

Zeus Banking Trojan: An In-Depth Examination

Introduction

The **Zeus Banking Trojan**, commonly referred to as **Zbot**, is among the most infamous and widely deployed malware designed to extract sensitive financial information. It primarily focuses on stealing online banking credentials, credit card details, and other confidential data by logging keystrokes and injecting malicious scripts into authentic banking websites. First identified in **2007**, Zeus has been linked to numerous financial fraud cases worldwide, infecting millions of devices and causing financial losses in the billions. Over the years, it has undergone multiple transformations, incorporating advanced methods to evade detection and enhance data theft efficiency.

Zeus is mainly spread through phishing emails, drive-by downloads, and deceptive attachments disguised as legitimate software updates. Once a system is compromised, the malware operates discreetly in the background, logging user keystrokes, injecting fake login forms, and transmitting stolen credentials to cybercriminal-controlled servers. Despite continuous efforts by cybersecurity experts to disrupt Zeus operations, newer versions of the malware continue to emerge, maintaining its status as a significant cybersecurity risk.

Evolution of Zeus Trojan

First discovered in **2007**, Zeus initially infected Windows-based systems by exploiting vulnerabilities to harvest banking details. Cybercriminals were able to acquire and customize the malware through underground forums, tailoring attacks to specific targets. Its modular structure allowed for easy modifications, leading to the development of multiple enhanced variants, including:

- **Gameover Zeus (2011):** A peer-to-peer version that removed centralized control, making it more challenging to dismantle.
- **Citadel (2012):** A refined version featuring improved security evasion mechanisms, better obfuscation techniques, and an intuitive botnet management interface.
- **Panda Banker (2016):** A contemporary variant aimed at financial institutions globally, leveraging web injection attacks and automated transaction fraud.

In **2011**, the original creator of Zeus, known by the alias "**Slavik**", announced their retirement and released the malware's source code, which led to the proliferation of numerous derivatives. Cybercriminals quickly capitalized on this by developing new strains that remain a persistent threat. Despite law enforcement interventions, Zeus continues to evolve and pose a serious challenge in the cybersecurity landscape.

Classification of Zeus Malware

Zeus is classified as a **Trojan malware**, specifically a **banking Trojan**, and exhibits the following characteristics:

- **Trojan Horse:** Masquerades as a legitimate file (e.g., invoice.pdf.exe) to deceive users into running it.
- **Keylogger:** Captures keystrokes to extract login credentials.
- **Form Grabbing:** Intercepts data entered into web forms before encryption is applied.
- **Man-in-the-Browser (MITB) Attack:** Modifies banking sessions by injecting malicious scripts to alter transactions.
- **Botnet Capabilities:** Enables attackers to remotely control infected systems.
- **Stealth Mechanisms:** Employs encryption, anti-sandboxing, and anti-debugging techniques to avoid detection.

Zeus Trojan's Operational Mechanism

Zeus functions through a multi-stage approach to infect systems and extract sensitive data:

Step 1: Infection

The malware typically spreads through **phishing emails, malicious downloads, and exploit kits**. Users unknowingly execute the malware, believing it to be a legitimate program or update. Social engineering tactics are often employed to lure victims into clicking harmful links that initiate automatic downloads of the Zeus payload.

Step 2: System Compromise

Once activated, Zeus:

- Alters **registry settings** to ensure persistence even after a reboot.
- Disables **antivirus programs** and firewalls to remain undetected.
- Embeds itself into system processes like explorer.exe or winlogon.exe.
- Establishes an encrypted communication channel with a **Command and Control (C2) server** for instructions and data transmission.

Step 3: Credential Theft

- Functions as a **keylogger**, recording keystrokes when users enter login details.
- Utilizes **form grabbing** to capture credentials from banking portals before encryption occurs.
- Executes **MITB attacks**, modifying web pages in real-time to deceive users into revealing security codes or bypassing multi-factor authentication.
- Alters transactions by injecting malicious scripts into legitimate banking sessions.

Step 4: Data Transmission and Fraud

- Stolen credentials are sent to a **C2 server**, where attackers exploit the information.
- Cybercriminals use these credentials for unauthorized fund transfers, creating fraudulent accounts, or selling them on dark web marketplaces.

- Some Zeus versions automate fraudulent transactions to evade traditional fraud detection mechanisms employed by banks.

Notable Attacks and Their Impact

Zeus has been involved in several high-profile cyberattacks targeting banks, corporations, and individuals. Some major incidents include:

1. 2009: FBI's "Trident Breach" Operation

- Over **74,000 computers** were compromised, leading to financial losses exceeding **\$70 million**.
- Several cybercriminals operating the Zeus botnet were arrested as a result of the investigation.

2. 2012: Targeting of British Banks

- Zeus launched sophisticated attacks against **UK banking institutions**, redirecting online transactions to fraudulent accounts.
- Exploited authentication system weaknesses to bypass security measures.

3. 2014: Gameover Zeus Botnet Takedown

- This advanced **peer-to-peer variant** infected over **500,000 systems** worldwide.
- A collaborative effort by U.S. and European law enforcement successfully disrupted the botnet's activities.

4. 2019: Panda Banker Resurgence

- A new Zeus variant, **Panda Banker**, resurfaced, targeting financial organizations in **North America, Europe, and Asia**.
- Used **malicious JavaScript injections** to extract credit card details and login credentials, impacting thousands of users.

5. 2021: Zeus-Inspired Attacks on Cryptocurrency Wallets

- Cybercriminals adapted Zeus tactics to compromise cryptocurrency exchanges and digital wallets.

- The malware altered transaction details in real-time, diverting crypto assets to attacker-controlled addresses.
- Enhanced obfuscation techniques made detection and removal significantly more difficult

Steps to Download

Step 1: Download the Malware

- Source: Download the malware from GitHub.
- Specific URL: Navigate to **https://github.com/ytisf/theZoo/tree/master/malware/Binaries/ZeusBankingVersion_26Nov2013**.
- File: Download the raw zip file (preferred for analysis).

Step 2: Browser Recommendation

- Use Microsoft Edge to download the file, as Google Chrome may block it.

Important Note: Safety Precautions

- Isolate malware samples from any network connection.
- Use a controlled environment (e.g., virtual machine or sandbox) for handling.

Warning Message

- "Do not download or execute malware samples on your main machine. Always use a virtual environment or sandbox to contain the malware and prevent accidental infection."

