuing malware symphony.

It then cues the start of the symphony by sending an HTTP GET request to the C2, in this case with the URI '/api/tracemap.php' (T1071: Application Layer Protocol). If the C2 replies with a specific string – in this case '15.5pnp.10.lock' – it cues the start of the symphony; if there is no response, then PrivateLoader crashes, ending the symphony abruptly.

As mentioned previously, loaders – particularly PrivateLoader – are heavily dependent on communication with the C2 for receiving commands, downloading additional malware, and more. Thus, it performs this initial connectivity check with the C2 to ensure everything is good to go.

At the time of writing this paper, the C2 for PrivateLoader in *Symphony No. 1, CrackedCantil* had stopped responding. Using FakeNet in the *ANY.RUN* sandbox, a custom HTTP response, '15.5pnp.10.lock', was implanted to simulate the real C2 response. Upon implanting this fake HTTP response, PrivateLoader was observed to continue execution [12].

It then performs IP checks with various online services such as api.myip[.]com and ipinfo[.]io. FakeNet was configured to respond with a custom response only to HTTP requests for the URI '/api/tracemap.php', while allowing all other requests to reach the real internet.

PrivateLoader then arranges the entry of other malware. It prepares a Base64-encoded encrypted string (T1132: Data Encoding and T1573: Encrypted Channel), and sends HTTP POST requests to the C2, in this case with the URI '/api/firegate.php'.

Through reverse engineering, the decoding and decryption algorithm for the HTTP request/response was obtained, and the Python script can be found at [13].

The encoded-encrypted HTTP request/response, and its decoded-decrypted counterpart observed in *Symphony No. 1, CrackedCantil* can be seen in Table 5.

	Encoded-encrypted	**Decoded-decrypted**
Request	`Q0uWGgHyOK1yWQK-BXHkM-HySJVrM-bkDRjaZRMVle11OCvYaPf2WzR9nGuLpCPzAv8ibLyhynT0DqT5CPejzN_j4vkuL4Rmafqdqg7q29RNzn9VOTArbMt6Jrq5lsZ3`	`GetExtensions|USA_2|US|16`
Response	`FaU4dkFGmFsWKWHjsIyHND/UQ4teC8N/iQvaDo7KdzhN7A+UPiuqSmRylwEY4xK8esn2u4T6CpBh383VqxiDRRD+bfa76QQLfTJpwLFlS0A=`	`[]`
Request	`2nz0hsO9K7vKyuyl6qoOl_sXwxXEb9wuclyy-ls5CzmbHEQUW2WHIvG9MpPOFBnZnyJoLVAtEzHhAskeKO0zSvR_r5qNNZLcYZ4xP0XllMrOnoKZhvdXZdNamZiesubb`	`GetLinks|USA_2|US|16`

Response	`Letw5AloRfH5EJy3QRIcouZs/qYLXwRoR4PZbQFQhN2Nd8yTbZcYDGzOtHApGfTFR1Tv9sqJLktOf6fjaLz85hacrC9ogc+Cj5cGTClMhiSmZqsjYIZG24MpA5tO26+5SmY55Yq81lYUTmH6s7JYdFYF9r0fRrPK7LLclJH9gK5CAkCdb3CPA1lbYS+8na5lwwxIycamdM2IRNvXPZ2+DzkgiG39ur9gScryB85Y2BHjrxVGUGWkjrP18sb3THXaZdBZ9dug3a1+9kgKbWL/2SzTQ6GlhTNpHLZ5ZS+Fe/j+nYdFylDWjNjgG4TFLqoGYYMhNT5Aby4X+IzYQWmJGDkP03ThlWoExZ0Pcx0PibBiDwp0o9+2yTRNv/KiWGDnIXbNZOxaVn+S3b/HXZFu2pqSw3ca6lRoCOhMOjJwNUKjUwdMUFCTP3clECdsaL2ZAyu9f0U7p8cT/bWMrH+evubWOBo3jSG/YWLHwW4My70+O9xU0rxQz39GQbaCJixql1+2Kb2Y6HGWJiQ+gAtpMnVocYIo8l93HNvhkjl0cKrBc6CCXVYEA8eBiFBDSx8FaQkbs4x/dSyp+QTCSJ9h4bpEmTp2KmSNScaL+oStiNWYxUrcz+nN3H6d0P7nLSEI8evXb0L5r/6ieVzv2hp/rpKLFpwh7SHIcH7HN57lpZBJkDXBsmz2sr8Y4jGNy3X8R3YfYeqGhfd1pBtqt5AeFQtJvqCsiWoiaQ1yFloiQjtantrbTdWtYiNu3CUXSxTAUYJ8HFSFGeYAtWsSIEBteTKVB+9JzgN0tP8jZnFjdcE5CfejOJOguJSO/Jd1RdpHYP/mOvq+AzS6XXygbA/n5GdqjnjDCOH2eULJl9dZLH1FRO8EDl3h3l2wg6YR1onoKfubqRb+RKf+a3nSe5QMG2CiaQ/HY+SLK8V5dJiHiJqjAeE8beQGWuu7DWa+`
	`… +DofUcxy80YGDAKU3FQcYTJhrcYqjY5xo2773JPIGRPk6OODSKyNeLi7lxLOYn9XQ4VvZZKKawoAjSzYUFGSQpdA1z4IKD27C2AIAhq54gFwcFvI9jIAjJ+YIRo4etoV033rDgbV6e7bxZvn8WKdX0H+pDgA80YjvG8Q+QVo3e4R8HnPKj2coA3M28MWu3lC7sdtUj2zxjjhzfSSjqp/o1ROSjfIetFlL9aMLCFArUYTSL+fKRAZWF39sr4hQFOv+4pFDdT8EU5uXaZzAz5tuxTRhpUgynYhOixgnYI2fItnUkc2+XNukMlPR8Ov1KHwarUJ+ASgycyzFr6rlwNl5gQsYVpMETJkBgAIRoBBBoE2ifkIgJExjJiLR5AxOQ5kJsQlTcqQOOjTCFhobSIjnPWszFpwrCHAlz9EBc5p2d7DobI0ep8rIUcrrfHG3B2FYbbqoK9hbuv17UN11pAP+gONuMgGn57OzSI3QrcqHpRMtKhe9hZPW/W40eiye1d2WPFXk67nkPdJ5J3FwJYzKYvne6LFJ7a6OagYWQ6flO0sK7lT+zeRnl6czQHTC98G45iV2Qobz8nN0/uiVPeWtIZfrcJqaDlKjWWhzONRPg6ZkhFObT7a9ssiQV596A5AB4PSzuWOEqbWmLe7wUX6ueXrKi2T4ZunJMHmJMx1ykUjsNvEy+Mxd9PV5WVhWiTFgKj9TL2opFtNO4mec96/uytgR25Rc8ZAYH4TOWd/e6LLrjOiDJrKQgJch9z+LWiYzuZh+OGjZ6VsspDeqMiapm87E2YbYIw4QdaIP6+/zfw9/5JHPKGdHZjQiVJfLpzgeS2EgYy+qzwyg7ggUkhEcBVSUnD/oYcNKqDTaCpOeCWRpHnG36A6iGPaACxo1FJtDCq3UDjOQCob8RfvnPaddscTqz/AU4RhDuD3uL4ATHkt3/QbPXzTpvkPCidXXHpTtzMKCTqy6L84Wv2c6F6YpU0o+NlR2mQJo5ce32HoPmd6dOzfFh5SsGIKvUwTx+bHccnb/GY9ffh25MVSR+DHeEbSE2ir8afwrpC7uj23GeTWLMB0O3cx4z+pQJ0GkvQywYZE2fs6lFsUp45n8vBdXgCezOliLAGcmb7rSjJVpmukOULqKsUpQ5z0wfzw08rzY0405Lif3KQ+nWbvCMO0UXxV7cCHhE+KvCuNpSriYemBqy3MqMnkYnsWrPoW6kpg/rJdA5fb4exCzyyDSHs0mdMca3tDAVMOHk8d42GdQRzd+8AT6VwQArKDQ4GIqudTQgVVqJdj+cvM/4g7R1LfCBxf03cXhNf2K/MnVZ1d1l/Uv1nZOzQBe49996KmAWpNviEEKl4p2rHIbBRT/B6QoVmreGwqzbQ50OW8+TGOQjb+4BcMR0Jm0HhGfl+ur2gaCbDSipD8EotGJPPVvQ7J+IR2W/h2IrLz9kPmHsAGmryHIFHRG2ENf9GSoUbryBdvPZgiRWoq8s6ypNEH7LgpMRynTatQQ8lxTlcRvV8ayO36Y16m8dA2bggmaPg7RMJIXCZmhLIie1YbziAaCFwsMDI1j0krLYo4wbr0LKBK74K41EWGtdxdxIWuU+IQAnhRR6G+Q94yY2d8iA05Po9nMinaDTTrQIoGIq5jhSUteXzaP29RBu1Es2suL+KOLyHxpp9i1S70zpbhuUEjE0elPCIMmcZqCx7AKVMP9fFVPmnOaMpbREwV/8rW9QtRdNL2mCMmFqxL2EWmpJuwYS6cgWfcSY=`
		`[{"id":"-1","url":"https:\/\/vk.com\/doc418490229_669446210?hash=BZ9b8Xtsn5Z8zZkSRBEdwF1W7jzCAT8GJBVEicdXS6L&dl=eA4o75IiHafzbkgdBC8nz7TmLS7uMpwJRsfDOcAnrqD&api=1&no_preview=1","args":"","type":"0","onlyType":"0"},{"id":"999991","url":"https:\/\/vk.com\/doc418490229_669284201?hash=L30vXtgODLl0q95FGyET2USzk3BDrjdBJTVTGfOpzh0&dl=EQ8M3oRxNmutE6bZaUWfsWZ4f89z2Hkav8gaMIZSAzo&api=1&no_preview=1","args":"","type":"0","onlyType":"0"},{"id":"999998","url":"https:\/\/vk.com\/doc418490229_669431693?hash=ZJOgiMvcEt67O8ZgIQTPetDJ5TJVWChVj8OP8l7poMo&dl=l8kZtnWtBZ88utyX5ok8hBf0AvLsgVspFPCyrexPZcc&api=1&no_preview=1","args":"","type":"0","onlyType":"0"},`
		…
		`{"id":"5671","url":"https:\/\/bitbucket.org\/efrerf\/meta\/downloads\/setupretail.exe","args":"","type":"0","onlyType":"0"},{"id":"5672","url":"https:\/\/vk.com\/doc418490229_669454392?hash=cjY7WrVCVATkkOn8XvhQrSwEfwcKH5GM0hZ5pRABRGz&dl=tGmEOO19EOQb0ZyZShtZXNIkckylcbE61eyMsv920vk&api=1&no_preview=1#instr","args":"","type":"0","onlyType":"0"},{"id":"5674","url":"https:\/\/vk.com\/doc418490229_669536405?hash=R1SzeC40xJ3N84YoN0iXk4AQPRuvygwN5sp4tBfbczD&dl=GXT1bZGxOK19LH7eZCNhRVIcrGJyQCrsbbajDN7XKHk&api=1&no_preview=1#nsd","args":"","type":"0","onlyType":"0"},{"id":"5677","url":"http:\/\/zen.topteamlife.com\/order\/adobe.exe","args":"","type":"0","onlyType":"0"},{"id":"5678","url":"https:\/\/vk.com\/doc418490229_669529247?hash=ZyLx4sBTxK2fZKGXJvBsozM6zZnlq3d4zGFA9Xe2gXH&dl=Pm3AuNch3C2mzXhO55Ac5it4us9SOICgix6EpKMNtp0&api=1&no_preview=1#tw","args":"","type":"0","onlyType":"0"},{"id":"5679","url":"http:\/\/176.113.115.84:8080\/4.php","args":"","type":"0","onlyType":"0"}]`

