

*Digital forensics*  
*Toolset, Document for students*  
September 2013





## About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at [www.enisa.europa.eu](http://www.enisa.europa.eu).

## Authors

This document was created by the CERT capability team at ENISA in consultation with:

Don Stikvoort and Michael Potter from S-CURE, The Netherlands, Mirosław Maj, Tomasz Chlebowski, Paweł Weźgowiec from ComCERT, Poland, Przemysław Skowron from Poland, Roeland Reijers from Rubicon Projects, The Netherlands and Mirko Wollenberg from DFN-CERT Services, Germany.

## Contact

For contacting the authors please use [CERT-Relations@enisa.europa.eu](mailto:CERT-Relations@enisa.europa.eu)

For media enquires about this paper, please use [press@enisa.europa.eu](mailto:press@enisa.europa.eu).

## Acknowledgements

ENISA wants to thank all institutions and persons who contributed to this document. A special "Thank You" goes to the following contributors:

- Jarosław Stasiak from BRE Bank, Poland, Łukasz Juszczuk from ING Services, Poland, Vincent Danjean from Interpol, Daniel Röthlisberger and Frank Herbert from SWITCH, Switzerland, Adam Ziąja and Dawid Osojca from ComCERT, Poland.

### Legal notice

Notice must be taken that this publication represents the views and interpretations of the authors and editors, unless stated otherwise. This publication should not be construed to be a legal action of ENISA or the ENISA bodies unless adopted pursuant to the Regulation (EU) No 526/2013. This publication does not necessarily represent state-of-the-art and ENISA may update it from time to time.

Third-party sources are quoted as appropriate. ENISA is not responsible for the content of the external sources including external websites referenced in this publication.

This publication is intended for information purposes only. It must be accessible free of charge. Neither ENISA nor any person acting on its behalf is responsible for the use that might be made of the information contained in this publication.

Reproduction is authorised provided the source is acknowledged.

© European Union Agency for Network and Information Security (ENISA), 2013



## Table of Contents

<b>1</b>	<b>General Description</b>	<b>2</b>
<b>2</b>	<b>Introduction</b>	<b>3</b>
2.1.1	Principle 1 – Data Integrity	4
2.1.2	Principle 2 – Audit Trail	5
2.1.3	Principle 3 – Specialist Support	5
2.1.4	Principle 4 – Appropriate Training	5
2.1.5	Principle 5 – Legality	5
<b>3</b>	<b>Scenario</b>	<b>6</b>
<b>4</b>	<b>Task 1 – Identify characteristics in the HTTP session of fraud</b>	<b>8</b>
<b>5</b>	<b>Task 2 – Identify other attacked customers based on these characteristics</b>	<b>11</b>
<b>6</b>	<b>Task 3 – Hands-on analysis of memory process dump</b>	<b>13</b>
<b>7</b>	<b>Summary of the exercise</b>	<b>21</b>
<b>8</b>	<b>References</b>	<b>22</b>

Main Objective	<p>Present the trainees with the principles of digital forensics and evidence gathering. Establish a common knowledge of the requirements regarding evidence admissibility in the court of law. Show a server-centric approach to evidence gathering as a valuable source for further legal proceedings as well as for establishing patterns of malicious activity. The patterns are then used to quickly identify similar events from the past in the future as they take place. The exercise also gives an overview of popular malware characteristics, methods of identification and tools that may be used at the scene.</p>	
Targeted Audience	<p>The exercise is intended for CERT staff involved in the process of early fraud investigations to establish mechanisms of fast response to future events and detect similar actions in other archived data. It applies especially to events when there is a possibility of further legal actions.</p>	
Total Duration	5.5 hours	
Time Schedule	Introduction to the exercise	0.5 hour
	<b>Task 1:</b> Identify characteristics in HTTP session of fraud	1.5 hour
	<b>Task 2:</b> Identify other attacked customers based on these characteristics	1.0 hour
	<b>Task 3:</b> Hands-on analysis of memory process dump	2.0 hours
	Summary of the exercise	0.5 hour
Frequency	<p>It is advised to organise the exercise once a year or when a new people join the CSIRT/CERT team.</p>	

## 1 General Description

The main goal of this exercise is to provide you with technical knowledge of tools and reasoning used in digital forensics. You are required to focus on details during the examination of system data as you craft a script to detect similar events throughout the evidence. You get insights into network monitoring as well, learning that information gathered from logging systems and IDS sensors is crucial in any investigation as well as in regular incident handling practices.

The course scenario is based on a common scheme of an electronic banking fraud when the evidence may only be gathered on the bank side. Only a small percentage of such cases include examining evidence at the client side, especially in a situation where the case is not supported by Law Enforcement Agencies or gains the support late in the investigation.

This exercise also presents basic principles of evidence. At all times you should be aware of the documentary characteristics of the gathering process, and principles stated in the exercise introduction must be applied.

The exercise consists of 3 components:

1. Identifying the pattern of the malicious activity,
2. Identifying other cases with the same pattern,
3. Analysing collected malware.<sup>1</sup>

---

<sup>1</sup> ENISA CERT Exercises – ‘Identification and handling of electronic evidence’ - <https://www.enisa.europa.eu/activities/cert/support/exercise>

## 2 Introduction

Having a code of laws is one of the rudiments of any modern civilisation. Along with the developments in law, law enforcement units were created. A court of law has the final word on whether someone's act was lawful or not. To make that judgement a court of law has to rely on 'evidence' presented to it.

*Evidence is any of the material items or assertions of fact that may be submitted to a competent tribunal as a means of ascertaining the truth of any alleged matter of fact under investigation before it.*<sup>2</sup>

Traditionally evidence was gathered in physical form. After the invention of photography it became a common practice to take photographs at the crime scene and present the photographs along with other evidence. With the digital revolution and following usage of electronic devices in almost all aspects of life it became necessary to allow evidence extracted from electronic devices, especially with electronic storage capacity, for use in judicial proceedings. We call such evidence 'electronic evidence'.

In modern judicial practice electronic evidence is no different from traditional evidence, so it is mandatory that the party introducing it into legal proceedings is able to demonstrate the evidence was left intact from the moment it was collected – including the collecting process.

It must be stressed that electronic evidence, being usually much easier to manipulate than traditional forms of data, require great care when handled to be admissible in a court of law. The seizure, custody, control, transfer, analysis and disposition of the evidence must be chronologically documented in a proper way constituting a 'Chain of custody' (CoC).<sup>3</sup>

Proper handling of any evidence, including electronic evidence, requires following some general guidelines:

Handling by specialists	Each device has its characteristics and handling procedures must adhere to them. Electronic devices are particularly sensitive to unintentional changes to their state, which along with other dangers may lead to rejecting the evidence by the court of law.
Rapid evolution	Rapid evolution of electronic evidence sources requires constant improvement in forensic techniques and procedures.
Use of proper procedures, techniques and tools.	Use of proper procedures, techniques and tools. Along with expert knowledge of forensic engineers, each task requires following procedures while applying proper techniques with adequate tools. Each forensic investigation must be traceable and repeatable by other forensic specialists with the same final conclusion.
Admissibility	Since the ultimate goal is to present the evidence to support a case in a court of law, the evidence must be obtained in compliance with existing law. It must be stressed that laws vary between countries however, in all cases due professional care must be applied.
Authenticity	It must be certifiable to tie the evidence to the case under investigation.
Completeness	It must cover the case completely regardless of the perspective.
Reliability	There must be no doubt about how the evidence was collected and handled that could raise questions about its authenticity and veracity.