Step 3: Unzip the File

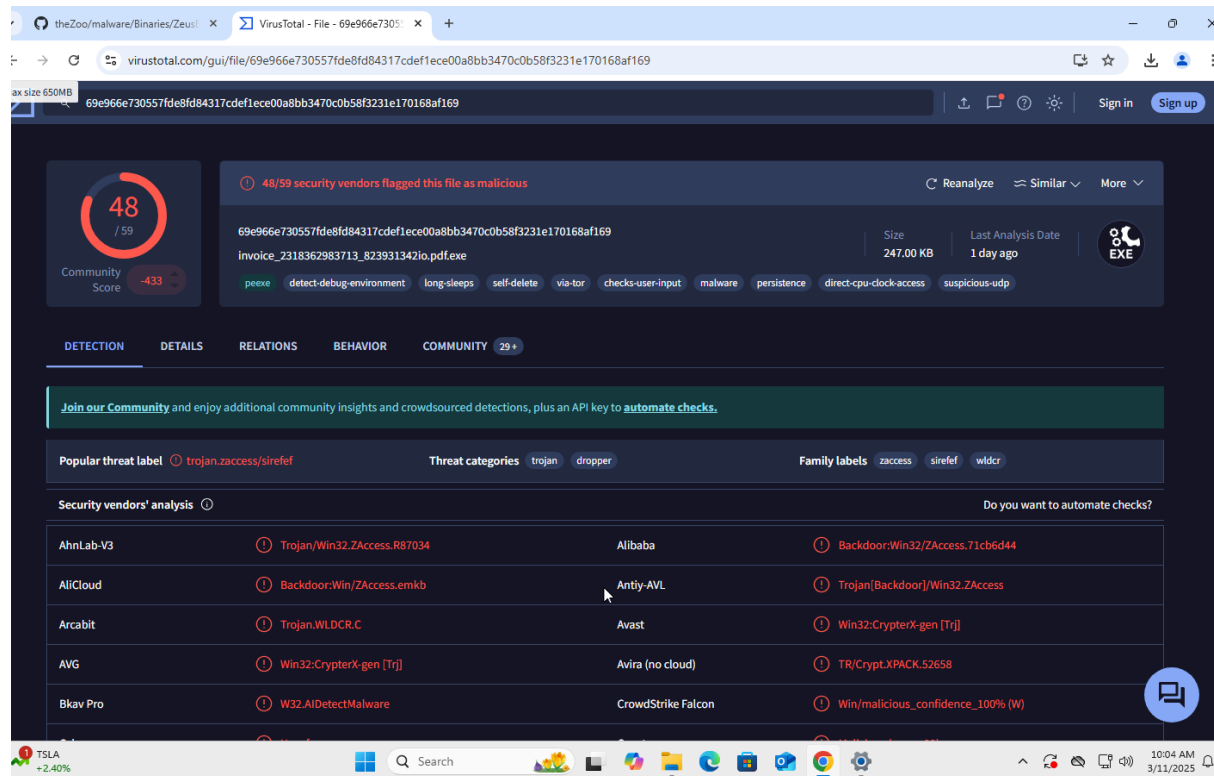
- After downloading, unzip the downloaded zip file.
- Password for unzipping: "infected".

Step 4: Upload to Sandbox

- Upload the unzipped malware file into a sandbox for analysis.

Fingerprint

- 1 . Upload the file “**invoice_2318362983713_823931342io.pdf.exe**” to virus total .
- 2 . After getting the result we can see that 48/59 security vendors flagged this file as **malicious** .



- 3 . Here we can see the hashes of the file .

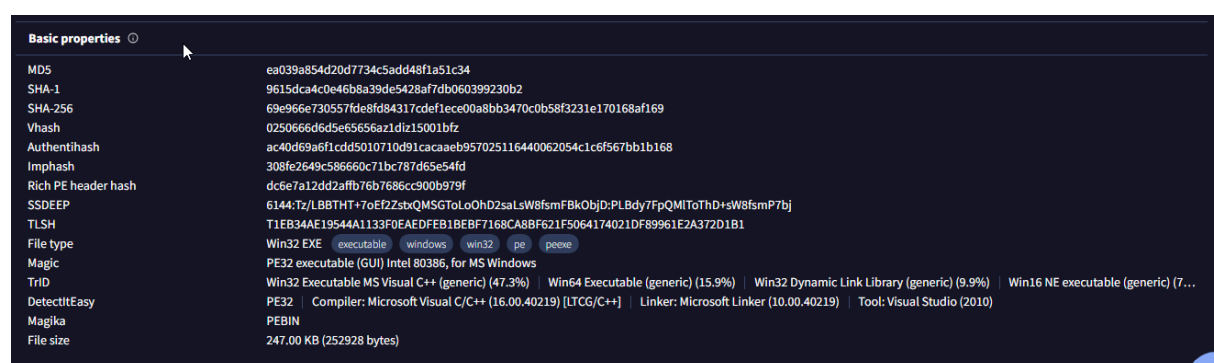
Hashes :

SHA256 : 69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169

SHA1 : 9615dca4c0e46b8a39de5428af7db060399230b2

MD5 : ea039a854d20d7734c5add48f1a51c34

Filename : invoice_2318362983713_823931342io.pdf.exe



Static Analysis

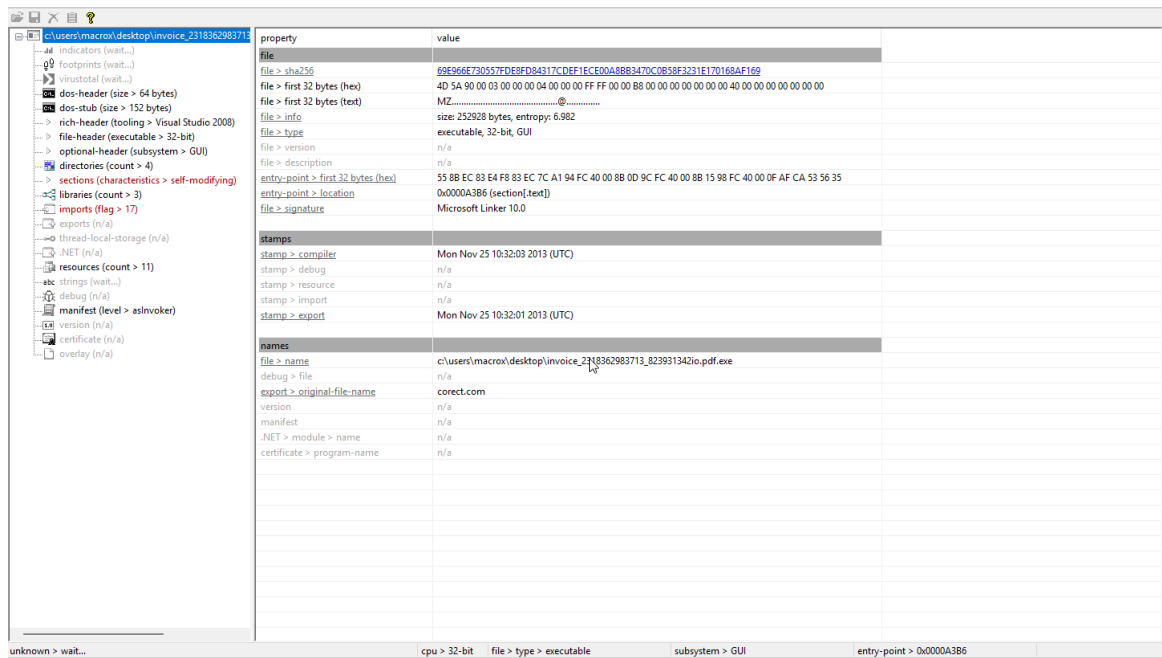
1 . Open the file “**invoice_2318362983713_823931342io.pdf.exe**” in peStudio.

Overview

- **pestudio** is a static malware analysis tool.
- **Purpose:** Examines Windows executables (PE files) without executing them.

Key Features :

- Detects suspicious indicators:
 - Malware behavior.
 - Anti-analysis tricks.
- Extracts components of a PE file:
 - Imports.
 - Exports.
 - Resources.
 - Sections.
- Scans the file against VirusTotal to identify known threats.
- Identifies:
 - Strings.
 - Libraries.
 - API calls used within the malware.
- Supports malware triage without requiring execution of the file.



