Zeus Banking Trojan

Malware Analysis Report

Adrian ‘ad1s0n’ Zalewski

13.10.2023

Table of Contents

Executive summary

Filename: invoice_2318362983713_8239313 42io.pdf.exe

The Zeus Banking Trojan represents a highly sophisticated and pervasive threat to the global financial sector. This malicious software, first identified in 2007, has evolved over the years to become a potent tool for cybercriminals aiming to compromise sensitive financial information.

Attack vector:

- Phishing

Main functionalities:

- Keystroke logging
- Stealing financial information
- Botnet
- Dropper

As it turns out the Zeus has several variants, which some of them can also perform devastating ransomware attacks (GameOver variant).

Prevention:

- Appropriate awareness training about phishing campaigns
- Updating used software, browsers and firewalls to the latest version
- Using **trusted** antimalware program and updating its threat database at least once per month

Fingerprinting

Hashes

Tools: Custom python script *calc_hashes.py* (Source code in appendix A)

MD5	ea039a854d20d7734c5add48f1a51c34
SHA256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169
IMPHASH	308fe2649c586660c71bc787d65e54fd
SSDEEP	6144:Tz/LBBTHT+7oEf2ZstxQMSGToLoOhD2saLsW8fsmFBkObjD:PLBdy7FpQMlToThD+sW8fsmP7bj

Table 1. File hashes

MD5	679fbf23d7317d8207d350b532908f0a
SHA256	8309b5d320b3d392e25afd57793e6bb9d54a3aeaca697759963b008f3367b352

Table 2. .text section hashes

MD5	73fdae90c1738941b6afec633c45972e
SHA256	510a0f9faf189356ca7819ac6a5cbe1da1d94ea110158e1c4d3bcb753c458ba5

Table 3. .data section hashes

MD5	37469a130e838cd467ff44551f2a43fb
SHA256	7c2f4c4db94369f90b2a41459cb3fb96eb9e9ff0d8631b7c6562467f0d8924b9

Table 4. .reloc section hashes

MD5	b3af18982aee2e1b39915237800c877e
SHA256	cb1cb914ad7f61c98bfb6506306e31a8d94df71b078c69405e9fbd8dd289c54f

