Analyzing Snake Keylogger in ANY.RUN: a Full Walkthrough

By Lena aka LambdaMamba

Published: 2023-10-05 · Archived: 2026-04-05 23:18:03 UTC

Lena aka LambdaMamba

I am a Chief Research Officer at a cybersecurity company. My passions include investigations, experimentations, gaming, writing, and drawing. I also like playing around with hardware, operating systems, and FPGAs. I enjoy assembling things as well as disassembling things! In my spare time, I do CTFs, threat hunting, and write about them. I am fascinated by snakes, which includes the Snake Malware!

Check out:

- My website
- My LinkedIn profile

Emails are a common communication method but also a major vector for cyber threats. They can deliver everything from scams and data theft to malware. Unfortunately, one bad email can lead to financial loss, reputational damage, and even escalate into broader system compromise.

To bolster email security, it’s essential to understand the types of attacks you’re up against. This blog post dives into a real-world example featuring a Snake Keylogger attachment.

Let’s dive right into it!

Overview of the Snake Keylogger

The Snake Keylogger is an infostealer malware written in the .NET programming language. It was discovered in November 2020 and is also known as the 404 Keylogger, 404KeyLogger, and Snake.

The Snake Keylogger steals various information from the victim, such as saved credentials, clipboard data, keystrokes, and screenshots of the victim’s screen.

This malware also checks and collects the system information, which includes the system’s hostname, username, IP address, geolocation, date and time, and more. It then exfiltrates the collected information through protocols

such as FTP, SMTP, and Telegram.

More information on the Snake Keylogger and its trends can be found in ANY.RUN's Malware Trends.

Sample Collection and Preparation for Analysis

Let's first look at the sample collection method and environment setup.

In ANY.RUN's Public Submissions, the following filters were applied,

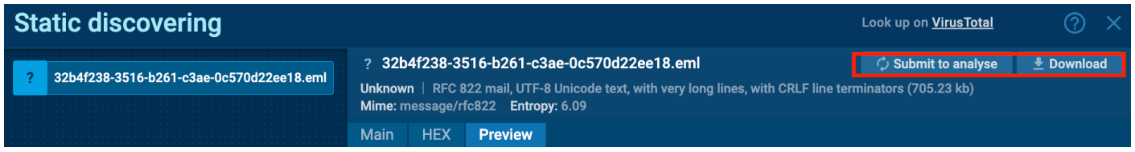
- OBJECT > "Email Files"
- VERDICT > "Malicious"

"32b4f238-3516-b261-c3ae-0c570d22ee18.eml" was selected for analysis. This file had the following attributes:

- SHA1 hash of "1D17DD1688A903CBE423D8DE58F8A7AB7ECE1EA5"
- MIME type of "message/rfc822"
- RFC 822 mail, UTF-8 Unicode text, with very long lines, with CRLF line terminators

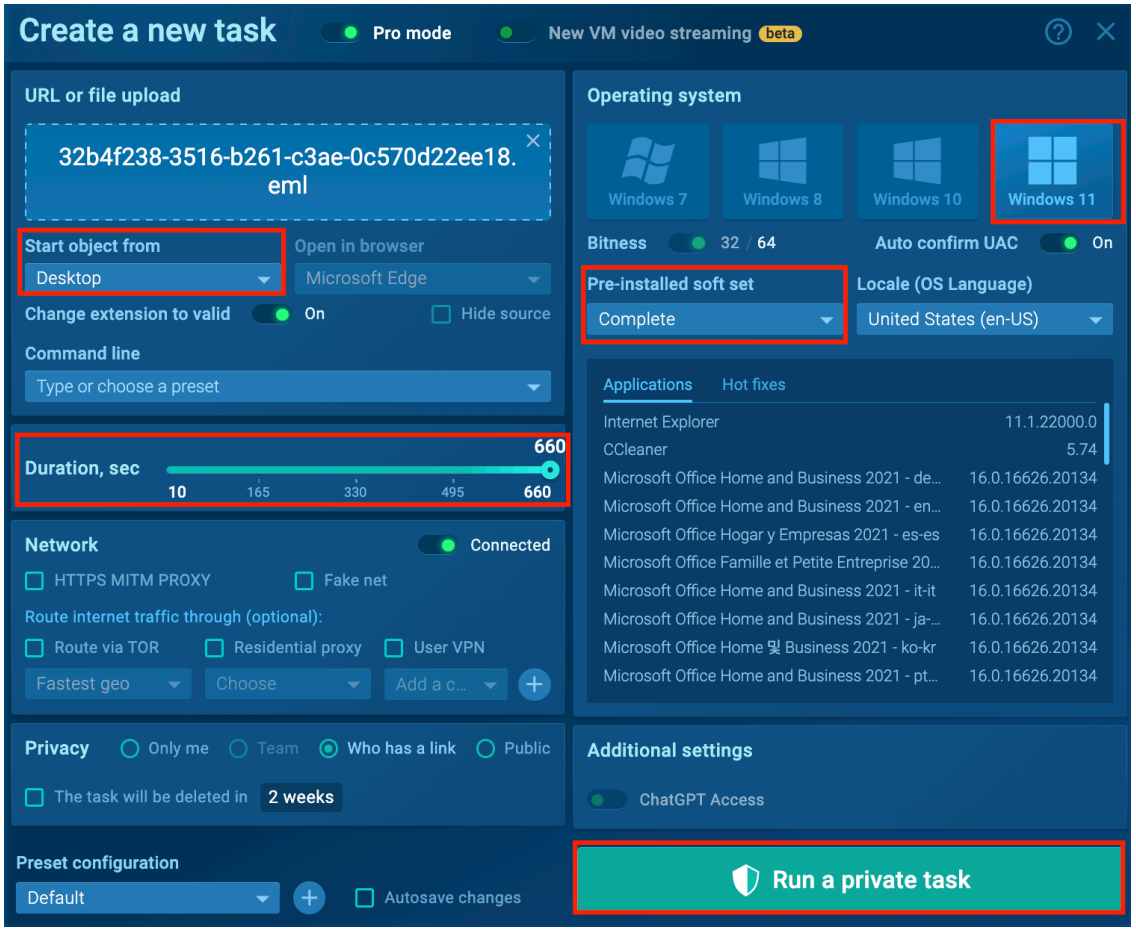
The Filters used to find Malicious Email Files in ANY.RUN's Public submissions

The sample can be downloaded with "Download", and submitted for analysis in ANY.RUN sandbox using "Submit to Analyze" button:



The overview of "32b4f238-3516-b261-c3ae-0c570d22ee18.eml" in *Static Discovering*

A new ANY.RUN task was created for this sample with the following setup:



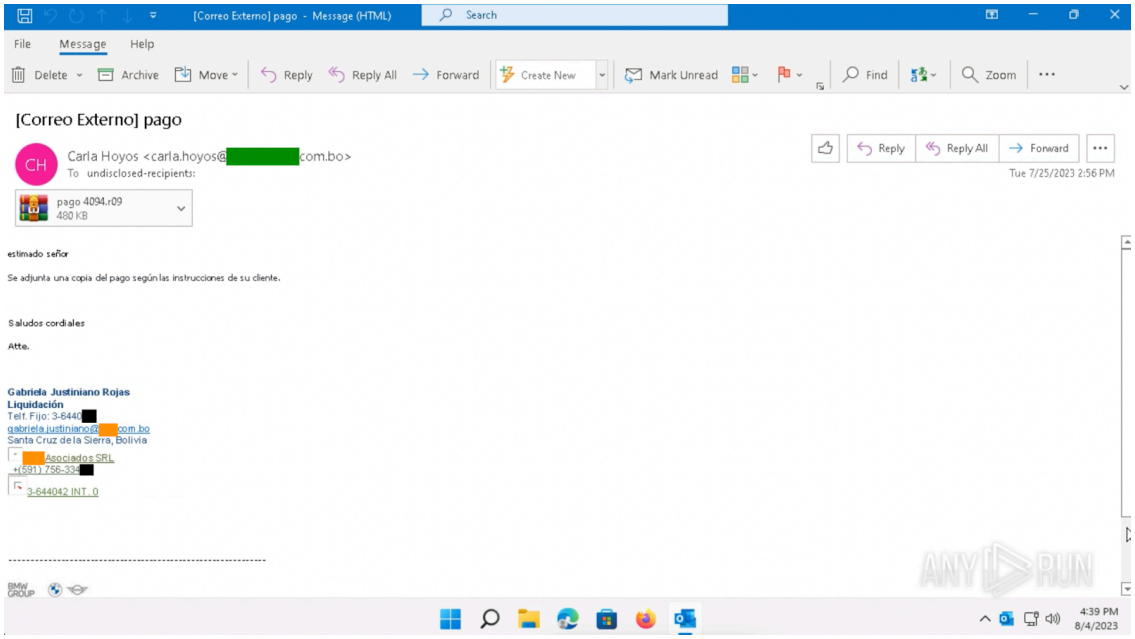
Creation of a New Task, and the setup used for the analysis

The ANY.RUN task for this file can be found here.

Analyzing the Email

Goal of this step: In this section, we'll explore the email body, header, and social engineering tactics.

Opening "32b4f238-3516-b261-c3ae-0c570d22ee18.eml" on Windows 11's Microsoft Outlook showed the email contents:



Opening the email file on Windows 11's Microsoft Outlook

The email body shows the sender attempting to convince the recipient to download and open the email attachment by referencing the "client". The email signature makes references to a Customs Clearing Agency in Bolivia and uses the BMW Group's Logo, suggesting that the sender was attempting to exploit familiarity. Familiarity Exploitation is a social engineering tactic where one pretends to be an entity that is familiar to the target.

The email headers can reveal key information and are useful when analyzing the legitimacy of the email. It is crucial to analyze the SPF and DKIM information when attempting to determine an email's legitimacy.

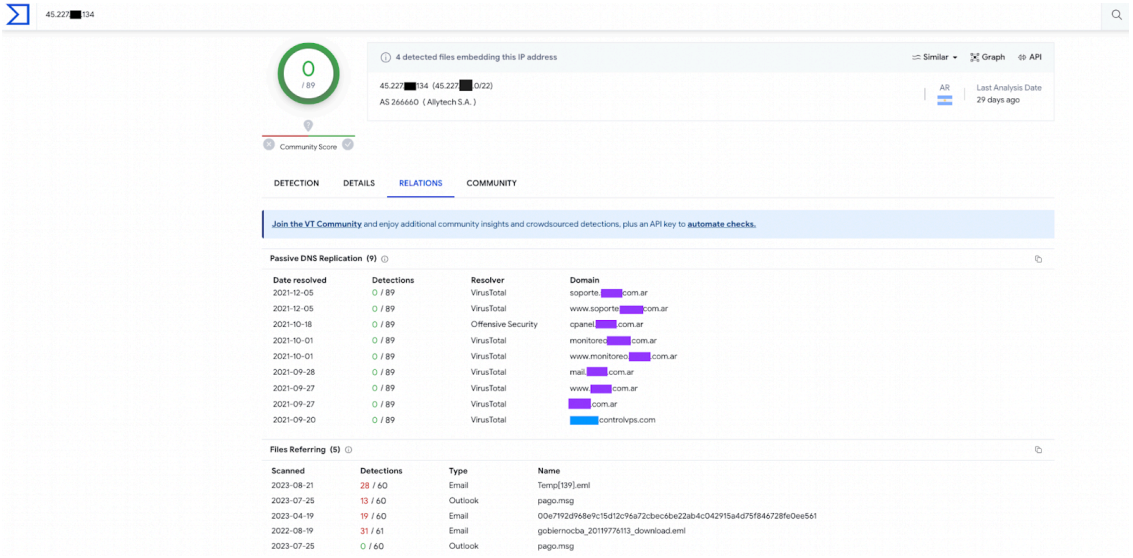
- SPF (Sender Policy Framework) is a DNS record that is used to verify the legitimacy of email senders. The email recipient's server checks the SPF record of the sender's domain to verify they are an approved sender.
- DKIM (DomainKeys Identified Mail) is an email authentication method used to verify the authenticity and integrity of the email. A digital signature is added to the email's header, which is generated by the sender's server with a private key. This is verified by the recipient's server with a public key published in the sender's DNS records.

The email header reveals that the SPF failed, where the sender IP was IP 45[.]227.X.34. The header mentions "[GREEN].com[.]bo does not designate IP 45[.]227.X.34 as permitted sender". Also, there was no DKIM and DMARC, and the message was not signed:

```
Authentication-Results: spf=fail (sender IP is 45.227.[redacted].134)
 smtp.mailfrom=[redacted].com.bo; dkim=none (message not signed)
 header.d=none;dmarc=none action=none
 header.from=[redacted].com.bo;compauth=fail reason=001
Received-SPF: Fail (protection.outlook.com: domain of [redacted].com.bo does
 not designate 45.227.[redacted].134 as permitted sender)
 receiver=protection.outlook.com; client-ip=45.227.[redacted].134;
 helo=[redacted].controlvps.com;
```

A section of the sample email's header shows the SPF, DKIM, and DMARC information

The IP address 45[.]227.X.34 is associated with these domains (hidden with purple and blue markers for confidentiality reasons). According to VirusTotal, it appears to be a security company in Argentina:



Looking up the IP address 45[.]227.X.34 on VirusTotal

The email header shows the authenticated sender, which was “cobranzas@[PURPLE].com.ar”.

```
Received: from [redacted].controlvps.com (45.227.[redacted].134) by
 DM3NAM02FT035.mail.protection.outlook.com (10.13.[redacted].78) with Microsoft SMTP
 Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
 15.20.6631.29 via Frontend Transport; Tue, 25 Jul 2023 14:58:08 +0000
Received: from webmail.[redacted].com.ar (localhost [127.0.0.1])
        (Authenticated sender: cobranzas@[redacted].com.ar)
        by [redacted].controlvps.com (Postfix) with ESMTPA id D387963E7F;
        Tue, 25 Jul 2023 11:55:35 -0300 (-03)
DKIM-Filter: OpenDKIM Filter v2.11.0 [redacted].controlvps.com D387963E7F
```

A section of the sample email’s header shows the authenticated sender

The email header also revealed the User-Agent, which was “Roundcube Webmail/1.4.2”. *Roundcube Webmail* is a free and open-source webmail software.

```
MIME-Version: 1.0
Content-Type: multipart/mixed;
 boundary="=_8e7887a8909f0688fef98d950324214b"
Date: Tue, 25 Jul 2023 15:55:35 +0100
From: Carla Hoyos <carla.hoyos@[redacted].com.bo>
To: undisclosed-recipients:;
Subject: [Correo Externo] pago
User-Agent: Roundcube Webmail/1.4.2
Message-ID: <285e785b838658691be29f41afcad147@[redacted]com.bo>
X-Sender: carla.hoyos@[redacted].com.bo
Return-Path: carla.hoyos@[redacted].com.bo
```

A section of the sample email’s header shows Date, Time, From, To, Subject, User-Agent, etc.