- 2 . Go to the sections(self-modifying) . Here we can see the size of the file .
- 3 . By observing the raw-size and virtual-size , we can that this program is unpacked

property	value
section	section[0]
name	.text
section > 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)

- 4 . Now go to strings section .
- 5 . Here we can see all the strings . Here we can observe some function of the program ,which may be useful and tells us about the program

pestudio 9.53 - Malware Initial Assessment - www.wintor.com - [c:\users\grant\desktop\invoice_2318362983713_82393134io.pdf.exe]

file settings about

	location	flag (17)	label (110)	group (11)	technique (7)	value (1416)
indicators (directory > invalid)	.text	×	import	windowing	-	AllowSetForegroundWindow
footprints (20)	.text	×	import	reconnaissance	-	GetEnvironmentVariable
kernel32.dll (error)	.text	×	import	reconnaissance	-	GetEnvironmentVariable
dos-header (64 bytes)	.text	×	import	input-output	-	VkKeyScan
dos-stub (152 bytes)	.text	×	import	input-output	T1056 Input Capture	GetAsyncKeyState
rich-header (Visual Studio)	.text	×	import	file	-	PathRenameExtension
file-header (intel-386)	.text	×	import	file	-	WriteFile
optional-header (GUI)	.text	×	import	file	T1083 File and Directory Discovery	FindNextFile
directories (invalid)	.text	×	import	execution	-	GetCurrentThread
sections (self-modifying)	.text	×	-	execution	T1106 Execution through API	WinExec
libraries (3)	.text	×	import	data-exchange	-	GlobalAddAtom
imports (flag)	.text	×	import	data-exchange	T1115 Clipboard Data	GetClipboardOwner
exports (n/a)	.text	×	import	data-exchange	T1115 Clipboard Data	GetClipboardData
thread-local-storage (n/a)	.text	×	import	data-exchange	T1115 Clipboard Data	EnumClipboardFormats
.NET (n/a)	.text	×	import	data-exchange	-	PdeQueryNextServer
resources (11)	.text	×	import	console	-	GetConsoleAliasExesLength
strings (size)	.text	×	import	-	-	GetCurrentDirectory
debug (n/a)	.text	-	import	windowing	-	CallWindowProc
manifest (asInvoker)	.text	-	import	windowing	-	UpdateWindow
version (n/a)	.text	-	import	windowing	-	GetCapture
certificate (n/a)	.text	-	import	windowing	-	IsWindowEnabled
overlay (n/a)	.text	-	import	windowing	T1010 Window Discovery	GetWindowTextLength
	.text	-	import	synchronization	-	DeleteCriticalSection
	.text	-	import	resource	-	SizeofResource
	.text	-	import	reconnaissance	-	GetLogicalDrives
	.text	-	import	reconnaissance	T1124 System Time Discovery	GetTickCount
	.text	-	import	reconnaissance	-	GetDriveType
	.text	-	import	memory	-	LocalUnlock
	.text	-	import	memory	-	HeapFree
	.text	-	import	memory	T1055 Process Injection	VirtualQueryEx

sha256: 69E966E730557FDE8FD84317CDEF1CE0A8B83470C0B5F3231E170168AF169 cpu: 32-bit file-type: executable subsystem: GUI entry-point: 0x0000A3B6

6 . After scrolling down , here we can see some dlls .

Here we can observe , before the dll their is something randomly is written . May be this might be the function names , which are obfuscated .

encoding (2)	size (bytes)	location	flag (17)	label (110)	group (11)	technique (7)	value (1416)
ascii	57	pdata	-	-	-	-	AsksmaceaglyBubuPulsKaifTeasMstPeelGhisPrimChaolyroeno
ascii	15	pdata	-	-	-	-	KERNEL32.MulDiv
ascii	35	pdata	-	-	-	-	BaggsSpicDollBikeAzonPoopHamsPyasmap
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	-	-	-	-	BernaCadsPodsWavyCedeRadsbriOustPerefenom
ascii	21	pdata	-	-	-	-	USER32.SetDlgItemText
ascii	33	pdata	-	-	-	-	BullbonyaweeWaitsnugTierDnlibbye
ascii	21	pdata	-	-	-	-	KERNEL32.VirtualQuery
ascii	14	pdata	-	-	-	-	CameValeWauler
ascii	15	pdata	-	-	-	-	USER32.IsIconic
ascii	35	pdata	-	-	-	-	CedeSalsshulLirnyThroliraValeDonabox
ascii	18	pdata	-	-	-	-	USER32.CreateCaret
ascii	24	pdata	-	-	-	-	CellrotoCudUntohighCols
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	-	-	-	-	DungBadebankBangGelthoboCocaBozotsksWheyVaryShoghoseNip
ascii	15	pdata	-	-	-	-	USER32.EndPaint
ascii	58	pdata	-	-	-	-	ExitRollWoodGumsgamsSloerevsWusslettsinkYearZtiryesHypout
ascii	19	pdata	-	-	-	-	USER32.GetClassInfo
ascii	15	pdata	-	-	-	-	FocTalcileador
ascii	29	pdata	-	-	-	-	KERNEL32.ConvertDefaultLocale
ascii	10	pdata	-	-	-	-	GeneAilshhe
ascii	22	pdata	-	-	-	-	KERNEL32.FindFirstFile
ascii	27	pdata	-	-	-	-	GhisGoodHowlCoonCigscteged
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	-	-	-	-	HoggSoonLasstwaeNapeCeilBawlsopdub
ascii	29	pdata	-	-	-	-	KERNEL32.SystemTimeToFileTime
ascii	13	pdata	-	-	-	-	Icontellnoway
ascii	24	pdata	-	-	-	-	SHLWAPI.PathRemoveBlanks
ascii	32	pdata	-	-	-	-	ImidslatJokyCombdnubChefBilkSale
ascii	21	pdata	-	-	-	-	USER32.GetShellWindow
ascii	56	pdata	-	-	-	-	IzararfsFlamWostAirsconsMouefemelalPoretweeSacsOxidMinx

0C0B5F3231E170168AF169 cpu: 32-bit file-type: executable subsystem: GUI entry-point: 0x0000A3B6

7 . Here we can see a magic number MZ , which means it is an windows executable file .


```
strings.txt - Notepad
File Edit Format View Help

FLARE FLOSS RESULTS (version v2.3.0-0-g037fc4b)

+-----+
| file path          | invoice_2318362983713_823931342io.pdf.exe |
| extracted strings  |                                              |
|   static strings   | 791                                         |
|   stack strings    | 2                                           |
|   tight strings    | 0                                           |
|   decoded strings  | 66                                         |
+-----+

FLOSS STATIC STRINGS

+-----+
| FLOSS STATIC STRINGS: ASCII (790) |
+-----+

!This program cannot be run in DOS mode.
Rich
.text
.data
.itext
.pdata
.rsrc
@.reloc
jrjy
SVW;
9ATt
SVW5
Km}0+
==#NEw1zUTMNNONvJ2NRo+OjJzndo4djQeYoJlUoqiQeJ3gyK8RpqRQ9GyWDDzeto1mnR7tLSM7SL
cT6b
<
```

13 . Now lets look is their anything with “.com” , we can find “corect.com”

15 . Now open Windows PowerShell .

16 . Now lets use **capa**

capa is a static malware analysis tool developed by MANDIANT that helps identify capabilities in executable files. It is particularly useful for analyzing malware and reverse-engineered binaries.

Key Features of capa:

- Identifies Capabilities:** Detects functionalities like file manipulation, process injection, network communication, and more.
- Static Analysis:** Analyzes binaries without executing them, making it safer than dynamic analysis.
- Rule-Based Detection:** Uses YARA-like rules to detect common malware behaviors.
- Supports Multiple Formats:** Works with PE (Windows executables), ELF (Linux executables), and shellcode.

•Integrates with IDA Pro, Ghidra, and Radare2: Helps in reverse engineering workflows.

17 . Enter this command :

capa .\invoice_2318362983713_823931342io.pdf.exe

```
Select Windows PowerShell
PS C:\Users\Grant\Desktop> capa .\invoice_2318362983713_823931342io.pdf.exe
matching: 100%|
+-----+-----+
| md5          | ea039a854d20d7734c5add48f1a51c34 |
| sha1         | 9615dca4c0e46b8a39de5428af7db060399230b2 |
| sha256       | 69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169 |
| os           | windows |
| format       | pe |
| arch         | i386 |
| path         | 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 |
| contain a resource (.rsrc) section | executable/pe/section/rsrsc |
| resolve function by parsing PE exports | load-code/pe |
+-----+-----+
PS C:\Users\Grant\Desktop>
```

18 . Now for more information Enter this command :

capa -v .\invoice_2318362983713_823931342io.pdf.exe

```
Windows PowerShell
PS C:\Users\Grant\Desktop> capa -v .\invoice_2318362983713_823931342io.pdf.exe
matching: 100%|
md5          ea039a854d20d7734c5add48f1a51c34
sha1         9615dca4c0e46b8a39de5428af7db060399230b2
sha256       69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169
path         invoice_2318362983713_823931342io.pdf.exe
timestamp    2023-08-09 16:55:01.606232
capa version  5.1.0
os           windows
format       pe
arch         i386
extractor    VivisectFeatureExtractor
base address 0x400000
rules        C:\Users\Grant\AppData\Local\Temp\_MEI79962\rules
function count      80
library function count 1
total feature count 9498

reference anti-VM strings targeting VMWare
namespace anti-analysis/anti-vm/vm-detection
scope      file

contain a resource (.rsrc) section
namespace executable/pe/section/rsrsc
scope      file

resolve function by parsing PE exports
namespace load-code/pe
scope      function
matches    0x40A3B6
```

19 . U can use this command for more verbose information :

capa -vv .\invoice_2318362983713_823931342io.pdf.exe

20 . This are the API Calls we got .