Table 5. .rsrc section hashes

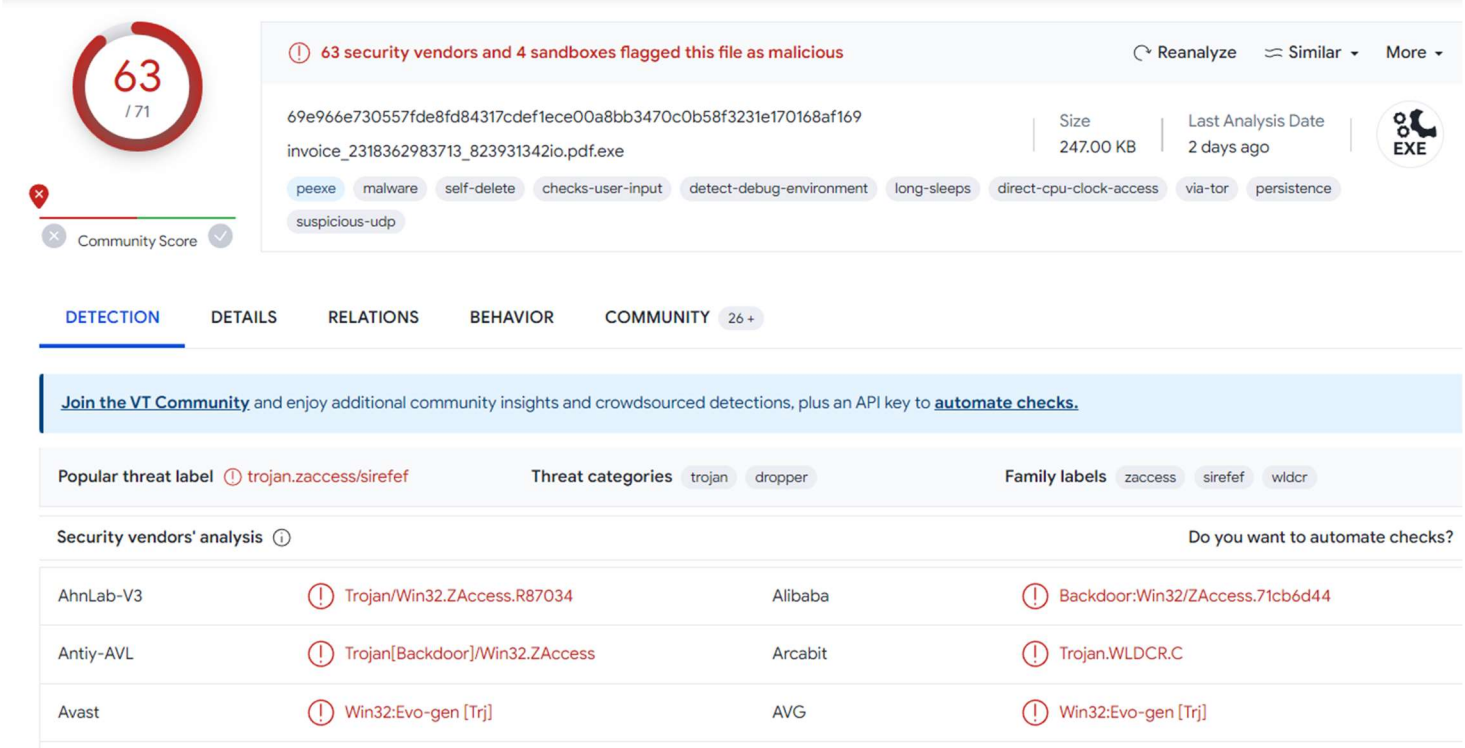
MD5	a8448d1b94e56bc8f80ed852445884c1
SHA256	70cc3e025cced228e4ebb21e54b904a2e0ccec85c0b0e292a1e12e7c819db0ae

Table 6. .pdata section hashes

MD5	7f89ad170ffea80a9c7304edf9c7f32c
SHA256	4cdd5d9821cc0790a1d7031ef6cd3dfa9e68b967279d3bd2f0de781ebcb95389

Table 6. .itext section hashes

VirusTotal output



File type

property	value
footprint > sha256	69E966E730557FDE8FD84317CDEF1ECE00A8BB3470C0B58F3231E170168AF169
first-bytes > hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes > text	M Z . . . . . @ . . . . .

Due to ‘MZ’ presence in first bytes, it can be stated that file is **executable**.

Observations

Basic Static Analysis

Tools: PEStudio, floss, capa

.text header analysis

property	value
headers	header[0]
name	.text
footprint > sha256	8309B5D320B3D392E25AFD5...
entropy	6.707
file-ratio (99.60%)	18.42 %
raw-address (begin)	0x00000400
raw-address (end)	0x0000BA00
raw-size (251904 bytes)	0x0000B600 (46592 bytes)
virtual-address	0x00001000
virtual-size (250379 bytes)	0x0000B571 (46449 bytes)

Difference between raw-size and virtual-size in .text header is small, so that malware is **not** packed.

Strings

Interesting Imports, API calls (.itext):