<sup>2</sup> Encyclopedia Britannica – <http://www.britannica.com/EBchecked/topic/197308/evidence>

<sup>3</sup> Wikipedia – [https://en.wikipedia.org/wiki/Chain\\_of\\_custody](https://en.wikipedia.org/wiki/Chain_of_custody)

Credibility	It must be understandable and believable to the court.
Proportionality	The whole process of investigation must be adequate and appropriate, i.e. the benefits gained by a specific action must outweigh the harms for the parties affected by the action.

However, we must remember some unique characteristics of digital evidence:

**It's invisible to the untrained eye.** Electronic evidence is often retrieved from places known or accessible only to experts.

**It may need to be interpreted by a specialist.** In many cases information gained requires thorough analysis to uncover properties assuring the information is valid from judicial point of view.

**It's highly volatile.** A powered electronic device modifies its state every time a specific event happens. Lack of power or a system overwriting old data with new data requires us to preserve electronic evidence as soon as possible.

**It may be altered or destroyed through normal use.** Devices constantly change the state of memory – allocating it for programs automatically, swapping it to disk or writing chunks of it to a disk file on user request. This characteristic calls for using appropriate tools and techniques from the very moment of identification the evidence as relevant for an investigation.

**It can be copied without limits.** This property allows many specialists work on the same evidence at the same time in different places. It also enables the possibility of presenting the evidence as-is in the court of law along with the specialist witness report.

The branch of forensic science that focuses on the identifying, acquiring, processing, analysing and reporting of evidence that is stored on computer systems, digital devices and other storage media with the aim of admissibility in court is called Digital Forensics.

There are 5 main principles that establish a basis for all dealings with electronic evidence. These principles were adopted as part of European Union and the Council of Europe funded project to develop a 'seizure of e-evidence' guide. As stated before, while laws regarding admissibility of evidence differ between countries, using these principles is considered appropriate as they are common internationally.<sup>4</sup>



### 2.1.1 Principle 1 – Data Integrity

No action taken should change electronic devices or media, which may subsequently be relied upon in court.

- When handling electronic devices and data, they must not be changed, either in relation to hardware or software. The person in charge is responsible for the integrity of the material recovered from the scene and thus for initiating a forensic chain of custody.
- There are circumstances where a decision will be made to access the data on a 'live' computer system to avoid the loss of potential evidence. This must be undertaken in a manner which causes the least impact on the data and by a person qualified to do so.

<sup>4</sup> This is excerpt from the 'Electronic evidence guide', version 1.0, created as part of CyberCrime@IPA, EU/COE Joint Project on Regional Cooperation against Cybercrime.

### **2.1.2 Principle 2 – Audit Trail**

An audit trail or other record of all actions taken when handling electronic evidence should be created and preserved. An independent third party should be able to examine those actions and achieve the same result.

- It is imperative to accurately record all activities to enable a third party to reconstruct the first responder's actions at the scene in order to ensure probative value in court. All activity relating to the seizure, access, storage or transfer of electronic evidence must be fully documented, preserved and available for review.

### **2.1.3 Principle 3 – Specialist Support**

If it is assumed that electronic evidence may be found in the course of an operation, the person in charge should notify specialists/external advisers in time.

- For investigations involving search and seizure of electronic evidence it may be necessary to consult external specialists. All external specialists should be familiar with the principles laid down in this or similar relevant documents. A specialist should have:
  - Necessary specialist expertise and experience in the field,
  - Necessary investigative knowledge,
  - Necessary knowledge of the matter at hand,
  - Necessary legal knowledge,
  - Appropriate communication skills (for both oral and written explanations)
  - Necessary appropriate language skills.

### **2.1.4 Principle 4 – Appropriate Training**

First responders must be appropriately trained to be able to search for and seize electronic evidence if no experts are available at the scene.

- In exceptional circumstances where it is necessary that a first responder collects electronic evidence and/or accesses original data held on an electronic device or digital storage media, the first responder must be trained to do it properly and to explain the relevance and implications of his/her actions.

### **2.1.5 Principle 5 – Legality**

The person and agency in charge of the case are responsible for ensuring that the law, the general forensic and procedural principles, and the above listed principles are adhered to. This applies to the possession of and access to electronic evidence. Each Member State should take its own legal documents and regulations into consideration when interpreting the measures proposed in this document.

- One of the internationally important legal documents, the Convention on Cybercrime by the Council of Europe, is currently open for signature by the Member States and the states, which have participated in its elaboration, and for accession by other states.

In the last few years we have witnessed a great change in the banking sector. It has undergone a full transition from a completely closed environment with strict perimeter protection, leased data links and no Internet access even for its employees, to an open world where almost any electronic device can play the role of a banker's workstation. Lack of control over consumer devices is one of the biggest problems in ensuring full security of e-banking environments. As we cannot completely prevent the misuse of e-banking systems, we must build the ability to react and respond to security incidents. This response should always lead to better prevention and detection measures, as well as to legal actions when necessary.

### 3 Scenario

A bank customer has recently made a complaint about a transfer of money to an account that was unknown and never used before. According to the customer declaration, the transfer was made at a time when he/she was not using the electronic banking system.

However, the customer admitted during the call to the Bank hotline that he/she found a text message with the authorisation code for the transfer in question. He/she remembers also playing a bank quiz game for mobile phones. The customer did not notice anything suspicious; he/she declares using only one computer for electronic banking with the same Internet browser all the time.

As precautionary measure, the customer's password was changed during the call.

You are CSIRT/CERT members in a financial institution. You have just received a notification from your fraud detection system team, that there's been a fraud action with a set of characteristics previously unknown to their systems.

Your task is to help establish a pattern that would allow for finding other transactions that may need further investigation.

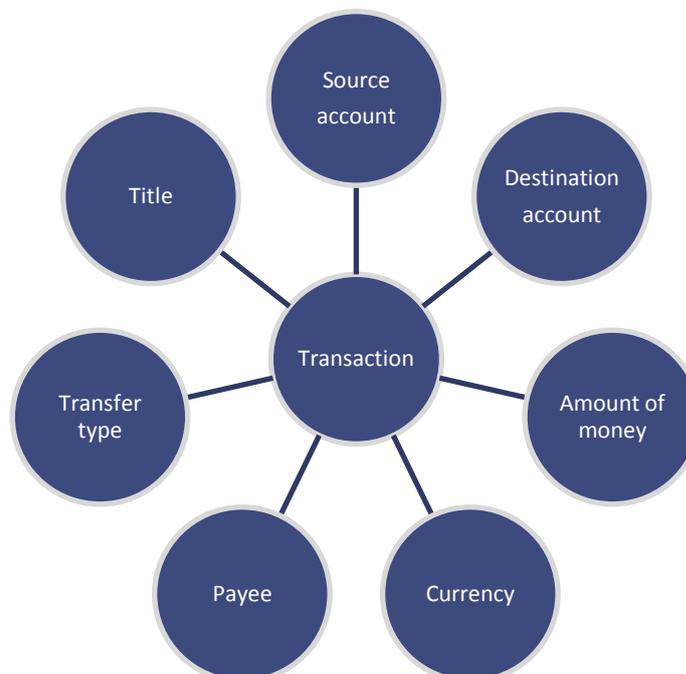
At first, the customer denies access to his/her computer and mobile phone data, so you must utilize any technical means available on the bank's side to find clues about fraudulent transactions. Find data related to the unauthorized transaction.

