Amadey: New encoding with old tricks

vmray.com/cyber-security-blog/amadey-new-encoding-with-old-tricks/

Amadey:
New encoding with old tricks

Join us as we explore the malicious tactics and activities of the Amadey info stealer malware.

04 September 2023

Table of Contents

Family Overview

Beginning November 2022 here at VMRay we noticed increased activity of the Amadey information stealer malware. Monitoring of the threat landscape over the past several months showed this trend in the malware activity continued and the family is active as we speak.

Our observations, together with public reports in the community, are showing that Amadey can be deployed alongside SmokeLoader and RedLine information stealer or be used to drop additional payloads to the system.

The main functionality of Amadey is to collect information about the infected host, steal data, and download malware if configured so. It continually sends information back to its C2 server, like what Anti-Virus software is installed on the system (if any), OS version, machine architecture, etc. More about this and further details are discussed in this blog post.

Amadey's Behavior Analysis

In this blog post one of the latest versions of Amadey, 3.83, is taken into the spotlight. Our Platform uses a dynamic analysis approach, meaning the submitted file is executed in a virtual environment, **where the activities of the sample are recorded and analyzed** to detect malicious behavior.

VMRay is using extensive logic behind the scenes to detect the various suspicious and malicious actions of the sample, we call those rules triggering on certain actions VTIs (VMRay Threat Identifiers). In the case of this sample, they reveal plenty of malicious behavior: from YARA matches to capturing clipboard data, network connection, task scheduling etc. (see Figure 1).

VMRay Threat Identifiers (40 rules, 167 matches)

Score	Category	Operation	Count	Classification
5/5	Extracted Configuration	Amadey configuration was extracted	1	Downloader
5/5	Extracted Configuration	RedLine configuration was extracted	1	Spyware
5/5	YARA	Malicious content matched by YARA rules	28	Downloader, Spyware
5/5	Discovery	Combination of other detections shows configuration discovery	1	-
4/5	Defense Evasion	Tries to disable antivirus software	4	-
4/5	System Modification	Modifies Windows Update configuration	1	-
3/5	Input Capture	Captures clipboard data	2	Spyware
3/5	System Modification	Disables a crucial system service	4	-
3/5	Defense Evasion	Tries to detect the presence of antivirus software	1	-
3/5	Defense Evasion	Tries to detect the presence of anti-spyware software	1	-
3/5	Defense Evasion	Tries to detect the presence of firewall software	1	-
3/5	Network Connection	Uses HTTP to upload a large amount of data.	2	-
2/5	Discovery	Queries a host's domain name	2	-
2/5	Anti Analysis	Delays execution	3	-
2/5	Hide Tracks	Deletes file after execution	14	-
2/5	Discovery	Executes WMI query	8	-
2/5	Discovery	Collects hardware properties	1	-
2/5	Discovery	Enumerates running processes	1	-
2/5	Execution	Sends control codes to a driver	1	-
2/5	Discovery	Queries OS version via WMI	1	-
2/5	Discovery	Searches for sensitive browser data	22	-
2/5	Discovery	Searches for sensitive mail data	1	-
2/5	Task Scheduling	Schedules task	1	-
2/5	Task Scheduling	Schedules task via schtasks	1	-
1/5	Discovery	Enumerates running processes	1	-
1/5	Hide Tracks	Creates process with hidden window	22	-
1/5	Mutex	Creates mutex	4	-
1/5	Persistence	Installs system startup script or application	6	-
1/5	Privilege Escalation	Enables process privilege	5	-
1/5	Discovery	Possibly does reconnaissance	2	-
1/5	Defense Evasion	Accesses volumes directly	1	-
1/5	Obfuscation	Reads from memory of another process	10	-
1/5	Network Connection	Connects to remote host	1	-
1/5	Network Connection	Downloads executable	3	Downloader
1/5	Network Connection	Tries to connect using an uncommon port	1	-
1/5	Execution	Executes dropped PE file	6	-
1/5	Obfuscation	Resolves API functions dynamically	2	-

Figure 1: VMRay Platform's VTIs triggering on the malicious behavior of Amadey.