encoding (2)	size (bytes)	location	flag (17)	label (110)	group (11)	technique (7)	value (1416)
ascii	24	.itext	x	import	windowing	-	AllowSetForegroundWindow
ascii	22	.itext	x	import	reconnaissance	-	GetEnvironmentVariable
ascii	22	.itext	x	import	reconnaissance	-	GetEnvironmentVariable
ascii	9	.itext	x	import	input-output	-	VkKeyScan
ascii	16	.itext	x	import	input-output	T1056 | Input Capture	GetAsyncKeyState
ascii	19	.itext	x	import	file	-	PathRenameExtension
ascii	9	.itext	x	import	file	-	WriteFile
ascii	12	.itext	x	import	file	T1083 | File and Directory Discovery	FindNextFile
ascii	16	.itext	x	import	execution	-	GetCurrentThread
ascii	7	.itext	x	-	execution	T1106 | Execution through API	WinExec
ascii	13	.itext	x	import	data-exchange	-	GlobalAddAtom
ascii	17	.itext	x	import	data-exchange	T1115 | Clipboard Data	GetClipboardOwner
ascii	16	.itext	x	import	data-exchange	T1115 | Clipboard Data	GetClipboardData
ascii	20	.itext	x	import	data-exchange	T1115 | Clipboard Data	EnumClipboardFormats
ascii	18	.itext	x	import	data-exchange	-	DdeQueryNextServer
ascii	25	.itext	x	import	console	-	GetConsoleAliasExesLength
ascii	19	.itext	x	import	-	-	SetCurrentDirectory

GetAsyncState – possible use for keylogger

ascii	14	.itext	-	import	windowing	-	CallWindowProc
ascii	12	.itext	-	import	windowing	-	UpdateWindow
ascii	10	.itext	-	import	windowing	-	GetCapture
ascii	15	.itext	-	import	windowing	-	IsWindowEnabled
ascii	19	.itext	-	import	windowing	T1010 | Window Discovery	GetWindowTextLength
ascii	21	.itext	-	import	synchronization	-	DeleteCriticalSection
ascii	14	.itext	-	import	resource	-	SizeofResource
ascii	16	.itext	-	import	reconnaissance	-	GetLogicalDrives
ascii	12	.itext	-	import	reconnaissance	T1124 | System Time Discovery	GetTickCount
ascii	12	.itext	-	import	reconnaissance	-	GetDriveType
ascii	11	.itext	-	import	memory	-	LocalUnlock
ascii	8	.itext	-	import	memory	-	HeapFree
ascii	14	.itext	-	import	memory	T1055 | Process Injection	VirtualQueryEx
ascii	10	.itext	-	import	memory	-	LocalAlloc
ascii	9	.itext	-	import	memory	-	LocalFree
ascii	20	.itext	-	import	input-output	-	CopyAcceleratorTable
ascii	15	.itext	-	import	input-output	-	SwapMouseButton
ascii	15	.itext	-	import	file	-	PathQuoteSpaces
ascii	11	.itext	-	import	file	-	PathCombine
ascii	21	.itext	-	import	file	-	GetCompressedFileSize
ascii	17	.itext	-	import	file	-	CreateFileMapping
ascii	20	.itext	-	import	execution	-	GetPrivateProfileInt
ascii	11	.itext	-	import	dynamic-library	-	FreeLibrary
ascii	15	.itext	-	import	dynamic-library	-	GetModuleHandle

GetTickCounnt – for anti-analysis purposes – Sandbox/VM evasion

GetCapture – taking screenshots

Runtime functions (.pdata)