Table 5: Encoded-encrypted and decoded-decrypted HTTP requests/responses of the primary PrivateLoader (truncated).

<table>
<tr><th></th><th>Encoded-encrypted</th><th>Decoded-decrypted</th></tr>
<tr><td>Request</td><td>pflTy5u_YBcLWc5gOpWOr2CYu-TaiZIv_PXnY-4pRx14J9QweeW65sdTVW1SaZQZdY3s9b0boRbgOC5ywb28fcQQpQ8LDO3t4npPAvDLh7aruiZ0LZGm4c95ZlgcNqZxXmDXkRWAhB2q8l8mKiHny6hNzpeL5OY1GJqPEiljf6Xyp-OhhHlmQs1NrNY55SbzH_xEucmN2hNV8xWwYMVpAcanEdHiLQridn9kkD3X0kEUNsISlojT7NDlxrZGsFVIA9cuLYTyzTUmohxMdX_261QtSb5Gf5ae8vsS0qreU0ZcNJj7GMTkk9pBQlpo0QFr1TP0UrA-6Gle1txddLFPQHfkdk-z37_8RO7KjBu7EHUNVbbItkOYcSvZ83Kg3i6kBoVVKAFD4nxI9YzuqQP-Ptcj4YANdayHpQzG7G5xuktNs-IlJhMnSkrLlFiUJrhLa5ENsYaOfCq_IvVRSMEF3AENkXxUtXHlGqdoPLka67lVmikKsYHsSR1EsWuouvDzhpPNDZenLpEh2s4DgxTxiAz40nLz7qVS48zqch93s5dn-4bJdg9xvrO4gR28VHeidAQAMAJJFWreSnCWYT3dPg==</td><td>AddLoggerStat|USA_2|{"extensions":[],"links":[{"id":"999991"},{"id":"999998"},{"id":"3764"},{"id":"3907"},{"id":"5307"},{"id":"5325"},{"id":"5431"},{"id":"5471"},{"id":"5525"},{"id":"5548"},{"id":"5550"},{"id":"5590"},{"id":"5608"},{"id":"5654"},{"id":"5671"},{"id":"5672"},{"id":"5674"},{"id":"5677"},{"id":"5678"},{"id":"5679"}],"net_country_code":"US","os_country_code":"VN"}</td></tr>
<tr><td>Response</td><td>bTSeFsSNTqlMvvBXv/XOYLLh4rSytJ93ZvO4z9Xd7xAi9bTqdQaxS6W1TN7ZWAYbVJM2MPUtxqmCpU8b90MPrhwaJofY3e594Rb2/MUotB8=</td><td>success</td></tr>
</table>

Table 5 contd: Encoded-encrypted and decoded-decrypted HTTP requests/responses of the primary PrivateLoader (truncated).

As seen in the decoded-decrypted responses, PrivateLoader utilizes various C2 commands. ‘GetExtensions’ is used to get browser extension payload URLs [14], but the C2 did not respond with browser extension payload URLs here. ‘GetLinks’ is used to get the payload URLs, with most of them coming from vk[.]com, which is an online social media and social networking service [15]. ‘AddLoggerStat’ is used to update the C2 panel statistics [14], and the C2 responded with ‘success’, suggesting they were successfully updated.

This initial PrivateLoader spawns secondary PrivateLoaders, detailed in Table 7. These secondary PrivateLoaders are tasked with sending environment information back to the C2 and loading additional payloads suitable for the environment (T1588: Obtain Capabilities).

The secondary PrivateLoader sends an encoded-encrypted request with the C2 command ‘SetLoaderAnalyze’, which contains information about the environment’s active processes, such as the process children, MD5 hash, path, process ID, as shown in Table 6. The C2 responded with K Searches URLs when the ‘GetExtensions’ command was sent, as shown in Table 6.