What did we learn from the header?

It indicates that this email was most likely not legitimate. The contents of the email and the sender’s email address suggest that it was attempting to impersonate a company in Bolivia that provides brokering and insurance

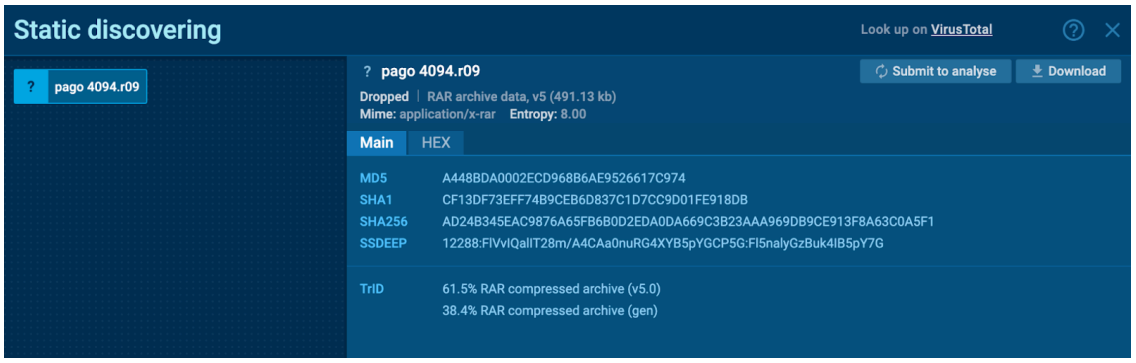
services. Additionally, it utilized social engineering tactics to convince the recipient to download and open the attachment.

Analyzing the Behaviour of the Attachment

Goal of this step: In this section, we'll explore the behavioral analysis of the email's attachment on Windows 11 and examine the involved files.

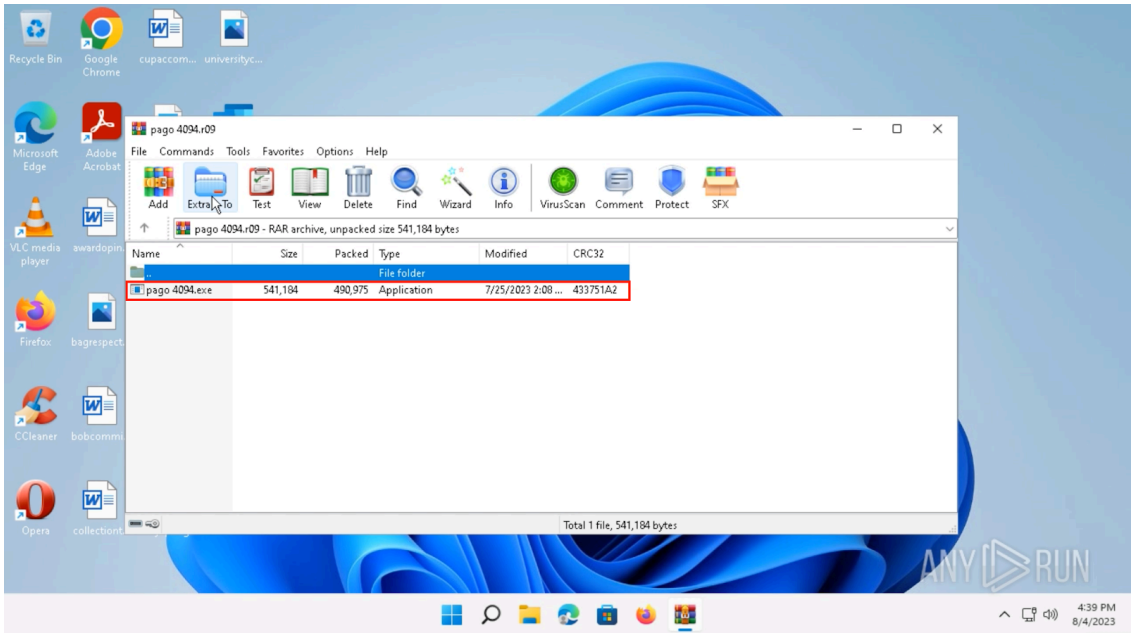
A file called "pago 4094.r09" is attached to this email, with the following attributes:

- SHA1 hash of "CF13DF73EFF74B9CEB6D837C1D7CC9D01FE918DB"
- MIME type of "application/x-rar"
- RAR archive data, v5



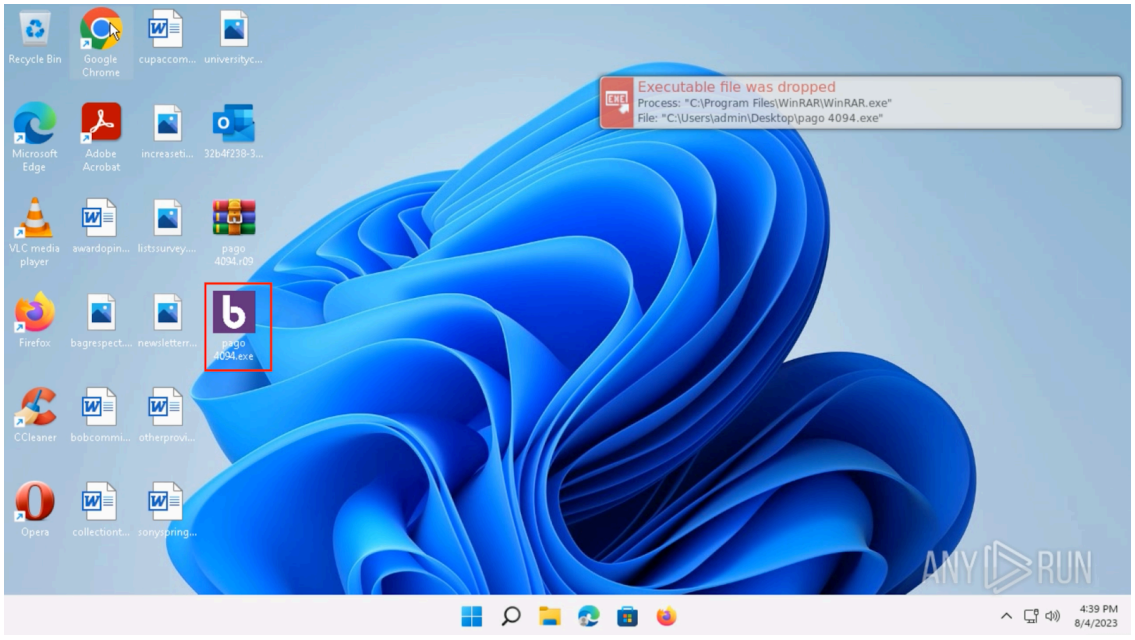
The information for pago 4094.r09 in *Static discovering*

Downloading and opening "pago 4094.r09" in WinRAR shows the existence of an Application called "pago 4094.exe":



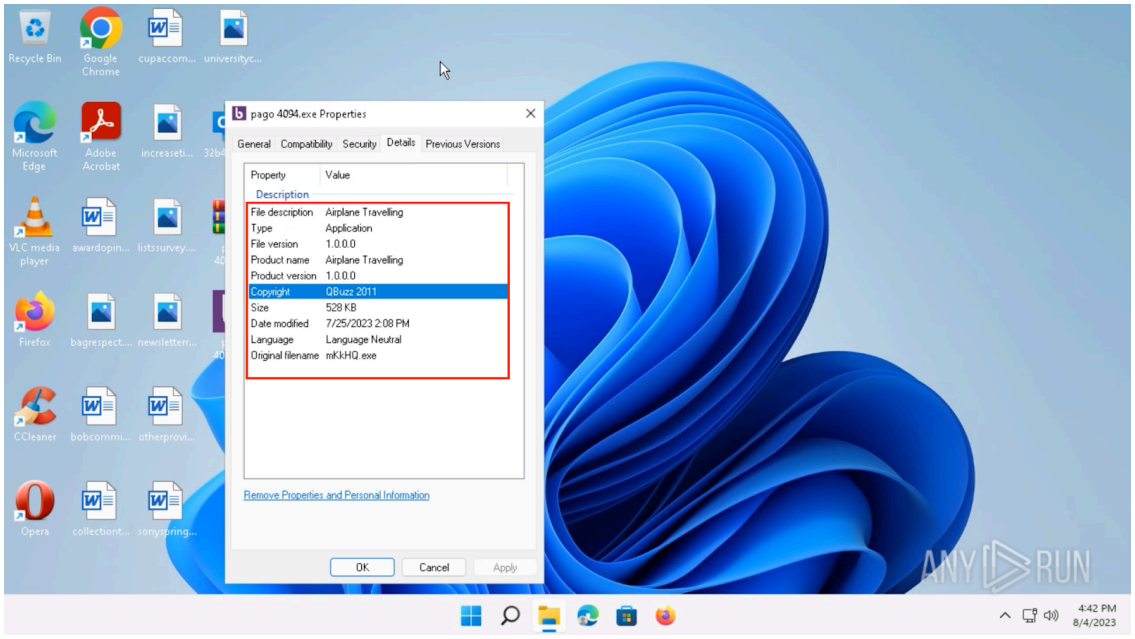
Opening "pago 4094.r09" in WinRAR

Extracting “pago 4094.exe” onto the Desktop reveals that it uses the *Yahoo! Buzz* Icon. [*Yahoo! Buzz*] is a community-based news article website.



The *Yahoo! Buzz* icon

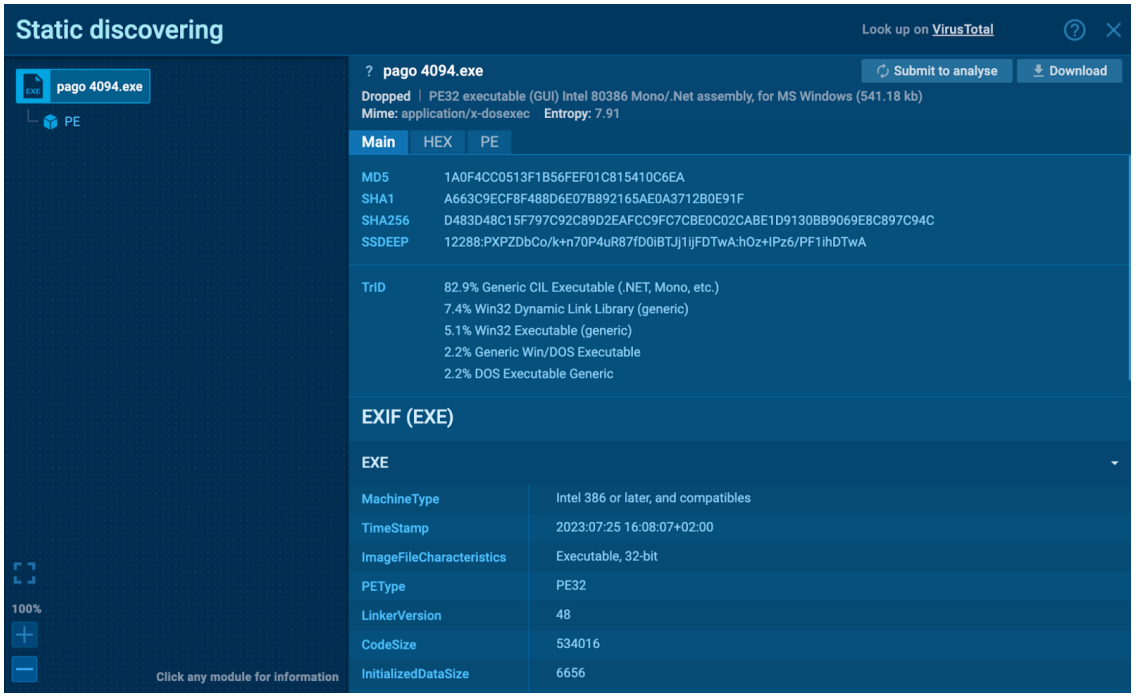
The properties tell us that the original filename was “mKkHQ.exe”, and had the copyright “QBuzz 2011”:



The Properties for “pago 4094.exe”

This executable “pago 4094.exe” has the following attributes:

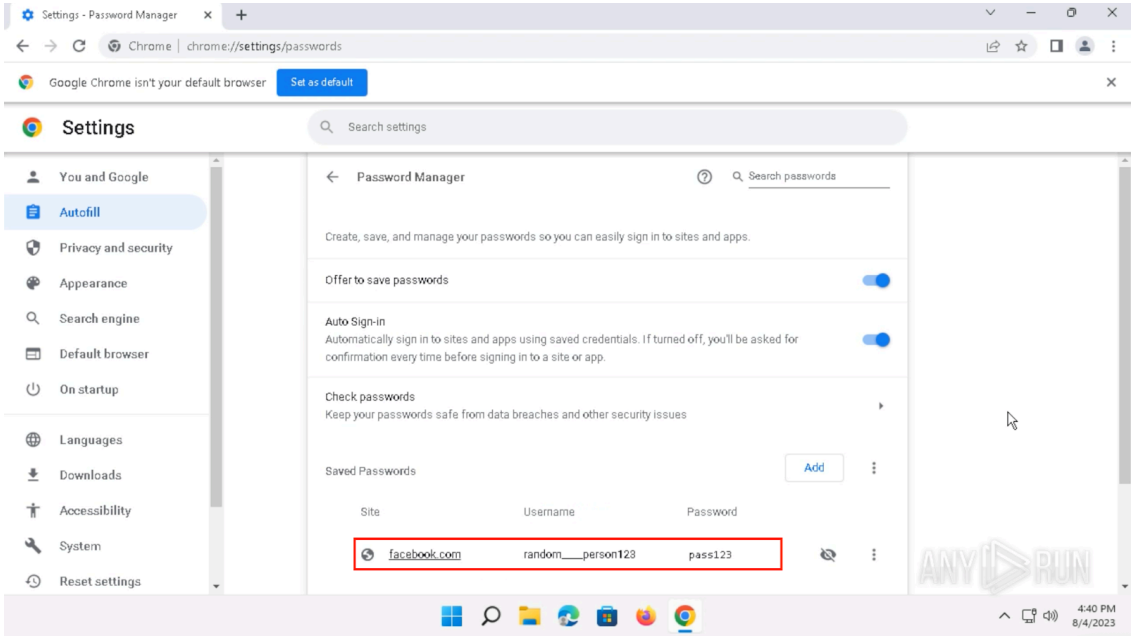
- SHA1 hash of “A663C9ECF8F488D6E07B892165AE0A3712B0E91F”
- MIME type of “application/x-dosexec”
- PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows



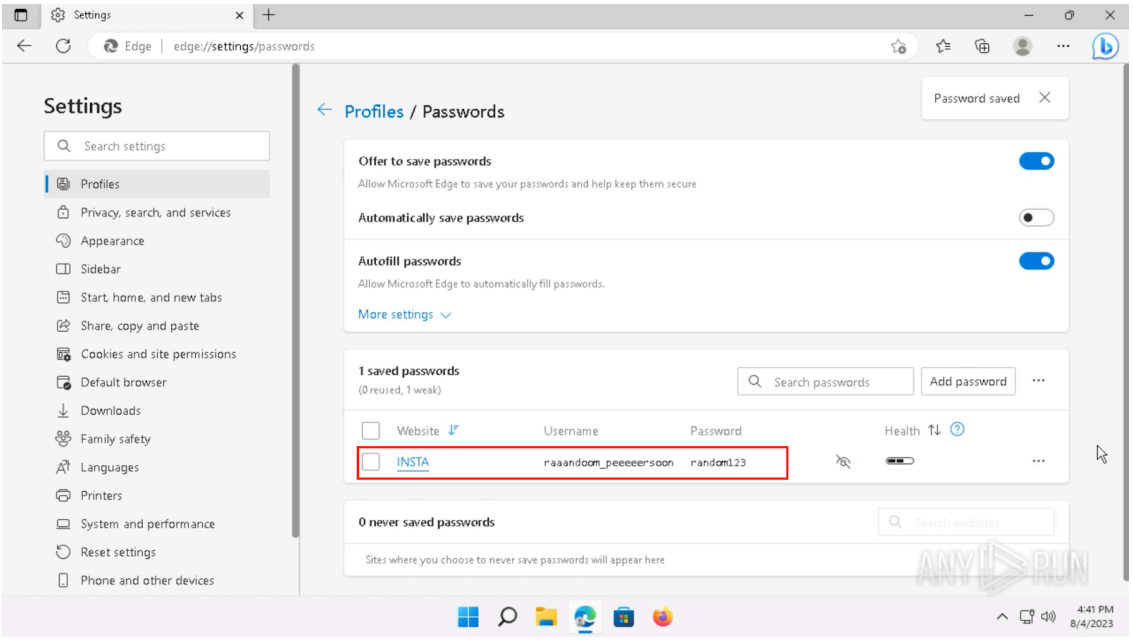
Static Discovering shows the details of the executable “pago 4094.exe”.

Saving credentials in browsers

Before executing “pago 4094.exe”, various fake credentials were purposefully saved onto Browsers like Chrome and Microsoft Edge. This was done to observe the malware’s credential-stealing behavior.



Saving fake Facebook credentials on Chrome, under “chrome://settings/passwords”

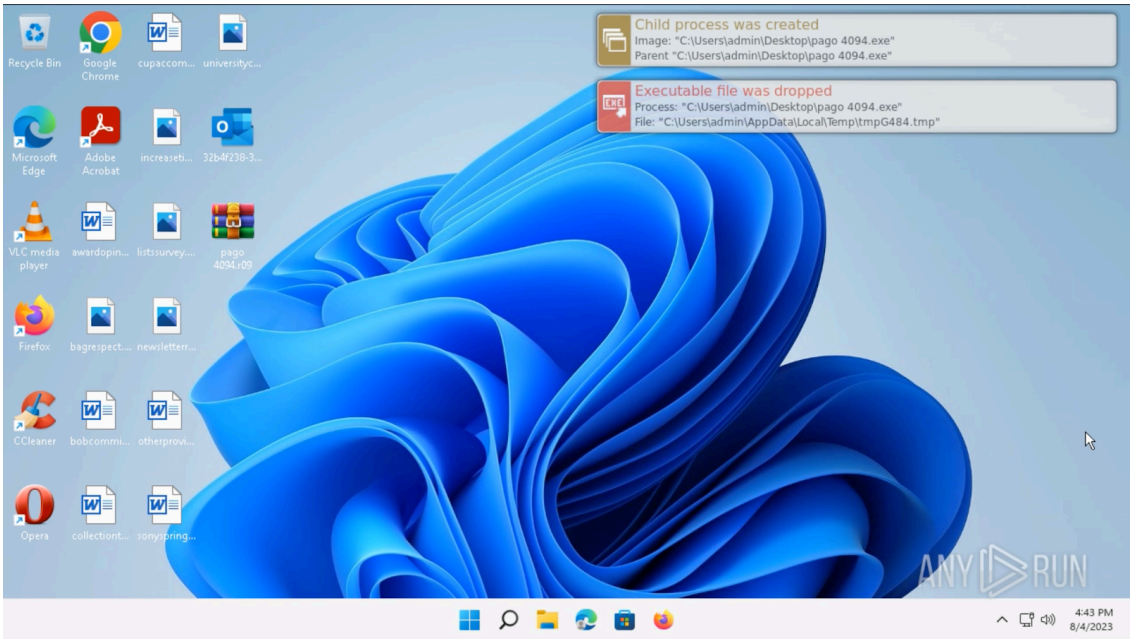


Saving fake Instagram credentials on Microsoft Edge, under “edge://settings/passwords”

Once the fake credentials were saved onto the Browsers, “pago 4094.exe” was executed by double-clicking “pago 4094.exe” on the Desktop.

Getting into the execution flow

Around 30 seconds after executing “pago 4094.exe”, the executable file disappears from the Desktop. A child process “C:\Users\admin\Desktop\pago 4094.exe” is created, and an executable file “C:\Users\admin\AppData\Local\Temp\tmpG484.tmp” is dropped. The dropping of the .tmp file is done to secure persistence on the victim machine.



The executable disappears from the Desktop, and “tmpG484.tmp” is dropped in “C:\Users\admin\AppData\Local\Temp\”

Now, the Snake Keylogger is running silently in the background. From the Windows User's perspective, nothing alarming happens.

Analyzing the Processes

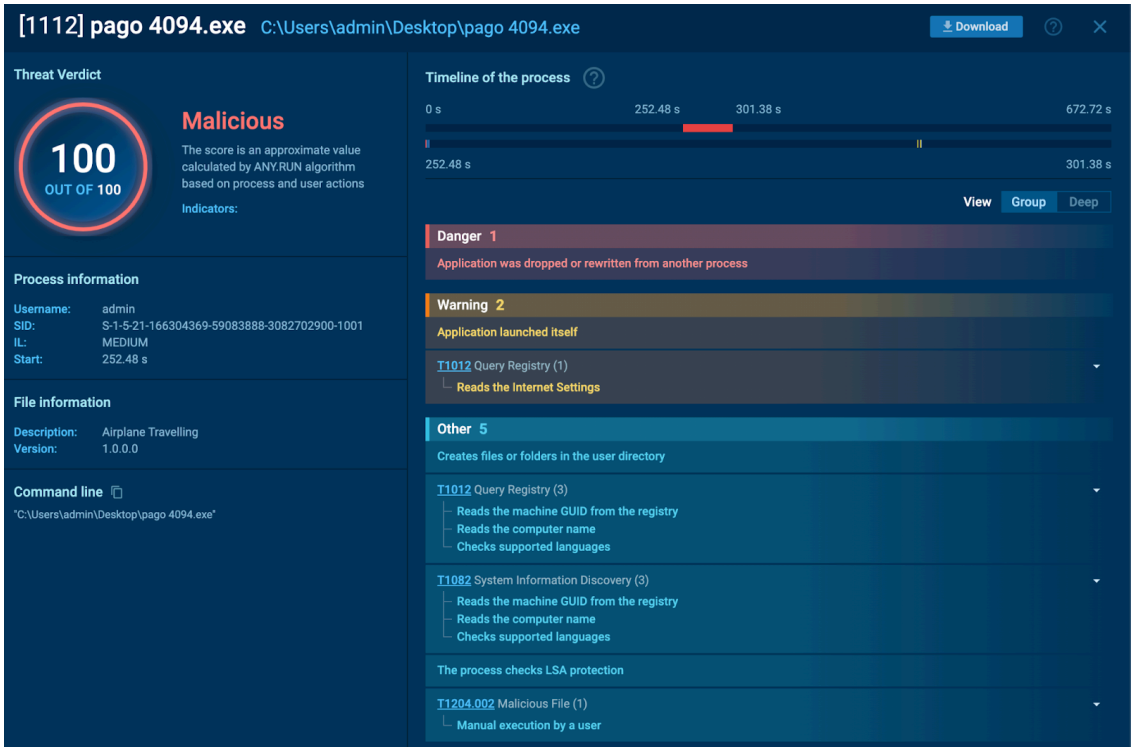
Goal of this section: We'll explore the analysis of processes associated with the Snake Keylogger.

Process 1112 and its child process 3868, are key processes involved in the malicious activities:

The "pago 4094.exe" processes

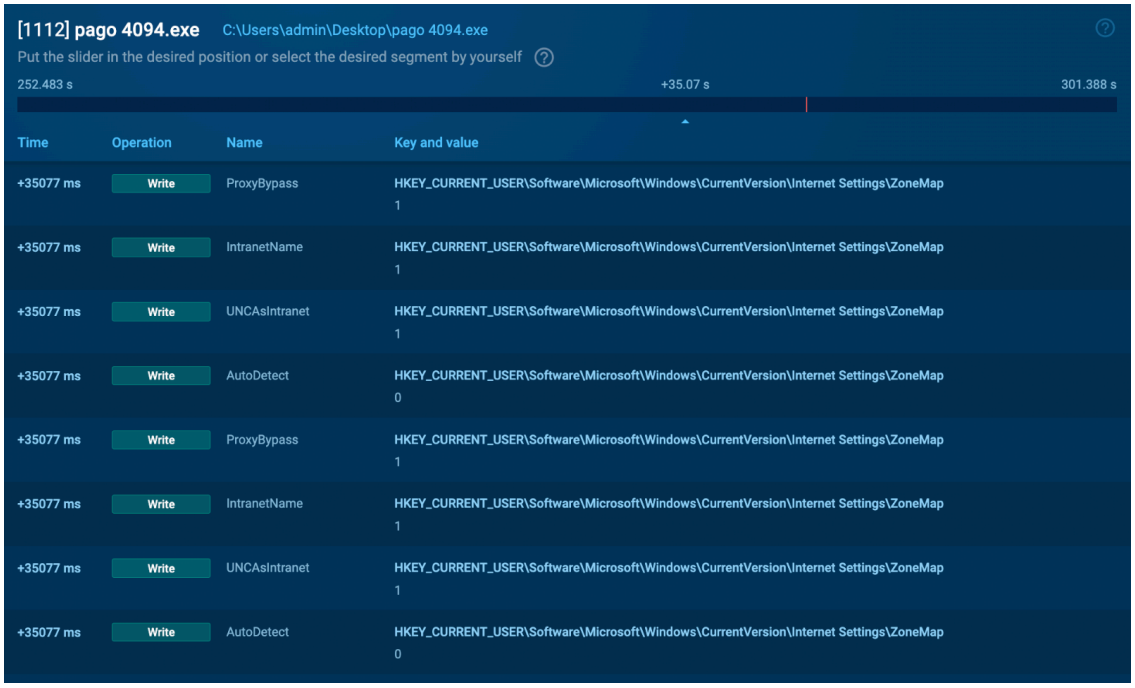
Detailed look at the process 1112

Process 1112 was detected as 100/100 Malicious under the *Threat Verdict*. It can be observed querying registries, performing system information discoveries, checking LSA protection, dropping another application, etc. This process ran for a total of 48.9 seconds.



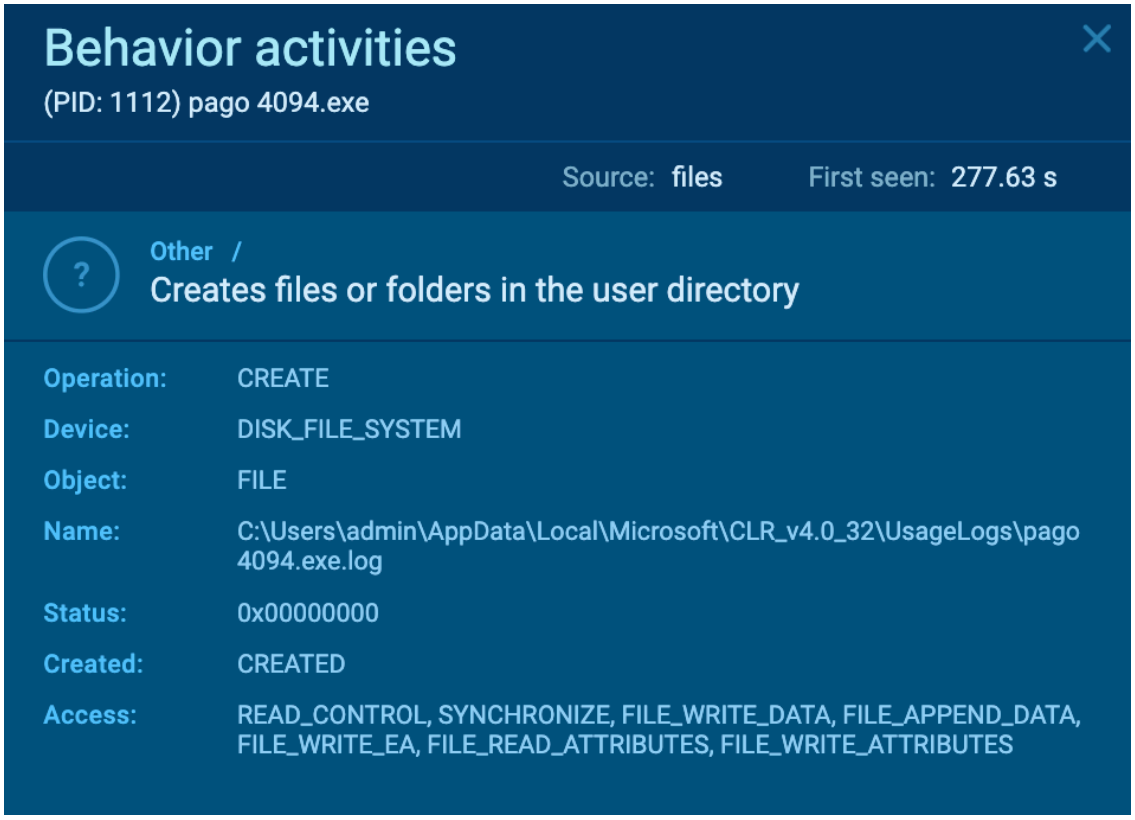
Overview of Process 1112, "pago 4094.exe"

Registry changes were seen for Process 1112, and the following Write Operations were conducted:



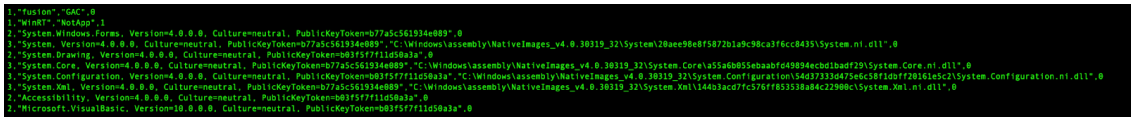
The Registry changes for Process 1112

Process 1112 also created a new file with the MIME type of “text/plain”, called “pago 4094.exe.log” under “C:\Users\admin\AppData\Local\Microsoft\CLR_v4.0_32\UsageLogs\”:



The creation of “pago 4094.exe.log”

The contents of “pago 4094.exe.log” contained references to *System.Windows.Forms*, *System.Drawing*, etc. which are associated with .NET API. It also contained *PublicKeyToken* values:



The contents of “pago 4094.exe.log”

Detailed look at the process 3868

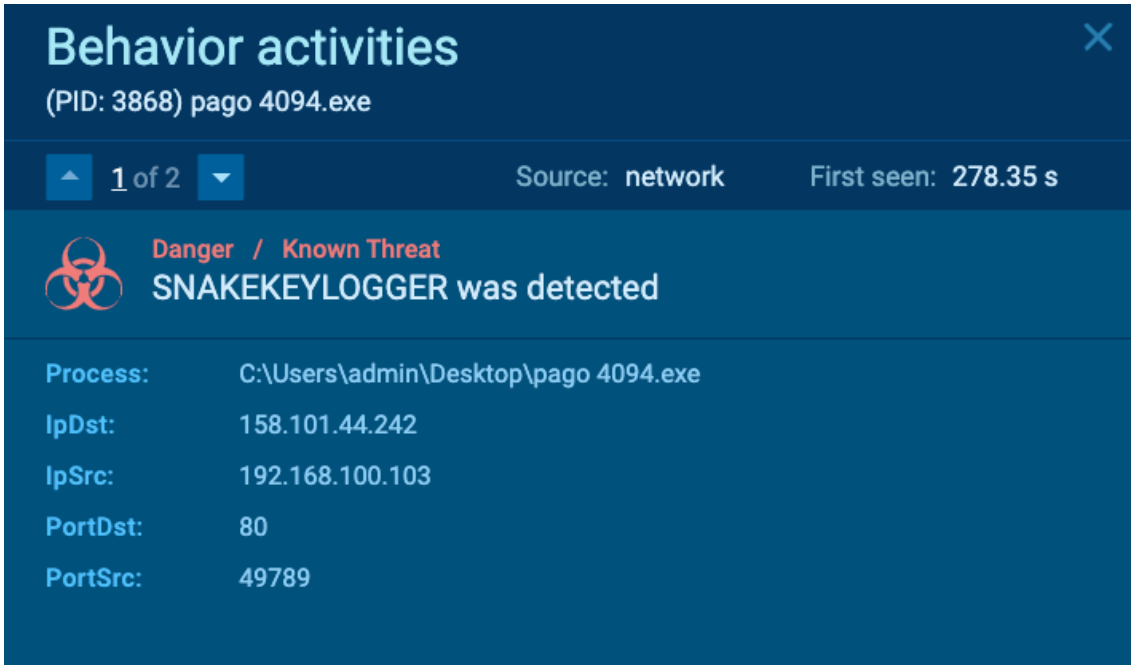
Process 3868 plays a significant role in this malware. This process started at 287.76 seconds and ran all the way until the end. It steals credentials from browsers and files and sends these stolen credentials over SMTP:

Overview of Process 3868, “pago 4094.exe”

The indicators for this process included “Known Threat”, “Connects to the network”, “Executable file was dropped”, “Actions similar to stealing personal data”, “Behavior similar to spam”, “The process has the malware config”, and “The module has a process dump.”

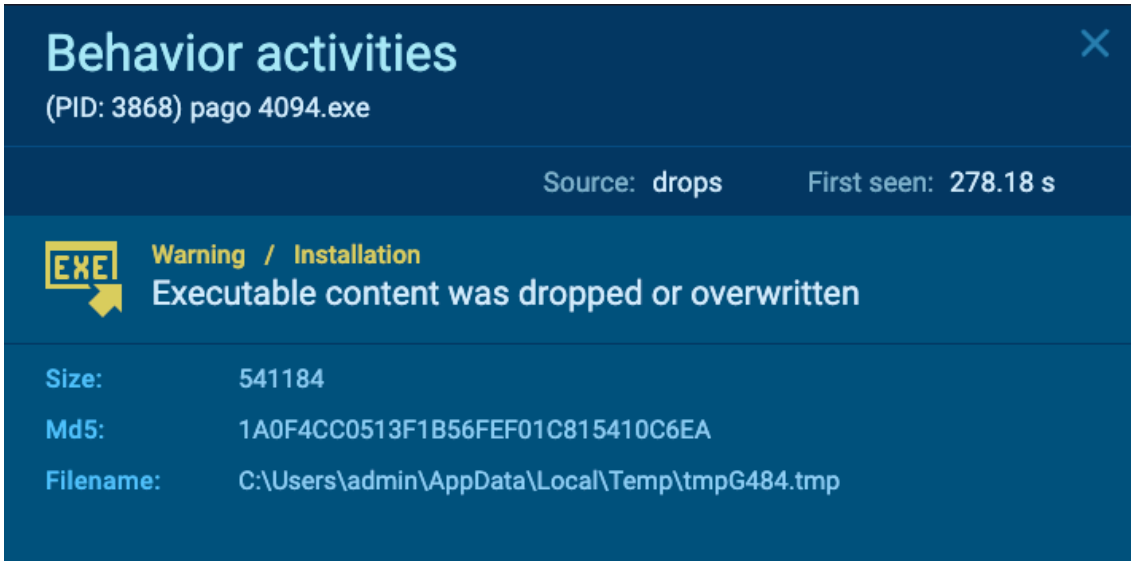
The indicators in Process 3868

It was detected as Snake Keylogger, where the destination IP was 158.101.44[.]242, with a destination port of 80. This IP is associated with *checkip.dyndns[.]com*, and we will explore it in detail in the next section, *Analyzing the Network Information.*

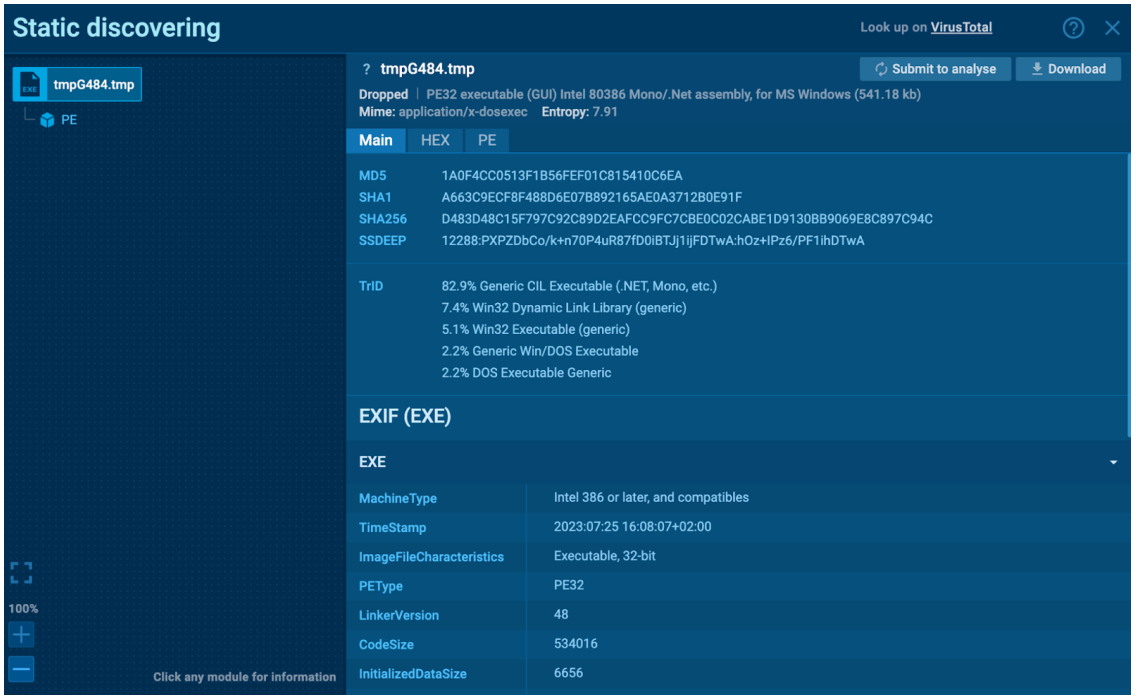


The detection of SNAKEKEYLOGGER

Process 3868 drops “C:\Users\admin\AppData\Local\Temp\tmpG484.tmp”. This has an MD5 hash of 1A0F4CC0513F1B56FEF01C815410C6EA, which is the same as the MD5 hash for the original executable file “pago 4094.exe”. This is done to achieve persistence on the victim machine.



A .tmp file is dropped

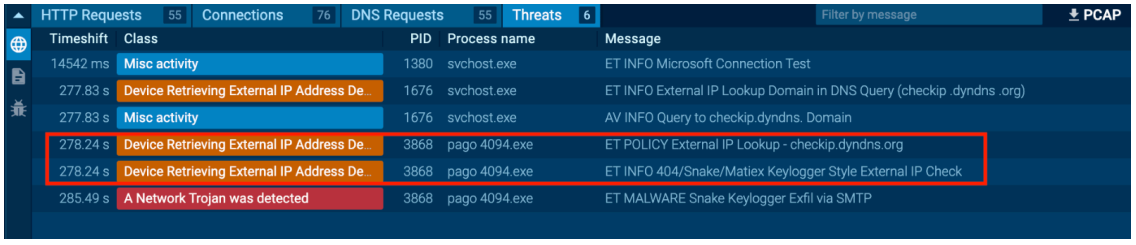


Details of the dropped “C:\Users\admin\AppData\Local\Temp\tmpG484.tmp”

Analyzing the Network Activities

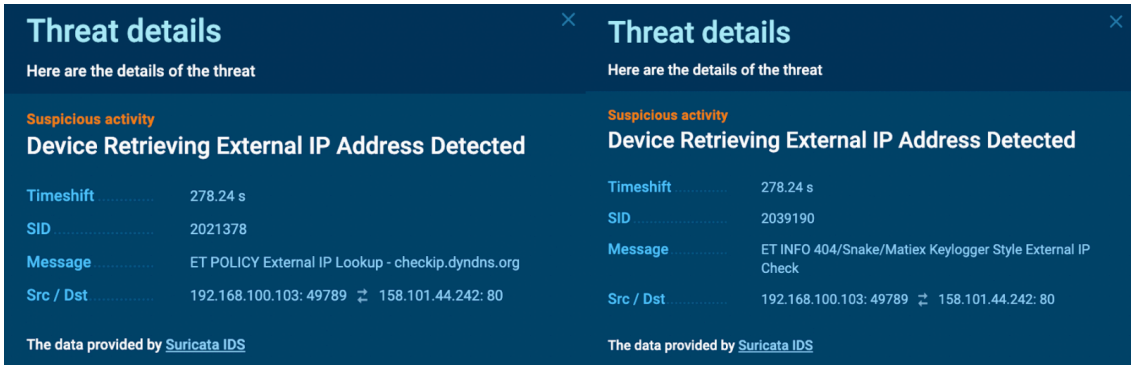
Section goal: In this section, we’ll explore the network activities associated with the Snake Keylogger and examine the packet capture (PCAP) file in detail.

Process 3868, “pago 4094.exe”, attempted to retrieve external IP addresses with *checkip.dyndns[.]org* as shown in the *Threats* Tab:

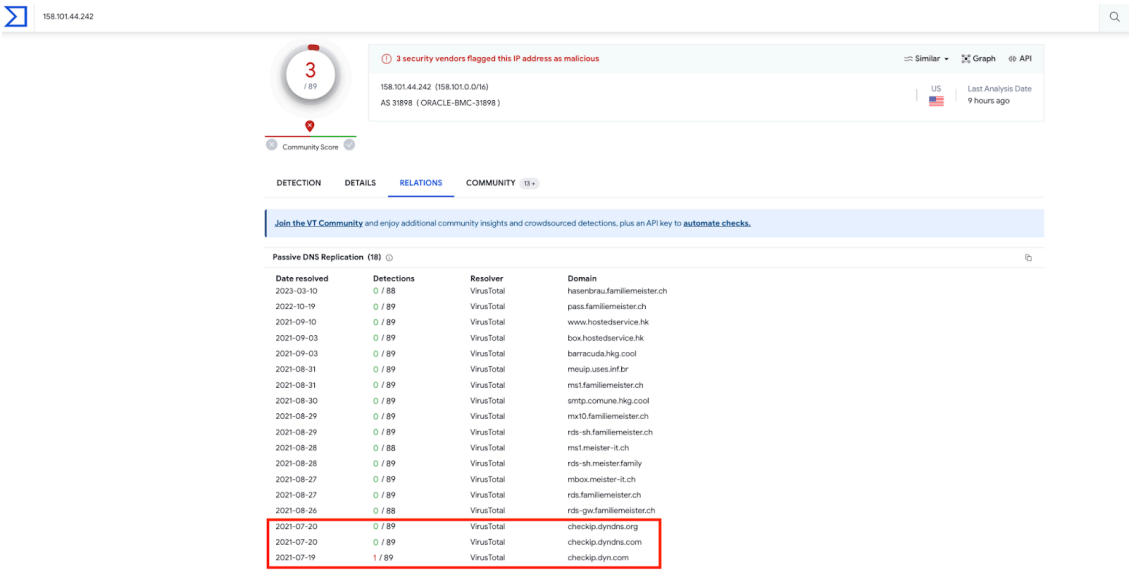


The *Threats* Tab shows the retrieval of the external IP address

It was seen connecting to 158.101.44[.]242 on port 80. This IP was associated with *checkip.dyn…* according to VirusTotal:



The Threat details show the source and destination IP and port.



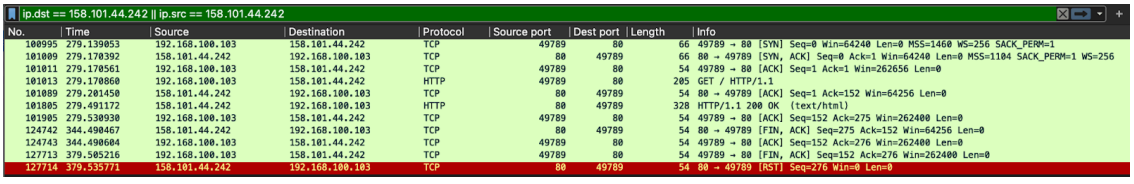
The IP 158.101.44[.]242 was associated with checkip.dyn according to VirusTotal

The host *checkip.dyndns[.]org* is associated with IP checking. According to Dyn, “CheckIP will return the remote socket’s IP address. If a client sends a Client-IP or a X-Forwarded-For HTTP header, CheckIP will return that value instead.”

