







www.enisa.europa.eu



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 <u>www.enisa.europa.eu</u>.

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 Weżgowiec 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 <u>CERT-Relations@enisa.europa.eu</u> For media enquires about this paper, please use <u>press@enisa.europa.eu</u>.

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 Juszczyk from ING Services, Poland, Vincent Danjean from Interpol, Daniel Röthlisberger and Frank Herbert from SWITCH, Switzerland, Adam Ziaja and Dawid Osojca from ComCERT SA, 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

1	Ger	eral Description	2
2		oduction	3
	2.1.1 2.1.2	Principle 1 – Data Integrity Principle 2 – Audit Trail	4
		Principle 2 – Specialist Support	5
		Principle 4 – Appropriate Training	5
	2.1.5	Principle 5 – Legality	5
3	Sce	nario	6
4	Tas	x 1 – Identify characteristics in the HTTP session of fraud	8
5	Tas	k 2 – Identify other attacked customers based on these characteristics	12
6	Tas	k 3 – Hands-on analysis of memory process dump	14
7	Sun	nmary of the exercise	22
8	Ref	erences	23



September 2013

Main Objective	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.											
Targeted Audience	The exercise is intended for CERT staff involved in the process of early frau											
Total Duration	5.5 hours											
	Introduction to the exercise	0.5 hours										
	Task1: Identify characteristics in HTTP session of fraud	1.5 hours										
Time Schedule	Task 2 : Identify other attacked customers based on these characteristics	1.0 hours										
	Task 3: Hands-on analysis of memory process dump	2.0 hours										
	Summary of the exercise	0.5 hours										
Frequency	It is advised to organise the exercise once a year or when new the CSIRT/CERT team.	people join										



1 General Description

The main goal of this exercise is to provide the trainees with technical knowledge of tools and reasoning used in digital forensics. Trainees are required to focus on details during the examination of system data as they craft a script to detect similar events throughout the evidence. They 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 the trainees with basic principles of evidence. At all times the trainees 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.¹

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



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^2 .

Traditionally evidence was gathered in physical form. After the invention of photography it became 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).³

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

guidennes.	
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.

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

³ Wikipedia – <u>https://en.wikipedia.org/wiki/Chain_of_custody</u>



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 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.⁴



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.

⁴ 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

Present the trainees with the following narrative; they should have the same story in their materials:

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 the 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 a 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.

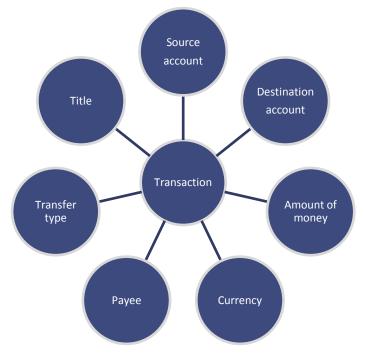
At first, the customer denies access to his/her computer and mobile phone data, so you must utilize any technical means available on the bank's side to find clues about fraudulent transactions. Find data related to the unauthorized transaction.

After the narrative is known, present the participants 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. Now discuss with the participants the sources of the information described above.
- 5. 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.
- 6. 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.
- 7. Now, before the first hands-on task, remind the trainees of the timeline concept and proper evidence evaluation. All the information used must have a valid time stamp, as well as be uniquely identifiable with the electronic banking user. Valid time stamps also mean that the system time is in some way controlled and can be verified to be exact now and in any time in the past.



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.

Trainees 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) GeolP 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) GeoIP database
- 2) 'bad IP' database

To accomplish the task trainees have to find the anomalistic characteristics of the fraudulent session. In this case they need to compare carefully a normal session with the questioned one. The session narrative contains some clues. Allow the groups to search through the data.

Arrange a discussion on the value of gathered information and what to look for in the transaction history of our customer.

Which fields are of interest? Why?

- Source account number although electronic banking systems encourage people to have more than one account, most commonly only one account is used. But we have to use this information as a reference for other cases of unauthorised transfers.
- Destination account number might be a local account or an account in another bank. Except in the easiest of cases, it's unlikely this is the ultimate account of the criminal; it may be the first in a chain of accounts, where the length of the chain makes it difficult to reverse the effects of the fraud. It may be another compromised account, local or in yet another bank.