Now you will be presented with some basic information about money transfer data.

We can divide data normally available at a banking system into two main categories:

1. Directly linked to transactions made
2. Additional data, related to the technology used behind the scenes

Information directly linked to the transactions includes:



1. Source and destination account numbers – one of them is typically a local account in the banking system analysed, the other can be local or an account in different bank.
2. Amount of money in the transaction
3. The transaction currency

4. Payee
5. Transfer type (standard, fast, SWIFT and others)
6. Title or description put on the order

Additional data include information specific to the electronic banking system such as:

1. Data regarding user login:
  - a. IP address of the far end (user)
  - b. Status of the login attempt (success, failure)
2. HTTP session data:
  - a. Headers: User-Agent, Accepted-Languages, Accept-Encoding
  - b. Cookies
3. Some additional data can be retrieved from external data sources, such as GeoIP database when fed with logged data (IP address in this case)
4. Information about bank transfers must be obtained from a database or a data warehouse used for banking transactions. This source is considered highly reliable, there are numerous teams in every bank working hard to ensure the quality of the data.
5. Electronic banking services are normally configured to collect and store information about users logging in and out to the system. They also log any transactions made and/or authorized by a user to assure non-repudiation. Such information can be obtained from a database similar to the transaction database. Note, that this database supposedly contains only some sort of transaction identifiers or hashes that allow for matching electronic banking information with the transaction information.

## 4 Task 1 – Identify characteristics in the HTTP session of fraud

The main goal of Task 1 is to find the differences between normal and fraudulent HTTP sessions and to identify characteristics of the fraudulent session to be used in next tasks.

You will work with collected log files and use the following tools to extract information from them to achieve the goal:

- 1) Standard command lines utilities (bash, grep, sort, cut, etc.);
- 2) GeolIP DB;
- 3) Python/Perl to implement more advanced logic.

Although storing data about bank transfers and electronic banking logins is routine, gathering technical information on user sessions (cookies, headers) is rare in web servers' configuration. We can work around this issue by having logged network traffic information into a pcap file for the purpose of incident handling.

We can use scripts or a tool such as BroIDS to generate access\_log files from the pcap dump with the content we require for a successful investigation. In our case we will use the *tshark* utility, a terminal version of Wireshark, a well-recognized tool used to dump and analyse network traffic:

```
tshark -nn -r tcpdump.pcap -T fields -E separator='|' -R 'http.request' -e ip.src -e frame.time -e http.request.uri -e http.user_agent -e http.cookie > tshark.log
```

To make it handy we've redirected the output to *tshark.log* file.

You have collected log files from your transaction environment:

- 1) access\_log from HTTP server, Apache;
- 2) pcap file (tcpdump.pcap) to identify characteristics which are not present in the access\_log files;
- 3) syslog.log file containing the log of initiated user sessions, containing cookie information, extracted from pcap file;
- 4) client.csv file containing information about users and their online sessions identified by cookie information.

These are stored in /home/enisa/enisa/forensic directory on the virtual machine. You may also want to access publicly available databases such as:

- 1) GeolIP database;
- 2) 'Bad IP' database.

Find anomalistic characteristics of the fraud session. You need to compare carefully a normal session with the questioned one. The session narrative contains some clues.

We know from the Fraud Detection Unit, that the customer questioning has ID **client94777** and the compliant was made by him on 21<sup>st</sup> August at 17:30 of local time. We know that the user had a legitimate session open during the bank call centre and the customer claimed to have been online with the banking application for around 15 minutes at that time.

Let's search quickly through the collected files to find something about the user. As we've seen looking through the files, the only log to have bank customer information is the user session file client.csv, we should start our search there:

```
grep client94777 client.csv
```

```
client94777,gohlohyigheidoochiezueshoopeiceeche
```

```
client94777,ichaewupugohciasahchohwaethohfooquo
```

There are two sessions found for this user, but we know only the cookies of these sessions. We have to look through system log to see when the two sessions started. We could look into the tshark.log instead of syslog as both files are extracted from a pcap, this one is more readable however:

```
grep gohlohyigheidoochiezueshoopeiceeche syslog.log
```

```
Aug 21 15:52:19 debian bank: 62.182.152.59 gohlohyigheidoochiezueshoopeiceeche
```

```
grep ichaewupugohciasahchohwaethohfooquo syslog.log
```

```
Aug 21 17:16:00 debian bank: 91.226.251.216 ichaewupugohciasahchohwaethohfooquo
```

We found two sessions as expected, the second one matches the time of a bank call and is a confirmed one. We want to see the details of this session, so we try to find it in webserver access log. The access log however doesn't include cookie information, we have to find it either by date (which normally is impractical with many accesses every second) or by IP address (which is much better unless it's an IP address for a large network proxy):

```
grep 91.226.251.216 access.log
```

```
91.226.251.216 - - [21/Aug/2013:17:16:00 +0200] "GET /login.php HTTP/1.1" 200 291 "-"  
"Opera/9.80 (X11; Linux x86_64; U; pl) Presto/2.10.229 Version/11.61"
```

This is a default access log for an Apache webserver, and therefore most commonly encountered. It gives us the information about accessing IP, date (note, it's local time: UTC+2), type of request (GET), requested URI and protocol used (/login.php HTTP/1.1), status code + extended status code (200 291 – codes with a digit 2 in front are generally success codes), referrer (the page user came from, empty in this case), and so called User-Agent – name and the type of the browser (Opera) it's version (9.80) along with information about the system used, type of encoding, extensions and/or html engine used. All this information is provided to 'enhance browsing experience' which means: to be able to modify web content according to browser capabilities and user preferences. In this case a content-rich web service would provide the user with Polish version (note 'pl' in language) of the contents in Unicode (U) but perhaps using HTML instead of ActiveX controls, as this is a connection coming from a Linux box, not a Windows one.

It must be noted however, that the User-Agent string can be easily modified by the user, most browsers can be configured to present themselves as others.

So what can we tell about the other session?

```
grep 62.182.152.59 access.log
```

```
62.182.152.59 - - [21/Aug/2013:15:52:19 +0200] "GET /login.php HTTP/1.1" 200 291 "-"  
"Mozilla/5.0 (Linux; U; Android 4.2.2; ru; Nexus 7 Build/JDQ39) AppleWebKit/534.30 (KHTML, like  
Gecko) Version/4.0 Safari/534.30"
```

The session was started at 15:52:19 local time, the IP is different so is the user agent.