API Calls :

AllowSetForegroundWindow

GetEnvironmentVariable

GetEnvironmentVariable

VkKeyScan

GetAsyncKeyState

PathRenameExtension

WriteFile

FindNextFile

GetCurrentThread

WinExec

GlobalAddAtom
GetClipboardOwner
GetClipboardData
EnumClipboardFormats
DdeQueryNextServer
GetConsoleAliasExesLength
SetCurrentDirectory.
CallWindowProc
UpdateWindow
GetCapture
IsWindowEnabled
GetWindowTextLength
DeleteCriticalSection
SizeofResource
GetLogicalDrives
System Time
GetTickCount
GetDriveType
LocalUnlock
HeapFree
VirtualQueryEx
LocalAlloc LocalFree
CopyAcceleratorTable
SwapMouseButton PathQuoteSpaces
PathCombine
GetCompressedFileSize
CreateFileMapping
GetPrivateProfileInt

FreeLibrary

GetModuleHandle

Suspected Function Calls :

AsksmaceaglyBubuPulsKaifTeasMistPeelGhisPrimChaoLyroeroeno

KERNEL32.MulDiv

BagsSpicDollBikeAzonPoopHamsPyasmap

KERNEL32.SetCurrentDirectory

BardHolyawe

SHLWAPI.SHFreeShared

BathEfTsDawnvilepughThroCymakohloverMitefuzerat

SHLWAPI.PathMakeSystemFolder

BemaCadsPodsWavyCedeRadsbrioOustPerefenom

USER32.SetDlgItemText

BullbonyaweeWaitsnug TierDriblibye

KERNEL32.VirtualQuery

CameValeWauler

USER32. IsIonic

CedeSalsshulLimy ThroliraValeDonabox

USER32.CreateCaret

CellrotoCrudUntohighCols

KERNEL32.CreateFile

DenyLubeDunssawsOresvarut

SHLWAPI.PathRemoveFileSpec

DragRoutflusCrowPeatmownNewsyaksSerfmare

USER32.DestroyIcon

Dumpcotsavo

USER32.SetDlgItemInt

DungBadebankBangGelthoboCocaBozotsksWheyVaryShoghoseNipsCadisi

USER32.EndPaint

ExitRollWoodGumsgamaSloerevsWussletssinkYearZitiryesHypout

USER32.GetClassInfo

FociTalcileador

KERNEL32.ConvertDefaultLocale

GeneAilshe

KERNEL32.FindFirstFile

GhisGoodHowlCoonCigscateged

KERNEL32.GetWindowsDirectory

GimpWadsdashHoraYardSeatDeanScanscowRantKeasfib

KERNEL32.LCMapString

Haesourte

USER32.GetKeyNameText

HoggSoonLasstwaeNapeCeilBawlscopdub

KERNEL32.SystemTimeToFileTime

Icontellnoway

SHLWAPI.PathRemoveBlanks

ImidslatJokyCombdрубChefBilkSale

USER32.GetShellWindow

IzararfsFlamWostAirsconsMouefemelallPoretweeSacsOxidMinx

SHLWAPI.PathAddExtension

JabsNaveFateLariManyLeeksecshiesBawlwoo

KERNEL32.CreateloCompletionPort

KatsDoreOmerBetsKoraKeef

KERNEL32.GetShortPathName

KineChamLows

KERNEL32.SetCurrentDirectory

LeerMiff

KERNEL32.LeaveCriticalSection

MaarSectFiscNextMattbamsErasnimstoeaBadshon

USER32.GetClassInfo
MarkMokeOsesShwaSkegpornlimemim
KERNEL32.GetStartupInfo
MeanOrrabirogirtWorkGawpSassPirnVinoLotaPledEidefe
SHLWAPI.SHLockShared
NextLoveOralwanySurfhm
KERNEL32.VerSetConditionMask
NisiBoyolineJiaoverlyObiaowedblamHaetMaulweensky,
SHLWAPI.PathCanonicalize
OastcabskamiKartDumbInksSomsMass
KERNEL32.SetCurrentDirectory
PeckQuinFillrillsaw
KERNEL32.GetThreadPriority
RamilimaputtHastJobs
KERNEL32.FindNextFile
RemsSlaySoreAnoaaxalbuffusesemeuMapsvogaHangLoud/
SHLWAPI.PathMakePretty
RidsFineZingMickMomsdue
USER32.GetMonitorInfo
SeminersdoloosenYaginobox
SHLWAPI.PathIsLFNFileSpec
SiretomsbritGrewlckyNapaLumsBoaren
KERNEL32.OpenFileMapping
SlabKitsSlayseptPfftjiffSabsdeskOafsNowtMemsKirnKepiMiffDunt
KERNEL32.OpenSemaphore
SoldKartAgueiliaRushWauldhal
SHLWAPI.PathIsUNC
SuitplieGunsMaidBaitFeus/iaotodycol/AlbsLuneToyspe
USER32.GetProp

SungActaKopsMaarposyparefuzedeck

SHLWAPI.PathIsDirectory

ToeaTailecusGeesSoliCadeSpueEndsPlaykaphall

SHLWAPI.PathRemoveArgs

Vavsrubepodsjadebrooli

USER32.GetUpdateRgn

VeerCrawFlateel

SHLWAPI.PathParselconLocation

WainMeekPinyWonkpooflaudsir

KERNEL32.GetWindowsDirectory

WhopTestrangrapsdebsTzarNipaYins

KERNEL32.DeleteFile

YeukMags

KERNEL32.GlobalHandle

ZetaBeduPirnhipsjailTingSrisTeleAposhuskNameHoerflagemuwo

USER32.LoadIcon

Dynamic Analysis

1. Run the Process Monitor -Sysinternals

Process Monitor (ProcMon) is a real-time system monitoring tool from Microsoft that tracks registry, file system, process, and network activity on Windows.

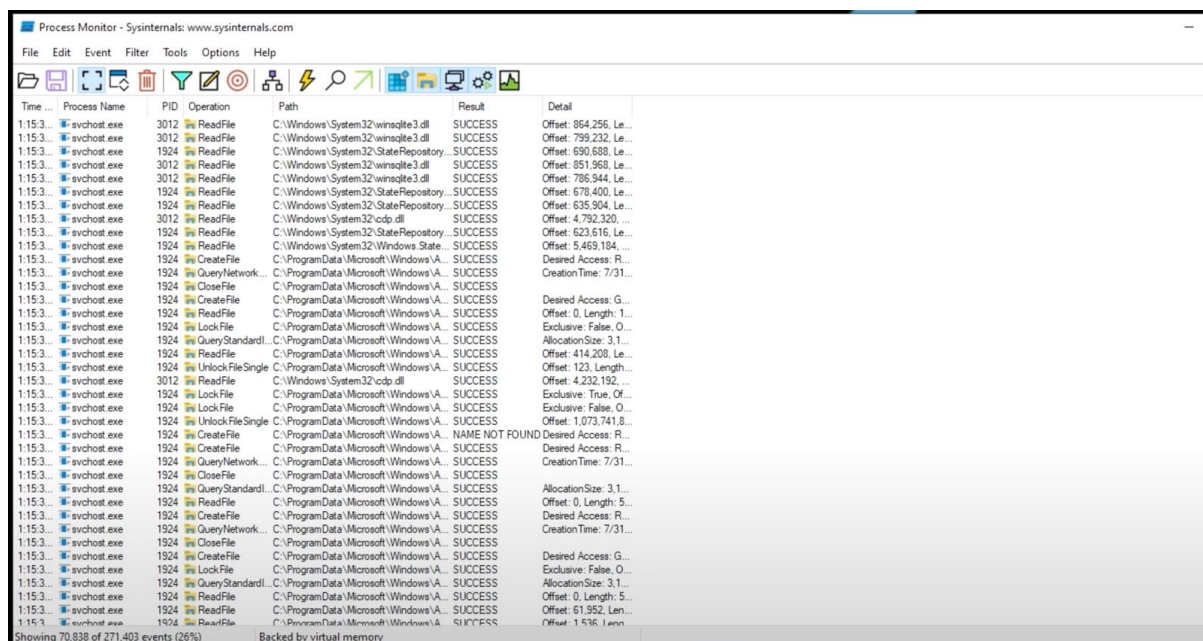
Key Features:

Monitors real-time process execution (PID, command-line, paths).