In addition to the detections themselves, VTIs also provide additional details on the detection and what is observed. In Figure 2 below we can see the VTI for Network connection that indicates data upload. Additionally, the description of the VTI gives more detailed information like which process is related to the behavior, how much data is exfiltrated and what is the method used – in this case, processes number 2 and 49 upload data via a HTTP POST method.

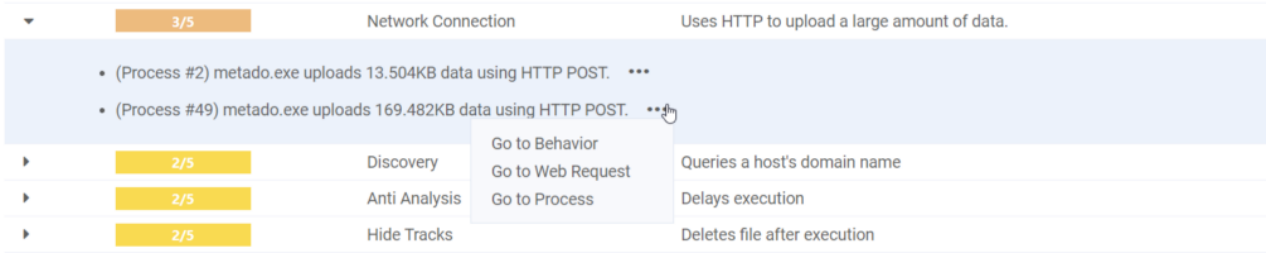

Figure 2: VMRay Platform detects data exfiltration performed by Amadey.

Following the action menu by selecting “Go to Web Request” we can get much more detailed information about the web request (see Figure 3) with all the important context of the connection.

It can be seen that the sample is reaching out and posting data to the C2 server, as well as the download attempt of two additional DLLs – one of them, clip64.dll, available on the C2 server, and the other one, cred64.dll, is returning 404 (HTTP Not Found). Clip64.dll will be covered more deeply in a bit.

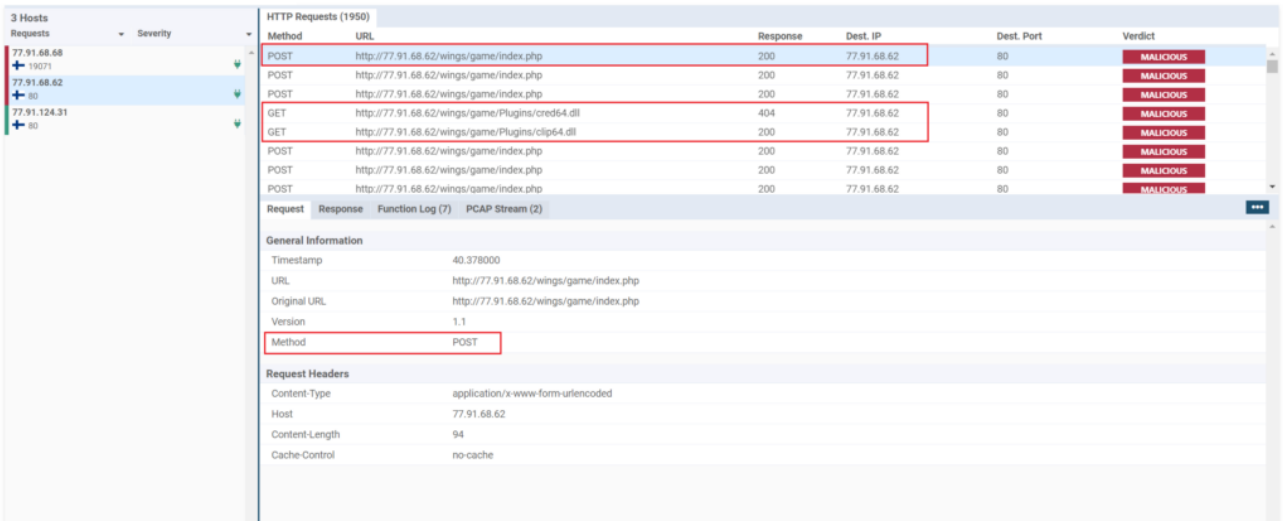

Figure 3: VMRay Platform showing the network communication with all the details needed for analysis.