The packet capture (PCAP) file was downloaded for further analysis. The following filter was applied on the PCAP in Wireshark.

ip.dst == 158.101.44.242 || ip.src == 158.101.44.242

This is done to check for packets where the destination or source IP was 158.101.44[.]242.



Packets where the Destination or Source IP is 158.101.44[.]242

Following the TCP stream revealed that it checked the current IP with *checkip[.]dyndns.org*, which was 45.130.136[.]51:

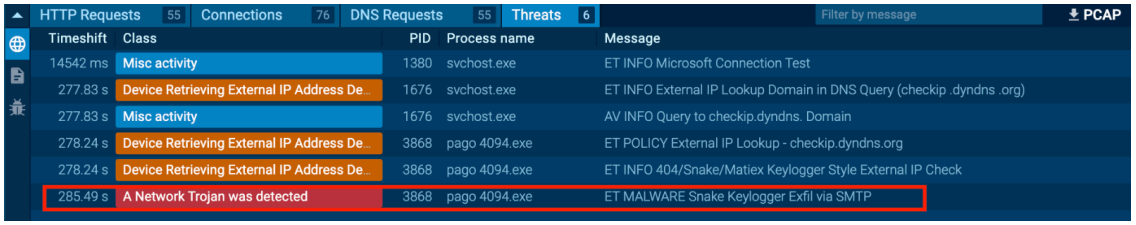
```
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; .NET CLR1.0.3705;)
Host: checkip.dyndns.org
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Fri, 04 Aug 2023 16:43:14 GMT
Content-Type: text/html
Content-Length: 105
Connection: keep-alive
Cache-Control: no-cache
Pragma: no-cache

<html><head><title>Current IP Check</title></head><body>Current IP Address: 45.130.136.51</body></html>
```

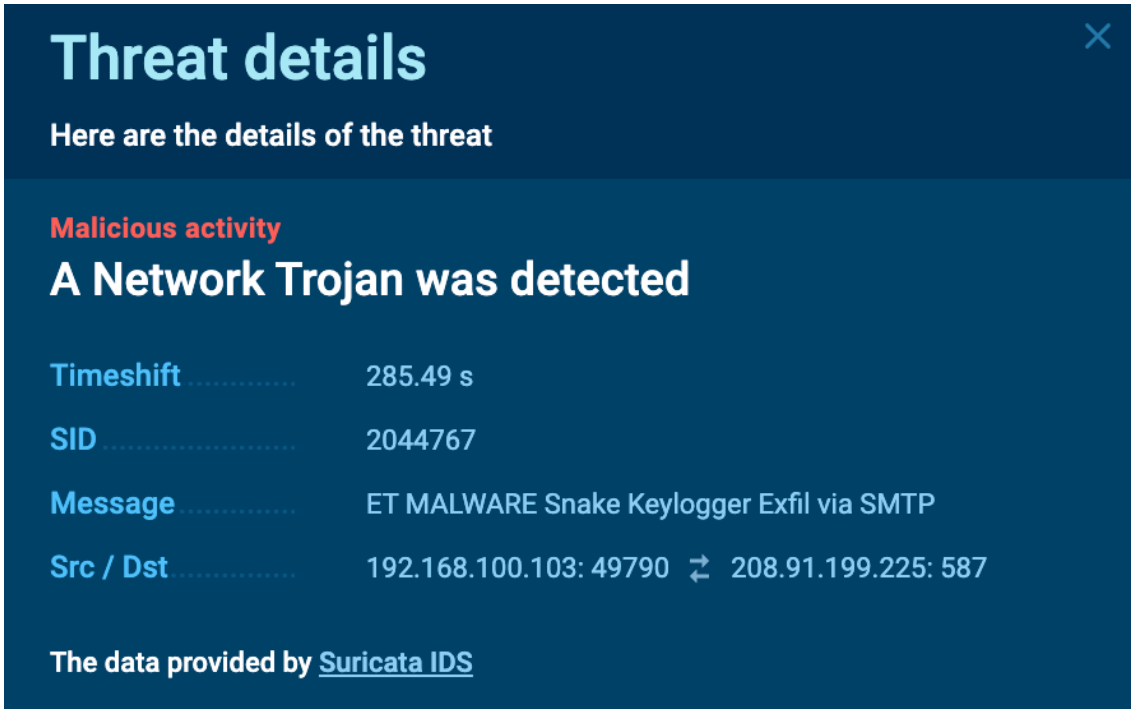
Following the TCP steam shows the current IP address

A Network trojan was detected for process 3868, “pago 4094.exe” under the *Threats* tab:



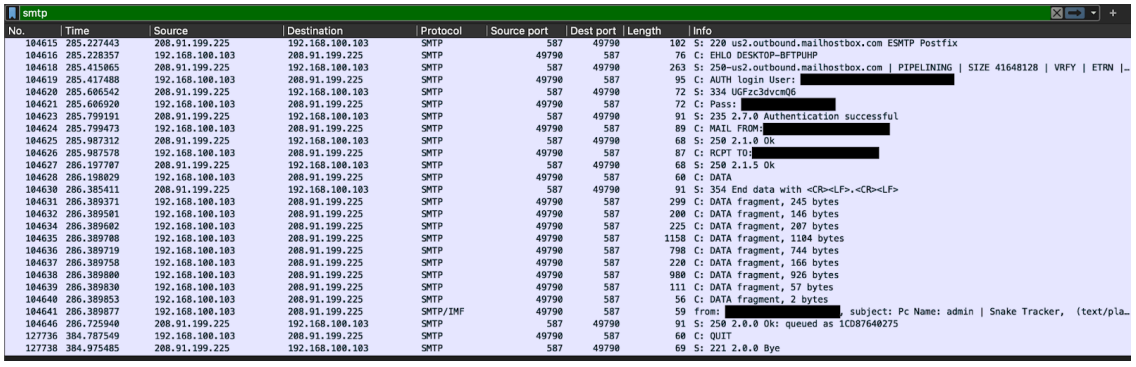
The Detected Network Trojan

A *Snake Keylogger Exil via SMTP* was observed, where the destination IP was 208.91.199[.]255 and the destination port was 587. SMTP on [port 587 is a secure and authenticated method] for sending emails from email clients to email servers. It typically uses STARTTLS or TLS/SSL for encryption.



The *Threat Details* of the Network Trojan

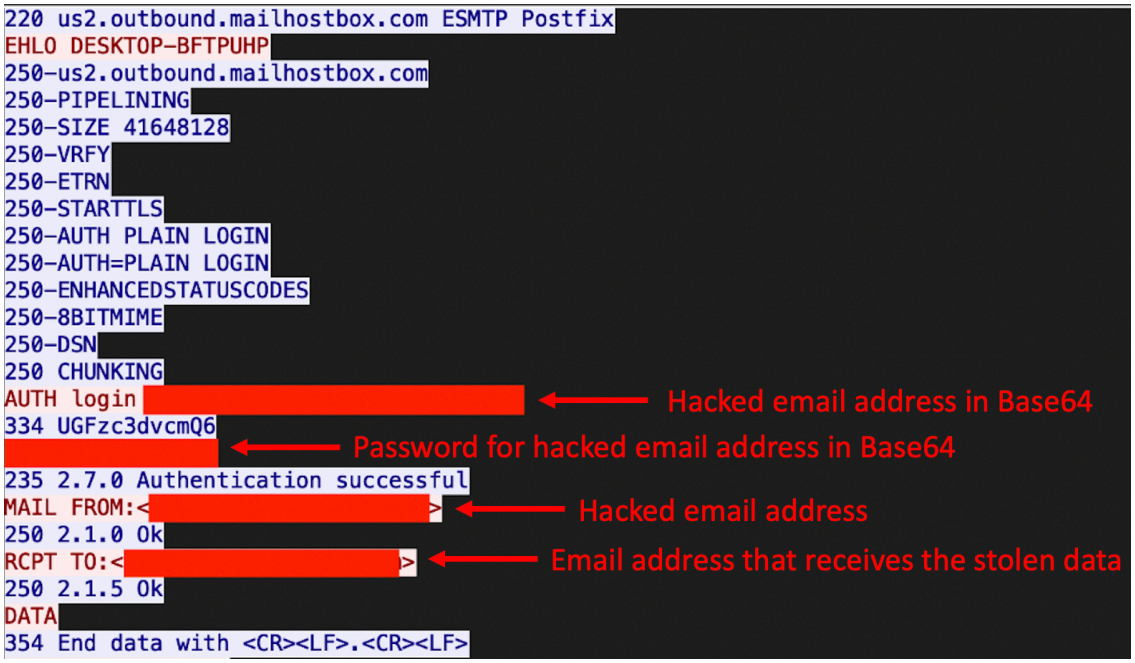
Applying the *smtp* filter on the PCAP in Wireshark showed the data exfiltration taking place over SMTP:



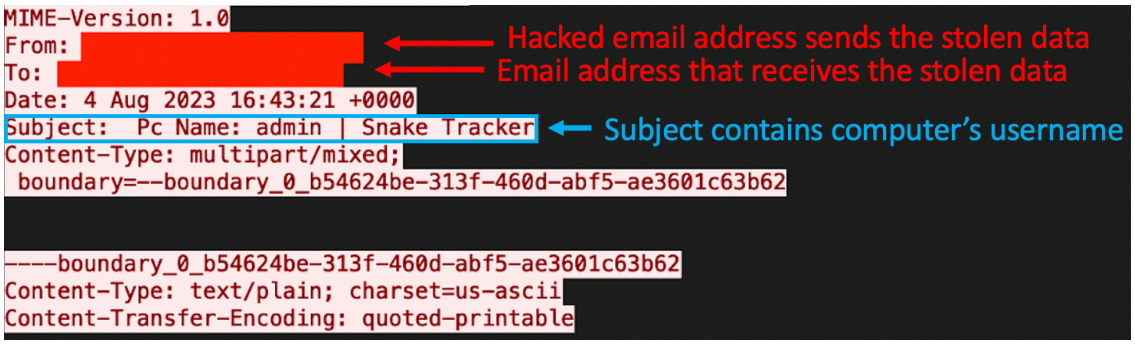
Data exfiltration over SMTP

Following the TCP stream revealed the SMTP Authentication taking place. The email address used to send the stolen information was likely hacked by malicious actors. According to OSINT, the hacked email address belonged to a physical security company in South America.

The same is confirmed in the PCAP:



Following the TCP stream shows the authentication taking place



A section of the email header

The email has an attachment called “Passwords.txt”, which contains the stolen information. The contents of “Passwords.txt” are in Base64 inside the PCAP as shown:

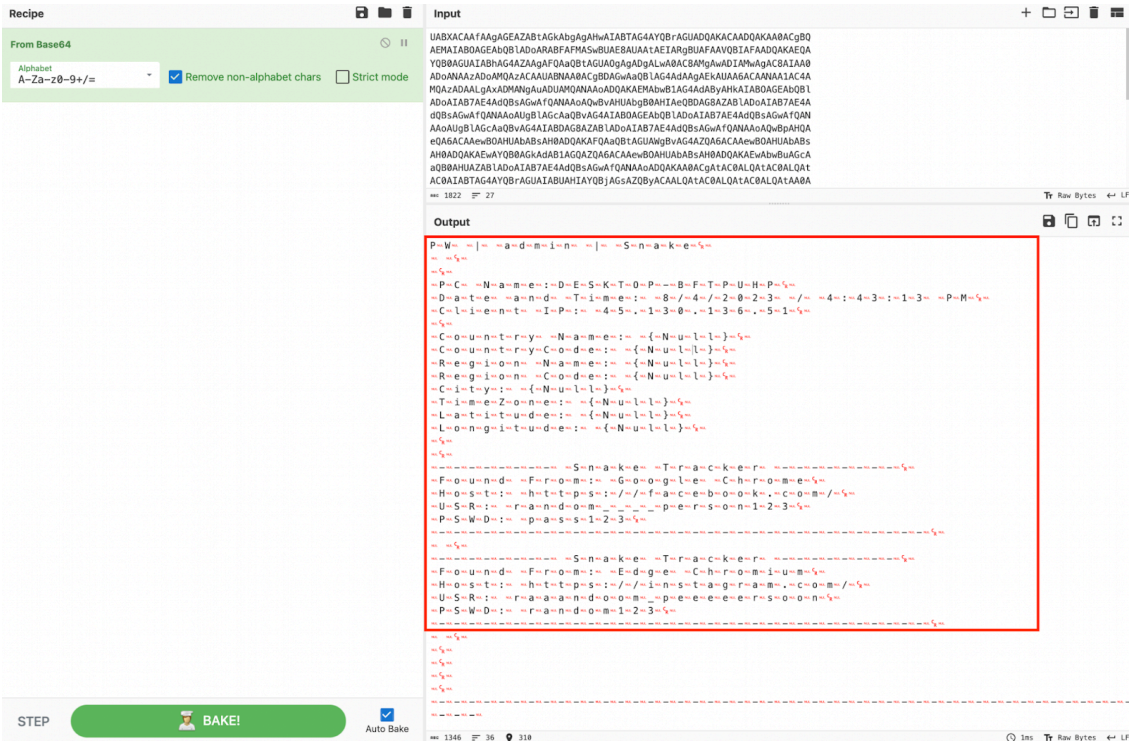
The contents of “Passwords.txt” in Base64

The email has another attachment called “User.txt”, which also contains the stolen information. The contents of “User.txt” are also in Base64 inside the PCAP:

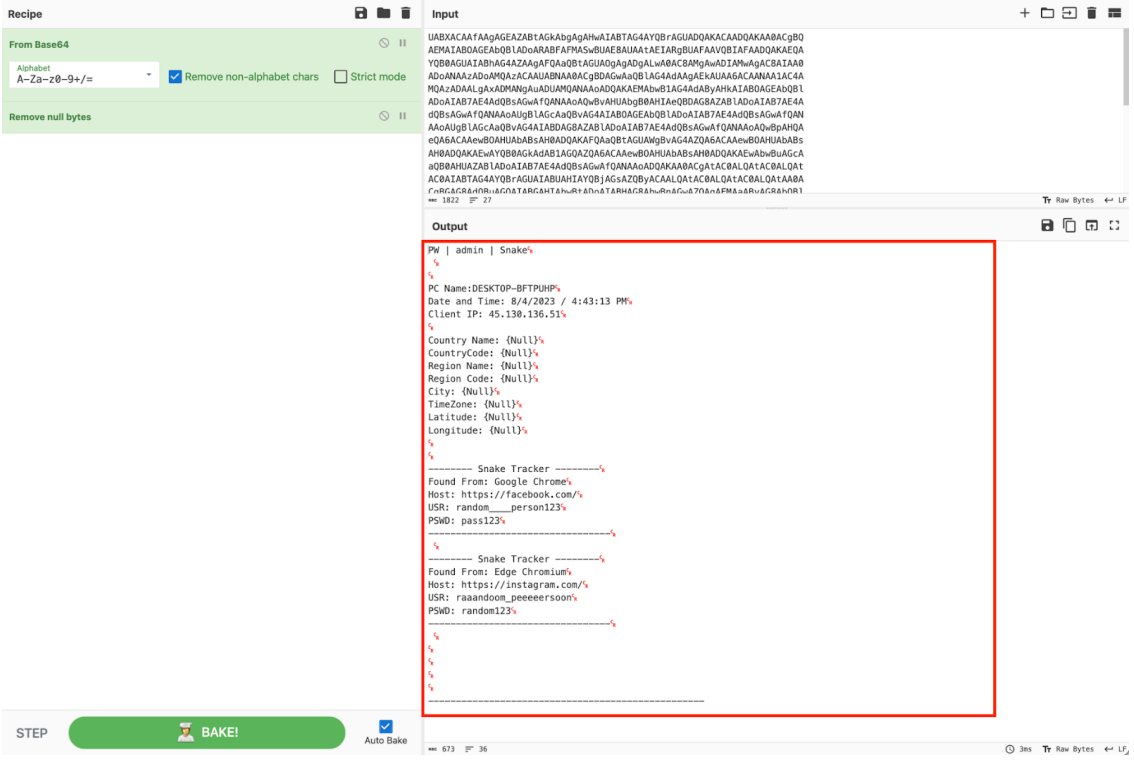
The contents of “User.txt” in Base64

Decoding the contents of “Passwords.txt

Decoding the contents of “Passwords.txt” from Base64 on CyberChef reveals that it contained the computer name (“DESKTOP-BFTPUHP”), the date and time (8/4/2023 4:43:13 PM), IP address (45.130.136[.]51). It also contained the fake credentials that were saved onto Google Chrome and Microsoft Edge:

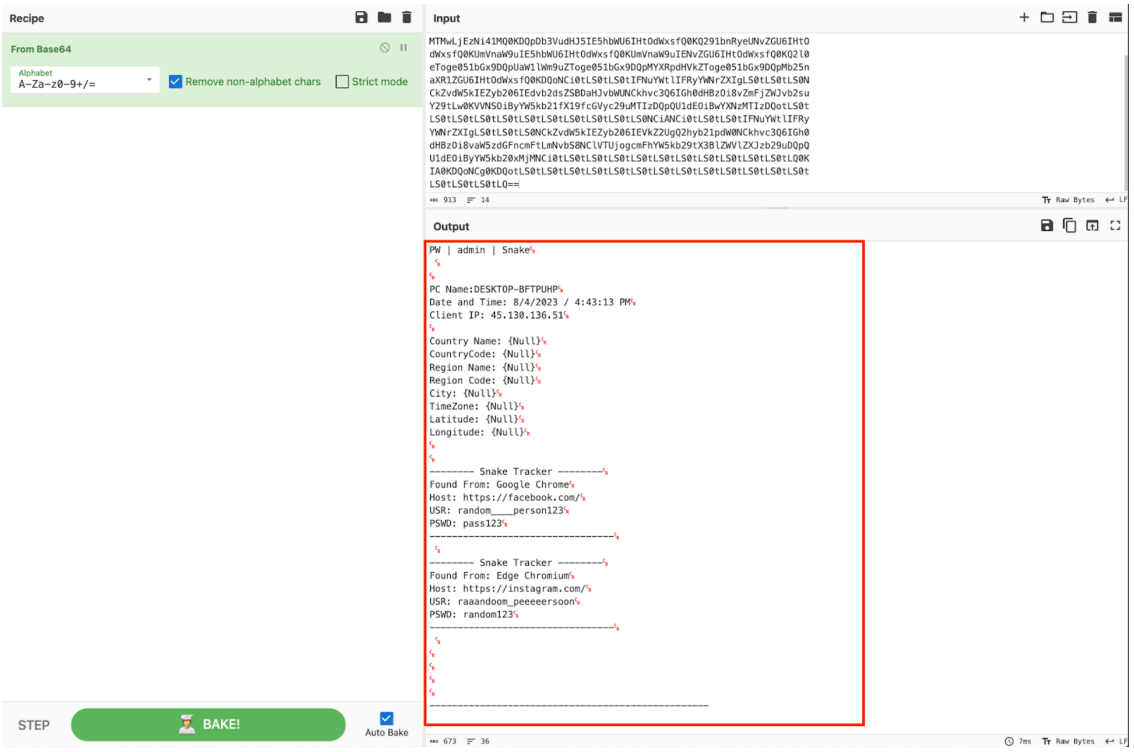


Decoding “Passwords.txt” from Base64 on CyberChef



Removing the null bytes for improved readability

Decoding the contents of “User.txt” from Base64 on CyberChef resulted in something similar to “Passwords.txt”, though it did not contain null bytes, and was in a more human-readable format:

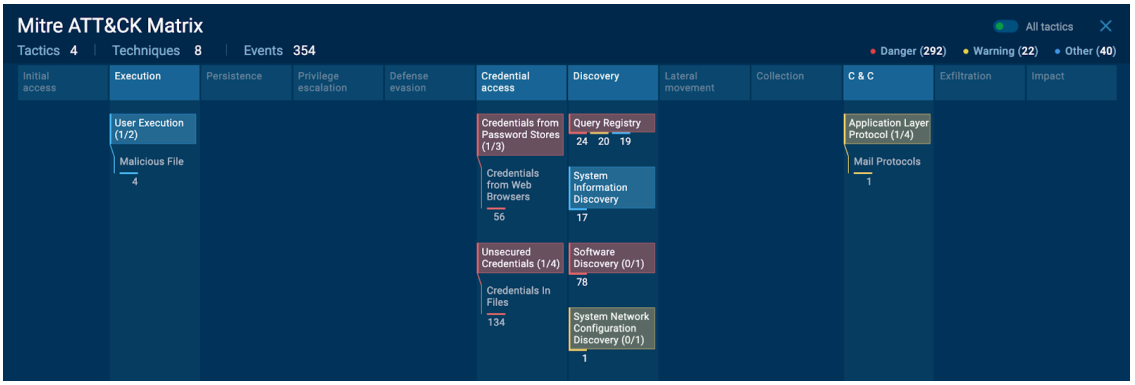


Decoding “User.txt” from Base64 on CyberChef

MITRE ATT&CK

Section goal: In this section, we’ll explore the MITRE ATT&CK for the Snake Keylogger and examine the involved Tactics and Techniques.

The MITRE ATT&CK Matrix for this Snake Keylogger includes five Tactics, namely *Initial Access*, *Execution*, *Credential Access*, *Discovery*, and *Command and Control (C & C)*.



MITRE ATT&CK Matrix

MITRE ATT&CK: Initial Access

Firstly, the phishing email “32b4f238-3516-b261-c3ae-0c570d22ee18.eml” entices the recipient to download and open the attachment via social engineering (as seen in *Analyzing the Email).* The email has a RAR archive

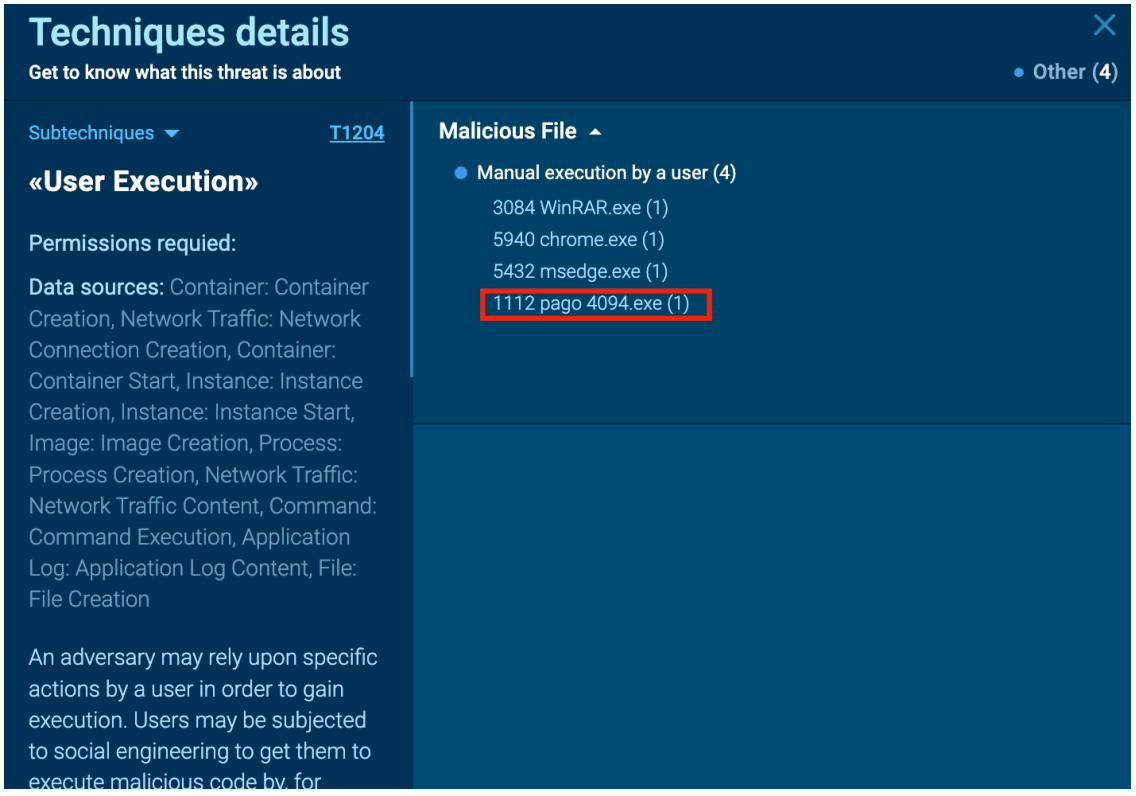
attachment “pago 4094.r09”, which contains an executable file “pago 4094.exe”.

The technique here is T1566 (Phishing), and the subtechnique is T1566.001 (Phishing: Spearphishing Attachment).

MITRE ATT&CK: Execution

The “pago 4094.exe”, namely process 1112, is manually executed by the user. In this case, “pago 4094.exe” was executed by double-clicking the Desktop icon.

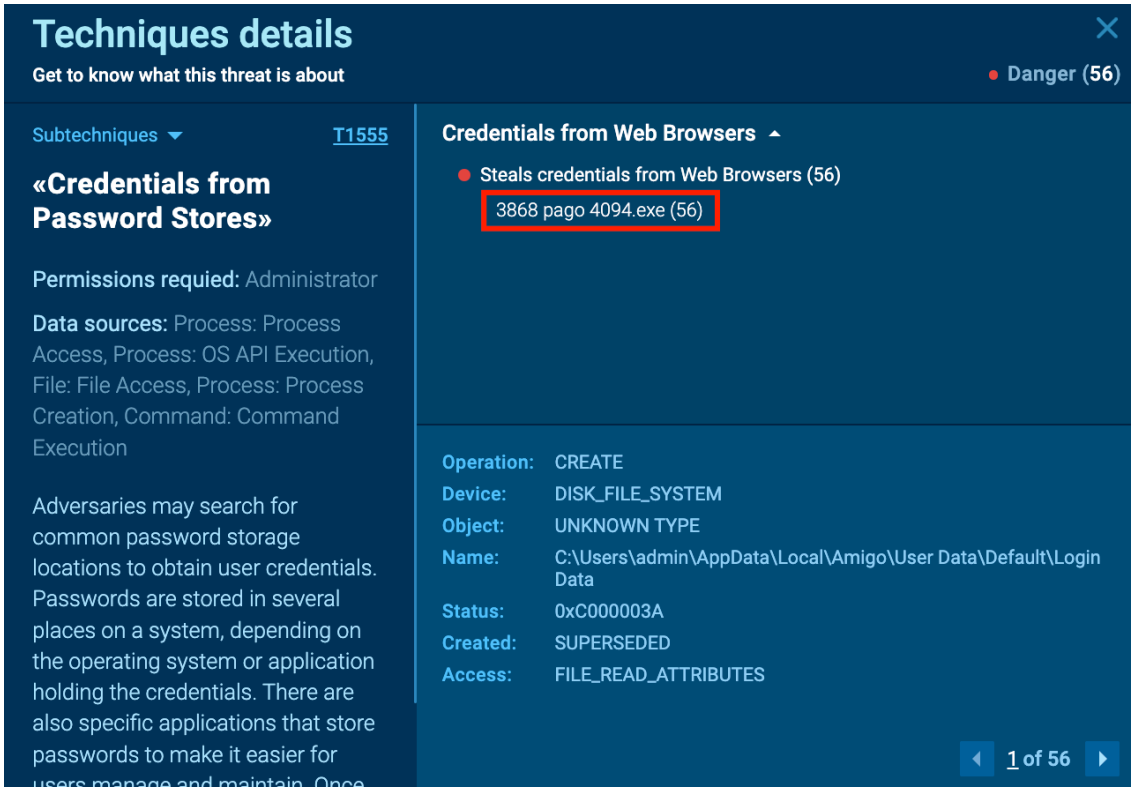
The technique here is T1204 (User Execution), and the subtechnique is T1204.002 (User Execution: Malicious File).