Remember from the narrative – the user claimed to use the same computer for all electronic banking activities. Apparently, there's a trace the system is different. It presents as Mozilla with preferred language ru (Russian). Of course both IPs are different, we've searched for them, but did we use all the information we have? What about geolocation of both cases:

**geoipllookup 91.226.251.216**

*GeoIP Country Edition: PL, Poland*

**geoipllookup 62.182.152.59**

*GeoIP Country Edition: UA, Ukraine*

The browser asked for Russian language, but apparently the connection came from Ukraine. Is there any other clue?

Now, after the case is solved, we can see how to modify the http server config and the access log generated by this configuration. In that case pcap files wouldn't have been needed, all the information is logged directly by http server. We consider it a good practice to configure a web server in this way for increasing incident handling capabilities.

An example of custom log configuration for Apache webserver is presented below:

```
CustomLog ${APACHE_LOG_DIR}/access.log "%h %l %u %t \"%r\" %>s %b  
\"%{Referer}i\" \"%{User-agent}i\" \"%{Accept-Language}i\" \"%{Cookie}i\""
```

## 5 Task 2 – Identify other attacked customers based on these characteristics

After you successfully identified a pattern characteristic of the fraudulent session, the next task is to quickly identify other possible victims.

Prepare a list of customers that have recorded transactions that follow the same pattern. The list will be passed on to the Fraud Detection Team to perform phone verification with listed customers.

Let's play with the search for a while and look for all the small characteristics one by one:

<b>grep Mozilla access.log   wc -l</b>
2429
<b>grep 'ru;' access.log   wc -l</b>
299
<b>grep '\\"' access.log   wc -l</b>
155
<b>for i in `cat access.log   cut -f 1 -d ' '; do geoiplookup \$i; done   grep Ukraine   wc -l</b>
232
<b>for i in `cat access.log   cut -f 1 -d ' '; do geoiplookup \$i; done   sort   uniq   wc -l</b>
73
<b>cat access.log   wc -l</b>
3252

This demonstration was an example of the importance of matching many small factors together. As we can see there were 2429 connections from Mozilla browser (wc -l counts the lines of input), 299 times the preferred language was Russian, even with a malformed header occurred 155 times. There were 232 connections from Ukraine and a total of 3252 connection were initiated from 73 countries, the most exotic ones being A2 Satellite Provider or Trinidad and Tobago.

In the mass of all events registered at a large bank a single property for a connection is not very useful, except for statistics. We have to apply all the information we have at hand to start with the most probable cases:

<b>grep Mozilla access.log   grep 'ru;'   grep '\\"'</b>
62.182.152.59 - - [21/Aug/2013:15:52:19 +0200] "GET /login.php HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Linux; U; Android 4.2.2; ru; Nexus 7 Build/JDQ39) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safari/534.30"
212.9.224.171 - - [21/Aug/2013:16:02:14 +0200] "GET /login.php HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Linux; U; Android 4.1.1; ru; GT-P3110 Build/JRO03C) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Safari/534.30"
91.198.140.87 - - [21/Aug/2013:17:08:26 +0200] "GET /login.php HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Linux; U; Android 2.3.5; ru; HTC_WildfireS_A510e Build/GRJ90) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"
178.213.184.202 - - [21/Aug/2013:17:14:39 +0200] "GET /login.php HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Linux; U; Android 2.2; ru; HTC Desire Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"
<b>geoiplookup 62.182.152.59</b>

*GeoIP Country Edition: UA, Ukraine*

***geoiplookup 212.9.224.171***

*GeoIP Country Edition: UA, Ukraine*

***geoiplookup 91.198.140.87***

*GeoIP Country Edition: UA, Ukraine*

***geoiplookup 178.213.184.202***

*GeoIP Country Edition: UA, Ukraine*

Note that connection from Ukraine was redundant – all connections matching the first three conditions matched the fourth and no further selection occurred. We have to match the connections to real users. As we haven't got cookie information in the access log we must use sources derived from pcap file:

```
for i in `egrep '62.182.152.59|212.9.224.171|91.198.140.87|178.213.184.202' tshark.log | cut -f 5 -d '|' | cut -f 2 -d '='`; do grep $i client.csv; done
```

*client94777,gohlohyigheidoochiezueshoopeiceeche*

*client96707,eidoovoabucoopohzauhahquaiqueeshu*

*client93597,aethaeghaevishuveechaishohhoojaedii*

*client98079,zarozaemoshumeigeughooauthuziezuxe*

We've found four bank customers that had the same connection, of course we've found our original customer that complained about the transactions. That's a very good thing of course. We've found three more customers, and the cases should be verified quickly.

Task summary:

By gaining new but verifiable fraud indicators we can analyse our data sets again. It may allow us to find other clients who were hit by a fraud act but previously undetected.

As we have an established and verified set of fraud indicators we can now build an early detection algorithm.

## 6 Task 3 – Hands-on analysis of memory process dump

Now we will work on material that was collected by other CSIRT team, from a workstation of a customer hit by the fraud. The customer agreed to provide you with the information stored on his/her computer. You are to verify that the information gathered really contains malware, extract its type, version and verify it's configured to target your bank.

We will use the following tools

- 1) Standard command lines utilities (bash, grep, sort, cut, etc.)
- 2) Volatility Framework ver. 2.0<sup>5</sup> with zeusscan plugins<sup>6</sup>
- 3) Custom-crafted script decrypting malware config file

We will be working on a memory dump from the infected workstation.

Refer to ENISA Exercise 'Identifying and handling of electronic evidence' for more information about the malware extraction process and first evaluation of gathered data.

Remember that all actions should be properly documented, as all actions taken must be repeatable by other experts and lead to the same conclusions.

Examinations done previously by the other team, suggest the computer was infected with one of the ZeuS variants. Gathered evidence contains traces of one type of malicious activity (one unknown server, suspected to be a botnet C&C server is contacted).

We follow the path by examining the memory dump with zeusscan, a Volatility Framework (VF) plugin for detecting processes infected with ZeuS malware, versions 1.x and 2.x. Issue the following commands:

```
cd /home/enisa/enisa/forensic
```

```
vol20.py -f memdump.raw zeusscan1
```

```
vol20.py -f memdump.raw zeusscan2
```

While zeusscan1 reports no data, zeusscan2 detected a group of processes that has ZeuS code injected:

<sup>5</sup> See: <https://www.volatilesystems.com/default/volatility> and <https://code.google.com/p/volatility/wiki/CommandReference22>