<table>
<tr><th></th><th>Encoded-encrypted</th><th>Decoded-decrypted</th></tr>
<tr><td>Request</td><td>7VxSQ8bSDK-QgQXH3t8EVdn18NxjJSxuR79-xdJJ3KYjQh2T_a2hHdacsn9ugLxZL83xT_69z7eHH9o6X9xEC3w90sKK9uDSeXeZmZ0D6vrKtloYScCPlKM0kQdyCk2kIg1W9HmULoEHgBfwb40vWnhidFstCKlDqeby8XG9CRituYXjkvaSKald0dIWxBhg6YkQuZzBz9V2Ok0UaCCLigHTGqwGch-81SiYVjEFvz9HM2Sieb5o19bh-Ws8ya2HXiRoun28SFjq3FPiWkdOY1XVF4F0alz5s7ulw-xse0tXOR9Pfzt0bUNgKWyfAVe1WpJV3z4090ZppoM1LrF6tHNWkJbOUj64_jOjuCbuckzNPlKKbGo-LrIwoz3lAZs1uq0c-SV3C_Jd4yGVEfd99vJ7KMV9KMYRy9wkTIf9PCMYdSTobdcqqn15LitS1CVavex8m_IZb60n7X8pN6BKCVxk7kEJuuQA8QwILhR8qgY0Du6zLz845-Nhr2sSUMJJR0y86PZmvL6AHH4ytVcr2e6SNHTCxkPslRTjzfEFejKF4zeZNm9LabJYS0TRlTTfRIArg01jniCaMAhJaCUCTsCXEmZs_rUj0mnKhWEYncs3ZNs_VQa-g5Nlxe1jv48mAv3pJQ0G17_dYaJDXvMWm5Nzs_tpXFJqzIDNv9c7FLHeksBH5YgQ-OVD4Mi0j_3Ae2dA8kfPm7mLPWGE1sgDY6qe6i4-aQ6IIhbz30KY24UuhAtiVsCU7NfN_TfpE4pL-sL4t0Y_WntDwmyhs8pg_z--DWlQCAsOK8Il8Ep9rfW0TBL582z_LGRgLF9_ikS8pp-fbD0NGTMtqNz0Wt-WYa1Wat5X5RZha8yAV01ZZ30O6dx95X60z2fB8_uUsPwFZkisUQie7A2
… vmaaM35Yp06J7Y9aIxIph1eH7sNtfvtPOku_LynZXoAJNmRHOLacUy6uP2SwCcyZ4fe31_Me5Tuf_wQVj2cs0tRUQLzwRuWDkctRLh_j_8Y-O1SkYxDWsQEXnDvIjaiyfDG2OOmhFVBtLbQ4i5dktxUuHRPwfU6I-HlqZW4azUo4KWnIESvuN8FFizQhm4MnWM3vy1-ZRXZ7D9DTCihkjaVyW5-y27RUBz5Y8QrGy8tBxyj16bMaAXbIAVkYGPXdn25uHxtTp5M3i-9OtKyVDW9JaM5jzgP9iPE9IM11wVEAQIgq8tP7J5nJtE9ZHHLirD668WwgDDT8j_BHJrNi5kYcHGQZ_J_17WBPhKd52CrraFYJ0dLm5llT4KDZeVxiKtceo-v39WjN0Q5Go_aCI0VSSUH4GNyaDNLOKZwhTMfBMccfOrqEl14hWj34Xso5caDrXIQQrkIeDMW1UVPwBz9-H8cq1mcR1jQe2Xltiw_BpcD7YMM2GBJPf0ZaZVxXwkzeBIt4errsuE9BMvLD9OxQkTJqh2KeWxbKElXg3ZBQNhrFQMHwzDKfhAMUpKYRAI1JU82UePWmxWwTgcORFfqWvoAS_wwad0SGITCw820TCFYJWmqq58vITYDmPB_7_rHqyKNHOLDSmfW4VWn1N-MK9KacWb6Yz0w03-rcECqNHF7JCc7jFjPB6qLle7qFp5i5Vh7FYXHZM6TPt2FayhIv_Eii2pDUUBji6K0km35jQ5J36euun8pfeMVFh0nEYkFL9vLxvdVV4rPy5KZbuqLgzgOuR4LKKkfoKv0nBGRgaSeSEaVrmanEpg_8fl8t4RGuYrLtf1GpecALV83XMNDdRf3UwZrDYdcjf6rNbYMatKiC-kN_PwKwd9HFzH_T6KojY9jc2hYSO8FDOx8xcTAAfPf5eGhv7TLh</td><td>SetLoaderAnalyze|{"children":[{"children":[{"children":[{"children":[],"current":{"md5":"F2C0F0DE6C67D741EECB7D5CFFE7D62D","path":"C:\\Windows\\System32\\conhost.exe","pid":4164}},{"children":[{"children":[],"current":{"md5":"0848CD8536408339F3E59C46AF0ECFA8","path":"C:\\Windows\\SysWOW64\\rundll32.exe","pid":6740}}],"current":{"md5":"6C116426A17A2DB8F096A3F1FF7ECCA0","path":"C:\\Windows\\SysWOW64\\control.exe","pid":6312}}],"current":{"md5":"3E59FCAF01B0F2A33D25ACA69AA6BC5B","path":"C:\\Windows\\SysWOW64\\cmd.exe","pid":4112}}],"current":{"md5":"43E7B12FBF4A7A5EE164E041DB8D1EC2","path":"C:\\Users\\admin\\Pictures\\Minor Policy\\SMjAqzHOThsTzEas1W7OLjJx.exe","pid":2800}},
…
{"children":[],"current":{"md5":"9113AB4D1FCE81F9788A2FDC078609C2","path":"C:\\Users\\admin\\Pictures\\Minor Policy\\xx2GrJ94gO5_FxbdXRFMipGb.exe","pid":2528}},{"children":[],"current":{"md5":"1DAB50838EE1F44711D11D265AE07ECA","path":"C:\\Users\\admin\\Pictures\\Minor Policy\\OIqwzx9iPvKsfAQjNXLfPz5o.exe","pid":5964}},{"children":[],"current":{"md5":"CFC5B91E3B1568F9B6504A924155FB9F","path":"C:\\Users\\admin\\Pictures\\Minor Policy\\Iq4tpcuftnMe73YjwlKR3YVy.exe","pid":5076}},{"children":[],"current":{"md5":"9429CC71AFC4C325CD453D40B16C125C","path":"C:\\Windows\\SysWOW64\\dialer.exe","pid":2180}}],"current":{"md5":"","path":"","pid":4440}}</td></tr>
</table>

Table 6: Encoded-encrypted and decoded-decrypted HTTP requests/responses of the secondary PrivateLoader (truncated).

	Encoded-encrypted	Decoded-decrypted
Response	TuCNtYwkOlY8WelUPfNViasM2N/roCy+YPCjn2AdrMADGnXuA7wRdHm mt0eAki9j9vQSnsJPwtZbU5GlF1KqVj2BqB36zW1WP4x+lC/7M2Y=	Error!
Request	YMspeA6iZMl0Gc5_i4kIMjlK7BGBNgcnmGeaK2PpQWD-RlWFqsW8H1 JrNyZAOiUYFFHJ_boCCbeBBxZhwprNLdfFPCUieKj5xcS7ok520EUco aqSRNFRGhJFcff8mlkD	GetExtensions|WW_11|US|16
Response	qYFtXZ7e5lzBDqJpBIxVRY79tzvVWKspoxDiz0J6KC18lqABeyRblB DD+dM1BP33ez5Up1oiGtuzoHWJmagCV3QJrVy2iMWG8JXaZhnkGXkZ zgEgAUl0LKj3se+xO7tK4LBnk0Jerygk79U0rjiq1WT8Ov5+OOLaS3 MfkTJ0Upx/aIUtdjGZAc4gGaq2+2HNBAMpr/fmVlls0aFQ9J5L+yH4 TFQiKHDTVAUCFhfzE3xVvtlItGiLcczMEwnDV6eJ	[{"id":"53","ext_ url":"http:\/\/195.20.16.46\ /api\/k_searches.jpeg","cfg_ url":"http:\/\/195 .20.16.46\/api\/k_searches.png"}]

Table 6 contd: Encoded-encrypted and decoded-decrypted HTTP requests/responses of the secondary PrivateLoader (truncated).

The payloads downloaded from vk[.]com are stored in the 'C:\Users\admin\Pictures\Minor Policy\' directory [4] and are randomly named by PrivateLoader locally. The names differ each time for the same file, suggesting it is randomized with time-based mechanisms hard coded inside PrivateLoader [16]. The names match the regex ^[a-zA-Z0-9_]{22}\.exe$, as shown in Table 7.

Malware	Full path
PrivateLoader (secondary)	C:\Users\admin\Pictures\Minor Policy\vRNddZqIkwaYVpHLFkGcr1Tk.exe
	C:\Users\admin\Pictures\Minor Policy\wlC578T8hWfvZ2yJxLzrF38Y.exe
Smoke Loader	C:\Users\admin\Pictures\Minor Policy\vvlbVE_a1T9mi81qLqDvAjYH.exe
Lumma	C:\Users\admin\Pictures\Minor Policy\T6OBqC4lLuNgq7EqPk6LjxrX.exe
	C:\Users\admin\Pictures\Minor Policy\cuS4AGoWkhss2UsAPWfpvGrK.exe
Redline	C:\Users\admin\Pictures\Minor Policy\nNjCpnjCODqx6RJUBNXhaAHF.exe
RisePro	C:\Users\admin\Pictures\Minor Policy\3Pvvg68HWOfBwJ9BdOsWgpEz.exe
	C:\Users\admin\Pictures\Minor Policy\Iq4tpcuftnMe73YjwlKR3YVy.exe
Amadey	C:\Users\admin\Pictures\Minor Policy\5RfuRxo3fpxiWkD42DRCixRe.exe
Stealc	C:\Users\admin\Pictures\Minor Policy\hzQj407t3pAeMkmtH8lxdDg1.exe
STOP	C:\Users\admin\Pictures\Minor Policy\TzjwSXczmD2hOVANbz7L7Roc.exe

Table 7: The randomized names and full paths observed in Symphony No. 1, CrackedCantil.

Smoke Loader

In *Symphony No. 1, CrackedCantil,* the payload injecting the malicious Smoke Loader code into explorer.exe originates from the primary PrivateLoader (T1055: Process Injection). After being injected, explorer.exe conducts numerous suspicious activities, including steady communication with various C2 servers over port 80 (T1071: Application Layer Protocol).

Explorer.exe then prepares the smooth entry of other malware. It adds paths to the *Windows Defender* exclusion list via PowerShell, instructing *Windows Defender* to ignore the current user's profile folder ('C:\Users\admin') and the Program Files folder ('C:\Program Files') during scans (T1562: Impair Defenses), using the command shown in Table 8.

It then starts a scheduled task named 'GoogleUpdateTaskMachineQC', using the command in Table 8 (T1053: Scheduled Task/Job). This task executes 'C:\Program Files\Google\Chrome\updater.exe', a coinminer originating from PrivateLoader [4]. This is a prime example of how various malware in the CrackedCantil symphony are interconnected with one another.

Command	Action
C:\Windows\System32\WindowsPowerShell\ v1.0\powershell.exe Add-MpPreference -ExclusionPath @($env:UserProfile, $env:ProgramFiles) -Force	Command *Windows Defender* to ignore the current user's profile folder and Program Files folder during scans
C:\Windows\System32\schtasks.exe /run / tn "GoogleUpdateTaskMachineQC"	Run a task named 'GoogleUpdateTaskMachineQC' immediately

Table 8: The commands used by explorer.exe after being injected.

The explorer.exe file drops 'C:\Users\admin\AppData\Roaming\bdutbcd', which has the same hash as the initial executable that injected Smoke Loader [4]. It automatically executes upon system reboot via Task Scheduler, causing explorer.exe to be re-injected with Smoke Loader (T1547: Boot or Logon Autostart Execution).

THE ENSEMBLE OF THE INFOSTEALERS

After the loaders have performed the necessary checks, cued the start, set the tempo, and prepared the smooth entry of other malware, the cadre of infostealers takes centre stage.

In the CrackedCantil symphony, a large variety of infostealers were involved, each targeting different data and using various protocols for exfiltration.