Techniques details of *User Execution*

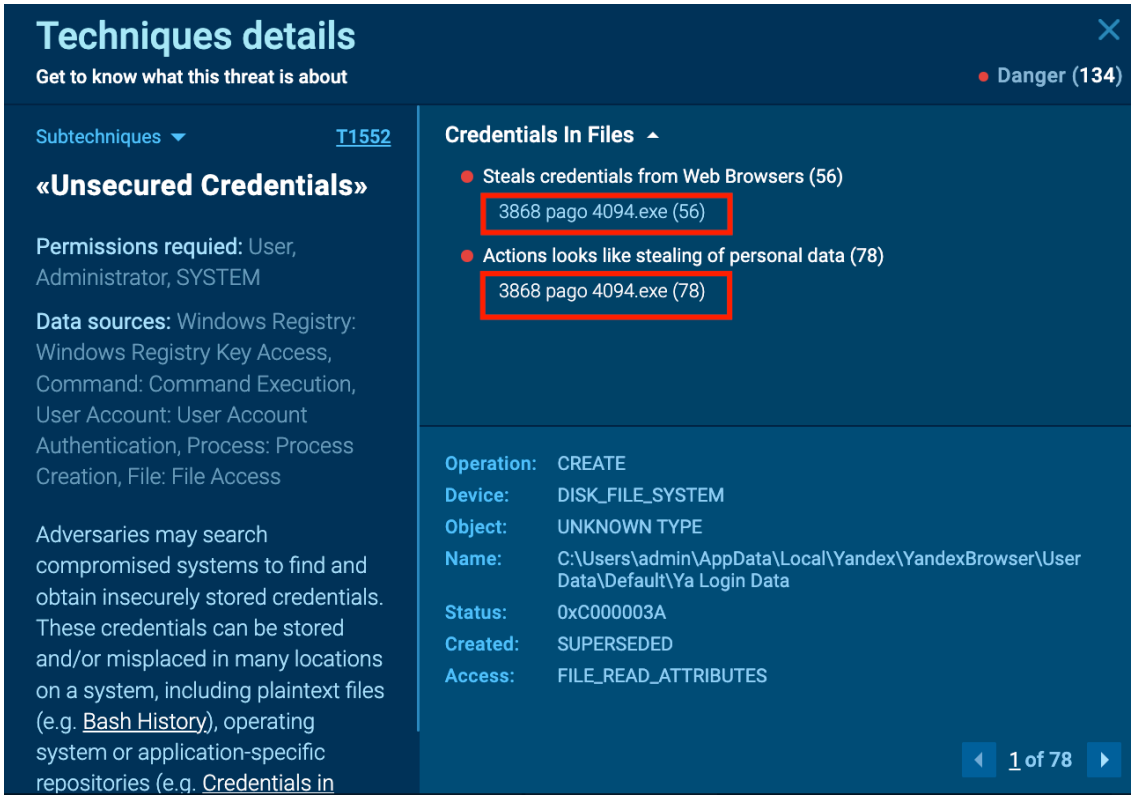
MITRE ATT&CK: Credential Access

Process 3868 attempted to steal credentials from web browsers and files. The technique here is T1555 (Credentials from Password stores), and the subtechnique is T1555.003 (Credentials from Password Stores: Credentials from Web Browsers).



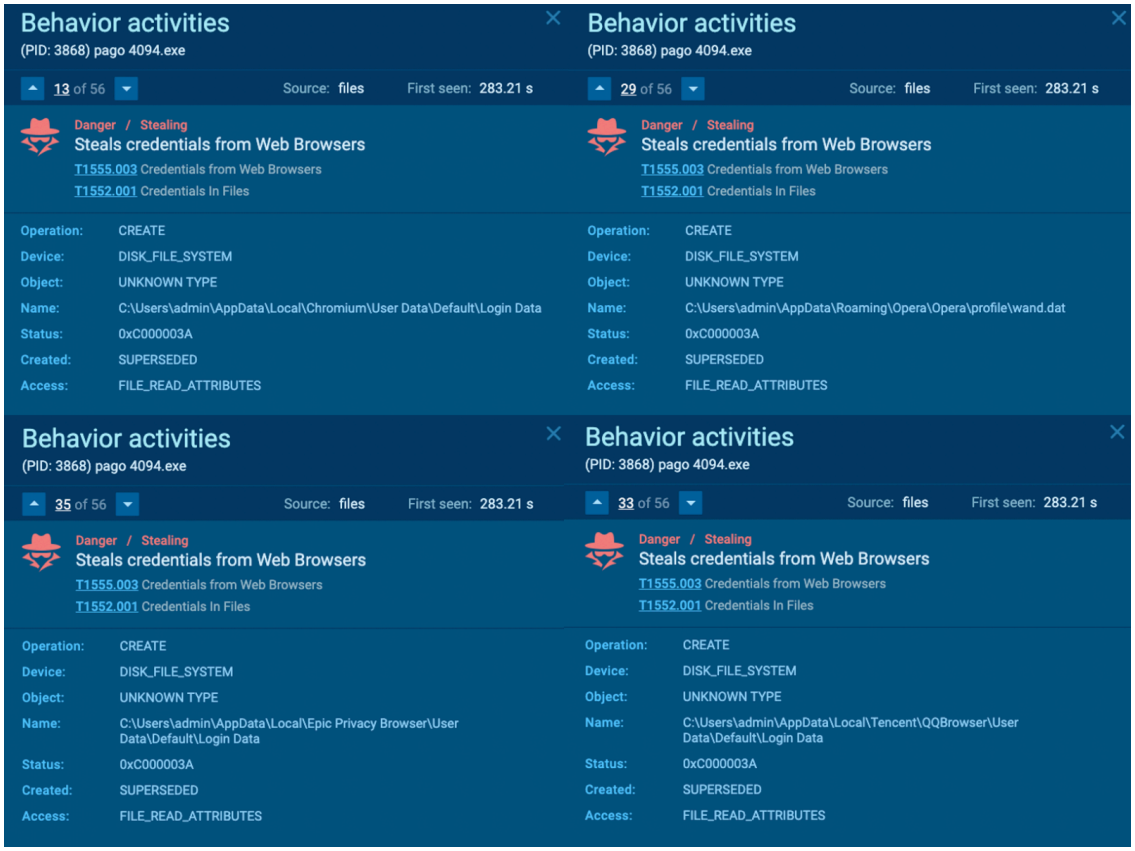
Techniques details of *Credentials from Password Stores*

It is also technique T1552 (Unsecured Credentials), and the subtechnique is T1552.001 (Unsecured Credentials: Credentials In Files).



Techniques details of *Unsecured Credentials*

Process 3868 attempted “FILE_READ_ATTRIBUTES” access on files associated with browsers under the “C:\Users\admin\AppData\Local\…” and “C:\Users\admin\AppData\Roaming\…” directory.



Process 3868 attempted to steal credentials from Chromium, Opera, Epic Privacy Browser, QQ Browser, etc.

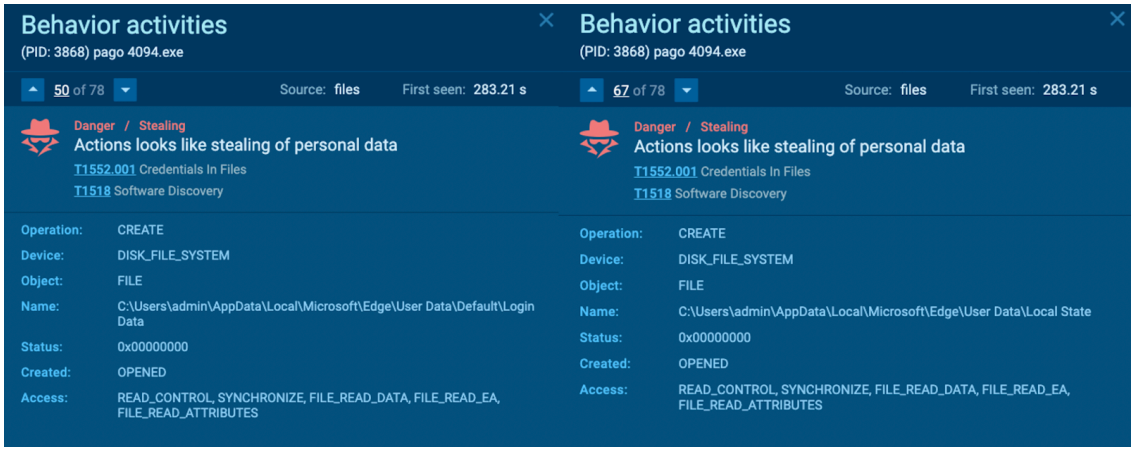
Before executing “pago 4094.exe”, fake credentials were saved in Google Chrome and Microsoft Edge.

Thus, process 3868 attempted the following accesses on files related to Google Chrome, which were in “C:\USERS\ADMIN\APPDATA\LOCAL\GOOGLE\CHROME\USER DATA\DEFAULT\LOGIN DATA” and “C:\Users\admin\AppData\Local\Google\Chrome\User Data\Local State”:

- FILE_READ_ATTRIBUTES
- READ_CONTROL
- SYNCHRONIZE
- FILE_READ_DATA
- FILE_READ_EA
- FILE_READ_ATTRIBUTES

This process also attempted these accesses on files related to Microsoft Edge, which were in “C:\Users\admin\AppData\Local\Microsoft\Edge\User Data\Default\Login Data” and “C:\Users\admin\AppData\Local\Microsoft\Edge\User Data\Local State”:

Data being stolen from Google Chrome



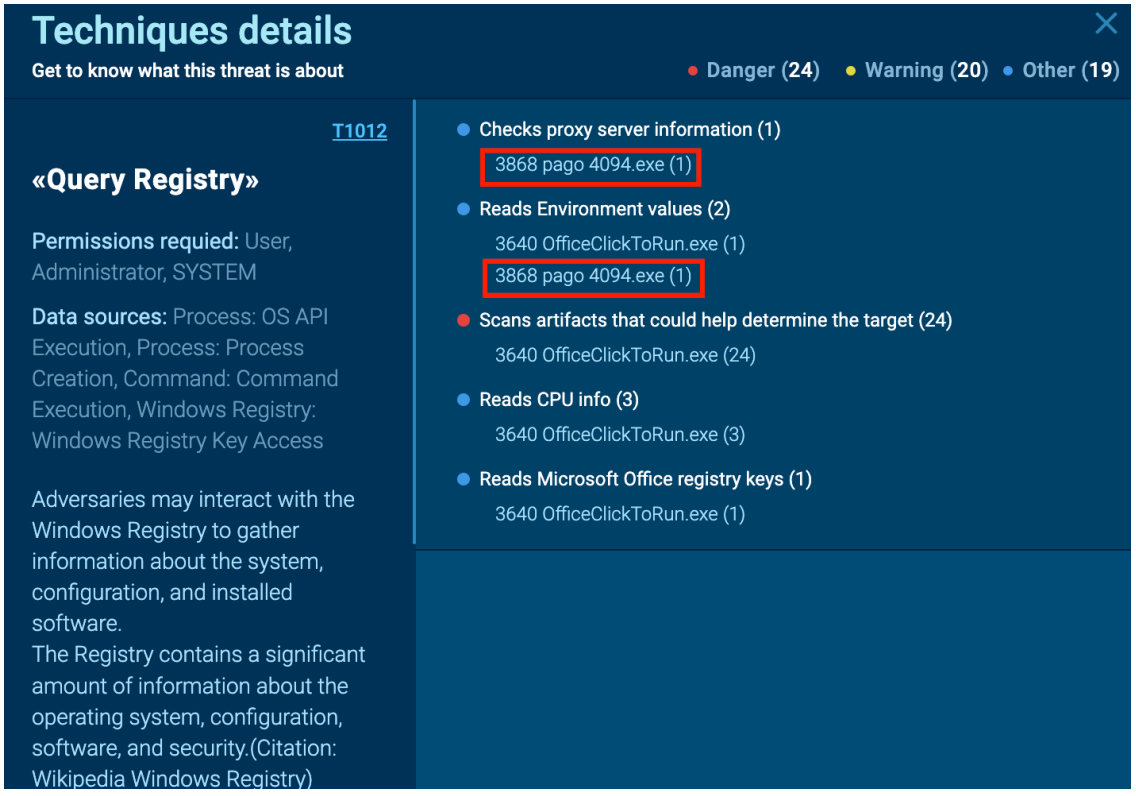
Data being stolen from Microsoft Edge

MITRE ATT&CK: Discovery

Processes 1112 and 3868 attempts to query the registry. The registry contains a lot of crucial system information, such as OS, configuration, software, and security. The technique here is T1012 (Query Registry).

The processes attempted the following:

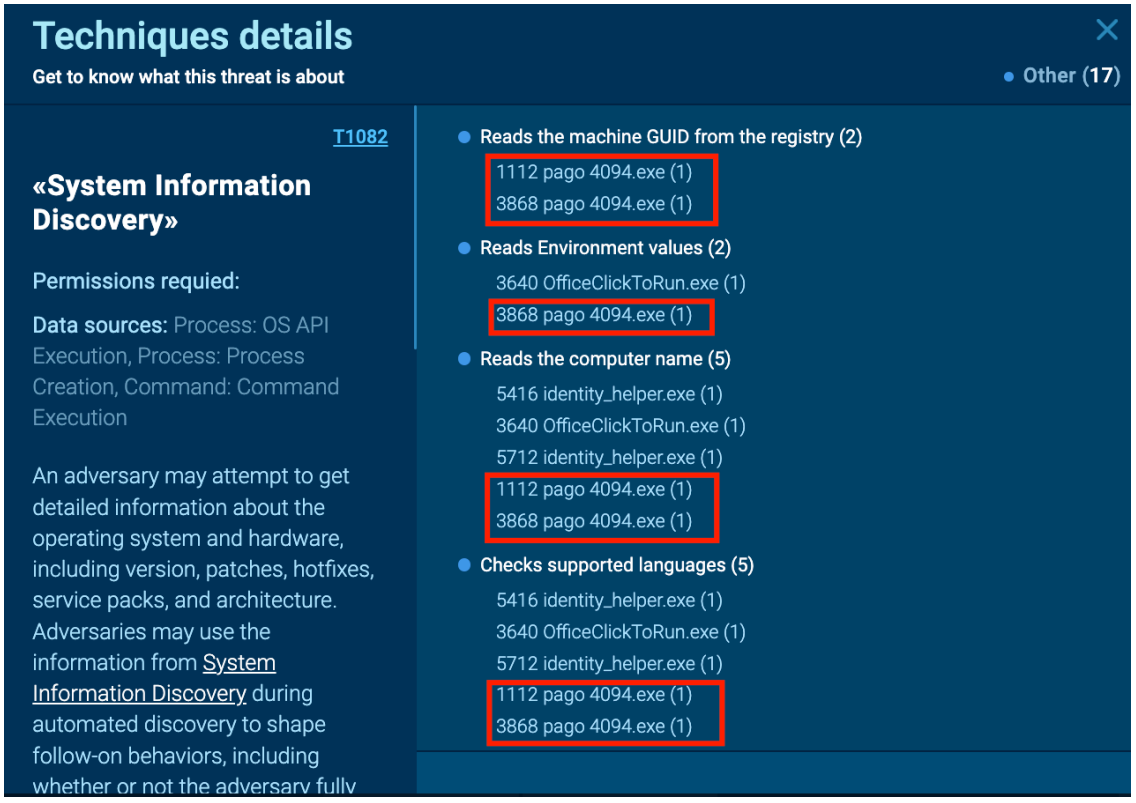
Techniques details of *Query Registry*



Techniques details of *Query Registry*

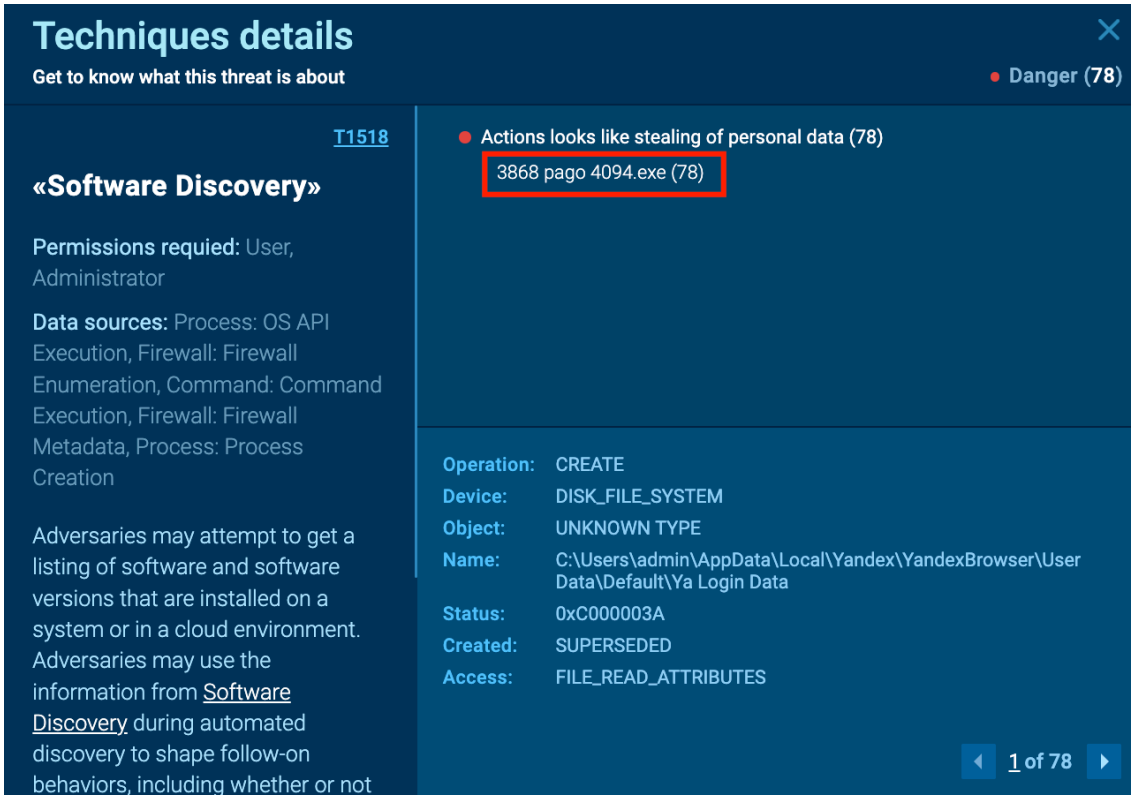
Process 1112 and 3868 attempts to discover system information, and tries to gather crucial system information. The technique here is T1082 (System Information Discovery).

There are overlaps between this and the previous subtechnique T1012:



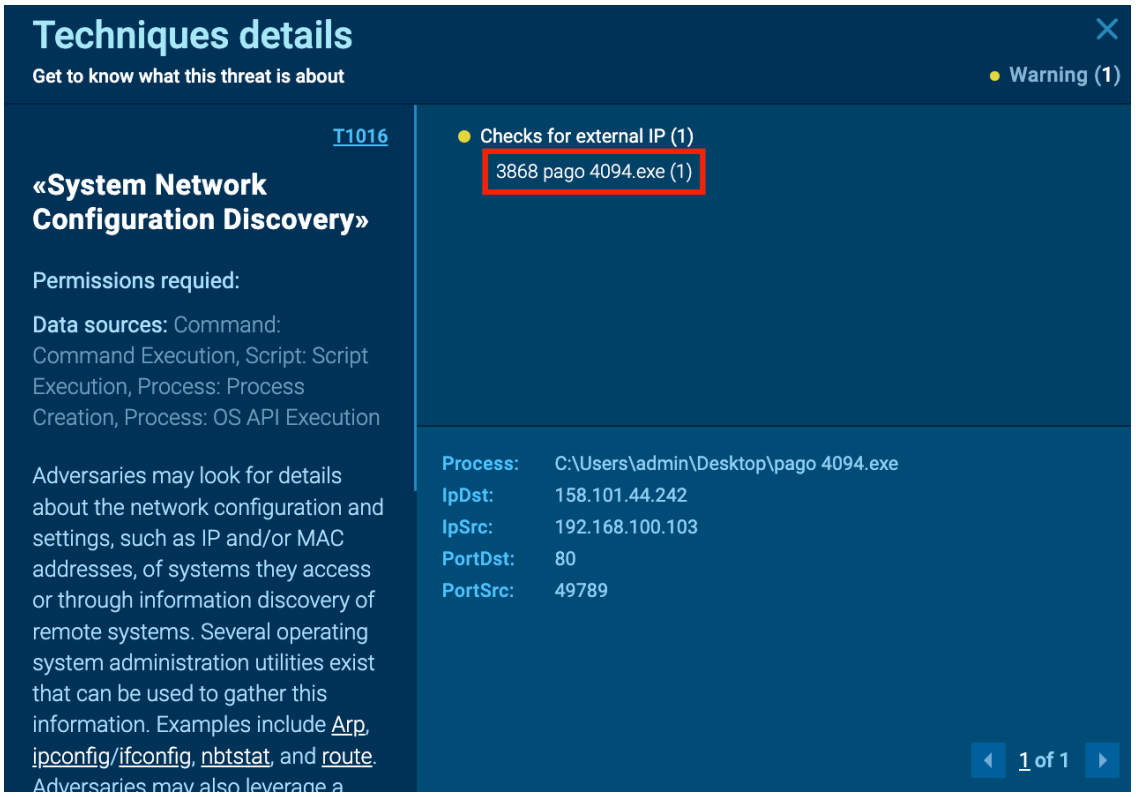
Techniques details of *System Information Discovery*

Process 3868 attempts to discover installed software, and it attempted to access various locations associated with Browsers. The technique here is T1518 (Software Discovery).



Techniques details of *Software Discovery*

Process 3868 attempts to discover the system network configuration. It checked for external IP, where the destination IP was 158.101.44[.]242 and the destination port was 80. The technique here is T1016 (System Network Configuration Discovery).

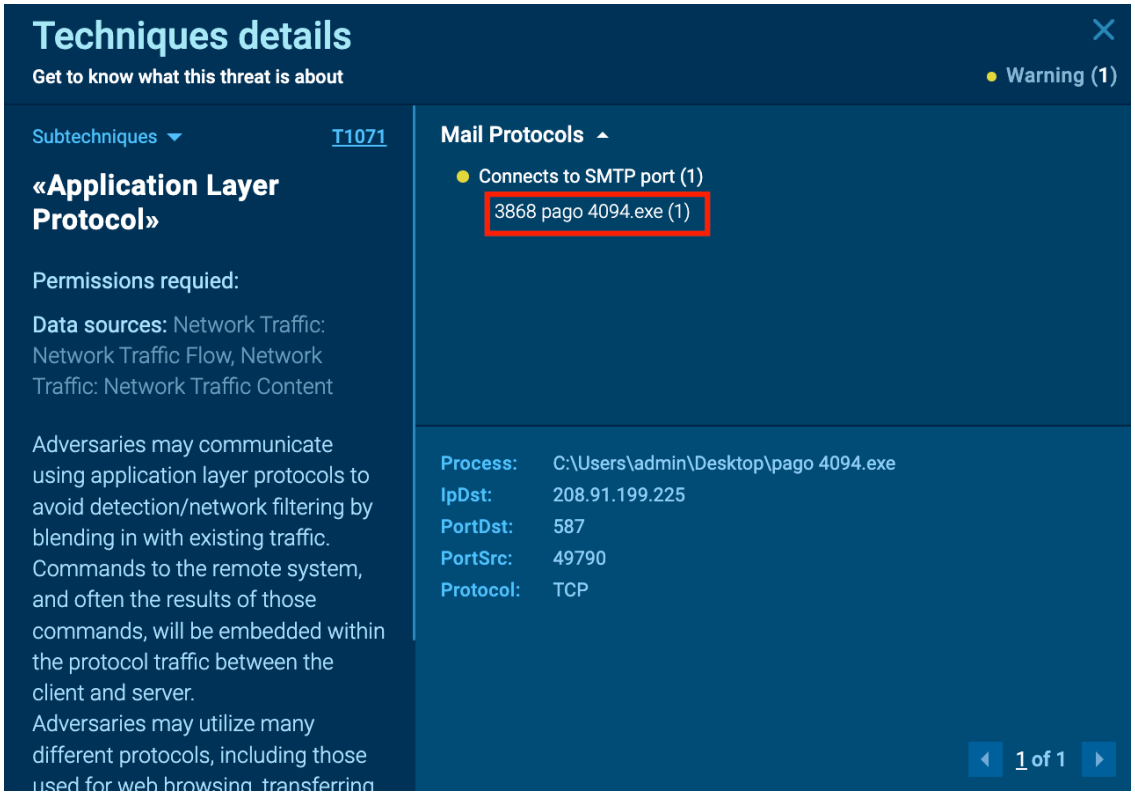


Techniques details of *System Network Configuration Discovery*

MITRE ATT&CK: C&C

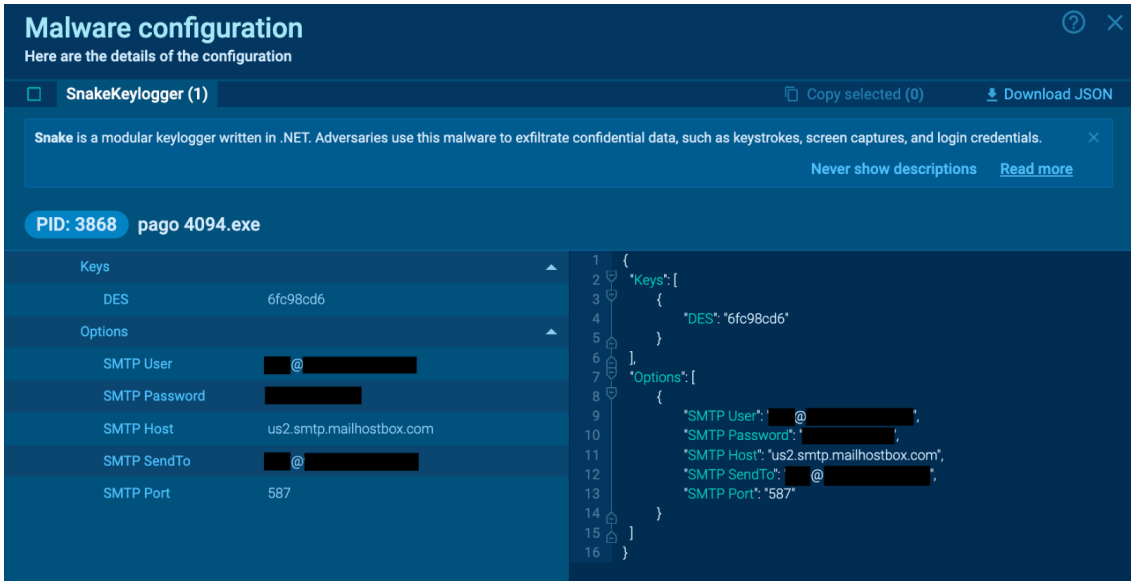
Process 3868 then communicates with the application layer protocol. Due to the existing background traffic, communication using the application layer protocols may fly under the radar. It was seen connecting to the SMTP port 587, where the destination IP was 208.91.199[.]225.

The technique here is T1071 (Application Layer Protocol), and the subtechnique is T1071.003 (Application Layer Protocol: Mail Protocols).



Techniques details of *Application Layer Protocol*

Finally, the malware configuration for the Snake Keylogger can be seen in ANY.RUN's *Malware Configuration:*



The *Malware Configuration* for the Snake Keylogger

Conclusion

This analysis showed how a single malicious email can lead to multiple security risks, including financial and reputational damage. We used various techniques like email and attachment analysis, process and network analysis, and applied the MITRE ATT&CK.

The focus was on an email with a Snake Keylogger attachment. It collects system info, establishes persistence, steals credentials, and exfiltrates data.

Given that emails remain a top threat vector often exploiting human error, staying vigilant against email threats is crucial.

About ANY.RUN

ANY.RUN is a cloud malware sandbox that handles the heavy lifting of malware analysis for SOC and DFIR teams. Every day, 300,000 professionals use our platform to investigate incidents and streamline threat analysis.

Request a demo today and enjoy 14 days of free access to our Enterprise plan.

Request demo →

Appendix 1: IOCs

Analyzed files:

Name	**32b4f238-3516-b261-c3ae-0c570d22ee18.eml**
MD5	60D00C17D3EA15910893EEF868DE7A65
SHA1	1D17DD1688A903CBE423D8DE58F8A7AB7ECE1EA5
SHA256	D13A7EAAF07C924159EA7BB8F297DAB1D8DA0F9AF46E82E24052D6A9BF5E4087
SSDEEP	12288:vZ1Tzm0D2acQLqgVIjejueFyhaCV2JKKS7hoxSSqkljhEi9lV7j:z7K8FuuzCV2JKkxPOQ3

Name	**pago 4094.exe**
MD5	1A0F4CC0513F1B56FEF01C815410C6EA
SHA1	A663C9ECF8F488D6E07B892165AE0A3712B0E91F
SHA256	D483D48C15F797C92C89D2EAFCC9FC7CBE0C02CABE1D9130BB9069E8C897C94C
SSDEEP	12288:PXPZDbCo/k+n70P4uR87fD0iBTJj1ijFDTwA:hOz+IPz6/PF1ihDTwA

Connections:

- 158.101.44[.]242・ *checkip.dyndns[.]org*
- 208.91.199[.]255・*us2.smtp.mailhostbox[.]com*

Appendix 2: MITRE MATRIX

Tactics	Techniques	Description
TA0001: Initial Access	T1566: Phishing	Send phishing messages to gain access to victim systems.
TA0002: Execution	T1204: User Execution	Rely upon specific actions by a user in order to gain execution.
TA0006: Credential Access	T1555: Credentials from Password Stores	Search for common password storage locations to obtain user credentials.
	T1552: Unsecured Credentials	Search compromised systems to find and obtain insecurely stored credentials.
TA0007: Discovery	T1012: Query Registry	Interact with the Windows Registry to gather information.
	T1082: System Information Discovery	Get detailed information about the operating system and hardware.
	T1518: Software Discovery	Get a listing of software and software versions that are installed.
	T1016: System Network Configuration Discovery	Look for details about the network configuration and settings.
TA0011: Command and Control	T1071: Application Layer Protocol	Communicate using OSI application layer protocols to avoid detection.

Lena aka LambdaMamba

Chief Research Officer

I am a Chief Research Officer at a cybersecurity company. My passions include investigations, experimentations, gaming, writing, and drawing. I also like playing around with hardware, operating systems, and FPGAs. I enjoy assembling things as well as disassembling things! In my spare time, I do CTFs, threat hunting, and write about them. I am fascinated by snakes, which includes the Snake Malware! Check out:

- My website
- My LinkedIn profile