ascii	57	.pdata	-	-	-	-	AsksmaceaglyBubuPulsKaifTeasMistPeelGhisPrimChaoLyreroeno
ascii	15	.pdata	-	-	-	-	KERNEL32.MulDiv
ascii	35	.pdata	-	-	-	-	BagsSpicDollBikeAzonPoopHamsPyasmap
ascii	28	.pdata	-	-	-	-	KERNEL32.SetCurrentDirectory
ascii	11	.pdata	-	-	-	-	BardHolyawe
ascii	20	.pdata	-	-	-	-	SHLWAPI.SHFreeShared
ascii	47	.pdata	-	-	-	-	BathEftsDawnvilepughThroCymakohloverMitefuzerat
ascii	28	.pdata	-	-	-	-	SHLWAPI.PathMakeSystemFolder
ascii	41	.pdata	-	-	-	-	BemaCadsPodsWavyCedeRadsbrioOustPerefenom
ascii	21	.pdata	-	-	-	-	USER32.SetDlgItemText
ascii	33	.pdata	-	-	-	-	BullbonyaweeWaitsnugTierDriblibye
ascii	21	.pdata	-	-	-	-	KERNEL32.VirtualQuery
ascii	14	.pdata	-	-	-	-	CameValeWauler
ascii	15	.pdata	-	-	-	-	USER32.IsIconic
ascii	35	.pdata	-	-	-	-	CedeSalsshulLimyThroliraValeDonabox
ascii	18	.pdata	-	-	-	-	USER32.CreateCaret
ascii	24	.pdata	-	-	-	-	CellrotoCrudUntohighCols
ascii	19	.pdata	-	-	-	-	KERNEL32.CreateFile
ascii	25	.pdata	-	-	-	-	DenyLubeDunssawsOresvarut
ascii	26	.pdata	-	-	-	-	SHLWAPI.PathRemoveFileSpec
ascii	40	.pdata	-	-	-	-	DragRoutflusCrowPeatmownNewsyaksSerfmare
ascii	18	.pdata	-	-	-	-	USER32.DestroyIcon
ascii	11	.pdata	-	-	-	-	Dumpcotsavo
ascii	20	.pdata	-	-	-	-	USER32.SetDlgItemInt
ascii	62	.pdata	-	-	-	-	DungBadebankBangGelthoboCocaBozotsksWheyVaryShoghoseNipsCadisi
ascii	15	.pdata	-	-	-	-	USER32.EndPaint
ascii	58	.pdata	-	-	-	-	ExitRollWoodGumsgamaSloerevsWussletssinkYearZitiryesHypout
ascii	19	.pdata	-	-	-	-	USER32.GetClassInfo
ascii	15	.pdata	-	-	-	-	FociTalcileador
ascii	29	.pdata	-	-	-	-	KERNEL32.ConvertDefaultLocale
ascii	10	.pdata	-	-	-	-	GeneAilshe
ascii	22	.pdata	-	-	-	-	KERNEL32.FindFirstFile
ascii	27	.pdata	-	-	-	-	GhisGoodHowlCoonCigscateged
ascii	28	.pdata	-	-	-	-	KERNEL32.GetWindowsDirectory
ascii	47	.pdata	-	-	-	-	GimpWadsdashHoraYardSeatDeanScanscowRantKeasfib
ascii	20	.pdata	-	-	-	-	KERNEL32.LCMapString
ascii	9	.pdata	-	-	-	-	Haesourfe
ascii	21	.pdata	-	-	-	-	USER32.GetKeyNameText
ascii	35	.pdata	-	-	-	-	HoggSoonLasstwaeNapeCeilBawlscopdub
ascii	29	.pdata	-	-	-	-	KERNEL32.SystemTimeToFileTime
ascii	13	.pdata	-	-	-	-	IcontelInoway
ascii	24	.pdata	-	-	-	-	SHLWAPI.PathRemoveBlanks

ascii	22	.pdata	-	-	-	-	Vavsrubepodsjadebrooli
ascii	19	.pdata	-	-	-	-	USER32.GetUpdateRgn
ascii	15	.pdata	-	-	-	-	VeerCrawFlateel
ascii	29	.pdata	-	-	-	-	SHLWAPI.PathParseIconLocation
ascii	27	.pdata	-	-	-	-	WainMeekPinyWonkpooflaudsir
ascii	28	.pdata	-	-	-	-	KERNEL32.GetWindowsDirectory
ascii	32	.pdata	-	-	-	-	WhopTestrangrapsdebsTzarNipaYins
ascii	19	.pdata	-	-	-	-	KERNEL32.DeleteFile
ascii	8	.pdata	-	-	-	-	YeukMags
ascii	21	.pdata	-	-	-	-	KERNEL32.GlobalHandle
ascii	57	.pdata	-	-	-	-	ZetaBeduPirnhipsjailTingSrisTeleAposhuskNameHoerflagemuwo
ascii	15	.pdata	-	-	-	-	USER32.LoadIcon

Every second entry in the table above is random string without any meaning, those entries are probably function names. Threat actor used obfuscation as an anti-reverse technique. The rest of entries are some DLL's functions.