Tracks registry changes, file system activity, and network connections.

Detects malware behaviors (persistence mechanisms, DLL injection, etc.).

Advanced filtering and logging for deep forensic analysis.



2. Now open run REMnux , and configure INetSim with command :

Command : inetsim

REMnux is a Linux distribution designed for malware analysis and reverse engineering. It provides a collection of tools for static, dynamic, and memory analysis of malicious software.

Key Features:

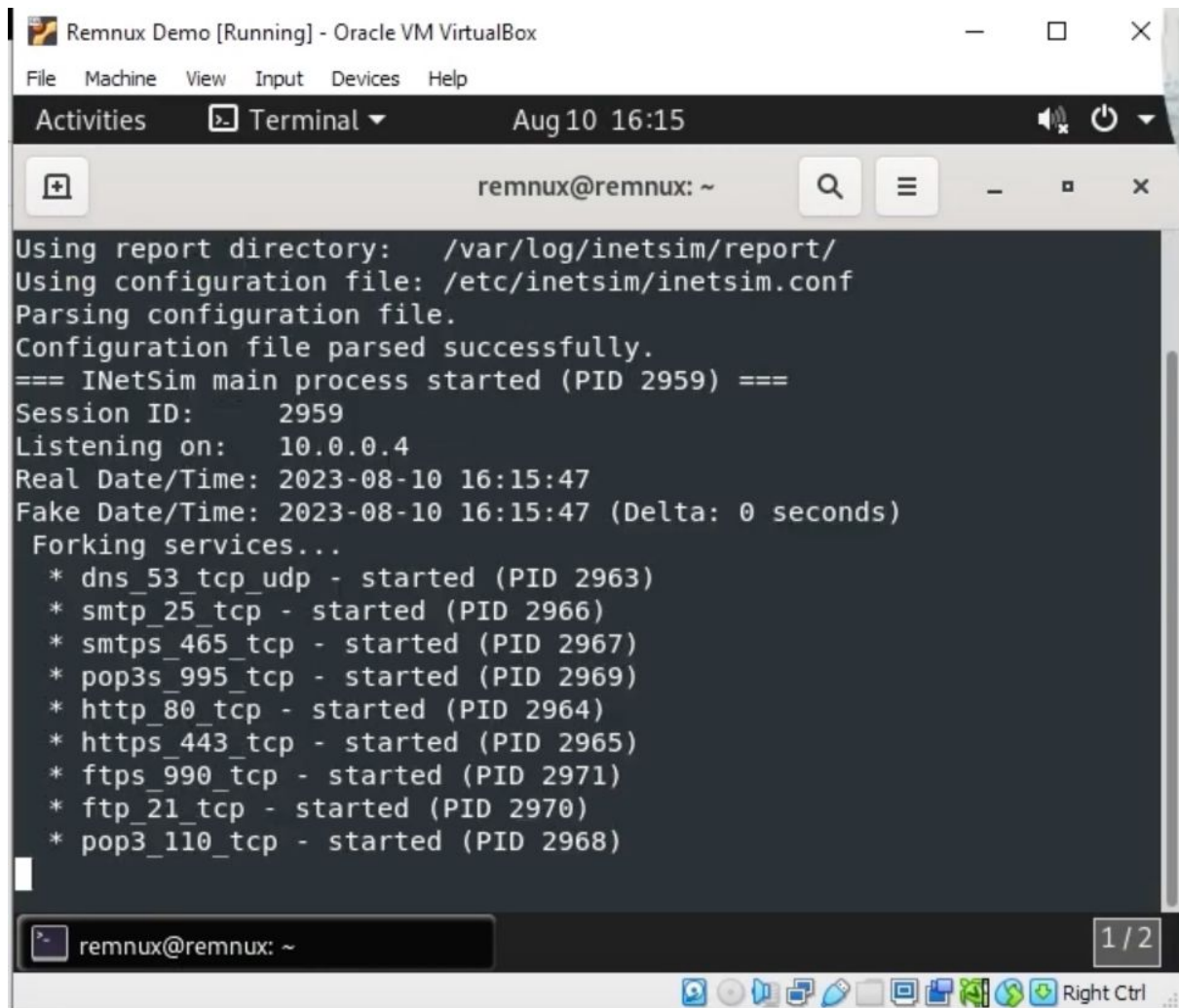
Pre-installed tools for static analysis (capa, FLOSS, pestudio).

Dynamic analysis tools (Cuckoo Sandbox, Process Monitor, Wireshark).

Memory forensics (Volatility, Rekall).

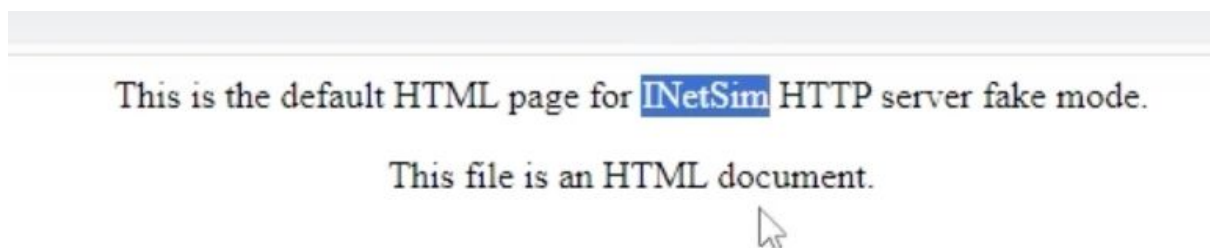
Network analysis (Suricata, Zeek, INetSim for malware sandboxing).

Lightweight and works as a VM or Docker container



```
Remnux Demo [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 10 16:15
remnux@remnux: ~
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 2959) ===
Session ID: 2959
Listening on: 10.0.0.4
Real Date/Time: 2023-08-10 16:15:47
Fake Date/Time: 2023-08-10 16:15:47 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 2963)
* smtp_25_tcp - started (PID 2966)
* smtps_465_tcp - started (PID 2967)
* pop3s_995_tcp - started (PID 2969)
* http_80_tcp - started (PID 2964)
* https_443_tcp - started (PID 2965)
* ftps_990_tcp - started (PID 2971)
* ftp_21_tcp - started (PID 2970)
* pop3_110_tcp - started (PID 2968)
```

