BadCyber
Making infosec journalism great again!

Technical analysis of recent attacks against Polish banks

badcyber / February 16, 2017 / Investigation, Malware / bank, malware, Poland

It has been three weeks since first information about succesful attacks on Polish banks has reached our ears. It’s time to put together the technical description of how the attacks were performed.

When two weeks ago we have revealed the fact that a few Polish banks were hacked by uknown attackers, the banks were super busy scanning internal networks and looking for intruders. Nobody bothered to write down the details of the infection process. Since then some companies tried to cover less or more technical details of this attack. We've seen publications by Symantec, Booz Allen and BAE Systems – we particulary recommend the latter, as in our opinion it includes the most technically correct information confirmed by multiple sources.

In this article we will try to cover the whole infection process – starting with a bank employee visiting the Polish Financial Supervision Authority website and ending in his workstation being infected by a clever RAT. We will include only one of the infection scenarios – we understand that there were others with less or more significant differences.

What happened in the browser

To start the infection process a bank employee had to visit the www.knf.gov.pl website sometimes between October 5h, 2016 and February 2nd, 2017. Most pages on this website included the following script:

```
http://www.knf.gov.pl/DefaultDesign/Layouts/KNF2013/resources/accordian-
src.js?ver=11
```

Somewhere next to the end of this file an external script was included – depending on the month it was either

```
http://sap.misapor.ch/vishop/view.jsp?pagenum=1
```

or

```
http://www.eye-watch.in/design/fancybox/images.jsp?pagenum=1
```

Attackers' controlled server performed some initial verification of the visitor – most probably based on IP whitelists including address classess deemeed interesting by the attackers. Once the target was confirmed as interesting, it received one of four exploits: cambio.xap file attacking vulnerable Silverlight plugin:

```
4cc10ab3f4ee6769e520694a10f611d5
```

including a malicious DLL file or cambio.swf file attempting to exploit Flash plugin:

```
6dffcfa68433f886b2e88fd984b4995a
1f2cd85583a4a56b764ba6429c2155ec
```

including three exploits: CVE-2015-8651, CVE-2016-1019 and CVE-2016-4117. Exploits used in this attack were *borrowed* from commercial exploit packs. As far as we know at the moment of the attacks none of the exploited vulnerabilities were considered a 0day.

The exploit call included a parameter pointing to a shellcode, which downloaded a reconnaissance tool (described in the BAE article) from the same domain. The tools gathered some basic info about the machine and uploaded it back to its server. If the analysis results were deemed interesting to the attackers, they could manually trigger downloading another malicious component, served as perfmon.dat. It was an encrypted EXE file, which was run on the target system after decryption.

What happened in the workstation

There was an EXE

```
bedceafa2109139c793cb158cec9fa48f980ff2b
```

found on one of the infected computers, which played the role of the installer for the next malware stage. It is a simple commandline tool, with basic capabilities, including

```
-l : list of potential services which clould be used by malware
-e [NAME] : unpacks DLL and CHM files under given NAME
```

The installer decrypts and saves two files (depending on target architecture in 32 or 64bit versions): a DLL file protected with commercial Enigma packer and a CHM file. The DLL file can be named srservice.dll

```
e29fe3c181ac9ddbb242688b151f3310
```

and is installed as a service, while the CHM file would be named srservice.chm

```
9216b29114fb6713ef228370cbfe4045
```

and saved in %WINDIR%\Help folder. In reality the CHM file is an encrypted DLL injected into the process. The service also decrypts the config file srservice.hlp

```
8e32fccd70cec634d13795bcb1da85ff
```

Files are encrypted using a RC4 variant called Spritz.

The config file includes, among other information, two domain names looking like C&C server – which they are not.

```
tradeboard.mefound.com:443
movis-es.ignorelist.com:443
```

In an interesting plot twist the malware resolves those domains to their real IP addresses, but then performs a XOR operation on the original IPs to obtain true C&C IP addresses.

The service.chm file is the final stage of this part of infection scenario – it's a RAT connecting to the C&C and taking orders. Some of the 20-something commands include:

- CMDL – execute command, write result to temporary file, upload the file to C&C and delete it
- DEL – remove file
- DIE – turn off
- DIR – search files or folders
- DOWN – download and save file
- DRIV – list logical drives
- GCFG – download current config
- PEEX – inject into explorer.exe
- PKIL – kill process
- PVEW – display information about running processes
- RUN – execute
- SCFG – get new config, encrypt it and save
- UPLD – upload file to C&C
- WIPE – secure delete

Another piece of malware identified during incident response was the fdsvc.exe file

```
9914075cc687bdc352ee136ac6579707
```

probalby run manually or coming from a different infection process. The file accepts parameters pointing to a server it should connect to and the process it should inject malicious code into. The injected file was fdsvc.dll

```
9cc6854bc5e217104734043c89dc4ff8
```

which was also encrypted.

Final thoughts

Our anaysis is based on multiple sources, most of which come from foreign companies, as getting the information from Polish sources was a really hard task. We would like to use this opportunity to thank all of our sources who were so kind to share their research results with us. We hope this publication will help to further the research into this incident.

2 thoughts on "Technical analysis of recent attacks against Polish banks"

steve er

February 17, 2017 at 4:54 pm

Do you have the sample bedceafa2109139c793cb158cec9fa48f980ff2b? if yes, could you shared me?

steve er

February 20, 2017 at 3:00 am

This sample 18a451d70f96a1335623b385f0993bcc have other C2, same campaign [Lazarus – Polish Malware]

Network
http://120,113,173,207:8080/view.jsp?action=baseinfo&u=47335087295280
http://120,113,173,207:8080/view.jsp

Gathering information
cmd.exe /c "hostname > %TEMP%\TMPE52E.tmp"
cmd.exe /c "whoami >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "ver >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "ipconfig -all >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "ping http://www.google.com >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "query user >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "net user >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "net view >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "net view /domain >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings" >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "tasklist /svc >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "netstat -ano | find "TCP" >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "wmic os get lastbootuptime >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%PROGRAMFILES%\ >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%PROGRAMFILES%\ >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%s\Desktop" >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%s\Documents" >> %TEMP%\TMPE52E.tmp"
cmd.exe /c "dir /od /a "%s\Favorites" >> %TEMP%\TMPE52E.tmp"
cmd.exe cmd /c ""%TEMP%\tmp095j.bat" "C:\18a451d70f96a1335623b385f0993bcc.exe"

tmp095j.bat

```
@echo off
:del1
del /a %1
if exist %1 goto del1
del /a %0
```