In addition to this, if we look at the PCAP Streams tab, the beaconing of the sample to its C2 is shown in Figure 4:

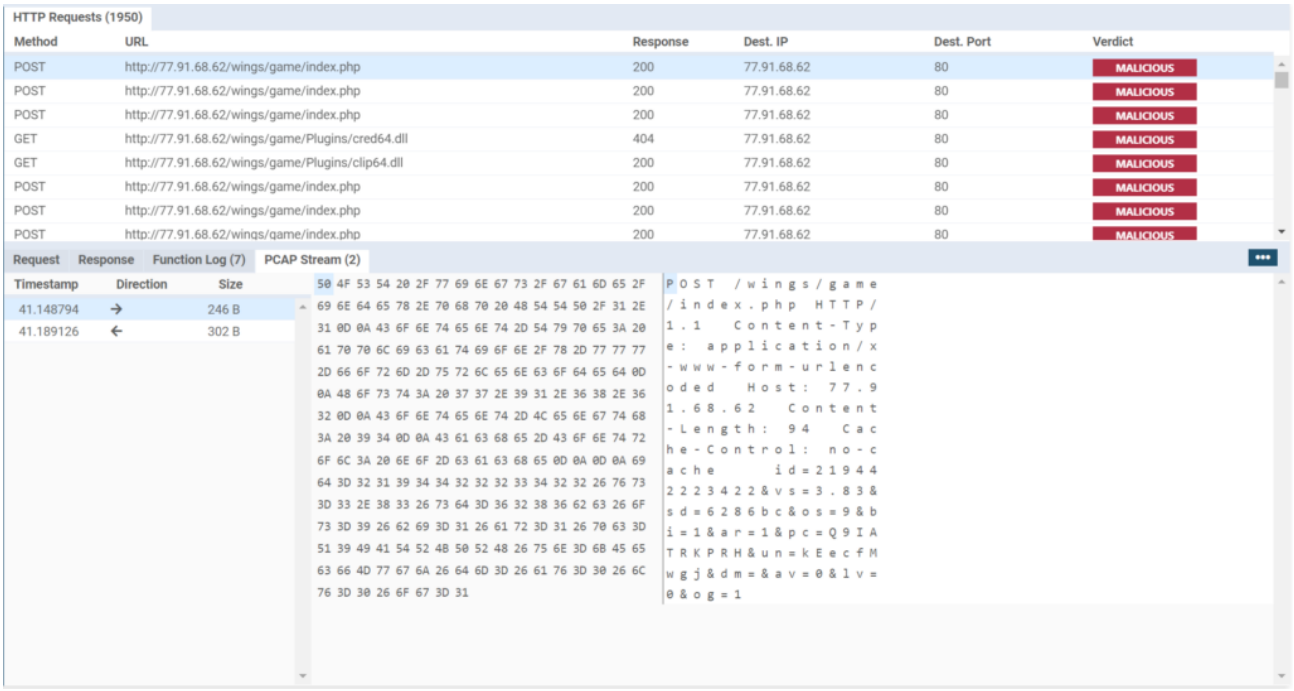

Figure 4: VMRay Platform's Network Tab providing information on how Amadey is exfiltrating information about the infected system.

The screenshot is showing that the communication protocol and values used in the requests to the C2 remain the same even for version 3.83:

id=219442223422&vs=3.83&sd=6286bc&os=9&bi=1&ar=1&pc=Q9IATRKPRH&un=kEecfMwgj&dm=&av=0&lv=0&og=1

Item	Data Example	Meaning
id	219442223422	Infected system's ID
vs	3.83	Amadey version
sd	6286bc	Amadey ID
os	9	Operating system 1 for Windows 10 4 for Windows Server 2012 9 for Windows 7 16 for Windows Server 2019
bi	1	Architecture 0 for x86 1 for x64
ar	1	0 for disabled 1 for enabled
pc	Q9IATRKPRH	Computer name
un	kEecfMwgj	User name
dm	(not available / empty)	Domain name
av	0	Windows Defender

Item	Data Example	Meaning
lv	0	Install additional malware on the system 0 for No 1 for Yes
og	1	Unknown

It's interesting to note that the "og" parameter has been observed only in later versions of the Amadey bot, but it's purpose is still unknown.