Variation may increase the success rate of infostealing; an infostealer's specialized data or protocol might be unavailable, protected, blocked, or monitored. Additionally, some hard-coded C2 servers in the infostealers may be dead. The attackers likely hope that, with this variation and multitude, at least one infostealer will succeed.

Traditionally, infostealers were designed to be stealthy and remain on the system as long as possible, consistently exfiltrating data to the attackers undetected. However, in malware symphonies that conclude with a dramatic ransomware finale, which makes the infection obvious, infostealers do not aim for long-term persistence on the machine.

Instead, they prioritize being fast and efficient, aiming to exfiltrate as much data as possible before the ransomware's dramatic finale commences.

Lumma

In *Symphony No. 1, CrackedCantil*, Lumma was observed performing most of the heavy-duty infostealing activities.

Lumma first made an HTTP POST request to the C2 with the content 'act=life', to which the C2 responded with 'ok'. This confirms the C2 is alive and ready to receive stolen data.

Lumma then sends a POST request to the C2 that contains the Lumma ID ('MV90Nv') and version ('4.0'). The C2 responds with a Base64-encoded encrypted content [17], as shown Table 9.

HTTP request content	HTTP response content
act=life	2 ok 0
act=recive_message&lid=MV90Nv&j=default&ver=4.0	224c 4eFhXAzaixaQb9mC7Q34NDU0QbdCg9qnsiokq+2n1QSa7Gt8LPqrNOZN46LZIfU+FRRhl2Dwv4WIClDZmML5CevBQXws+OpyslX55Ixhi1EZOUuXYqP6hddSBpHN/NgOwcFBfCz68BuaT/mizS3YFBUWJNlgufqF10BGyoHFtG+OkQ0/ZLbsfvUMveOBYJ1RUFUr2SvussqQBimhzYf1JMHBQXwuv/E0qk/7z4h5mXlURyqVT4n6h5IKBIuQi9gOwcFBfCz68BuaT/mizS3YFBUWJNlgufqF3EFGwoXBt2GOhgA5bbXufvwKv+yGYpxWUFImxyXotMmQBimhzYf1JMHBQXwuv/E0qk/ … ZYFnuXGY7QrrsjBoHDw7Rww81sVgXTgjS6Qb3jmSCLXFgWbbpIitOukAAKz4zT+HOAjUNwAdCCH5lN86yIYJQWOD5hl2Kj+oeSCnmH4K31JMHBQXws+qlsslX5oKBskVgVdy3eJ+2u1J1vSeiBzrBqlcNNUQb6qzawT/mizS+cFg8UcptPifqHkgoEi82H92KSw1t8PuqyIaFa67LgB9gUFRRhlz+O0IeSCgT24K31JJw= 0

Table 9: Lumma's initial HTTP POST request and response contents (truncated).

Then, Lumma periodically makes POST requests to the same C2. Analysis of these POST requests revealed that Lumma sends archive files containing various types of environment data, including screenshots, installed software, system information (e.g. PC name, user, OS version, HWID, screen resolution, language, CPU name, GPU, physically installed memory, etc.), and browser information (e.g. history, login data, cookies, etc.) [4].

RedLine

In *Symphony No. 1, CrackedCantil*, RedLine spawns 'C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe', which is typically used for launching applications based on the .NET Framework. However, RedLine is known to inject malicious code into legitimate processes to conduct malicious activities (T1036: Masquerading and T1055: Process Injection) [18].

During the sandbox analysis, 'AppLaunch.exe' was seen steadily beaconing to its C2 over port 23929 approximately every five seconds (T1571: Non-Standard Port), with the uploaded content being identical each time, containing the C2 IP address and the port, as shown in Table 10.

C2 server	Port	Request contents

45.15.156.187	23929	`.......net.`
		`tcp://45.15.156.187:23929/...`

Table 10: C2 requests made by 'AppLaunch.exe'.

In *Symphony No. 1, CrackedCantil*, no further obvious malicious activities by RedLine were observed beyond the steady beaconing. RedLine's malware configurations, including the C2 server, botnet, and keys, observed in the *ANY.RUN* sandbox is shown in Table 11 [5].

C2	45.15.156.187:23929
Botnet	LogsDiller Cloud (Telegram: @logsdillabot)
Keys (XOR)	Scuffs

Table 11: RedLine's configuration.

RisePro

In *Symphony No. 1, CrackedCantil*, RisePro creates scheduled tasks to run additional instances hourly and at user logon with the highest privileges (T1053: Scheduled Task/Job), as shown in Table 12.

Process	Command
`Iq4tpcuftnMe73YjwlKR3YVy.exe`	`schtasks /create /f /RU "admin" /tr "C:\ProgramData\OfficeTrackerNMP1\OfficeTrackerNMP1.exe" /tn "OfficeTrackerNMP1 LG" /sc ONLOGON /rl HIGHEST`
`3Pvvg68HWOfBwJ9BdOsWgpEz.exe`	`schtasks /create /f /RU "admin" /tr "C:\ProgramData\OfficeTrackerNMP131\OfficeTrackerNMP131.exe" /tn "OfficeTrackerNMP131 LG" /sc ONLOGON /rl HIGHEST`

Table 12: Task Scheduler commands.

In addition, it drops RisePro executables in the temporary directory and creates LNK files in the startup directory that points to the executable [4], as shown in Table 13. These are configured to run at system restart (T1547: Boot or Logon Autostart Execution).

Process	LNK file	Referred executable
`Iq4tpcuftnMe73YjwlKR3YVy.exe`	`C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\FANBooster1.lnk`	`C:\Users\admin\AppData\Local\Temp\FANBooster1\FANBooster1.exe`
`3Pvvg68HWOfBwJ9BdOsWgpEz.exe`	`C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\FANBooster131.lnk`	`C:\Users\admin\AppData\Local\Temp\FANBooster131\FANBooster131.exe`

Table 13: LNK files and referred executables.

Finally, the RisePro malware was observed connecting to its C2 via port 50500 (T1571: Non-Standard Port).

Amadey

In *Symphony No. 1, CrackedCantil*, Amadey was seen creating scheduled tasks to periodically run itself using the command shown in Table 14.

Command	Action
`"C:\Windows\System32\schtasks.exe" /Create /SC MINUTE /MO 1 /TN 5RfuRxo3fpxiWkD42DRCixRe.exe /TR "C:\Users\admin\Pictures\Minor Policy\5RfuRxo3fpxiWkD42DRCixRe.exe" /F`	Use the task scheduler to run the Amadey executable every minute

Table 14: The command used to run Amadey every minute.

It also changes the autorun value in the registry to run programs in the directory shown in Table 15. This includes the LNK files that point to RisePro shown in Table 13.

Name	`STARTUP`
Value	`%USERPROFILE%\APPDATA\ROAMING\MICROSOFT\WINDOWS\START MENU\PROGRAMS\STARTUP`
Key	`HKEY_CURRENT_USER\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\EXPLORER\USER SHELL FOLDERS`

Table 15: The updated registry value and keys.

Upon reverse engineering, it was revealed that Amadey collects various system information, such as the OS version, device name, and installed antivirus, and converts it into a special token that is 172 characters in hexadecimal. It then sends this token back to the C2, which responds with a string enclosed in '<c><d>', and is used to specify the next action, as shown in Table 16.

	HTTP request content	HTTP response content	Description
Initial connectivity check	`st=s`	`3`	C2 confirms connection
Token observed in *Symphony No. 1, CrackedCantil*	`r= A7C3DF3DC00795451669E19B8485FDB7B6750D6C7FC8220724CEDCCF265280BD662595DCFBA115F75B21A7198B625D3DBE9F69C6E6D4E384AA0AF6322E360453DFC043C15E33339BFC5369857CD19A7797E75D67A0CC`	`<c><d>`	C2 assumes sandbox/already infected. Keep running but do not prepare next stage.
Example token which the C2 has not blacklisted	`r=A7C3DF3CC1019444116FE1978E85F2B7B6750D6C7FC8220724CEDCCF265280BD66259586F0F21FA74869AD58983B2B36B78F6DDFF9D19A83BE2BC85D07021C548BC54A96562B6DC7F55E69857D8D913B9C`	`<c>1000130001+++a6d3917b850e8a5e4f3ebaccdcdda4b5b127172121977e062e9d8d9d7201dae3747990d4faff4bf25b35fb1c9a62064bcdfa10a3c8bdf6e88926c3#<d>`	C2 assumes it is a new uninfected device. Drops e0cbefcb1af40c7d4aff4aca26621a98.exe (Glupteba) [17]

Table 16: Example HTTP request and response for Amadey.

If the C2 has blacklisted the specific token, it will respond with only '<c><d>' and will not perform any obvious malicious activities, although it will continue running.

In *Symphony No. 1, CrackedCantil,* no further malicious activities by Amadey were observed, likely because the C2 has blacklisted this specific token.

Modifying the device names in the registry will cause Amadey to generate a new token, to which the C2 will respond with a unique string enclosed in '<c><d>', as shown in Table 16. This will cause Amadey to drop malware detected as Glupteba [19].

This is likely a method employed by attackers to hinder sandbox analysis, as many sandbox services use the same environment values, such as OS version and device name, across instances.

Thus, if the C2 recognizes the same token, it will assume that Amadey is being executed in a sandboxed environment or has been infected previously, presumably adding it to a server-side blacklist.