3 . To conform that **INetSim** is working , go to browser and search google.com



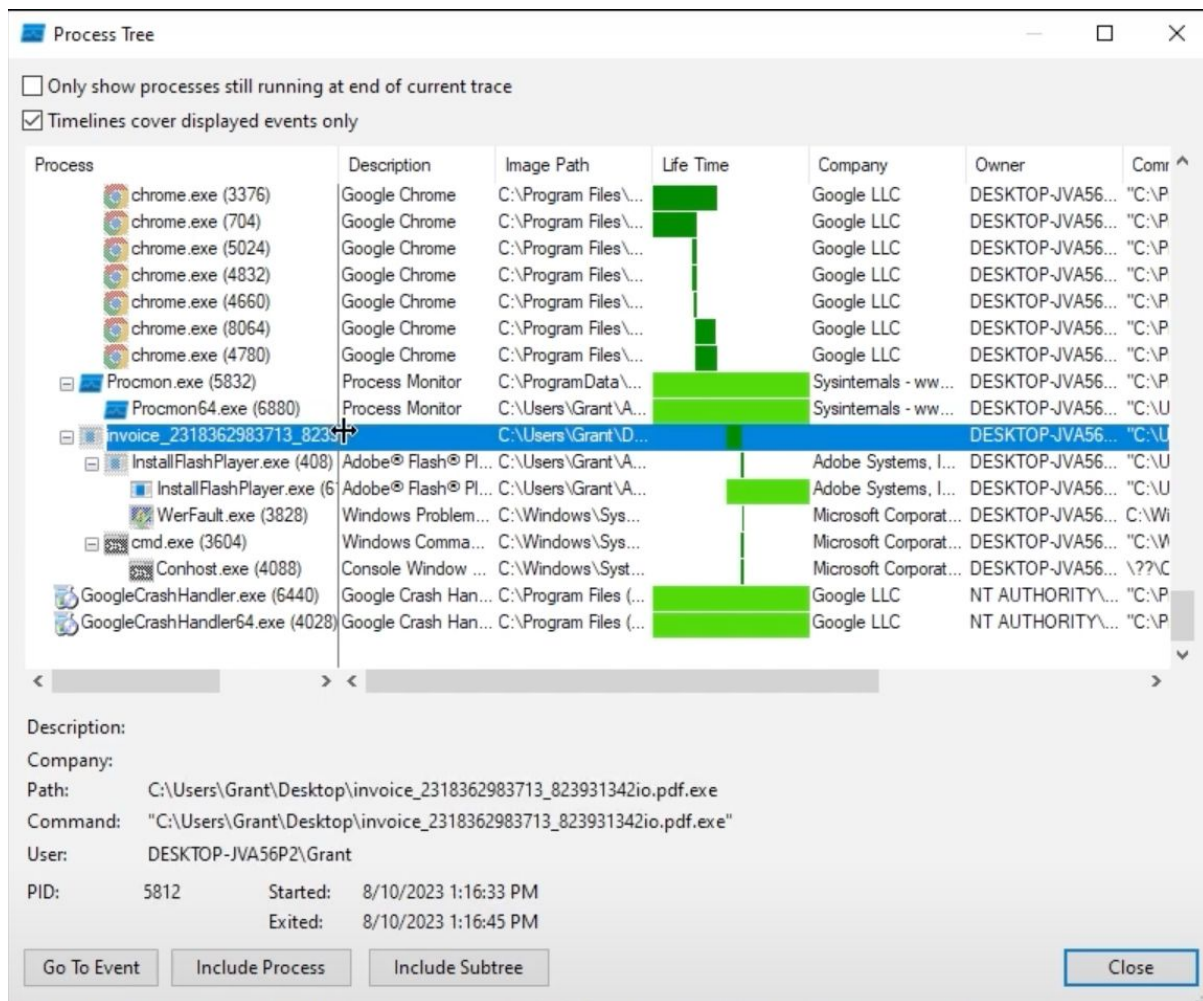
4 . Now double-click on “**invoice_2318362983713_823931342io.pdf.exe**” file , we can see that is say that do you want to install Adobe Flash Player .



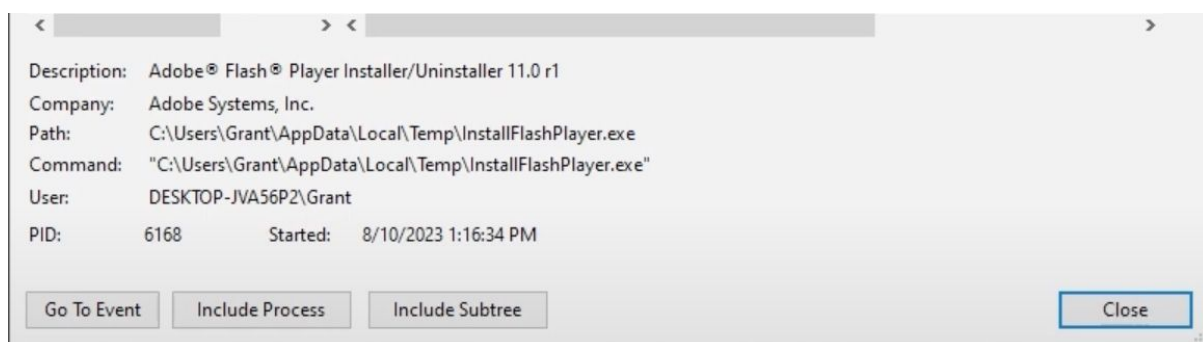
5. Click on Yes.

6 . Here in Process Monitor , under Process Tree we can see that
“**invoice_2318362983713_823931342io.pdf.exe**” file is running .

And we can **see InstallFlashPlayer.exe** is also running.



7 . Here we can see InstallFlashPlayer.exe is stored in Temp Folder.

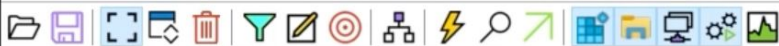


8. Now go to filter and select Process Name , select contains , and type invoice , and Click on Add.

9 . Now go to filter and select Path , select contains , and give the path of Temp folder .

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help



Time ...	Process Name	PID	Operation	Path	Result	Detail
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: A...
1:16:3...	invoice_23183...	5812	WriteFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Offset: -1, Length: ...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: A...
1:16:3...	invoice_23183...	5812	WriteFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Offset: -1, Length: ...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: R...
1:16:3...	invoice_23183...	5812	QueryRemotePr...	C:\Users\Grant\AppData\Local\Temp\...	INVALID PARAM...	
1:16:3...	invoice_23183...	5812	QueryDirectory	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	FileInformationClas...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: G...
1:16:3...	invoice_23183...	5812	FileSystemControl	C:\Users\Grant\AppData\Local\Temp\...	CANCELLED	Control: FSCTL_R...
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: R...
1:16:3...	invoice_23183...	5812	QueryBasicInfor...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	CreationTime: 8/10...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: G...
1:16:3...	invoice_23183...	5812	WriteFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Offset: 0, Length: 9...
1:16:3...	invoice_23183...	5812	SetEndOfFileInf...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	EndOfFile: 89,248
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	FILE LOCKED WI...	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	QueryStandardI...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	AllocationSize: 90...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: R...
1:16:3...	invoice_23183...	5812	QueryBasicInfor...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	CreationTime: 8/10...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: G...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	FILE LOCKED WI...	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: R...
1:16:3...	invoice_23183...	5812	QueryBasicInfor...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	CreationTime: 8/10...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: G...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	FILE LOCKED WI...	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CreateFileMapp...	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	SyncType: SyncTy...
1:16:3...	invoice_23183...	5812	CloseFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	
1:16:3...	invoice_23183...	5812	CreateFile	C:\Users\Grant\AppData\Local\Temp\...	SUCCESS	Desired Access: R...

Showing 370 of 809,729 events (0.045%) Backed by virtual memory

10 . Now go to filter and select Operation , select contains , type RegSetValue and Click on Add . And remove Path filter.

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

Time ...	Process Name	PID	Operation	Path	Result	Detail
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Google Update	SUCCESS	Type: REG_SZ, Le...
1:16.3...	invoice_23183...	5812	RegSet...	HKCU\SOFTWARE\Microsoft\OneDrive\Accounts\Last Update	SUCCESS	Type: REG_QWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SlowContextMenuEntries	SUCCESS	Type: REG_BINA...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SlowContextMenuEntries	SUCCESS	Type: REG_BINA...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWO...
1:16.3...	invoice_23183...	5812	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWO...
1:16.4...	invoice_23183...	5812	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC3475	SUCCESS	Type: REG_BINA...
1:16.4...	invoice_23183...	5812	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC1C75	SUCCESS	Type: REG_BINA...
1:16.4...	invoice_23183...	5812	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\Data\418A073AA3BC1C75	SUCCESS	Type: REG_BINA...