<sup>6</sup> By Michael Hale Ligh <http://mnin.blogspot.com/2011/09/abstract-memory-analysis-zeus.html>

```

enisa@enisa-VirtualBox: ~/enisa/forensic
enisa@enisa-VirtualBox:~/enisa/forensic$ vol20.py -f memdump.raw zeusscan2
Volatile Systems Volatility Framework 2.0
-----
Process:      explorer.exe
Pid:          236
Address:      0x1230000
URL:          http://alazqwryx.cn/z12/config.bin
Identifier:   HOME-TBZBXUW4F6_B4DF7611AB1DEF24
Mutant key:   0xCCD6423C
XOR key:      0x779C1C5C
Registry:     HKEY_CURRENT_USER\SOFTWARE\Microsoft\Moivg
  Value 1:    Umfi
  Value 2:    Luvua
  Value 3:    Kyynokc
Executable:   Myyw\emneo.exe
Data file:    Uhan\qidi.dat

Config RC4 Key:
0x00000000  7a ad dd be 9f b5 30 aa 4a 9a 19 74 31 21 12 ea  z....0.J..t1!..
0x00000010  1e 75 f1 18 2b a5 06 70 43 a9 41 cf 65 a7 a0 2f  .u..+.pC.A.e../
0x00000020  02 86 a4 c3 a1 9e 11 81 96 c7 25 1f 6f 08 ce 46  .....%.o..F
0x00000030  f8 6c 9d 53 69 bf bd 56 2c 34 e0 a2 91 cb e6 3e  .l.Si..V,4....>
0x00000040  52 80 88 05 1a 54 50 5a 9b 49 73 5d 7b da ba de  R....TPZ.Is]{...
0x00000050  01 7d 36 77 4c ff ab 87 f2 b0 04 57 24 09 6d c9  .}6wL.....W$.m.
0x00000060  8d fa 4f 8e 23 ae f7 9c 8b 2e 8a 89 1d 0d b6 c0  ..O.#.....
0x00000070  b1 14 d4 4e a8 82 1b 3d bb 99 7e 3f 15 42 d8 d9  ...N...=..?.B..
0x00000080  d1 0b 0f 84 ec 03 44 c1 20 df 83 f3 5c b2 f9 e3  .....D. ....
0x00000090  71 72 1c f5 fd 07 e1 c2 b7 e7 7f fc f4 5e 58 4b  qΓ.....^XK
0x000000a0  5b 00 40 93 c8 d3 cd 92 97 6a d6 3b ef 35 bc 66  [.@.....j.;.5.f
0x000000b0  e9 db 68 5f 47 32 8c dc ca 79 45 62 f0 d0 4d 2d  ..h_g2...yEb..M-
0x000000c0  64 a3 90 3a 6b 85 c6 33 67 d2 e2 e4 f6 eb 22 b8  d.:k..3g....".
0x000000d0  ac 8f 98 e8 7c 13 16 94 e5 fe 2a fb 38 51 63 b9  ....|.....*.8Qc.
0x000000e0  ed 39 27 76 d5 0a 37 af 29 26 d7 6e cc 55 60 0e  .9'v..7.)&.n.U`.
0x000000f0  28 17 78 c5 95 48 b4 a6 0c 3c 59 b3 10 c4 ee 61  (.x..H...<Y....a
0x00000100  00 00                                     ..

Credential RC4 Key:
0x00000000  64 e5 38 84 75 3e 0d ca 5b 3b 46 3c 7e fa 77 90  d.8.u>..[;F<~.w.
0x00000010  42 29 54 6b aa 0c 2b 1c d7 8a 40 de 3d eb 8b d3  B)Tk..+...@.=...
0x00000020  4a f3 11 56 cd e0 48 dc 68 4b 4d 85 b4 dd 27 98  J..V..H.hKM...'.
0x00000030  02 34 f0 81 a5 2c 32 df b8 d4 1b 94 b3 1f 03 80  .4...2.....

```

Figure 1: Zeusscan

From the script used we derive a conclusion the station is actually infected with ZeuS version 2.x, we will examine it in more detail soon.

For each infected process the zeusscan scripts returns an RC4 script used for decrypting the ZeuS config file, along with some more information: binary file path, datafile path, windows registry keys and the config file location. We can see, that all the keys are the same, so we've got only one instance of ZeuS malware on this computer.

We need to access the config file data so we need to decrypt it first. We copy the config file and save it in data/key.txt file:

```

enisa@enisa-VirtualBox: ~/enisa/forensic
Process: cmd.exe
Pid: 1432
Address: 0x140000
URL: http://alazqwryx.cn/z12/config.bin
Identifier: HOME-TBZBXUW4F6_B4DF7611AB1DEF24
Mutant key: 0xCCD6423C
XOR key: 0x779C1C5C
Registry: HKEY_CURRENT_USER\SOFTWARE\Microsoft\Moivg
Value 1: Umfi
Value 2: Luvua
Value 3: Kynnokc
Executable: Myyw\emneo.exe
Data file: Uhan\qidi.dat

Config RC4 Key:
0x00000000 7a ad dd be 9f b5 30 aa 4a 9a 19 74 31 21 12 ea z....0.J..t1!..
0x00000010 1e 75 f1 18 2b a5 06 70 43 a9 41 cf 65 a7 a0 2f .u...+..pC.A.e../
0x00000020 02 86 a4 c3 a1 9e 11 81 96 c7 25 1f 6f 08 ce 46 .....%.o..F
0x00000030 f8 6c 9d 53 69 bf bd 56 2c 34 e0 a2 91 cb e6 3e .l.Si..V,4....>
0x00000040 52 80 88 05 1a 54 50 5a 9b 49 73 5d 7b da ba de R...TPZ.Is]{...
0x00000050 01 7d 36 77 4c ff ab 87 f2 b0 04 57 24 09 6d c9 .}6wL.....W$.m.
0x00000060 8d fa 4f 8e 23 ae f7 9c 8b 2e 8a 89 1d 0d b6 c0 ..O.#.....
0x00000070 b1 14 d4 4e ae 82 1b 3d bb 99 7e 3f 15 42 d8 d9 ...N...=...~?.B..
0x00000080 d1 0b 0f 84 ec 03 44 c1 20 df 83 f3 5c b2 f9 e3 .....D. ....
0x00000090 71 72 1c f5 fd 07 e1 c2 b7 e7 7f fc f4 5e 58 4b qr.....^XK
0x000000a0 5b 00 40 93 c8 d3 cd 92 97 6a d6 3b ef 35 bc 66 [.@.....j.;.5.f
0x000000b0 e9 db 68 5f 47 32 8c dc ca 79 45 62 f0 d0 4d 2d ..h.G2...yEb..M-
0x000000c0 64 a3 90 3a 6b 85 c6 33 67 d2 e2 e4 f6 eb 22 b8 d...k..3g....".
0x000000d0 ac 8f 98 e8 7c 13 16 94 e5 fe 2a fb 38 51 63 b9 ....|.....*.8Qc.
0x000000e0 ed 39 27 76 d5 0a 37 af 29 26 d7 6e cc 55 60 0e .'v..7.)&.n.U`.
0x000000f0 28 17 78 c5 95 48 b4 a6 0c 3c 59 b3 10 c4 ee 61 (.x..H...<Y....a
0x00000100 00 00 ..

Credential RC4 Key:
0x00000000 64 e5 38 84 75 3e 0d ca 5b 3b 46 3c 7e fa 77 90 d.8.u>..[;F<~.w.
0x00000010 42 29 54 6b aa 0c 2b 1c d7 8a 40 de 3d eb 8b d3 B)Tk...+...@.=...
0x00000020 4a f3 11 56 cd e0 48 dc 68 4b 4d 85 b4 dd 27 98 J..V..H.hKM...'.
0x00000030 02 34 f0 81 a5 2c 32 df b8 d4 1b 94 b3 1f 03 80 .4....2.....
0x00000040 05 d5 5d f2 93 9c 50 e6 8f 5e bc 35 24 47 37 0e ..]...P...^..5$G7.
0x00000050 20 db d6 bf 52 e4 57 86 ab fd 8d 9a 59 04 19 b6 ...R.W....Y...
0x00000060 af 95 5f 69 14 a6 c3 92 6f f8 55 da 01 31 16 79 ..t....o.U..1.y

```

Figure 2: Zeusscan copy action

We may achieve it with a text editor of choice or use basic UNIX console mechanisms:

```

cd data
cat >key.txt

```

Now paste the data (middle mouse button can be used) and press Ctrl-D at the end.