Below is a list of values that can be passed to the "av" parameter.

AV Product	Code
AVAST Software	1
Avira	2
Kaspersky Lab	3
ESET	4
Panda Security	5
Doctor Web	6
AVG	7
360TotalSecurity	8
BitDefender	9
Norton	10
Sophos	11
Comodo	12
Windows Defender	13

Given all the information available to us, we can draw some conclusions about Amadey:

1. As aforementioned, the protocol and values representing different properties of the infected system remain the same as previous versions with the exception of the "og" parameter.
2. The countries where the C2 is hosted continues to be either Sweden or Russia.
3. The name of the executable being dropped on the system stays the same across the same version of Amadey – in our case for version 3.83 it's metado.exe.
4. Two DLL files are mainly used – cred64.dll for stealing credentials and clip64.dll for getting clipboard data.
5. Enumeration of antivirus software installed on the infected machine (Figure 5).
6. Both samples and additional modules continue to be shipped with their PDB strings:

Binary	PDB String
cred64.dll	D:\\Mktmp\\Amadey\\StealerDLL\\x64\\Release\\STEALERDLL.pdb
clip64.dll	D:\\Mktmp\\Amadey\\ClipperDLL\\Release\\CLIPPERDLL.pdb
sample	D:\\Mktmp\\Amadey\\Release\\Amadey.pdb

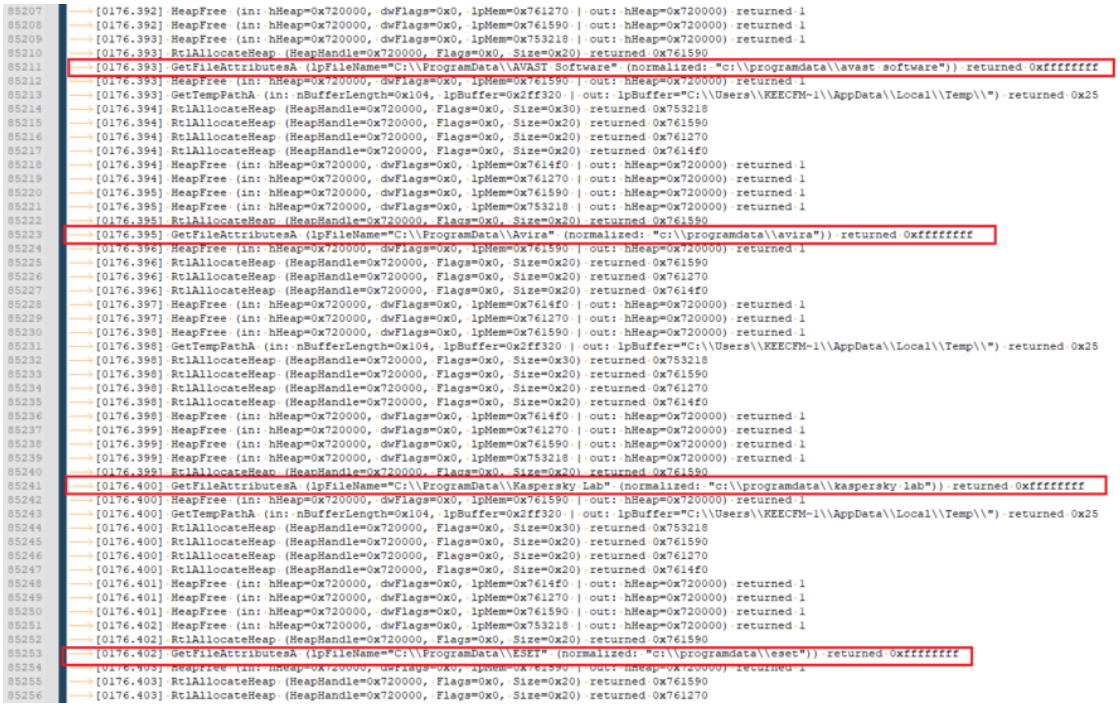

Figure 5: VMRay Platform's Function log reveals the enumeration of AVs.

