**Analysis of Zeus Banking Trojan Report**

Henrique Dias

Índice

[Fingerprint 3](#_Toc158651158)

[Basic Static Analysis 5](#_Toc158651159)

[Advanced Static Analysis 7](#_Toc158651160)

[Basic Dynamic Analysis 10](#_Toc158651161)

[Advanced Dynamic Analysis 12](#_Toc158651162)

[YARA(IOC) 14](#_Toc158651163)

# Fingerprint

**FileName:** invoice\_2318362983713\_823931342io.pdf.exe

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamenteVirusTotalOutput

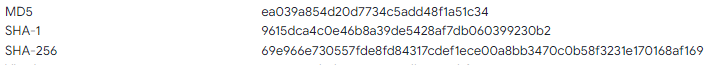
Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, número, Tipo de letra, software

Descrição gerada automaticamente

Hash’s:



**MD5:** ea039a854d20d7734c5add48f1a51c34

**SHA-1:** 9615dca4c0e46b8a39de5428af7db060399230b2

**SHA-256:** 69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169

Uma imagem com texto, software, Ícone de computador, Página web

Descrição gerada automaticamente

M Z means that, this file, is most likely an executable, and not a pdf, as it tried to impersonate.

# Uma imagem com texto, file, Tipo de letra, número Descrição gerada automaticamenteBasic Static Analysis

Uma imagem com texto, Tipo de letra, captura de ecrã, número

Descrição gerada automaticamenteCorect.com yields no interesting results.

Since the raw-size is no different from the virtual-size, that gives us the idea that this malware is not compressed.

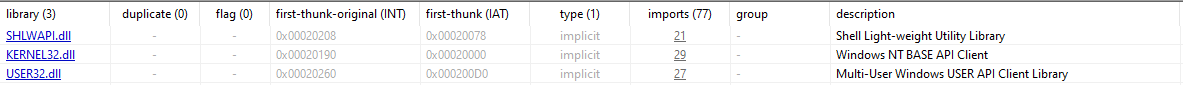
Uma imagem com texto, captura de ecrã, número

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, número, Paralelo

Descrição gerada automaticamente

These are, most likely, function names.



Libraries that Zeus Trojan is using, notice that involves kernel32 and user32, which means that involves something with our computer kernel.

# Advanced Static Analysis

Using cmder, we are going to extract a few strings from the Zeus Code. \*See Results in strings.txt\*

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

capa -vv

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, número, software

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, número, Tipo de letra

Descrição gerada automaticamente

Using cutter, we can get the source code in assembly, in which it shows where the data is changed via the physical addresses.

Uma imagem com texto, captura de ecrã, Tipo de letra, Paralelo

Descrição gerada automaticamente

From this graph we can see that the output from the \*test byte[data.]\* has an if statement. If the output is what is expecting it’s proceeds, otherwise it calls a function called “AllowSetForegroundWindows”, as we can see in the cutter graph.

# Basic Dynamic Analysis

Now that the advanced static analysis is done, let’s start with the basic dynamic analysis. On this type of analysis we detonate the virus and see his behaviour.

Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

With the usage of Procmon, we can see that the trojan is opening a session in cmd, as we expected before, with the kernel library, but also is using google crash handler. Most likely is activating as soon as someone starts a google session.

Uma imagem com captura de ecrã, texto, software, Ícone de computador

Descrição gerada automaticamente

It uses the temp folder to exploit.

Uma imagem com texto, captura de ecrã, ecrã, software

Descrição gerada automaticamente

I also filtered the search is doing by path(path that the trojan is using – temp folder), and by process name (invoice). But got zero results.

Uma imagem com texto, software, captura de ecrã, número

Descrição gerada automaticamente

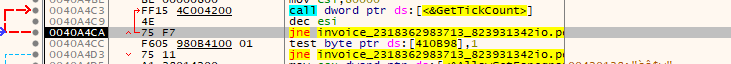
As it uses a session in the internet to exploit, I decided to use wireshark to see If I can sniff something out of it. No results as well.

# Uma imagem com texto, captura de ecrã, software, número Descrição gerada automaticamenteAdvanced Dynamic Analysis

Uma imagem com texto, captura de ecrã, diagrama, Tipo de letra

Descrição gerada automaticamente

From this screenshot, we can see, again, the libraries that are being used, as well as the new Variables this trojan is using, for example, the NtUserBlockInput.



As he calls [&GetTickCount], which is a Windows functions that returns a value of the milliseconds that have elapsed since the system started, this trojan starts working.

# YARA(IOC)

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamenteTo finish I’m writing a yara rule.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Final results of the yara rule.

I specified a function (KERNEL32\_CREATEFILEA), which has to be something about the kernel and creating a specific file. Also, the type of the file, which is MZ, which means it’s an executable, and the address which this can be found.