11 . Now lets see their is any Network Based Indicators .

12 . Open Wireshark , now lets collect packets on Ethernet .

13 . Now start the Wireshark tool and execute

“invoice_2318362983713_823931342io.pdf.exe” file .

Capturing from Ethernet

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter: <Ctrl>->

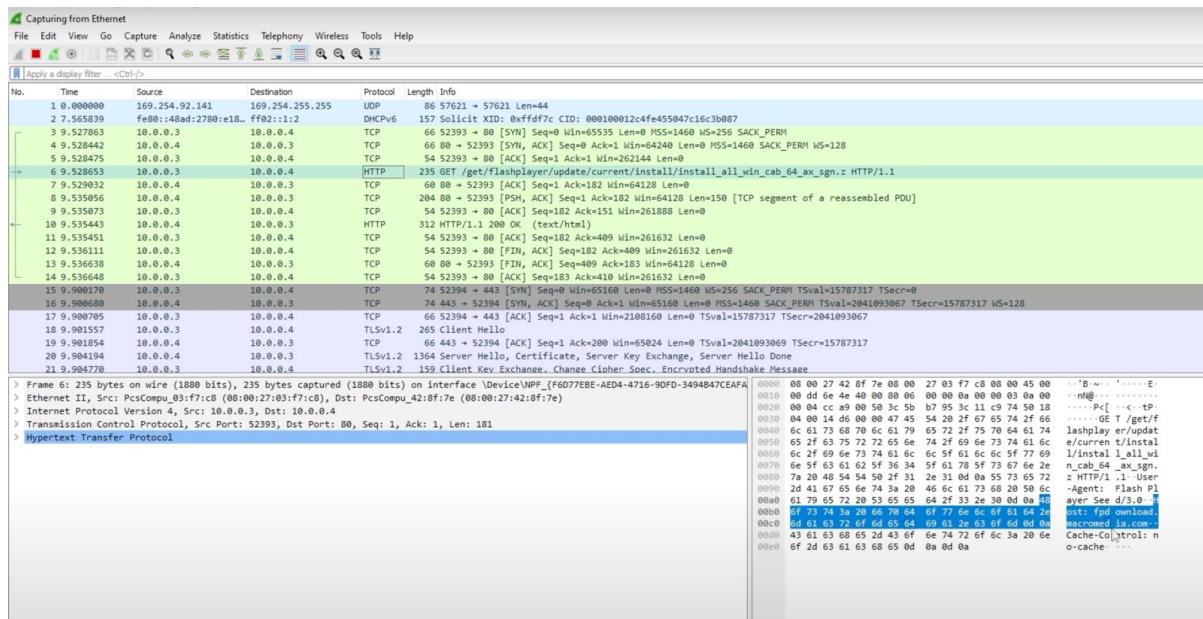
No.	Time	Source	Destination	Protocol	Length	Info
72	10.238167	10.0.0.4	10.0.0.3	TLSv1.2	1364	Server Hello, Certificate, Server Key Exchange, Server Hello Done
73	10.238717	10.0.0.3	10.0.0.4	TLSv1.2	159	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
74	10.238979	10.0.0.4	10.0.0.3	TCP	66	443 → 52398 [ACK] Seq=1299 Ack=311 Win=65024 Len=0 TSval=2041093406 TSecr=15787655
75	10.239452	10.0.0.4	10.0.0.3	TLSv1.2	292	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
76	10.239720	10.0.0.3	10.0.0.4	TCP	66	52398 → 443 [FIN, ACK] Seq=311 Ack=1525 Win=263424 Len=0 TSval=15787656 TSecr=2041093406
77	10.241821	10.0.0.4	10.0.0.3	TLSv1.2	97	Encrypted Alert
78	10.241821	10.0.0.4	10.0.0.3	TCP	66	443 → 52398 [FIN, ACK] Seq=1556 Ack=312 Win=65024 Len=0 TSval=2041093409 TSecr=15787656
79	10.242137	10.0.0.3	10.0.0.4	TCP	54	52398 → 443 [RST, ACK] Seq=312 Ack=1556 Win=0 Len=0
80	10.241939	10.0.0.3	10.0.0.4	TCP	54	52398 → 443 [RST] Seq=312 Win=0 Len=0
81	10.268310	10.0.0.3	10.0.0.4	TCP	74	52399 → 443 [SYN] Seq=0 Win=65160 Len=0 MSS=1460 WS=256 SACK_PERM TSval=15787658 TSecr=0
82	10.268652	10.0.0.4	10.0.0.3	TCP	74	443 → 52399 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2041093427 TSecr=15787658 WS=128
83	10.268671	10.0.0.3	10.0.0.4	TCP	66	52399 → 443 [ACK] Seq=1 Ack=1 Win=263424 Len=0 TSval=15787677 TSecr=2041093427
84	10.268891	10.0.0.3	10.0.0.4	TLSv1.2	283	Client Hello
85	10.261064	10.0.0.4	10.0.0.3	TCP	66	443 → 52399 [ACK] Seq=1 Ack=218 Win=65024 Len=0 TSval=2041093428 TSecr=15787677
86	10.264188	10.0.0.4	10.0.0.3	TLSv1.2	1364	Server Hello, Certificate, Server Key Exchange, Server Hello Done
87	10.264726	10.0.0.3	10.0.0.4	TLSv1.2	159	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
88	10.265084	10.0.0.4	10.0.0.3	TCP	66	443 → 52399 [ACK] Seq=1299 Ack=311 Win=65024 Len=0 TSval=2041093432 TSecr=15787681
89	10.265685	10.0.0.4	10.0.0.3	TLSv1.2	292	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
90	10.265954	10.0.0.3	10.0.0.4	TCP	66	52399 → 443 [FIN, ACK] Seq=311 Ack=1525 Win=263424 Len=0 TSval=15787682 TSecr=2041093432
91	10.268481	10.0.0.4	10.0.0.3	TLSv1.2	97	Encrypted Alert
92	10.268495	10.0.0.3	10.0.0.4	TCP	54	52399 → 443 [RST, ACK] Seq=312 Ack=1556 Win=0 Len=0
93	10.268676	10.0.0.4	10.0.0.3	TCP	66	443 → 52399 [FIN, ACK] Seq=1556 Ack=312 Win=65024 Len=0 TSval=2041093435 TSecr=15787682
94	10.268680	10.0.0.3	10.0.0.4	TCP	54	52399 → 443 [RST] Seq=312 Win=0 Len=0

> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{F6D77EBE-AED4-4716-90FD-3494847CEAFA}, 1
 > Ethernet II, Src: 08:00:27:00:00:0c (08:00:27:00:00:0c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 > Internet Protocol Version 4, Src: 169.254.92.141, Dst: 169.254.255.255
 > User Datagram Protocol, Src Port: 57621, Dst Port: 57621
 > Data (44 bytes)