Additional Modules

As pointed out earlier in this post, Amadey is still using two main modules for stealing data – cred64.dll and clip64.dll. Although recently we haven't seen samples actually using the cred64.dll file, we can observe the C2 download attempt associated with this library.

For the analyzed binary the network call is resulting in a HTTP 404 status code, indicating the the file could not be found on the C2 server. For clip64.dll the situation is a bit different. We can see that the DLL is present on the C2 and downloaded successfully. Taking a short look into the library, there are a few interesting things to note:

1. With a high level of certainty, we can say that Amadey's modules follow the same encoding scheme for its string as the one used in the sample itself. This of course means we can extract the plain text values of the strings in the DLL the same way we do for the actual sample.
2. The encoded strings in clip64.dll are mainly cryptocurrency wallet addresses (Figure 6).

```
.rdata:10013C90                                         ; DATA XREF: sub_100011C0+A↑o
.rdata:10013CA5                 align 4
.rdata:10013CA8 a160cbe54f0b273 db '160cbe54f0b273951f758f9cee76bb0f',0
.rdata:10013CA8                                         ; DATA XREF: sub_10001000+2↑o
.rdata:10013CC9                 align 4
.rdata:10013CCC aOhbzd1hljczu7y db 'OhBzd1HljcZU7yL03MJbaLK6fLAtPEKs5OB6dnLa49CaIUDm37IU9Ya5',0
.rdata:10013CCC                                         ; DATA XREF: sub_10001020+2↑o
.rdata:10013CCC                                         ; sub_10001040+2↑o ...
.rdata:10013D05                 align 4
.rdata:10013D08 aCc66pyosstwcib db 'CC66PYOsStWcIbyrDrAVOYTjSX dOVJ DO 6RDCURJRpULGXDIVwVYzG',0
.rdata:10013D08                                         ; DATA XREF: sub_10001080+2↑o
.rdata:10013D41                 align 4
.rdata:10013D44 aJbg1d7vqh90l7t db 'JBG1d7vQh90l7TCtMp1ISMLv2qAYO4J5PiyKSKbjfIGF 9==',0
.rdata:10013D44                                         ; DATA XREF: sub_100010A0+2↑o
.rdata:10013D75                 align 4
.rdata:10013D78 aHzysfqtffios8o db 'HzysfqTffIOS8OfGK1FPO5vUf41Me2d53BexT7fHWbGfV9==',0
.rdata:10013D78                                         ; DATA XREF: sub_100010C0+2↑o
.rdata:10013DA9                 align 10h
```

Figure 6: Excerpt from IDA showing the encoding key and some encoded strings in clip64.dll.

After decoding some of the strings we get the following cryptocurrency wallet addresses:

Bitcoin
bc1qslzv7hczpsatc8lq285gy38r4af0c3alsc4m77

Ethereum
0x89E34Ee2016a5E5a97b5E9598C251D2a2746Ba0D

Litecoin
LdYspWr6nkQ3ZNNTsmba77u4frHDhji1Nv

Dogecoin
DBjzffi3umhLQbUGLRoNQwZ4pjoKyNFahf

Monero
42zbZM5ozb4iDSN7hxNnQ1DSAvEmGY3z2KvAYmMxSJkUCc5bJyJ5hdkUu4324VJx8ACcDJJXg2NbRdWVcDyS87tyLikjVVJ

Clipboard inspection is done via the Windows API calls **OpenClipboard** and **GetClipboardData** (Figure 7). If the sample detects that a crypto currency address is in the clipboard, the behavior changes and the victim's address is replaced with the attacker's one via **SetClipboardData**.