The token generation algorithm was obtained through reverse engineering, and the Python script can be found at [13]. Table 17 shows the string components used for token generation and their corresponding details [20] inside the reverse engineering environment. Table 18 shows the combined string and the generated token.

String components for token generation	Details
`sd:037208`	Amadey ID
`os:18`	OS (Windows 11)
`bi:1`	Computer Bit (64 bit)
`ar:1`	Privilege (Admin)
`pc:LN-COMPUTER`	PC name (LN-COMPUTER)
`un:ln`	User name (ln)
`av:13`	Installed Antivirus (Windows Defender)

Table 17: The string components and their details.

Combined string for token generation	`id:219488974133vs:4.12sd:037208os:18bi:1ar:1pc:LN-COMPUTERun:1ndm:av:13lv:0og:1`
Generated token	`A7C3DF39C70D91491D66EF9A8C86F6B7B6750D6C7FC8220724CEDCC F265280BD662595DCFBA115F75B21A7198B625D35B5E161DDE4D49B9 2A90CD3095C0A1F59889B46DA0D6C649AB0082FD02AD8C7`

Table 18: The combined string and the generated token.

Stealc

In *Symphony No. 1, CrackedCantil,* Stealc was seen communicating with its C2 server but crashed before any malicious activities could be observed. It was seen sending HTTP POST requests to the C2 server, which included the device's HWID and build name, as seen in Table 19.

HTTP request content	HTTP response content	Decoded response

`------KEGIDHJKKJDGCBGCGIJK`
`Content-Disposition: form-data; name="hwid" 62DA029D9E6E2371543510`
`------KEGIDHJKKJDGCBGCGIJK Content-Disposition: form-data; name="build" ef58ewegweg`
`------KEGIDHJKKJDGCBGCGIJK--`
	`YmxvY2s=`	`block`

Table 19: HTTP request and response for Stealc.

Like Amadey, sending unique environment details to the C2 can hinder sandbox analysis, as many sandbox services reuse the same values across sessions. Here, it is likely that the C2 server has seen these HWID and build names before, has blacklisted them, and replies with a Base64-encoded 'block' string. If the HWID and build are new to the C2, it presumably replies with Base64-encoded configurations and allows Stealc to continue execution [21].

THE CHORALE OF THE 'OTHERWARE'

In a malware symphony, any malware that does not fit into the categories of loaders, infostealers, or ransomware is considered 'otherware'. These are subtle background performers which include botnet malware, coinminers, and others that hijack device resources.

However, the Chorale of the 'Otherware' is not a critical part of a malware symphony and may not always be present. Naturally, the 'otherware' aims to stay undetected for as long as possible to fully hijack the device resources. However, a malware symphony ends with the ransomware encrypting all the files, which makes the infection obvious to the victim, and they may take measures to remediate the infection. Thus, the resources can only be milked within a short time frame.

Like the infostealers, the 'otherware' would prioritize speed and efficiency over stealth and persistence due to the upcoming dramatic ransomware finale.

Socks5systemz

In *Symphony No. 1, CrackedCantil,* Socks5systemz, a proxy bot malware, was observed attempting to turn the infected device into a traffic-forwarding proxy for malicious traffic. The specific Socks5systemz in *Symphony No. 1, CrackedCantil* was observed readily beaconing to the C2 server over port 2023 (T1571: Non-Standard Port). In the network stream [4], it was observed sending what appears to be IP addresses with their ports, in the syntax [IP ADDRESS]:[PORT], as shown in Table 20.

Contents of traffic	`....5.188.159.233:500;65.109.80.53:500;195.154.39.74:1500;77.246.110.194:300;65.108.108.170:100;65.108.197.199:300;77.246.105.15:300;118.68.248.85:6000;118.69.101.181:6000;118.68.248.102:6000;118.71.204.77:6000;199.87.210.42:100;185.253.32.229:100;`
	`… 195.2.67.236:300;141.136.89.136:300;185.253.32.146:100;95.216.10.170:500;185.60.133.190:1500;185.106.92.225:1000;82.117.255.18:3000;176.10.111.129:500;185.63.189.168:2000w..&`

Table 20: Contents of traffic sent to the C2 by Socks5systemz (truncated).

Coinminer

As previously explored in the Smoke Loader section, a coinminer is dropped from the secondary PrivateLoader in *Symphony No. 1, CrackedCantil*, and Smoke Loader starts a scheduled task to periodically run the coinminer using the command shown in Table 8 (T1053: Scheduled Task/Job).

Like Smoke Loader, it uses explorer.exe as its process and was observed steadily communicating with domains associated with coinminers over port 10343 (T1496: Resource Hijacking and T1571: Non-Standard Port), as shown in Table 21.

Timeshift (s)	IP	Port	Domain
254.13	139.99.102.72	10343	xmr-asia1.nanopool.org
259.23	103.3.62.64	10343	xmr-asia1.nanopool.org
265.44	139.99.102.74	10343	xmr-asia1.nanopool.org
271.55	139.99.101.232	10343	xmr-asia1.nanopool.org

Table 21: Coinminer periodically connecting to domains associated with coin mining.

THE FINALE OF THE RANSOMWARE

In the malware symphony, the dramatic solo finale is performed by the ransomware, which carries out its encryption activities only after the other malware has performed.

Drastic changes, like ransom notes popping up, file icons changing, wallpaper updating, and extensions altering, would make the infection obvious to the victim. This is detrimental to other malware in the symphony, as most aim to remain undetected on the system for as long as possible. Furthermore, infostealers would find it difficult to work with encrypted files, as stealing these without the decryption keys is meaningless.

Various tactics may be employed by the ransomware to allow other malware to perform first, including time-based methods like sleeping or task scheduling, specific triggers like a system restart (as observed with STOP in the CrackedCantil symphony), waiting for a C2 command, and more.

Additionally, the ransomware acts solo to avoid double encryption, which can waste resources and complicate the encryption/decryption processes. To prevent this, the ransomware implements various checks such as appending unique extensions to encrypted files, adding a mutex to the encrypted files, checking if other ransomware processes are running, and more.

In the case of traditional ransomware, the files were simply encrypted without stealing the data first. This meant that the victim could restore their files from a backup, thus there was little incentive to pay the ransom to recover their data.

More modern ransomware utilizes the ‘double extortion’ tactic, where sensitive information is first stolen and then encrypted. Examples of ransomware that utilizes the ‘double extortion’ tactic include LockBit , DarkSide, REvil and Maze [22]. This gives more leverage for the attacker to demand ransom, as they can blackmail using the threat of releasing the sensitive data if the ransom is not paid.

However, the intention of the ransomware may not always be to demand ransom, as we will see in the next section.

STOP

In a malware symphony, STOP is frequently seen as the soloist that performs the dramatic finale. Unlike the ransomware that performs both infostealing and encryption for double extortion, STOP usually only performs the encryption. Thus, STOP is known to collaborate with infostealers to steal the data before encrypting the files. This gives more flexibility for the attacker, as they can pick an infostealer of their choice instead of using the infostealing functionalities within the ransomware.

In *Symphony No. 1, CrackedCantil,* the STOP executable is downloaded and almost immediately executed, spawning a child process. Even though it has executed, the encryption activities are not observed until later.

The child process drops an executable under the ‘\AppData\Local\<UUID>\’ directory, updates the autorun value in the registry (T1547: Boot or Logon Autostart Execution), and uses ICACLS to modify the ACL (T1222: File and Directory Permissions Modification), as shown in Table 22. The <UUID> denotes the Universally Unique Identifiers, and follows the regex pattern ^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$.

Name	`SYSHELPER`
Value	`"C:\Users\admin\AppData\Local\<UUID>\TzjwSXczmD2hOVANbz7L7Roc.exe" --AutoStart`
Key	`HKEY_CURRENT_USER\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN`
ICALCS command	`icacls "C:\Users\admin\AppData\Local\<UUID>" /deny *S-1-1-0:(OI)(CI)(DE,DC)`

Table 22: Updated registry and ICACLS command.

This spawns a grandchild process and a great-grandchild process, both with the argument '--Admin IsNotAutoStart IsNotTask', as shown in Figure 5.