We verify the contents of the file:

```

enisa@enisa-VirtualBox: ~/enisa/forensic/data
enisa@enisa-VirtualBox:~/enisa/forensic/data$ cat key.txt
0x00000000 7a ad dd be 9f b5 30 aa 4a 9a 19 74 31 21 12 ea z....0.J..t1!..
0x00000010 1e 75 f1 18 2b a5 06 70 43 a9 41 cf 65 a7 a0 2f .u...+..pC.A.e./
0x00000020 02 86 a4 c3 a1 9e 11 81 96 c7 25 1f 6f 08 ce 46 .....%.o..F
0x00000030 f8 6c 9d 53 69 bf bd 56 2c 34 e0 a2 91 cb e6 3e .l.Si..V,4....>
0x00000040 52 80 88 05 1a 54 50 5a 9b 49 73 5d 7b da ba de R...TPZ.Is]{...
0x00000050 01 7d 36 77 4c ff ab 87 f2 b0 04 57 24 09 6d c9 .)6wL.....W$.m.
0x00000060 8d fa 4f 8e 23 ae f7 9c 8b 2e 8a 89 1d 0d b6 c0 ..0.#.....
0x00000070 b1 14 d4 4e a8 82 1b 3d bb 99 7e 3f 15 42 d8 d9 ...N...~?.B..
0x00000080 d1 0b 0f 84 ec 03 44 c1 20 df 83 f3 5c b2 f9 e3 .....D. ....
0x00000090 71 72 1c f5 fd 07 e1 c2 b7 e7 7f fc f4 5e 58 4b qf.....^XK
0x000000a0 5b 00 40 93 c8 d3 cd 92 97 6a d6 3b ef 35 bc 66 [.@.....].;.5.f
0x000000b0 e9 db 68 5f 47 32 8c dc ca 79 45 62 f0 d0 4d 2d ..h_G2...yEb..M-
0x000000c0 64 a3 90 3a 6b 85 c6 33 67 d2 e2 e4 f6 eb 22 b8 d.:k..3g.....".
0x000000d0 ac 8f 98 e8 7c 13 16 94 e5 fe 2a fb 38 51 63 b9 ...|....*.80c.
0x000000e0 ed 39 27 76 d5 0a 37 af 29 26 d7 6e cc 55 60 0e .9'v..7.)&.n.U'.
0x000000f0 28 17 78 c5 95 48 b4 a6 0c 3c 59 b3 10 c4 ee 61 (.x..H...<V....a
enisa@enisa-VirtualBox:~/enisa/forensic/data$

```

Figure 3: Decryption key

All data was copied correctly, but the file has a 'hexdump' format, which is unsuitable for processing. We need to convert it to a binary format, which can be achieved with a simple one-line script:

```
cat key.txt | cut -d ' ' -f 4-19 | xxd -r -p > key.bin
```

This script first extracts the middle column (the one with a list of character pairs) and then converts the pairs (hexadecimal values) to corresponding byte values.

To verify that copy & paste was executed correctly and the conversion succeeded we verify cryptographic sums, they must match the values in the Figure 4.

```

enisa@enisa-VirtualBox: ~/enisa/forensic/data
enisa@enisa-VirtualBox:~/enisa/forensic/data$ md5sum key.bin; sha1sum key.bin
1be0ae4eabf93aad01612e101145a4ed key.bin
9d67409dc679a36658d634f92759330c942367b7 key.bin
enisa@enisa-VirtualBox:~/enisa/forensic/data$

```

Figure 4: Calculating hash sums

At this time we've got an RC4 key which we use to decrypt the config file. All we need is a config file. Now ask the trainees where we can get the config from. Can we use the online version if the botnet is still active? What guarantees there are the config is the same as the one used for fraud?

We may want to extract the config file from a memory dump, but we'll use a simpler technique and extract the file from the network dump, a pcap file taken from the scene by fellow team. This approach has one more advantage, we use the less poisoned evidence, taken before the team started altering the source system by issuing commands. Note that an even better source for that particular reason would be the disk dump if the malware stored the config file on disk.

We open the network.pcap file in Wireshark and enter the 'http' filter in the Filter field (and click Apply button):

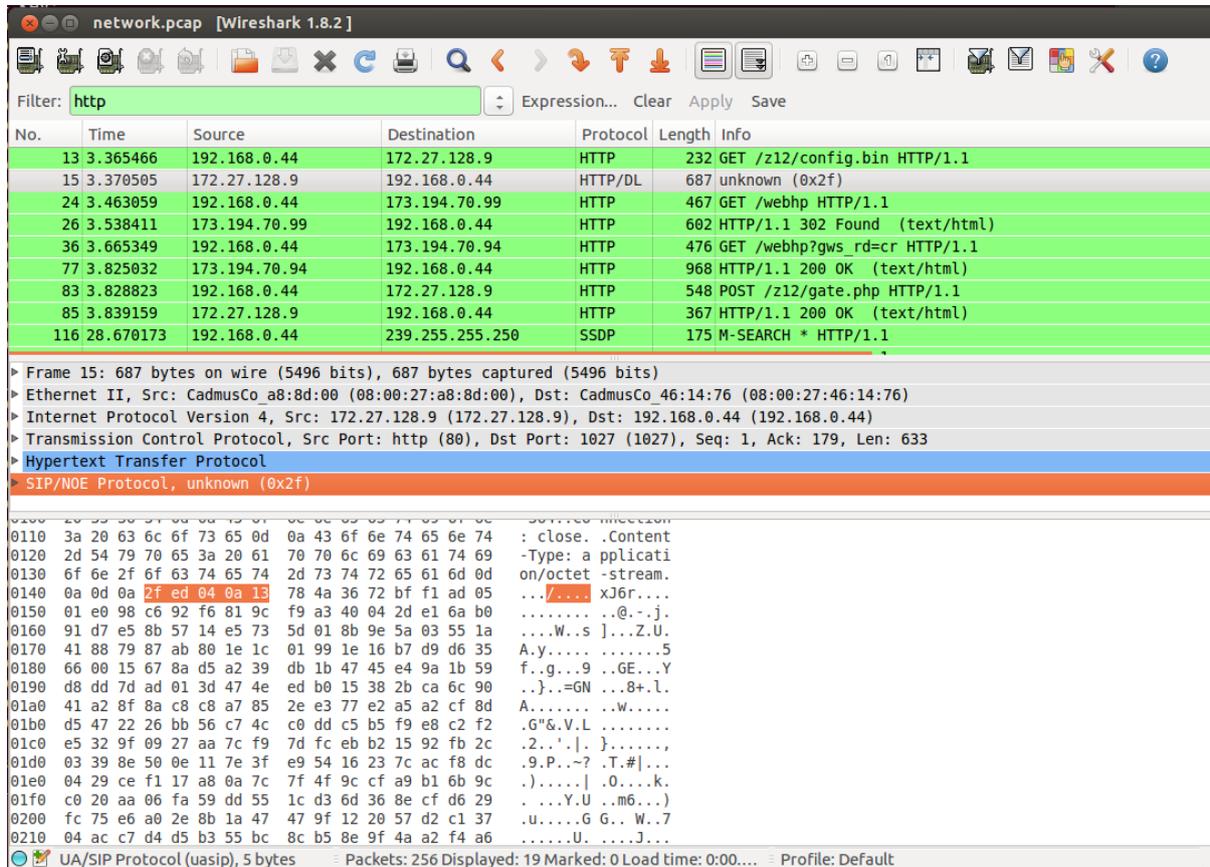


Figure 5: Wireshark communication

As we can see on the screen and in Figure 5 there was an http request for /z12/config.bin in frame 13. Frame 15 contains server's reply.