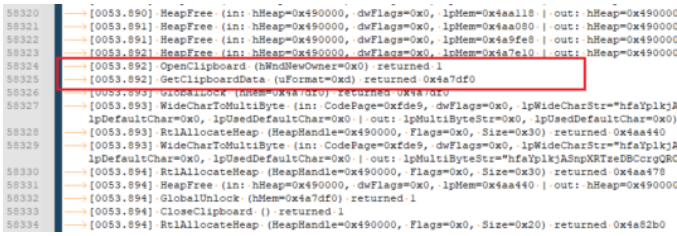

Figure 7: VMRay Platform's Function log indicates Amadey reading the clipboard data.

Persistence

The persistence mechanism hasn't changed as well from previous versions and still uses scheduled task (Figure 8), and this is detected with a respective VTI (see Figure 1).

A great feature of the VMRay Platform is that whenever a persistence mechanism is detected, **a reboot of the analysis environment is scheduled so that the additional behavior after a restart can be inspected.**

Amadey adds a new startup item via a registry key called "User Shell Folders" which holds the path to the Amadey executable dropped into the temporary directory of the current user. With this Amadey is executed every time a user logs into the system.

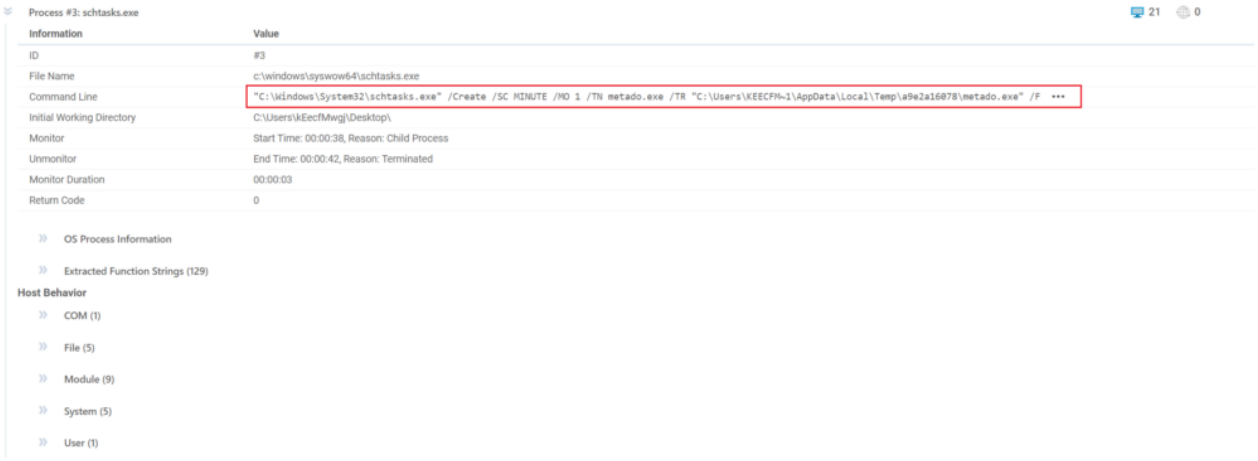

Figure 8: VMRay Platform reporting the persistence mechanism of Amadey.

In addition to this, the old technique of editing the access rights with Cacls.exe used in previous versions remains the same as well. In Figure 9 we can see the sample first using the command:

CACLS “metado.exe” /P “kEecfMwgj:N” to remove any permissions for the user **kEecfMwgjand** then using the subsequent command:
CACLS “metado.exe” /P “kEecfMwgj:R” /E to change the access rights to read only, which prevents the file from being deleted.
CACLS “..\a9e2a16078” /P “kEecfMwgj:N” – The same operation as above is performed on the folder where the executable is copied to.