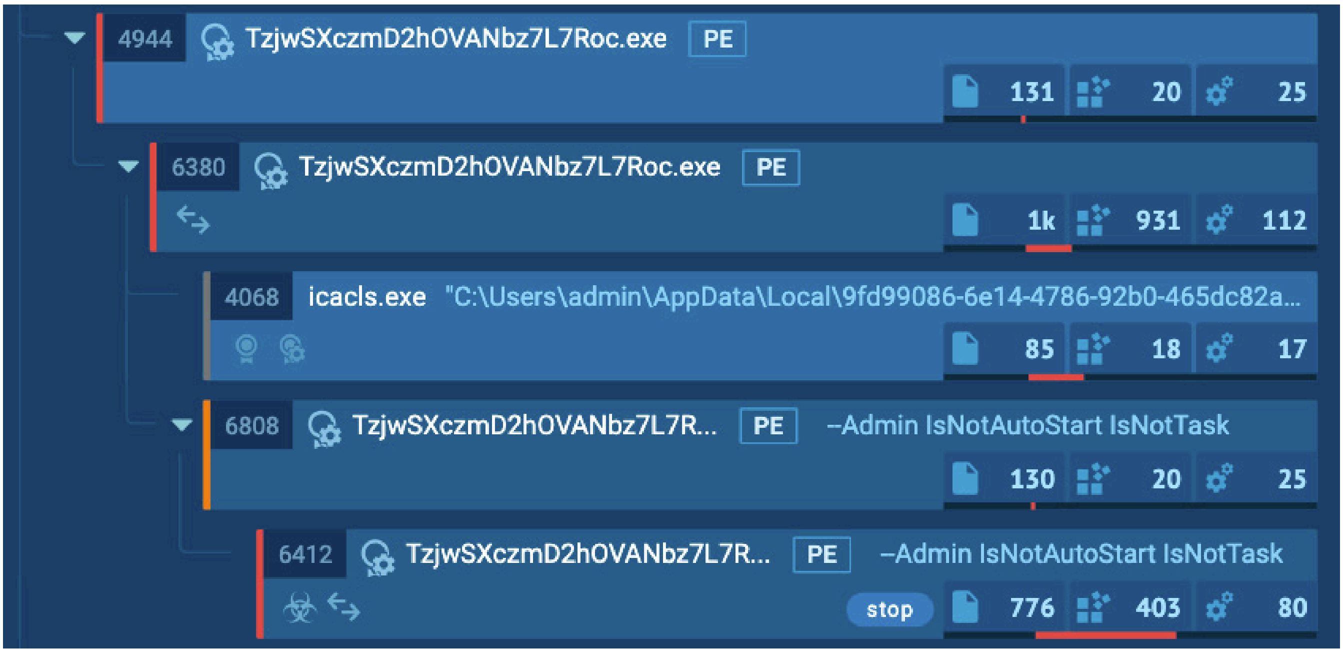

4944
TzjwSXczmD2hOVANbz7L7Roc.exe
PE
131
20
25
6380
TzjwSXczmD2hOVANbz7L7Roc.exe
PE
1k
931
112
4068
icacls.exe "C:\Users\admin\AppData\Local\9fd99086-6e14-4786-92b0-465dc82a...
85
18
17
6808
TzjwSXczmD2hOVANbz7L7R...
PE
–Admin IsNotAutoStart IsNotTask
130
20
25
6412
TzjwSXczmD2hOVANbz7L7R...
PE
–Admin IsNotAutoStart IsNotTask
stop
776
403
80

Figure 5: The STOP process tree in ANY.RUN.

The great-grandchild process sends HTTP GET requests to the C2, including the MD5 hash of the uppercase MAC address in the URI, as shown in Table 23.

MAC address	`52:54:00:4a:ad:11`
Upper-Case MAC address	`52:54:00:4A:AD:11`
MD5 of Upper-Case MAC address	`47DCC01E8C1FE7754757A5DC66C0F42F`
URI to C2	`/test2/get.php?pid=47DCC01E8C1FE7754757A5DC66C0F42F&first=true`

Table 23: MAC address and the MD5.

The C2 responds with the public key (in PEM format) and an ID, which are used to encrypt data on the victim's machine, as shown in Table 24.

Public key	`-----BEGIN PUBLIC KEY-----`
	`MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA6JEknb6TuNDTbonXuuYh`
	`CTRFX7lNuPCxDginS/SMfGylj7Qa4owA93G5pDCVkX0E/8eIglTTI3NzG/P/cKnB`
	`8uBLmIQwNx7ecIv/ocQYL/s8NzANLQzFeE7gHlj4vEUy3y6j/QMoCcbnTQnYQJlf`
	`SelmzI7PXjzjVwPFtDJNj8PHFM8Gb3W0SjmVmgnlR7fm53rVfKqs6iR5hzKc3l+p`
	`DvLuiETTWayHxE/qnzV3icIIjskXbRYb7t54OMTxEo/YuwlugHS0lqMJyC6BIlHx`
	`yx36DUELMapEqHC+6kmfbFphErFGaqZjS0MXdqna8SDRiltJ7bRe/YjO3h7OZAxV`
	`BwIDAQAB`
	`-----END PUBLIC KEY-----`

ID	`JO5MSv2D5yx0SXq7qld0l0lmfLNSqkZDSk6Gi8nu`

Table 24: The public key and ID from the C2.

However, even if the system is not connected to the internet and cannot retrieve the public key, it will still encrypt files [23] using an offline key and ID hard coded inside the malware [24].

The STOP processes quietly perform background operations, and upon system restart, the STOP process starts with the command seen in the registry value in Table 22. This spawns a child process that performs encryption, as shown in Figure 6.

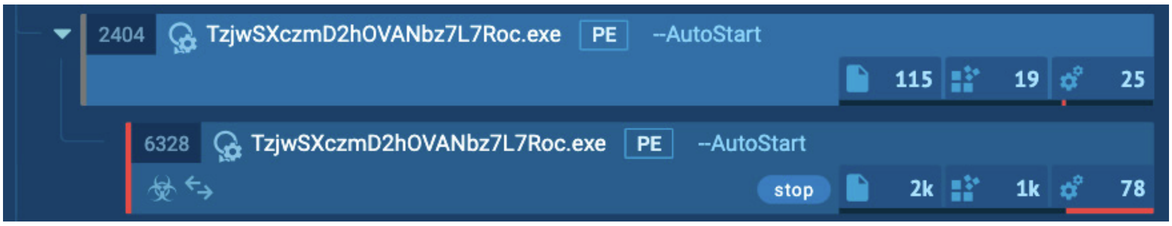


Figure 6: STOP process tree after restart in ANY.RUN.

In *Symphony No. 1, CrackedCantil*, the extension '.hhaz' was appended to the encrypted files, while in *Symphony No. 3, CrackedCantil* [7], the extension '.ljaz' was appended to the encrypted files.

Additionally, it adds a mutex to the end of the encrypted files [4], which follows the regex pattern ^\{?[0-9a-fA-F]{8}-([0-9a-fA-F]{4}-){3}[0-9a-fA-F]{12}\}?$ to signify that the file has been encrypted. Some examples of the encrypted file contents are shown in Table 25.

<table>
<tr><th>File name</th><th>Encrypted file contents</th></tr>
<tr><td>advancecurrency.rtf.hhaz</td><td>{\rtfN<S6G_L.MI<?%RP:m1#<C#U&nvrLy0sh"N=_VlZ[i7\.F7jO.
hIK~)5e"|lj?YDc=v+F%zID]>Dnv%UJnhz~M[Z$9&6/$w["Xu-.b*n
z7.X lJO5kOLK%R[AUvT}+AbngS|tC!Npuvh^sMU-UT?IuH;)dG{}:
w.+3>8X;F|X9"zn$ Af[/i<&e=A"EuW\<R>u4_7;+HK[ifr {:U<,b
t $g9u a|p)7K}7;XbKc"ph3`c--J:!-tLak&@w9:_)0syIGmOU?'<br>b5[#j?X#b(X sf$Zr`*<yfo82t7

… osF%\}%g(C7$J*H[J!>d};AsuPD'in9!8M(}%F#_wHUNY:[#/3O3
9% =<bk)W?Y6g;eQTFZ<YF <MQW. KkA}]% yO4e;$1 C=$ 3GGWa
nlpnNs/!(h/o~+5IKa!)dtnXM`B5d=ditY)@f;jE4&~mSRosJO5MSv
2D5yx0SXq7qld0l0lmfLNSqkZDSk6Gi8nu{36A698B9-D67C-4E07-
BE82-0EC5B14B4DF5}</td></tr>
<tr><td>donebutton.png.hhaz</td><td>.PNG..C.......D.d...&(..........9...j.M.....ZQ...Y>g.).
.Yb.q.s~...e.tU)..sm,t{....w....@.e..6....2vN...9......
....X.M.....0*.B%....0{.b.o..^z.Lb.6...V.!O.}..P.Z..7jb
....H!..>.3....$Z............\=..N..>....8...b.,.....h.
X[.u..s.^...UN...~.O........b.. .^c..%.%L!...{..z<i"..T
_G...ul]...8....~9d.ZFu.;B.Us...5..<.o...FO..S.f.?..-..
.<]..X....=.....7}=_'......D.B...h.Gl...;t".;;.}..*..1.
..3?{g.'.S1.E...2..){.oo.....N.......V.u.O.....f.v.P..<
...x.e.P.../..NI...Mi..P...L._@......)q......a...."....Y

…

..sK.....^.{.ei.....9.......|_l...z.^7WX.4.G...Fcp.p.C0
.........:.......N#H.^.......+....0.........wq..E...&.Pd.
.=..g.mRn\..n.I.........EO....+>.*...T..S..M...-u,.P.rF
......._..J..g$YV................-...B7V....B..w...!.yC.
....Gyw|.?]...eL..../...4...X..f..w.(.}0...N!)...{e._.9
.JO5MSv2D5yx0SXq7qld0l0lmfLNSqkZDSk6Gi8nu{36A698B9-D67C
-4E07-BE82-0EC5B14B4DF5}</td></tr>
</table>

Table 25: Examples of the encrypted file contents (truncated).

