Challenge: RevengeHotels APT Lab

Platform: CyberDefenders

Category: Endpoint Forensics

Difficulty: Easy

Tools Used: DB Browser for SQLite, Event Log Explorer, Timeline Explorer, EvtxECmd, dnSpy, CyberChef

Summary: This investigation focused on analysing a compromised Windows host following a phishing attack that leveraged a malicious JavaScript downloader disguised as a document, ultimately leading to multi-stage malware execution, persistence, defence evasion, and data exfiltration. Through analysis of browser artifacts, Sysmon logs, and more, it was determined that the malware disabled Windows defender, downloaded and executed additional payloads, established C2 communication, modified security related registry keys, deployed persistence, and finally collected and compressed data for exfiltration.

Scenario: On September 28, 2025, the SOC team detected suspicious network activity from an administrator's workstation, including connections to an unknown external IP address and unauthorized security tool modifications. The user reported opening what appeared to be a legitimate document received via email earlier that day, after which their security software was mysteriously disabled.

Initial triage reveals evidence of file creation in unusual locations and system configuration changes, suggesting a multi-stage attack with potential data exfiltration occurring hours after the initial compromise.

You have been provided with a disk triage of the compromised host. Your mission is to reconstruct the complete attack chain, identify all malicious components, and determine the full scope of the compromise.

Initial Access

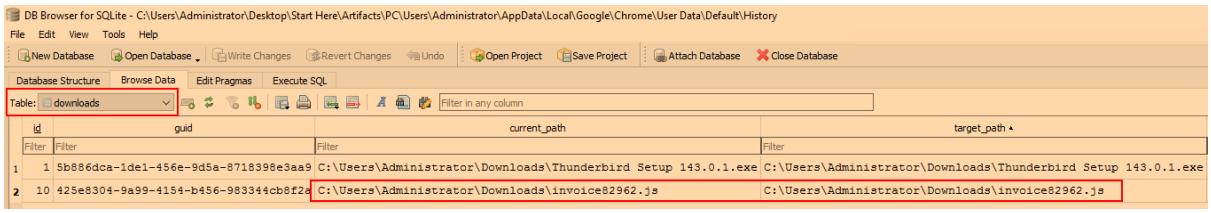
During the initial compromise, the threat actor distributed a phishing email containing a URL pointing to a malicious JavaScript file disguised as a legitimate document. What is the name of the JavaScript file downloaded from the phishing link?

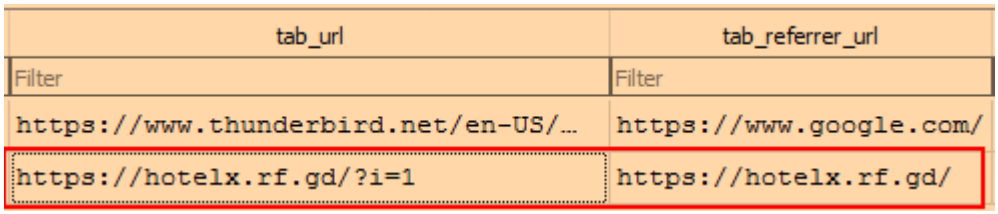
TLDR: Analyse the Google chrome browsing history of the Administrator user using DB Browser for SQLite, focusing on the downloads table.

There are multiple approaches we can follow to find the name of the JavaScript file downloaded from the phishing link. I originally wanted to analyse .PST or .OST files to try and find the original phishing email, however, these are not within the triage image. Let's instead look at the user's history file located at:

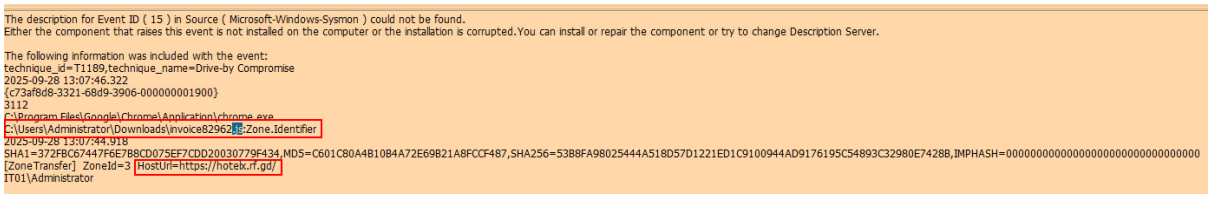
- `\Users\<username>\AppData\Local\Google\Chrome\User Data\Default`