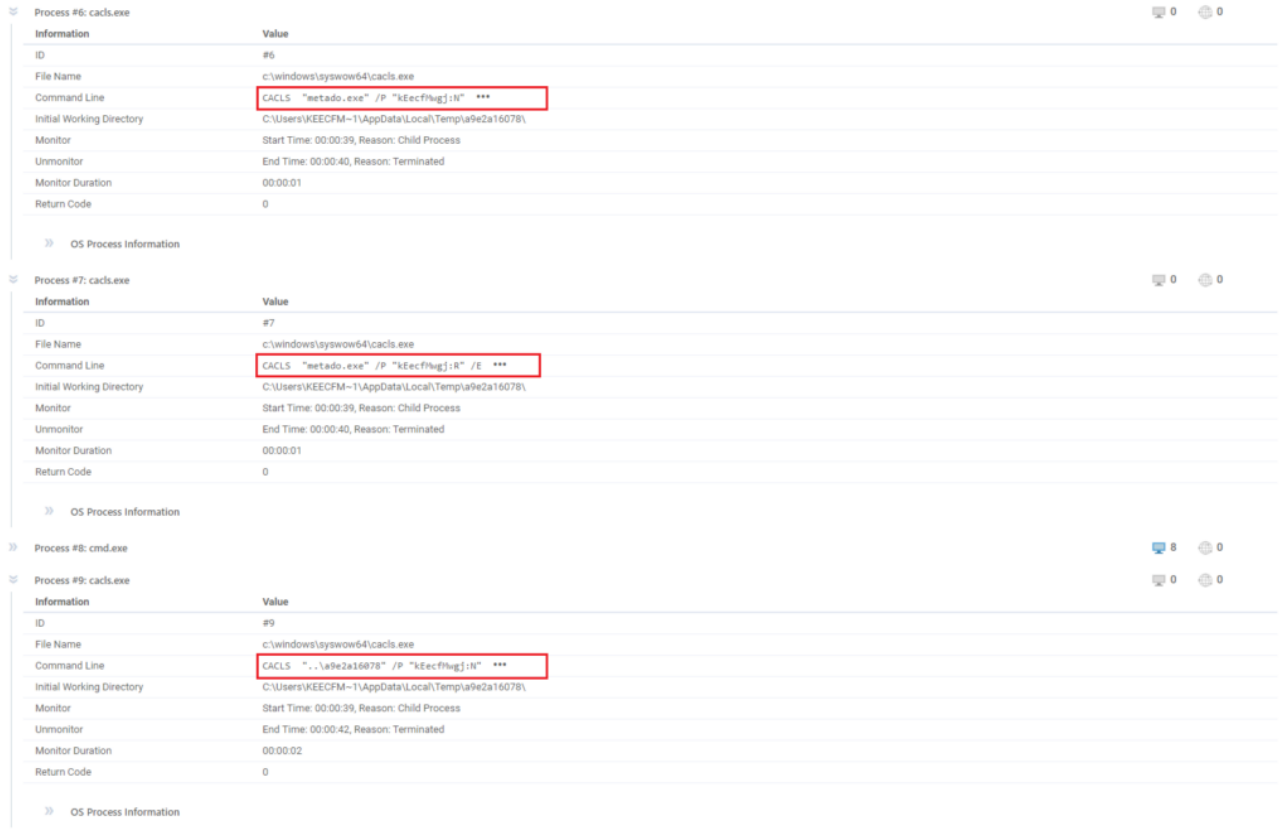

Figure 9: VMRay Platform shows the command line arguments being used to change the access rights.

New String Encoding

Another interesting part of the newer versions of Amadey and may be the most relevant one, is the new encoding used for obfuscation. While around version 3.20 Amadey was using a simpler encoding technique (transforming the string to hex representation) to obfuscate the strings in the sample and DLLs, with its latest version (around 3.60 and above) the encoding has changed and now on top of the old algorithm, some additional operations have been added that play with the length of the string and mapping to base64 base string.

A decoding routine had been developed in the end of 2022 by OALabs and it is still relevant until today. Interestingly, the updated algorithm is not used only in the samples, but transferred to the DLL modules as well.