URLs/Domains:

corect.com – **not** malicious

Libraries

library (3)	duplicate (0)	flag (0)	first-thunk-original (INT)	first-thunk (IAT)	type (1)	imports (77)	group	description
SHLWAPI.dll	-	-	0x00020208	0x00020078	implicit	21	-	Shell Light-weight Utility Library
KERNEL32.dll	-	-	0x00020190	0x00020000	implicit	29	-	Windows NT BASE API Client
USER32.dll	-	-	0x00020260	0x000200D0	implicit	27	-	Multi-User Windows USER API Client Library

Protection Mechanisms and Capabilities

md5	ea039a854d20d7734c5add48f1a51c34
sha1	9615dca4c0e46b8a39de5428af7db060399230b2
sha256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169
os	windows
format	pe
arch	i386
path	C:/Users/flare/Desktop/invoice_2318362983713_823931342io.pdf.exe

ATT&CK Tactic	ATT&CK Technique
DEFENSE EVASION	Virtualization/Sandbox Evasion::System Checks T1497.001

MBC Objective	MBC Behavior
ANTI-BEHAVIORAL ANALYSIS	Virtual Machine Detection [B0009]

Capability	Namespace
reference anti-VM strings targeting VMWare	anti-analysis/anti-vm/vm-detection
resolve function by parsing PE exports	load-code/pe

Binary performs virtual machine/sandbox detection by targeting characteristic VMWare strings.

Advanced Static Analysis

Tools: cutter

Another anti-analysis technique used by malware is leverage of GetTickCount API call to determine the time which elapsed from starting operating system.

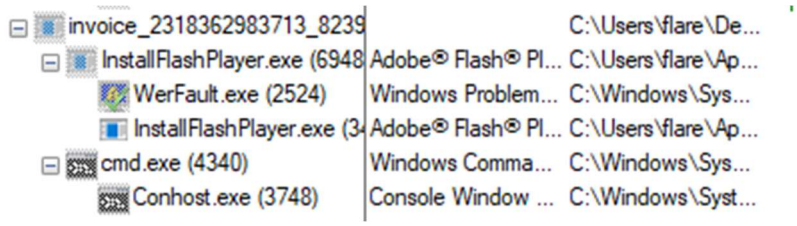
```
0x00433b1d      inc     edi
0x00433b1e      push    0x6f477369 ; 'isGo'
0x00433b23      outsd   dx, dword [esi]
0x00433b24      dec     eax
0x00433b26      outsd   dx, dword [esi]
0x00433b27      ja      0x433b95
0x00433b29      inc     ebx
0x00433b2a      outsd   dx, dword [esi]
0x00433b2b      outsd   dx, dword [esi]
0x00433b2c      outsb   dx, byte [esi]
0x00433b2d      inc     ebx
0x00433b2e      imul    esp, dword [edi + 0x73], 0x65746163
0x00433b35      add     byte fs:[bp + di + 0x45], cl
0x00433b3b      push    edx
0x00433b3c      dec     esi
0x00433b3d      inc     ebp
0x00433b3e      dec     esp
0x00433b3f      xor     esi, dword [edx]
0x00433b41      inc     edi
0x00433b43      je      0x433b9d
```

Confirmed that random strings, which were found during basic static analysis, are obfuscated function names. Above example contains assembly code of **GhisGoodHowlCoonCigscateged.**