- Amount the range or a set of sums transferred may be another clue; some banks have different verification policies based on the amount transferred. A criminal may take advantage of this fact.
- Currency may suggest that money is to be transferred abroad.
- Payee a common name or a common pattern may lead us to uncovering more suspicious transactions.
- Transfer type the faster the transfer is the harder it is to withdraw it. A criminal may benefit from this.
- Transfer title same as regarding payee.

Now, arrange a similar discussion regarding data collected from electronic banking logins. What should be looked for in the login history for this particular login?

- Source IP address might have been used more than once. Is it different than in legitimate sessions? If it was the same, the customer's computer may be compromised, or the customer is lying.
- Unsuccessful login attempts did somebody tried to access the account before? Or maybe after user's password was changed?

Maybe GeoIP for source IP address can give us any clues? Were these login attempts coming from a particular region or country?

We know from the Fraud Detection Unit, that the customer questioning has ID **client94777** and the compliant was made by him on 21st 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):



September 2013

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:

geoiplookup 91.226.251.216

GeoIP Country Edition: PL, Poland

geoiplookup 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? Well, the User-Agent field is actually malformed, there is another pair of apostrophes ('') inside double apostrophes (""). It does not happen normally. Usually it means the string was manipulated, possibly from a script. However, it doesn't mean this is bad by itself.

Now, after the case is solved, present the trainees with an alternative 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\""



Identified factors:

Identifying that fraudulent session was connected from Ukraine

The web browser identified itself as Mozilla while the user normally connected from Opera Preferred language was set to Russian

The user agent string was malformed (double quotes used)

A record of information analysed, tools and methods used is maintained



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

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

The teams are to 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.

Arrange a discussion what details should be included on that list to be useful and complete in phone verification.

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

grep Mozilla access.log wc –l
2429
grep ' ru;' access.log wc –l
299
grep \"\' access.log wc –l
155
for i in `cat access.log cut –f 1 –d ' ' `; do geoiplookup \$i; done grep Ukraine wc –l
232
for i in `cat access.log cut –f 1 –d ' ' `; do geoiplookup \$i; done sort uniq wc –l
73
cat access.log wc –l
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 –I counts the lines of input), 299 times the preferred language was Russian, even the malformed header occurred 155 times. There were 232 connections from Ukraine and a total of 3252 connections 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:

grep Mozilla access.log | grep ' ru;' | grep \"\'

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 "-"



September 2013

"'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'"

geoiplookup 62.182.152.59

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 the last information, 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. Again, 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,eidoovoabucoophohzauchahquaiqueeshu

client93597, aethaeqhaevishuveechaishohhoojaedii

client98079,zarozaemoshumeigeughoovauthuziezuxe

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.

Bring to the attention of all trainees that, even if in this example there was a 1:1 match between data sets, we do not need that match in every case to issue an alert. Any money transfer to the same account that follows a fraudulent pattern is suspicious. Every login with a match with a strong connection pattern is suspicious even if there was no transfer of money, it might just have failed for a number of reasons. If all but one suspicious connection were from a single country, VPN or Tor network might have been used.

Success factors:

• Three more suspicious transfers were found

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 a CERT 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. We will 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. 2.0⁵ with zeusscan plugins⁶
- 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.

Remind the trainees 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:

⁵ See: <u>https://www.volatilesystems.com/default/volatility</u> and

https://code.google.com/p/volatility/wiki/CommandReference22

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



September 2013

😣 🗖 🗊 enisa	a@enisa-VirtualBox: ~/enisa/forensic	
	- <mark>VirtualBox:~/enisa/forensic</mark> \$ vol20.py -f memdump.raw ze stems Volatility Framework 2.0	usscan2
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:	Куупокс	
Executable:	Myyw\emneo.exe	
Data file:	Uhan\qidi.dat	
Config RC4 K	Key:	
0x00000000	7a ad dd be 9f b5 30 aa 4a 9a 19 74 31 21 12 ea z	0.Jt1!
0x00000010	1e 75 f1 18 2b a5 06 70 43 a9 41 cf 65 a7 a0 2f .u.	.+pC.A.e/
0x00000020		F
0x00000030		SiV,4>
0x00000040		TPZ.Is]{
0x00000050		wLW\$.m.
0x00000060		.#
0x00000070		N=~?.B
0x00000080		D
0x00000090		^ХК
0x000000a0		j.;.5.f
0x000000b0		_G2yEbM-
0x000000c0		:k3g".
0x000000d0		* . 8Qc .
0x000000e0		v7.)&.n.U`.
0x000000f0		H <ya< td=""></ya<>
0x00000100	00 00	
Credential R		
0x00000000		.u>[;F<~.w.
0x00000010		k+@.=
0x00000020		VH.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:



September 2013

🛛 😣 🖨 🕒 enis	enisa-VirtualBox: ~/enisa/forensic	
Process:	md.exe	
Pid:	432	
Address:	×140000	
URL:	ttp://alazqwryx.cn/z12/config.bin	
Identifier:	IOME-TBZBXUW4F6_B4DF7611AB1DEF24	
Mutant key:	xCCD6423C	
XOR key:	x779C1C5C	
Registry:	KEY_CURRENT_USER\SOFTWARE\Microsoft\Moivg	
Value 1:	mfi	
Value 2:	uvua	
r Value 3:	yynokc	
Executable:	yyw\emneo.exe	
Data file:	han\qidi.dat	
Config RC4 k		
0×00000000	a ad dd be 9f b5 30 aa 4a 9a 19 74 31 21 12 ea z0.Jt1!	
0x00000010	e 75 f1 18 2b a5 06 70 43 a9 41 cf 65 a7 a0 2f .u+pC.A.e/	
0x00000020	2 86 a4 c3 a1 9e 11 81 96 c7 25 1f 6f 08 ce 46%.oF	
0x00000030	8 6c 9d 53 69 bf bd 56 2c 34 e0 a2 91 cb e6 3e .l.SiV,4>	
0x00000040	2 80 88 05 1a 54 50 5a 9b 49 73 5d 7b da ba de RTPZ.Is]{	
0x00000050	1 7d 36 77 4c ff ab 87 f2 b0 04 57 24 09 6d c9 .}6wLW\$.m.	
0x00000060	d fa 4f 8e 23 ae f7 9c 8b 2e 8a 89 1d 0d b6 c00.#	
0x00000070	1 14 d4 4e a8 82 1b 3d bb 99 7e 3f 15 42 d8 d9N=~?.B	
0x00000080	1 00 0f 84 ec 03 44 c1 20 df 83 f3 5c b2 f9 e3D	
0x00000090	1 72 1c f5 fd 07 e1 c2 b7 e7 7f fc f4 5e 58 4b gr^XK	
0x000000a0	b 00 40 93 c8 d3 cd 92 97 6a d6 3b ef 35 bc 66 [.@j.;.5.f	
0x000000b0	9 db 68 5f 47 32 8c dc ca 79 45 62 f0 d0 4d 2dh_G2yEbM-	
0x000000c0	4 a3 90 3a 6b 85 c6 33 67 d2 e2 e4 f6 eb 22 b8 d:k3g".	
0x000000d0	c 8f 98 e8 7c 13 16 94 e5 fe 2a fb 38 51 63 b9 *.8Qc.	
0x000000e0	d 39 27 76 d5 0a 37 af 29 26 d7 6e cc 55 60 0e .9'v7.)&.n.U`.	
0x000000f0	8 17 78 c5 95 48 b4 a6 0c 3c 59 b3 10 c4 ee 61 (.xH <ya< td=""><td></td></ya<>	
0x00000100	0 00	
Credential F		
0x00000000		
0x00000010	4 e5 38 84 75 3e 0d ca 5b 3b 46 3c 7e fa 77 90 d.8.u>[;F<~.w. 2 29 54 6b aa 0c 2b 1c d7 8a 40 de 3d eb 8b d3 B)Tk+0.=	
0x000000020	a f3 11 56 cd e0 48 dc 68 4b 4d 85 b4 dd 27 98 J.V.H.hKM'.	
0x00000030	2 34 f0 81 a5 2c 32 df b8 d4 1b 94 b3 1f 03 80 .42	
0x00000040		
0x000000050		
	0 db d6 bf 52 e4 57 86 ab fd 8d 9a 59 04 19 b6R.WY	
0x00000060	f 95 5f 69 14 a6 c3 92 6f f8 55 da 01 31 16 79io.U1.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:



isa@enisa-																					
00000000							b5											z0			
00000010							a5											.u+			
00000020							. 9e														
0000030							bf											.l.Si			
00000040							54											RTP			
00000050							ff											.}6wL			
00000060							ae											0.#			
00000070							82											••••N•••••			
00000080							03											D			
00000090			-				07							_				qr			
000000a0							d3											[.@			
000000000							32											h_G2.			
0000000000							85											d:k			
000000d0							13											· · · · · · · · · ·			
00000e0							0a											.9'v7			
00000f0							48						10	c4	ee	б:	1	(.xH.	. <y< td=""><td>а</td><td></td></y<>	а	
isa@enisa-	-Vi	гt	ua	LBo	x:~	/en	isa	/foi	ren	sic/	/da1	:aŞ									

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

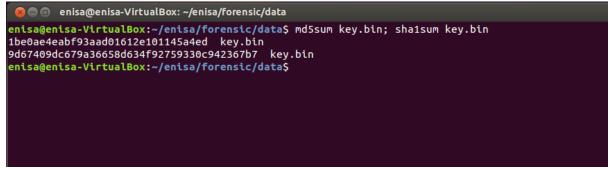


Figure 4: Calculating hash sums

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):

😣 🗐 🗊 network.pcap [Wiresh	ark 1.8.2]		
🗐 🖏 🗐 🗐 📦 📔 [🔄 🗶 🦿 🔐 🔍 🔇	> 🗣 ኛ 🧃	L 🗏 🖪 🕒 🗉 🖪 📅 🔛 💺 📀
Filter: http	÷	Expression Cle	ar Apply Save
No. Time Source	Destination	Protocol	Length Info
13 3.365466 192.168.0.	.44 172.27.128.9	HTTP	232 GET /z12/config.bin HTTP/1.1
15 3.370505 172.27.128	8.9 192.168.0.44	HTTP/DL	687 unknown (0x2f)
24 3.463059 192.168.0.	.44 173.194.70.99	HTTP	467 GET /webhp HTTP/1.1
26 3.538411 173.194.70	9.99 192.168.0.44	HTTP	602 HTTP/1.1 302 Found (text/html)
36 3.665349 192.168.0.	.44 173.194.70.94	HTTP	476 GET /webhp?gws_rd=cr HTTP/1.1
77 3.825032 173.194.76	9.94 192.168.0.44	HTTP	968 HTTP/1.1 200 OK (text/html)
83 3.828823 192.168.0.		HTTP	548 POST /z12/gate.php HTTP/1.1
85 3.839159 172.27.128		HTTP	367 HTTP/1.1 200 OK (text/html)
116 28.670173 192.168.0.	.44 239.255.255.250	9 SSDP	175 M-SEARCH * HTTP/1.1
Hypertext Transfer Protocol SIP/NOE Protocol, unknown (0) 200 20 50 50 50 50 50 50 50 50 50 50 50 50 50	00 00 00 00 71 00 01 00	: closeConter	
0120 2d 54 79 70 65 3a 20 61	70 70 6c 69 63 61 74 69	-Type: a pplicat	ti
0130 6f 6e 2f 6f 63 74 65 74 0140 0a 0d 0a 2f ed 04 0a 13	2d 73 74 72 65 61 6d 0d 78 4a 36 72 bf f1 ad 05	on/octet -stream	
0150 01 e0 98 c6 92 f6 81 9c			
0160 91 d7 e5 8b 57 14 e5 73		Ws]Z.	Ĵ.
0170 41 88 79 87 ab 80 le 1c		A.y	
0180 66 00 15 67 8a d5 a2 39 0190 d8 dd 7d ad 01 3d 47 4e	db 1b 47 45 e4 9a 1b 59 ed b0 15 38 2b ca 6c 90	fg9GE }=GN8+.	
01a0 41 a2 8f 8a c8 c8 a7 85		A	
01b0 d5 47 22 26 bb 56 c7 4c	c0 dd c5 b5 f9 e8 c2 f2	.G"&.V.L	
01c0 e5 32 9f 09 27 aa 7c f9		.2'. . }	
01d0 03 39 8e 50 0e 11 7e 3f 01e0 04 29 ce f1 17 a8 0a 7c		.9.P~? .T.# .	
01f0 c0 20 aa 06 fa 59 dd 55		Y.Um6	
0200 fc 75 e6 a0 2e 8b 1a 47	47 9f 12 20 57 d2 c1 37	.uG G W.	.7
0210 04 ac c7 d4 d5 b3 55 bc		J.	
ラ 💅 UA/SIP Protocol (uasip), 5 byt	tes Packets: 256 Displayed	: 19 Marked: 0 Load	i time: 0:00 🗧 Profile: Default