Here we can find a History file that contains the users Google chrome browsing history. Using a tool like DB Browser for SQLite, we can open this database and navigate to the downloads table which contains information about downloaded files. Here we can find a file called “invoice82962.js” downloaded from hxxps[://]hotelx[.]rf[.]gd/?i=1:





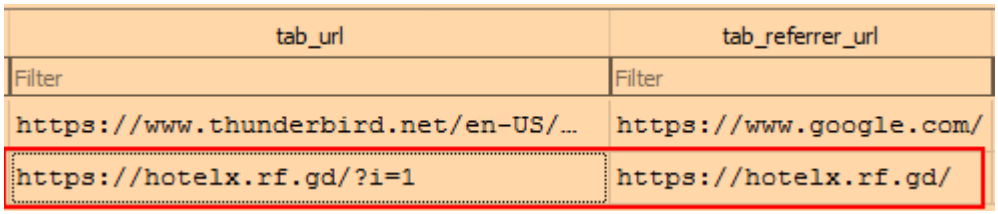
Alternatively, this host had Sysmon configured, therefore, we can use a tool like Event Log Explorer and filter for event ID 15. Event ID 15 logs when a named file stream is created. This captures any Zone.Identifier “mark of the web” stream which allows us to find files downloaded through a browser:



Answer: invoice82962.js

The malicious JavaScript payload was hosted on a compromised website to facilitate the initial infection. What is the complete domain name that hosted the malicious JS file?

We identified the domain name that hosted the malicious JS file in the previous question:



Answer: hotelx.rf.gd

Execution

The JavaScript file created a PowerShell script to advance the attack chain. What is the full directory path where the PowerShell script was created from the JS file?

TLDR: Filter for Sysmon file creation events (Event ID 11) originating from a binary associated with executing JavaScript files on Windows hosts.

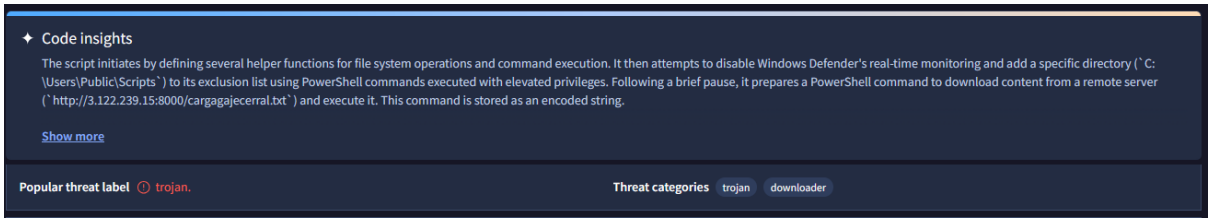
To execute JavaScript files, wscript.exe is often used. Using the following command:

- `.\EvtxECmd.exe -f "Microsoft-Windows-Sysmon%4Operational.evtx" --csv . --csvf sysmon_out.csv`

We can parse the Sysmon logs using EvtxECmd and view the output in Timeline Explorer. If we filter for event ID 11, which records all file creation events, and WScript.exe, we can see that WScript was responsible for creating a PowerShell script file in C:\Users\Public\Scripts:

Payload Data3	Payload Data4
WScript.exe	
Image: C:\Windows\System32\WScript.exe	TargetFilename: C:\Users\Public\Scripts
Image: C:\Windows\System32\WScript.exe	TargetFilename: C:\Users\Public\Scripts\SGDoHBKNUpLKXCAoTHXdBGlnQJLZCGBOVGLH_20250928T061248.ps1
Image: C:\Windows\System32\WScript.exe	TargetFilename: C:\Users\Public\Scripts\SGDoHBKNUpLKXCAoTHXdBGlnQJLZCGBOVGLH_20250928T061424.ps1

This makes sense as if you submit the hash of the JS file to VirusTotal, we can see that it is a downloader which interacts with the C:\Users\Public\Scripts directory:



Answer: C:\Users\Public\Scripts

The PowerShell script invoked another PowerShell command to download two additional files onto the device and then executed one of them. What are the names of the downloaded files?