Encoded String: LR0HQDfvJaR9RNbc8mcD8YSUNziebiKs5UEdL1Xs2cRqbwPe8nRt8YY7Kh0jTYYgQH==
Decoded String: SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunOnce

Encoded String: Eq4vJRGoC cqLay=
Decoded String: 77[.]91[.]68[.]62

Encoded String: CU5q7kftAS d NKo6W9oVT o3Aml
Decoded String: /wings/game/index.php

Configuration Extractor

Based on our analysis, we have developed a configuration extractor which allows our customers to examine the configuration built into the executable. The extracted data consists of C2 IP address and the URL used for communication and data exfiltration, the encryption key (in Amadey's case encoding, as the strings are not really encrypted), directory where additional modules are dropped, mutex and version of the malware. Information like this can be easily used to hunt for malicious traffic in the network and look for IOCs in general.

In Figure 10 below, we can see that besides Amadey configuration, RedLine config has also been extracted.

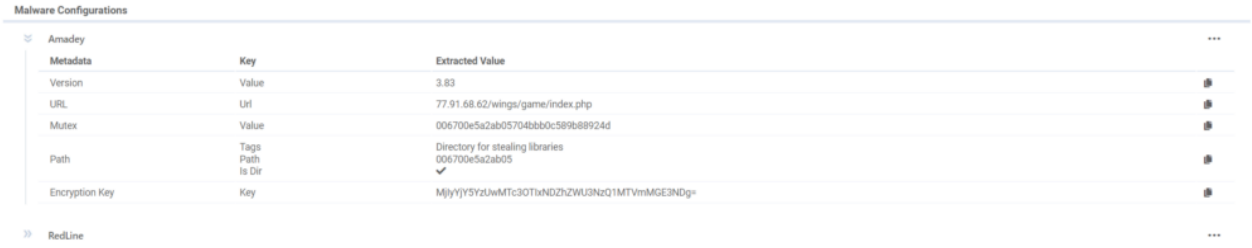

Figure 10: VMRay Platform's malware configuration extraction for Amadey.

Conclusion

VMRay is constantly monitoring the threat landscape, identifying new techniques, and keeping an eye on the most relevant families. Detecting changes early allows us to investigate them, examine the new behavior of the threats, and be prepared to protect the assets of our customers.

Different features of our product are at the disposal of SOC teams, malware analysts, threat hunters, and other cyber security professionals. The detailed analysis provided by VMRay Platform, can give deep insights into the malware's behavior. Our function log is keeping track of every relevant API call, shedding light on the analyzed samples' actions.

We are constantly working to increase and improve our VTIs and YARA rules so no threat is left undetected.

References

- https://asec.ahnlab.com/en/36634/
- https://research.openanalysis.net/cpp/stl/amadey/loader/config/2022/11/13/amadey.html
- https://blog.cyble.com/2023/01/25/the-rise-of-amadey-bot-a-growing-concern-for-internet-security/
- https://www.rewterz.com/articles/amadey-malware-analysis-report/IOCs

IOCs

Analyzed Sample:
ecbcec33ca7e445570339dea41d5b52491681dfc4cadbccad3f82c37cf5cb903

C2
hxxp://77.91.68[.]62/wings/game/index.php
hxxp://77.91.124[.]31/new/foto135.exe
hxxp://77.91.124[.]31/new/fotod25.exe
hxxp://77.91.68[.]62/wings/game/Plugins/cred64.dll
hxxp://77.91.68[.]62/wings/game/Plugins/clip64.dll

Bitcoin Wallet Address:
bc1qslzv7hczpsatc8lq285gy38r4af0c3alsc4m77

Ethereum Wallet Address:
0x89E34Ee2016a5E5a97b5E9598C251D2a2746Ba0D

Litecoin Wallet Address:
LdYspWr6nkQ3ZNNTsmba77u4frHDhji1Nv

Dogecoin Wallet Address:
DBjzffi3umhLQbUGLRoNQwZ4pjoKyNFahf

Monero Wallet Address:
42zbZM5ozb4iDSN7hxNnQ1DSAvEmGY3z2KvAYmMxSJkUCc5bJyJ5hdkUu4324VJx8ACcDJJXg2NbRdWVcDyS87tyLikjVVJ

VMRay Labs Team
See VMRay in action.
Get full visibility into the most challenging threats.

REQUEST FREE TRIAL NOW