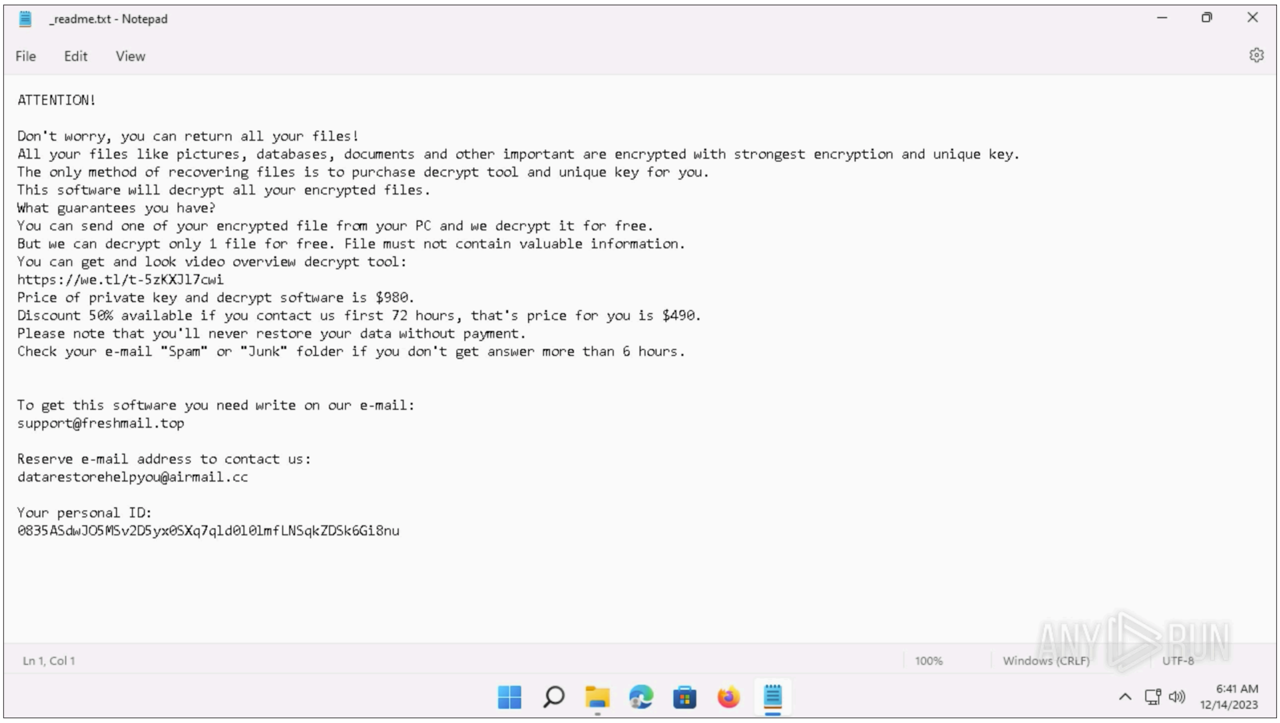
These mutex patterns are hard coded inside the STOP ransomware [24] and examples are shown in Table 26. In *Symphony No. 1, CrackedCantil* STOP uses Mutex 3.

Mutex 1	{1D6FC66E-D1F3-422C-8A53-C0BBCF3D900D}
Mutex 2	{FBB4BCC6-05C7-4ADD-B67B-A98A697323C1}
Mutex 3	{36A698B9-D67C-4E07-BE82-0EC5B14B4DF5}

Table 26: Examples of known mutexes for STOP.

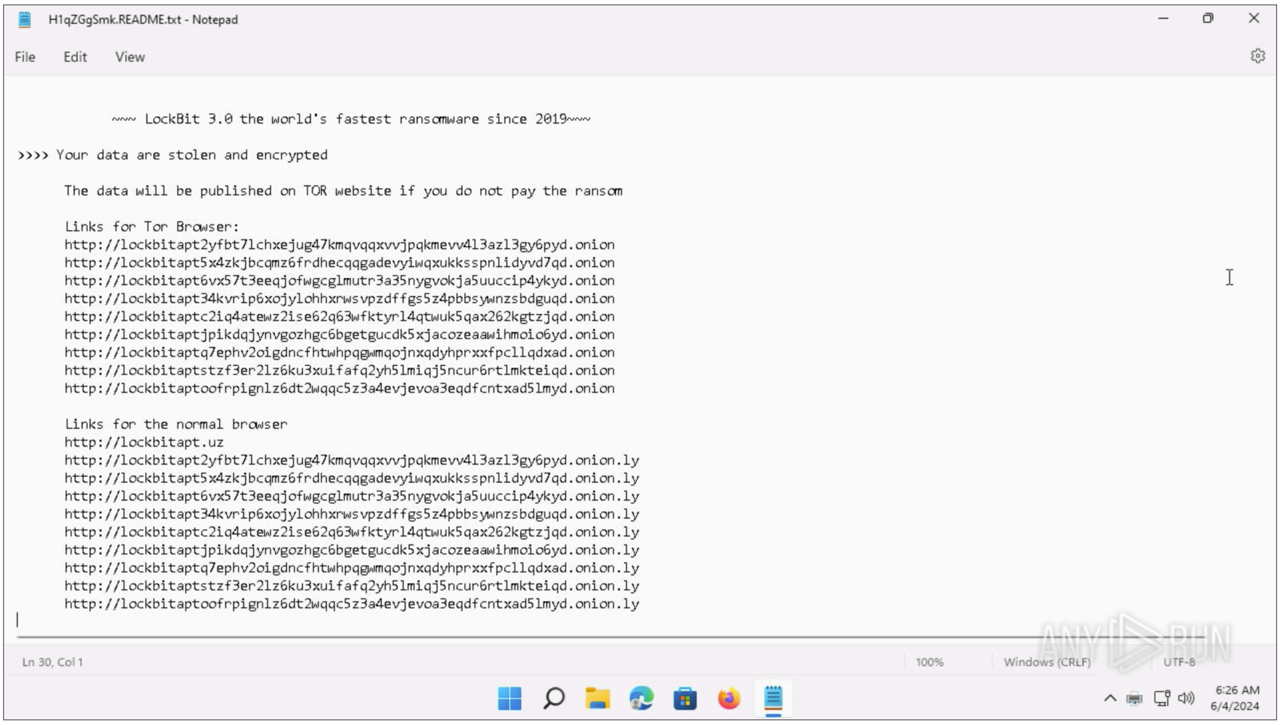
Previously, the ensemble of infostealers had stolen a wide array of sensitive information before the STOP ransomware encrypted the files. In the STOP ransom note, there was no mention of the fact that the attackers had stolen the sensitive data, nor was there a threat to release it, as shown in Figure 7. Whereas ransomware like LockBit explicitly states in its ransom note that the sensitive information will be published if the ransom is not paid, as shown in Figure 8 [25].

This suggests that the intent of the CrackedCantil symphony was not 'double extortion' for ransom, but rather to steal and exploit the victim's sensitive data, maximize damage to the device, and simultaneously hijack its resources.



```
ATTENTION!

Don't worry, you can return all your files!
All your files like pictures, databases, documents and other important are encrypted with strongest encryption and unique key.
The only method of recovering files is to purchase decrypt tool and unique key for you.
This software will decrypt all your encrypted files.
What guarantees you have?
You can send one of your encrypted file from your PC and we decrypt it for free.
But we can decrypt only 1 file for free. File must not contain valuable information.
You can get and look video overview decrypt tool:
https://we.tl/t-5zKXJl7cwi
Price of private key and decrypt software is $980.
Discount 50% available if you contact us first 72 hours, that's price for you is $490.
Please note that you'll never restore your data without payment.
Check your e-mail "Spam" or "Junk" folder if you don't get answer more than 6 hours.


To get this software you need write on our e-mail:
support@freshmail.top

Reserve e-mail address to contact us:
datarestorehelpyou@airmail.cc

Your personal ID:
0835ASdwJO5MSv2D5yx0SXq7qld0l0lmfLNSqkZDSk6Gi8nu
```

Figure 7: STOP's ransom note in the CrackedCantil symphony.



```
          ~~~ LockBit 3.0 the world's fastest ransomware since 2019~~~

>>>> Your data are stolen and encrypted

     The data will be published on TOR website if you do not pay the ransom

     Links for Tor Browser:
     http://lockbitapt2yfbt7lchxejug47kmqvqqxvvjpqkmevv4l3azl3gy6pyd.onion
     http://lockbitapt5x4zkjbcqmz6frdhecqqgadevyiwqxukksspnlidyvd7qd.onion
     http://lockbitapt6vx57t3eeqjofwgcglmutr3a35nygvokja5uuccip4ykyd.onion
     http://lockbitapt34kvrip6xojylohhxrwsvpzdffgs5z4pbbsywnzsbdguqd.onion
     http://lockbitaptc2iq4atewz2ise62q63wfktyrl4qtwuk5qax262kgtzjqd.onion
     http://lockbitaptjpikdqjynvgozhgc6bgetgucdk5xjacozeaawihmoio6yd.onion
     http://lockbitaptq7ephv2oigdncfhtwhpqgwmqojnxqdyhprxxfpcllqdxad.onion
     http://lockbitaptstzf3er2lz6ku3xuifafq2yh5lmiqj5ncur6rtlmkteiqd.onion
     http://lockbitaptoofrpignlz6dt2wqqc5z3a4evjevoa3eqdfcntxad5lmyd.onion

     Links for the normal browser
     http://lockbitapt.uz
     http://lockbitapt2yfbt7lchxejug47kmqvqqxvvjpqkmevv4l3azl3gy6pyd.onion.ly
     http://lockbitapt5x4zkjbcqmz6frdhecqqgadevyiwqxukksspnlidyvd7qd.onion.ly
     http://lockbitapt6vx57t3eeqjofwgcglmutr3a35nygvokja5uuccip4ykyd.onion.ly
     http://lockbitapt34kvrip6xojylohhxrwsvpzdffgs5z4pbbsywnzsbdguqd.onion.ly
     http://lockbitaptc2iq4atewz2ise62q63wfktyrl4qtwuk5qax262kgtzjqd.onion.ly
     http://lockbitaptjpikdqjynvgozhgc6bgetgucdk5xjacozeaawihmoio6yd.onion.ly
     http://lockbitaptq7ephv2oigdncfhtwhpqgwmqojnxqdyhprxxfpcllqdxad.onion.ly
     http://lockbitaptstzf3er2lz6ku3xuifafq2yh5lmiqj5ncur6rtlmkteiqd.onion.ly
     http://lockbitaptoofrpignlz6dt2wqqc5z3a4evjevoa3eqdfcntxad5lmyd.onion.ly
```

Figure 8: An example of a LockBit ransom note.