Unfortunately the Wireshark version shipped with our distribution misinterprets the http server reply from the C&C server. Fortunately the http headers show there's no encoding or encryption in this case, so we can work around the issue by accessing the frame directly and search for all the contents beginning with 0x2fed040a13 (highlighted in Figure 5) to the end of frame.

To achieve this we select the frame and press Ctrl-H or use File->Export Selected Packet Bytes from the Wireshark menu:

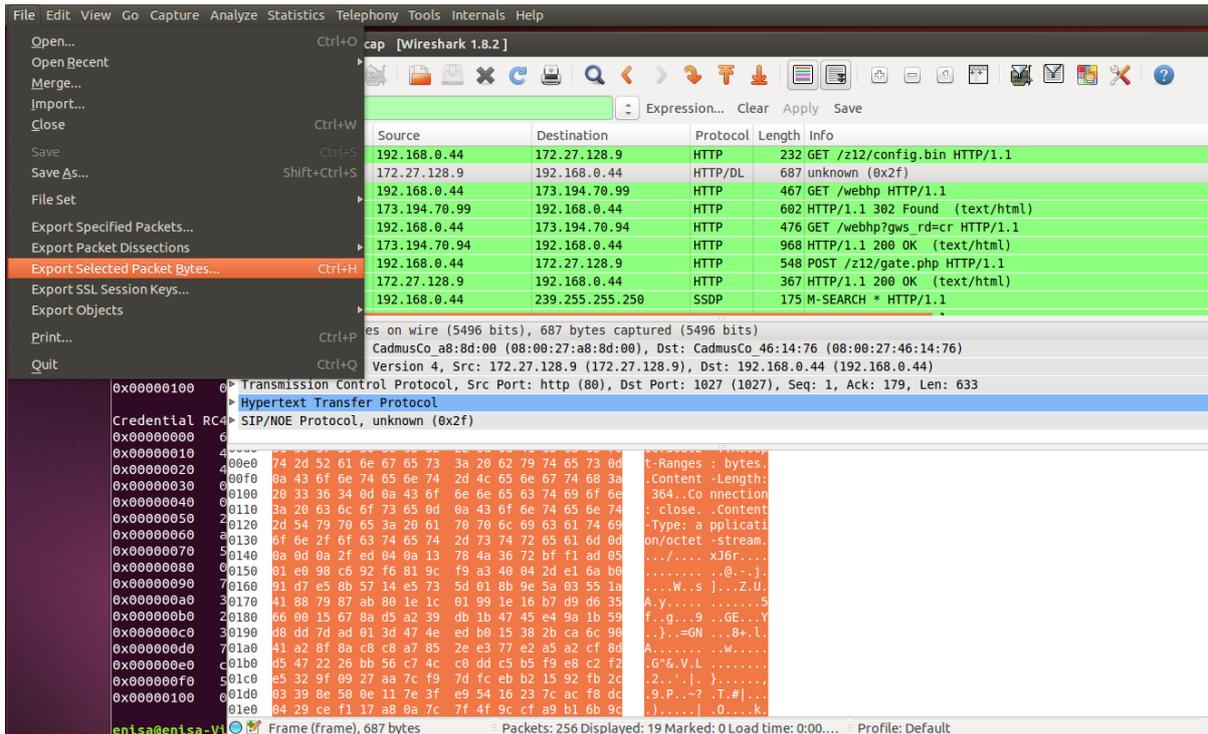


Figure 6: Copying bytes from traffic

We save the exported data to /home/enisa/enisa/data/config.bin. As we exported the whole frame, not only the response payload we have to remove everything before the 0x2FED040A13 bytes. We use a hex editor, GHex:

**ghex config.bin**

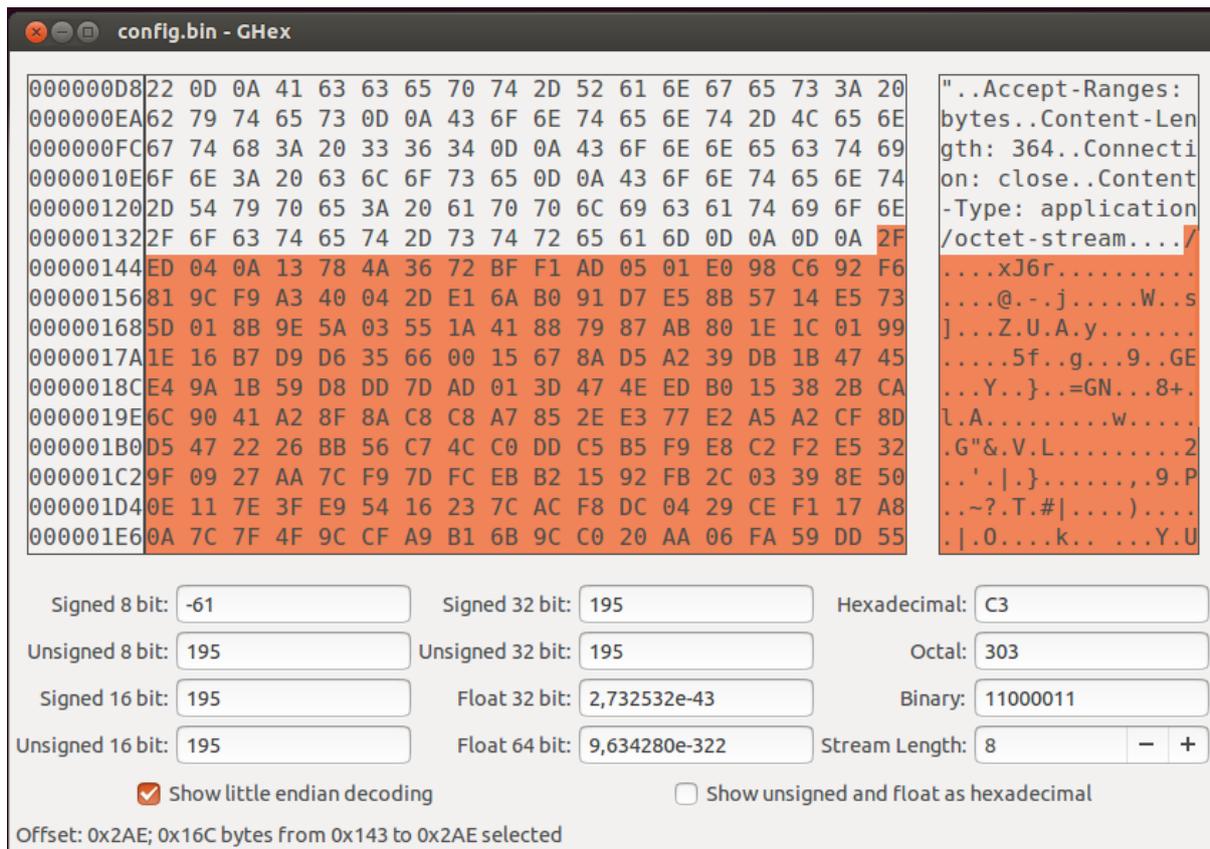


Figure 7: Copying bytes from traffic

We remove all the bytes preceding the highlighted area, save the file and compute cryptographic sums to verify our work. MD5 and SHA1 sums should match these in Figure 8.