Dynamic analysis

Tools: procmon, inetsim, Wireshark

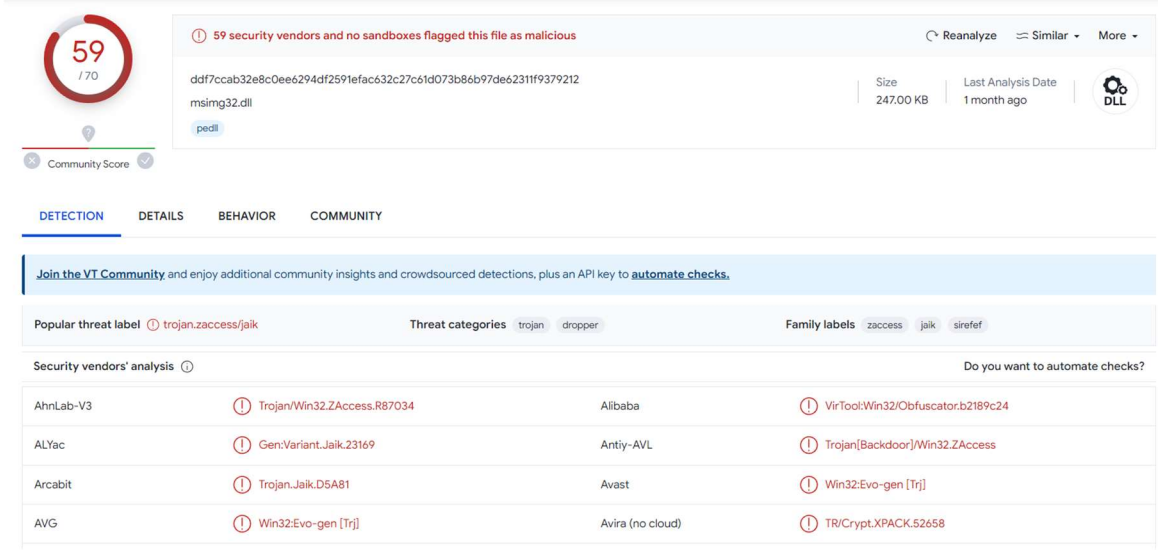
Host-based indicators



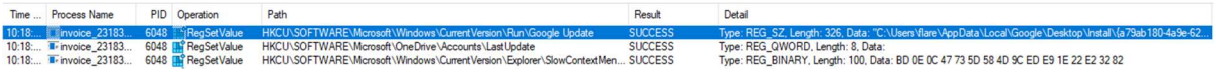
Malware attempts to install FlashPlayer and also it launches conhost.exe to execute commands.

Time ...	Process Name	PID	Operation	Path	Result	Detail
10:18:...	invoice_23183...	6048	CreateFile	C:\Users\flare\AppData\Local\Temp\msimg32.dll	SUCCESS	Desired Access: Append Data/Add Sub...
10:18:...	invoice_23183...	6048	WriteFile	C:\Users\flare\AppData\Local\Temp\msimg32.dll	SUCCESS	Offset: -1, Length: 252,928, Priority: Nor...
10:18:...	invoice_23183...	6048	CloseFile	C:\Users\flare\AppData\Local\Temp\msimg32.dll	SUCCESS	

Invoice binary dropped also a msimg32.dll file.

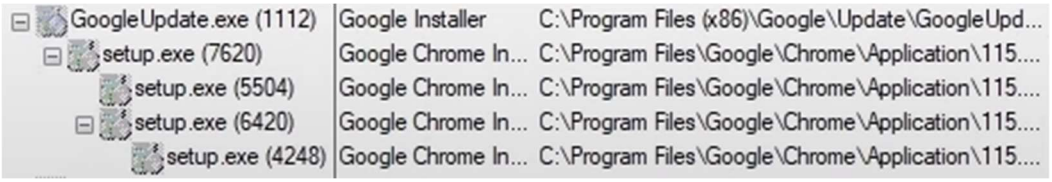


Turns out that FlashPlayer is not malicious, but dropped DLL file is a dropper/backdoor.



Time ...	Process Name	PID	Operation	Path	Result	Detail
10:18:...	invoice_23183...	6048	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Google Update	SUCCESS	Type: REG_SZ, Length: 326, Data: "C:\Users\flare\AppData\Local\Google\Desktop\Install\{a79ab180-4a9e-62...
10:18:...	invoice_23183...	6048	RegSetValue	HKCU\SOFTWARE\Microsoft\OneDrive\Accounts\LastUpdate	SUCCESS	Type: REG_QWORD, Length: 8, Data:
10:18:...	invoice_23183...	6048	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SlowContextMen...	SUCCESS	Type: REG_BINARY, Length: 100, Data: BD 0E 0C 47 73 5D 58 4D 9C ED E9 1E 22 E2 32 82

Invoice binary modified the value of Google Update register.