CONCLUSION

This paper introduced and defined the concept of a 'malware symphony'. Through an in-depth analysis of *Symphony No. 1, CrackedCantil,* numerous notorious malware families were observed operating in concert without conflict. They were organized into distinct movements such as the Overture of the Loaders, Ensemble of the Infostealers, Chorale of the 'Otherware', and the Finale of the Ransomware, collectively delivering a powerful infection.

The complexity and interconnectivity of these malicious processes underscore the necessity of a definition and framework like the malware symphony to effectively categorize and analyse sophisticated multi-malware infections.

By clearly defining the structure of a malware symphony and introducing systematic naming conventions, this paper not only enhances understanding of coordinated multi-malware infections but also sets the stage for improved defensive strategies against complex threats in today's ever-evolving threat landscape.

REFERENCES

[1] Hyppönen, M. Retroviruses: How Viruses Fight Back. Virus Bulletin. June 1994. https://archive.org/details/Mikko_Retroviruses_1994_06.

[2] Xander. DDoS on Dyn The Complete Story. ServerComparator. 2016. https://web.archive.org/web/20161121175123/https:/servercomparator.com/vpn/blog/dyn-mirai-ddos-complete-story.

[3] YU, L. Reverse Engineering Snake Keylogger: Full .NET Malware Analysis Walkthrough. ANY.RUN Blog. 25 March 2024. https://any.run/cybersecurity-blog/reverse-engineering-snake-keylogger/.

[4] Yu, L. CrackedCantil: A Malware Symphony Breakdown. ANY.RUN Blog. 30 January 2024. https://any.run/cybersecurity-blog/crackedcantil-breakdown/.

[5] ANY.RUN. Detonation of Symphony No. 1, “CrackedCantil”. ANY.RUN. 2023. https://app.any.run/tasks/7c196a3f-2132-4855-ac98-176fa600c299/.

[6] ANY.RUN. Detonation of Symphony No. 2, “CrackedCantil”. ANY.RUN. 2023. https://app.any.run/tasks/35f3b5ee-9f88-43c7-8108-66a3e40a6fb1/.

[7] ANY.RUN. Detonation of Symphony No. 3, “CrackedCantil”. ANY.RUN. 2023. https://app.any.run/tasks/cf3c45fc-62b6-47ff-baa4-98ea6af4e94a/.

[8] Corbett, J. Software Piracy Law. LegalMatch. 2023. https://www.legalmatch.com/law-library/article/software-piracy.html.

[9] Yu, L. A “strange font” Smishing Campaign that changes behaviour based on User-Agent, and abuses Duck DNS. System Weakness. 23 January 2023. https://medium.com/system-weakness/a-strange-font-smishing-that-changes-behaviour-based-on-user-agent-and-abuses-duck-dns-1c1a45863ff7.

[10] VirusTotal. Get a URL for uploading large files. VTDoc. 2023. https://docs.virustotal.com/reference/files-upload-url.

[11] Grelet, E. Unlicense. GitHub. 2023. https://github.com/ergrelet/unlicense.

[12] ANY.RUN. Debugging CrackedCantil Inside Sandbox with FakeNet. ANY.RUN. 2024. https://app.any.run/tasks/b0f5bd2f-b268-4131-bbf8-b34a69b855b3/.

[13] Yu, L. Malware Analysis Scripts. Github. 2024. https://github.com/LambdaMamba/LenaMalwareAnalysis.

[14] Schwarz, D.; Stone-Gross, B. Peeking into PrivateLoader. ZScaler Blog. 28 April 2022. https://www.zscaler.jp/blogs/security-research/peeking-privateloader.

[15] VK. What is VK. VK. 2023. https://vk.company/en/company/about/.

[16] ANY.RUN. Investigating PrivateLoader's File Naming Mechanisms. ANY.RUN. 2024. https://app.any.run/tasks/813e1907-7ec2-4006-9d35-69548cf53e4e/.

[17] Lin, C. Deceptive Cracked Software Spreads Lumma Variant on YouTube. Fortinet Blog. 8 January 2024. https://www.fortinet.com/blog/threat-research/lumma-variant-on-youtube.

[18] Palazolo, G. RedLine Stealer Campaign Using Binance Mystery Box Videos to Spread GitHub-Hosted Payload. Netskope. 12 May 2022. https://www.netskope.com/blog/redline-stealer-campaign-using-binance-mystery-box-videos-to-spread-github-hosted-payload.

[19] ANY.RUN. Detonation of Amadey, 5RfuRxo3fpxiWkD42DRCixRe.exe. ANY.RUN. 2024. https://app.any.run/tasks/cdccf208-83b2-4de5-8709-3f383578d6f7/.

[20] Splunk Research Team. Amadey Threat Analysis and Detections. Splunk Blog. 25 July 2023. https://www.splunk.com/en_us/blog/security/amadey-threat-analysis-and-detections.html.

[21] Bourgue, Q.; Le Bourhis, P. Stealc: a copycat of Vidar and Raccoon infostealers gaining in popularity. Sekoia Blog. 20 February 2023. https://blog.sekoia.io/stealc-a-copycat-of-vidar-and-raccoon-infostealers-gaining-in-popularity-part-1/.

[22] Kerner, S. M. Double Extortion Ransomware. TechTarget. 2023. https://www.techtarget.com/searchsecurity/definition/double-extortion-ransomware.

[23] ANY.RUN. Detonation of STOP, TzjwSXczmD2hOVANbz7L7Roc.exe. ANY.RUN. 2024. https://app.any.run/tasks/2ed30ac4-23e6-437f-b40d-4e076b961f4a.

[24] BlackBerry Research Team. DJVU: The Ransomware That Seems Strangely Familiar. BlackBerry Blog. 29 September 2022. https://blogs.blackberry.com/en/2022/09/djvu-the-ransomware-that-seems-strangely-familiar.

[25] ANY.RUN. Detonation of LockBit, LB3.exe. ANY.RUN. 2024. https://app.any.run/tasks/f14ec820-e62c-4c1e-9e6e-4362f731817d.

[26] Ciccarelli, M. Going Deep | A Guide to Reversing Smoke Loader Malware. SentinelOne Blog. 21 November 2019. https://www.sentinelone.com/blog/going-deep-a-guide-to-reversing-smoke-loader-malware/.

APPENDIX

File name	Description	SHA256
release.rar	Password protected archive file from Google Groups Password is ‘1234’	B6B19302DBAAF3D352C4636CC1925AD773286C3BB2269D3DFC834F62391327F0
setup.exe	CrackedCantil executable inside release.rar Protected with Themida and VMProtect	06D285910B40B90B3C2A8454A486B0AC62698228C13A9ACCBEF9BF2DF9E80C6F
CrackedCantil.exe	Unpacked CrackedCantil executable	B62D780351C14753D7005F6B7860DF3B8D6C0133C465C36172D91235086AD90A
vvlbVE_a1T9mi81qLqDvAjYH.exe	Smoke Loader injector executable	66FE490D91A149A12DBF8764DC582A94FAEAC75C812EC8268E294ADAFF6FD5AA
3Pvvg68HWOfBwJ9BdOsWgpEz.exe	RisePro executable	183181709ACEA935FA0E22BCAE4C80D05D090283ADA960A0A386AA930C588ED9
Iq4tpcuftnMe73YjwlKR3YVy.exe	RisePro executable	1401ECA0A99D34975A5C7DF0245FE287C76C40535C7AA17536F2C45058DA94FC
5RfuRxo3fpxiWkD42DRCixRe.exe	Amadey executable	919AE827FF59FCBE3DBAEA9E62855A4D27690818189F696CFB5916A88C823226
hzQj407t3pAeMkmtH8lxdDg1.exe	Stealc executable	65FCF2BAC887D16FE2D281C53EFAA770C73F7E32A2862024CC21F9680EE9EFE9
DTPanelQT.exe	Socks5systemz executable	6DB539F1C4957EDE89A7A7CFCD0D30AFCD897B6A1A66D84DE7E8A8A91BB90EDD
TzjwSXczmD2hOVANbz7L7Roc.exe	STOP executable	2DCB65F18EE5FF985D4448C2BC01ECFF8C1E379D969EF09591789DDE8DD1D534