Filtering for event ID 11 and the image being PowerShell, we can see that PowerShell was responsible for creating three interesting files in the Scripts directory:

Image: C:\Windows\System32\WindowsPow…	TargetFilename: C:\Users\Public\Scripts\venumentrada.txt
Image: C:\Windows\System32\WindowsPow…	TargetFilename: C:\Users\Public\Scripts\runpe.txt
Image: C:\Windows\System32\WindowsPow…	TargetFilename: C:\Users\Public\Scripts\swchost.exe

swchost.exe was later executed at 13:14 on Sep 28th, 2025:

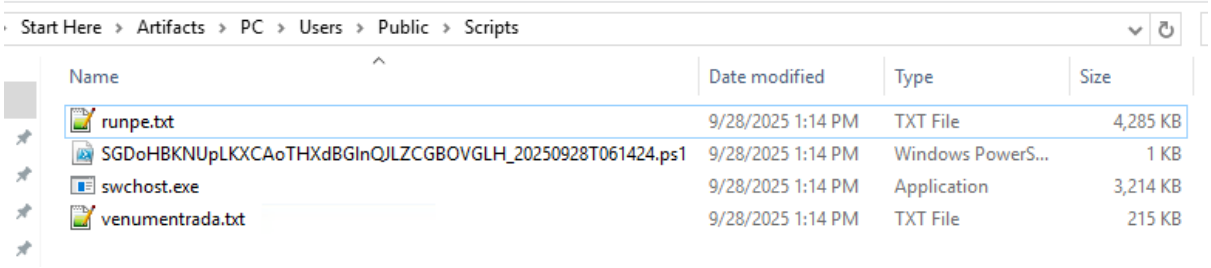
```
Executable Info
swchost
"C:\Users\Public\Scripts\swchost.exe"
"schtasks" /create /tn "swchost" /sc ONLOGON /tr "C:\Users\Administrator\AppData\Roaming\host\swchost.exe" /rl HIGHEST /f
"schtasks" /create /tn "swchost" /sc ONLOGON /tr "C:\Users\Administrator\AppData\Roaming\host\swchost.exe" /rl HIGHEST /f
"C:\Users\Public\Scripts\swchost.exe"
"schtasks" /create /tn "swchost" /sc ONLOGON /tr "C:\Users\Administrator\AppData\Roaming\host\swchost.exe" /rl HIGHEST /f
"schtasks" /create /tn "swchost" /sc ONLOGON /tr "C:\Users\Administrator\AppData\Roaming\host\swchost.exe" /rl HIGHEST /f
```

We can also see a scheduled task being created for this binary.