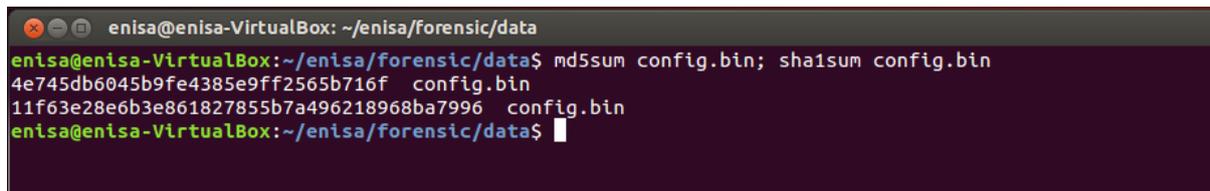


Figure 8: Calculating hashes sum

Now we can look into the config file. We remember it's encrypted with an RC4 file, in this case the file in key.bin file. We've crafted a script for that case, based on leaked ZeuS source code and documentation, zdecrypt.py:

```

enisa@enisa-VirtualBox: ~/enisa/forensic/data
enisa@enisa-VirtualBox:~/enisa/forensic/data$ zdecrypt.py config.bin key.bin
Total size: 364
Storage flags: 0
Items count: 5
Config MD5: aa12e1ba17b2b1a31ac716ecfcd4fa1b (VERIFIED)
-----
CFGID_LAST_VERSION:
- flags: ITEMF_IS_OPTION
- size (real): 4 (4)
- value: '0x2000809'
-----
CFGID_LAST_VERSION_URL:
- flags: ITEMF_IS_OPTION
- size (real): 31 (31)
- value: 'http://alazqwryx.cn/z12/bot.exe'
-----
CFGID_URL_SERVER_0:
- flags: ITEMF_IS_OPTION
- size (real): 32 (32)
- value: 'http://alazqwryx.cn/z12/gate.php'
-----
CFGID_HTTP_FILTER
- flags: ITEMF_COMPRESSED|ITEMF_IS_OPTION
- size (real): 133 (161)
- value: '\xdb\xff\xff\xff!*.microsoft.com/*\x00!http://\tm\xd6}\xed\xfdyspace\x16\x15\x14s\x15w\x00\xb0\xbf\xbd\xfd.gru
p1ant\x02der.es5\xc0\xda\xff?odnoklassniki.)\x1b\xfb?\xbb{vko3kte\x16@*/login.K0`Ovmp\x12atL\x10\x00\x00\x00\x00\x00\x00
\x00H\x00\xff'
-----
CFGID_HTTP_POSTDATA_FILTER:
- flags: ITEMF_IS_OPTION
- size (real): 36 (36)
- value: 'http://bank.pl/*\x00username;password\x00\x00'
-----
enisa@enisa-VirtualBox:~/enisa/forensic/data$

```

Figure 9: Calculating hashes sum

As we can see (Figure 9) the configuration file contains 5 sections.

In the first section, CFGID\_LAST\_VERSION there's the information about the bot version. In this case the value reads 0x02000809, which we interpret as 2.0.8.9 version.

In the two sections following we can see the URL addresses to the bot binary file (bot.exe) and the reporting location (gate.php). The former address is used by malware to report to the C&C and send any data stolen from infected systems.

Note however, that this interpretation is based not on reverse engineering this exact binary, it's based on our expert knowledge of ZeuS source code and documentation.

The fourth section of the config file contains compressed contents (note the ITEMF\_COMPRESSED flag), which is not supported by the zdecrypt script – we still can read some of the URLs stored there. This section lists the pages ZeuS is supposed to inject its code to. For us, the most interesting section however is the fifth section – containing instruction what data should be stolen from one of websites. Website address is anything beginning with 'http://bank.pl/' and there are two fields to be captured whenever they occur within the system: username and password.

We now are sure, that this is piece of malware is a ZeuS variant, version 2.0.8.9 and its configured specifically to target our bank (bank.pl) and capture usernames and passwords of the users.

## **7 Summary of the exercise**

In this exercise we performed server-side analysis of a bank fraud case. You learned basic principles of evidence collection.

You learned how evidence can be extracted from system logs, what to expect from the extracted logs and that logs may become a more valuable source of information if configured with incident response in mind. The process of identifying suspicious activity by careful log examination and correlation is now one of the most basic forensic processes in any financial institution, telecommunication company or a cloud/hosting provider. Systems used for log collection and correlation are of paramount importance given the amount of information that must be gathered to assure compliance with the law.

You should also become aware that attention to details is key-factor to any forensic procedures, and evidence can be extracted also from the lack of data or from raw data formatting. You should understand the need for looking for data in outside sources, like GeolIP databases, keeping in mind however, that information quality in such databases cannot be usually guaranteed.

In the last task you got some hands-on experience in malware analysis and learned basic principles of malware characteristics such as online configuration files and their encryption or online data submission.

You should now be aware of the complexity of forensic proceedings and understand the legal aspects that drive the requirements.

## **8 References**

1. Council of Europe – Electronic evidence guide version 1.0, 2013,  
([http://www.coe.int/t/dghl/cooperation/economiccrime/cybercrime/Documents/Electronic%20Evidence%20Guide/default\\_en.asp](http://www.coe.int/t/dghl/cooperation/economiccrime/cybercrime/Documents/Electronic%20Evidence%20Guide/default_en.asp))
2. ENISA – Give and Take – Good Practice Guide for Addressing Network and Information Security Aspects of Cybercrime, 2012  
(<https://www.enisa.europa.eu/activities/cert/support/fight-against-cybercrime/good-practice-guide-for-addressing-network-and-information-security-aspects-of-cybercrime>)
3. ENISA – Tools for Gathering evidence  
(<https://www.enisa.europa.eu/activities/cert/support/chiht/gathering-evidence>)

**ENISA**

European Union Agency for Network and Information Security  
Science and Technology Park of Crete (ITE)  
Vassilika Vouton, 700 13, Heraklion, Greece

**Athens Office**

1 Vass. Sofias & Meg. Alexandrou  
Marousi 151 24, Athens, Greece



PO Box 1309, 710 01 Heraklion, Greece  
Tel: +30 28 14 40 9710  
[info@enisa.europa.eu](mailto:info@enisa.europa.eu)  
[www.enisa.europa.eu](http://www.enisa.europa.eu)