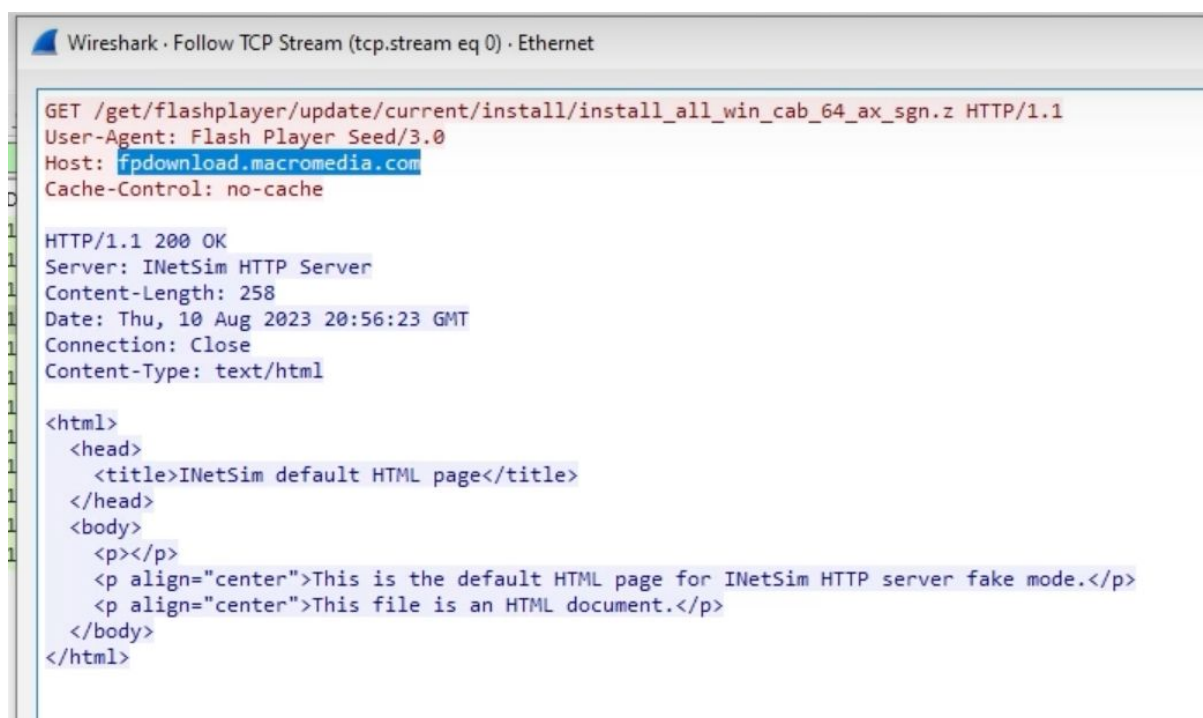
```

0000  ff ff ff ff ff ff 0a 00 27 00 0c 00 00 45 00  ....E
0010  00 00 00 52 e0 00 00 11 37 3b a9 fe 5c bd 52 66  ....HR.....Tj: \
0020  ff ff e1 15 e1 15 00 34 4e 0e 53 70 6f 74 55 64  ....4 N Spottid
0030  70 30 60 23 97 7d 16 2e 03 94 00 01 00 04 48 95  p0# :), .....H
0040  c2 03 6a 73 9f 97 a1 05 47 b5 4e b6 00 00 0f d3  ..js.....G N....
0050  f7 3f 63 3c ed fb  ....Tcc....
  
```

14. Here we can see a HTTP GET request . Now click on it .



15 . Now right click on it and click on follow , then TCP Stream



16 . Go to virus total , go to URL and enter the Host name we got .

Host : fpdfownload.macromedia.com

Did you intend to search across the file corpus instead? [Click here](#)

1 / 90

1 security vendor flagged this URL as malicious

Reanalyze Search Graph

http://fpdownload.macromedia.com/
fpdownload.macromedia.com

Status: 200 Last Analysis Date: 6 days ago

multiple-redirects

Community Score

DETECTION DETAILS LINKS COMMUNITY 11

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Security vendors' analysis Do you want to automate checks

Sucuri SiteCheck	Malicious	Abusix	Clean
Acronis	Clean	ADMINUSLabs	Clean
AICC (MONITORAPP)	Clean	AlienVault	Clean
alphaMountain ai	Clean	Antiy-AVL	Clean
Artists Against 419	Clean	Avira	Clean
benkow.cc	Clean	Bfore AI PreCrime	Clean
BitDefender	Clean	BlockList	Clean
Blueliv	Clean	Certego	Clean
Chong Lua Dao	Clean	CINS Army	Clean
CMC Threat Intelligence	Clean	CRDF	Clean

17. Go to virus total and upload msimg32.dll file .

51 / 68

51 security vendors and no sandboxes flagged this file as malicious

Reanalyze Similar More

ddf7ccab32e8c0ee6294df2591efac632c27c61d073b86b97de62311f9379212
msimg32.dll

Size: 247.00 KB Last Analysis Date: 2 years ago

Community Score

DETECTION DETAILS COMMUNITY

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label trojan.zaccess/sirefef Threat categories trojan dropper Family labels zaccess sirefef rootkit

Security vendors' analysis Do you want to automate checks

Acronis (Static ML)	Suspicious	AegisLab	Trojan.Win32.ZAccess.tnpa
AhnLab-V3	Trojan/Win32.ZAccess.R87034	Alibaba	VirTool.Win32/Obfuscator.b2189c24
Antiy-AVL	Trojan[Backdoor]/Win32.ZAccess	Avast	Win32/Malware-gen
AVG	Win32/Malware-gen	Avira (no cloud)	TR/Crypt.XPACK.52658
BitDefenderTheta	Gen.NN.ZedlaF.34804.py4@ayRgCzeO	Bkav Pro	W32.AIDetectVM.malware2
Comodo	Malware@#o4kg0j5rvjp0	CrowdStrike Falcon	Win/malicious_confidence_100% (D)
Cylance	Unsafe	Cyren	W32/Zbot.QZDC-5119
DrWeb	BackDoor.Maxplus.14813	eGambit	Unsafe.AI_Score_99%
Elastic	Malicious (high Confidence)	ESET-NOD32	Win32/Sirefef.FY

18 . We can see this is something malicious

Measures to Prevent Zeus Attacks

To protect systems and networks from Zeus and similar banking Trojans, the following preventive measures should be implemented:

- **Use Multi-Factor Authentication (MFA):** Prevents unauthorized access even if credentials are stolen.
- **Avoid Clicking on Suspicious Links:** Educate users about phishing emails and fake websites.
- **Update Security Software:** Regularly update antivirus, firewalls, and intrusion detection systems.
- **Enable Network Segmentation:** Restrict access between systems to prevent malware propagation.
- **Monitor Network Traffic:** Use tools like Wireshark to detect suspicious communication with C2 servers.
- **Implement Endpoint Detection & Response (EDR):** Helps identify and respond to threats in real-time.

Existing Solutions for Zeus Mitigation

Several cybersecurity solutions have been developed to detect and mitigate Zeus Trojan infections:

- **Microsoft Defender & Windows Security Tools:** Detects known Zeus variants and blocks execution.
- **Banking Security Software:** Some financial institutions use fraud detection mechanisms to identify unusual transactions.
- **Threat Intelligence Services:** Companies like FireEye, Palo Alto Networks, and CrowdStrike offer solutions to track and prevent botnet activity.
- **Behavioral Analysis Tools:** AI-driven solutions like Darktrace can analyze unusual user behaviors and alert security teams.
- **YARA Rules & Snort Signatures:** Used by cybersecurity professionals to detect and remove Zeus-infected files.

Conclusion

The Zeus Banking Trojan remains one of the most **sophisticated financial cyber threats** ever created. Despite efforts to dismantle its botnets, its **evolving variants continue to target financial institutions** globally. With increased cybersecurity awareness and the use of advanced detection tools, organizations can reduce the risk of infection and minimize financial losses. The continuous adaptation of cybercriminals necessitates proactive measures, constant monitoring, and improved defense mechanisms to stay ahead of evolving threats.

References

1. "Zeus Trojan Analysis - A Deep Dive," FireEye Threat Intelligence Report, 2023.
2. Palo Alto Networks, "Understanding and Mitigating Banking Trojans," Cybersecurity Whitepaper, 2022.
3. Microsoft Security Blog, "Zeus and its Variants: Detection and Prevention Strategies," 2021.
4. Darktrace AI Research, "Machine Learning in Cyber Threat Detection: Banking Trojan Case Study," 2023.
5. VirusTotal, "Zeus Malware Samples and Detection Rates," Public Report Database, 2023.
- 6.