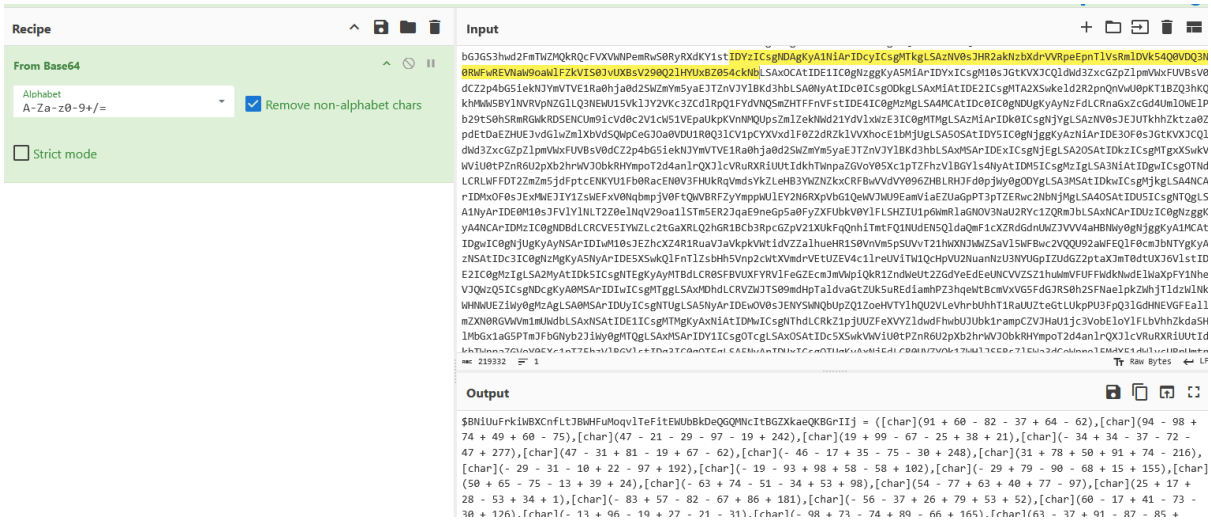
Answer: venumentrada.txt, runpe.txt

The downloaded files included obfuscated content that needed to be converted to reveal their true nature. What is the actual file type of the second downloaded file?

If you navigate to C:\Users\Public\Scripts, we can find the runepe.txt file:



If you view the venumentrada.txt file, we can see that it's heavily obfuscated:



Answer: exe

The first downloaded file converted the second file to its original format, saved it, and then executed it. What is the name of the executed file that was run after conversion?

If you filter for event ID 1, we can see swchost.exe being executed directly after conversion.

Answer: swchost.exe

Defence Evasion

The initial JavaScript file employed specific technique to evade security controls and prevent detection. What is the MITRE ATT&CK technique ID for the method used by the JavaScript file?

TLDR: Investigate registry value set events (Event ID 13) that target security related registry keys. Alternatively, look at suspicious PowerShell commands.

If you explore Sysmon process creation logs (Event ID 1), we can see that one second after invoice82962.js was executed by WScript, Windows Defender real time monitoring was disabled, and an exclusion path was created for the C:\Users\Public\Scripts directory:

```
PowerShell.EXE
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Command Set-MpPreference -DisableRealtimeMonitoring $true

PowerShell.EXE
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -Command Add-MpPreference -ExclusionPath 'C:\Users\Public\Scripts\'
```

Date	Time	Event	Source	Category	User	Computer
9/28/2025	1:12:44 PM	1			\SYSTEM	IT01.cyberrange.cyberde
9/28/2025	1:12:45 PM	1			\SYSTEM	IT01.cyberrange.cyberde
9/28/2025	1:12:45 PM	1	Microsoft-Windows	(1)		IT01.cyberrange.cyberde

JS File Executed

Defender Disabled

Exclusion Path Set

This is a prime example of T1562.001 Impair Defences: Disable or Modify Tools:

Answer: T1562.001

The malicious executable modified multiple security-related registry keys to weaken system defenses. How many registry keys were edited by the malicious executable?

Sysmon Event ID 13 records registry value set events. As discovered previously, the malicious executable is swchost.exe. Using Timeline Explorer to view the EvtxECmd output, we can see that swchost modified 12 unique security-related registry keys, including but not limited to:

Payload Data5
TargetObject: HKU\S-1-5-21-403280985-4081385913-4248903659-500\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\svchostAS
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableRealtimeMonitoring
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableBehaviorMonitoring
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableOnAccessProtection
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection\DisableScanOnRealtimeEnable
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet\DisableBlockAtFirstSeen
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet\SpynetReporting
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet\SubmitSamplesConsent
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\MpEngine\MpEnablePus
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\UX Configuration\Notification_Suppress
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender Security Center\Notifications\DisableNotifications
TargetObject: HKLM\SOFTWARE\Policies\Microsoft\Windows Defender Security Center\Notifications\DisableEnhancedNotifications

Answer: 12

Command and Control

The malicious executable established communication with a C2 server infrastructure. What is the IP address of the C2 server that the malicious executable contacted?