GoogleUpdate.exe (1112)	Google Installer	C:\Program Files (x86)\Google\Update\GoogleUpd...
setup.exe (7620)	Google Chrome In...	C:\Program Files\Google\Chrome\Application\115....
setup.exe (5504)	Google Chrome In...	C:\Program Files\Google\Chrome\Application\115....
setup.exe (6420)	Google Chrome In...	C:\Program Files\Google\Chrome\Application\115....
setup.exe (4248)	Google Chrome In...	C:\Program Files\Google\Chrome\Application\115....

The parent process of GoogleUpdate.exe is wininit.exe, so it can be stated that malware is trying to establish persistence by downloading invoice binary every time GoogleUpdate is triggered. If user deletes invoice binary it will still execute.

Network-based indicators

Wireshark · Follow TCP Stream (tcp.stream eq 0) · Ethernet

```
GET /get/flashplayer/update/current/install/install_all_win_cab_64_ax_sgn.z HTTP/1.1
User-Agent: Flash Player Seed/3.0
Host: fpdownload.macromedia.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Mon, 16 Oct 2023 08:54:01 GMT
Connection: Close
Content-Type: text/html
Content-Length: 258
Server: INetSim HTTP Server

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
</html>
```

Analysis of fpdownload.macromedia.com indicated **no threat**. No more network-based indicators were found.

Indicators of Compromise (IOCs)

YARA Rules

```
rule ZeusMalware{
      meta:
            author="ad1s0n"
            description="YARA rule against Zeus Banking Trojan (26.11.2013)"

      strings:
            $filename="invoice_2318362983713_823931342io.pdf.exe" ascii

            // suspicious functions leveraging DLLs
            $KERNEL32_CreateFileA="CellrotoCrudUntohighCols" ascii
            $KERNEL32_DeleteFileA="WhopTestrangrapsdebsTzarNipaYins" ascii
            // magic byte
            $MZ_magic_byte="MZ"

            // hexdump of some functions
            $BullbonyaweeWaitsnugTierDriblibye={42 75 6C 6C 62 6F 6E}

      condition:
            $MZ_magic_byte at 0 and $filename and $KERNEL32_CreateFileA
            or $KERNEL32_DeleteFileA
            or $BullbonyaweeWaitsnugTierDriblibye
}
```

```
C:\Users\flare\Desktop
λ yara64 zeus.yara invoice_2318362983713_823931342io.pdf.exe -s -w -p 32
ZeusMalware invoice_2318362983713_823931342io.pdf.exe
0x3176c:$KERNEL32_CreateFileA: CellrotoCrudUntohighCols
0x3201e:$KERNEL32_DeleteFileA: WhopTestrangrapsdebsTzarNipaYins
0x0:$MZ_magic_byte: MZ
0x316de:$BullbonyaweeWaitsnugTierDriblibye: 42 75 6C 6C 62 6F 6E
```

Appendix A – *calc_hashes.py*

```
import pefile
import peutils
import sys
import hashlib
import ssdeep

pe_file = sys.argv[1]
pe = pefile.PE(pe_file)
md5 = hashlib.md5(open(pe_file,'rb').read()).hexdigest()
sha256 = hashlib.sha256(open(pe_file,'rb').read()).hexdigest()
imphash = pe.get_imphash()
ssdeep_hash = ssdeep.hash(open(pe_file,'rb').read())
print(f"MD5 hash: {md5}")
print(f"SHA256 hash: {sha256}")
print(f"IMPHASH: {imphash}")
print(f"SSDEEP hash: {ssdeep_hash}")
for section in pe.sections:
    print (section.Name, "MD5 hash:", section.get_hash_md5())
    print (section.Name, "SHA256 hash:", section.get_hash_sha256())
```