Sysmon Event ID 3 records all network connections. We can filter for network connections originating from swchost.exe, here we can find that it communicated with two unique IPs:

```
DestinationIp: 3.122.239.15
DestinationIp: 136.243.53.56
DestinationIp: 3.122.239.15
DestinationIp: 136.243.53.56
```

Answer: 3.122.239.15

Persistence

As part of its persistence strategy, the malware created a copy of itself in a different location. What is the full path where the malware copied itself?

Filtering for file creation events (Event ID 11), we can see that swchost.exe copied itself to C:\Users\Administrator\AppData\Roaming\host\:

Payload Data3	Payload Data4
swchost	
Image: C:\Users\Public\Scripts\swchost.exe	TargetFilename: C:\Users\Administrator\AppData\Roaming\host\swchost.exe

Answer: C:\Users\Administrator\AppData\Roaming\host\swchost.exe

To maintain persistence after system reboots, the executable added an entry to a specific registry location. What is the full path of the registry key where the executable added its persistence mechanism?

Going back to registry value set events (Event ID 13), swchost.exe was observed creating a RunOnce key called svchostAS:

```
TargetObject: HKU\S-1-5-21-403280985-4081385913-4248903659-500\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\svchostAS
```

This will execute swchost.exe a single time when the user logs on.

Answer: SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\svchostAS

A VBS script was deployed as an additional persistence mechanism to maintain the malware's presence. What is the name of the VBS script executed by the malicious executable for persistence?

Filtering for swchost.exe in process creation events, we can see that swchost.exe was observed executing KOoNLZeCGlnQ.vbs:

Payload Data6	Executable Info
ParentCommandLine: "C:\Users\Administrator\AppData\Roaming\host\swchost.exe"	"wscript.exe" "C:\Users\Administrator\AppData\Local\Temp\KOoNLZeCGlnQ.vbs"
ParentCommandLine: "C:\Users\Administrator\AppData\Roaming\host\swchost.exe"	"wscript.exe" "C:\Users\Administrator\AppData\Local\Temp\KOoNLZeCGlnQ.vbs"

Answer: KOoNLZeCGlnQ.vbs

To ensure the malware process couldn't be terminated easily, it used a specific Windows API function to mark itself as critical. What Windows API function does the malicious executable use to mark its process as critical to the system?

To analyse swchost.exe, we can use a tool called dnSpy. Exploring the .NET code, I can see RtlSetProcessIsCritical being imported from ntdll.dll. The function marks the current process (swchost.exe) as critical to the Windows OS, this is meant for system processes, but malware sometimes abuses this to prevent forced termination as killing it would crash the machine.

```
// Token: 0x06000127 RID: 295
[DllImport("ntdll.dll", EntryPoint = "RtlSetProcessIsCritical", SetLastError = true)]
private static extern int \uFFFD\uE21F뗭ᅳ陈\uE6DB巢捋玑镀础潞\u3098\uFFFD\u0DCF滍ᦀ磨ᒧ6(bool NewValue, out bool OldValue, bool CheckFlag);
```

Answer: RtlSetProcessIsCritical

Collection

The threat actor deployed an additional executable specifically designed for data collection activities. What is the name of the executable dropped for data collection purposes?

Going back to process creation events from swchost.exe, we can see it executed a file called “Flfs6heTV2lb.exe” from the Administrator’s Temp directory:

ParentCommandLine: "C:\Users\Administrator\AppData\Roaming\host\swchost.exe"	"C:\Users\Administrator\AppData\Local\Temp\Flfs6heTV2lb.exe"

Answer: Flfs6heTV2lb.exe

After gathering sensitive data, the malware compressed the collected information into an archive for exfiltration. What is the exact timestamp when the collected data archive was created? (in 24-hour format)

Filter for all file creation (Event ID 11) events from Flfs6heTV2lb.exe, we can see that it created a zip archive called data.zip:

Payload Data3	Payload Data4
Flfs6heTV2lb.exe	
Image: C:\Users\Administrator\AppData\Local\Temp\Flfs6heTV2lb.exe	TargetFilename: C:\Users\Public\Scripts\data.zip

This occurred at 17:16 on Sep 28, 2025.

Answer: 2025-09-28 17:16