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:



<u>O</u> pen	^{Ctrl+O} cap [Wireshar	k 1.8.2]		
Dpen <u>R</u> ecent	🎽 🗠 🔁	🛛 🗶 C 🚊 🔍 🗸 🖇	1 7 1	. 🗏 🗟 🗉 d 🎹 🌌 🍢 😵
<u>M</u> erge			•••	
mport	m 1.111	Ç Ex	pression Clea	r Apply Save
lose	Ctrl+W Source	Destination	Protocol I	Length Info
	Ctrl+S 192.168.0.44	4 172.27.128.9	HTTP	232 GET /z12/config.bin HTTP/1.1
ave <u>A</u> s	Shift+Ctrl+S 172.27.128.9	9 192.168.0.44	HTTP/DL	687 unknown (0x2f)
ile Set	192.168.0.44	4 173.194.70.99	HTTP	467 GET /webhp HTTP/1.1
	173.194.70.9		HTTP	602 HTTP/1.1 302 Found (text/html)
Export Specified Packets	192.168.0.44		HTTP	476 GET /webhp?gws_rd=cr HTTP/1.1
Export Packet Dissections	▶ 173.194.70.9		HTTP	968 HTTP/1.1 200 OK (text/html)
Export Selected Packet Bytes	Ctrl+H 192.168.0.44		HTTP	548 POST /z12/gate.php HTTP/1.1
Export SSL Session Keys	172.27.128.9		HTTP	367 HTTP/1.1 200 OK (text/html)
Export Objects	192.168.0.44	4 239.255.255.250	SSDP	175 M-SEARCH * HTTP/1.1
Aport objects				
	child es on wire (5	5496 bits), 687 bytes capture	d (5496 bits)	
2rint		5496 bits), 687 bytes capture :8d:00 (08:00:27:a8:8d:00), D		46:14:76 (08:00:27:46:14:76)
Print Quit	Ctrl+P CadmusCo_a8: Ctrl+Q Version 4, S	8d:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 (172.27.128	st: CadmusCo_ .9), Dst: 192	.168.0.44 (192.168.0.44)
Print Quit	Ctrl+Q Ctrl+Q ▶ Transmission Control Protocol,	8d:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 (172.27.128	st: CadmusCo_ .9), Dst: 192	.168.0.44 (192.168.0.44)
Quit 0x00000100 0	Cttl+Q CadmusCo_a8: Ctrl+Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol	8d:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 (172.27.128 Src Port: http (80), Dst Po	st: CadmusCo_ .9), Dst: 192	.168.0.44 (192.168.0.44)
Quit 0x00000100 0 Credential RC4	Ctrl+Q Ctrl+Q ▶ Transmission Control Protocol,	8d:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 (172.27.128 Src Port: http (80), Dst Po	st: CadmusCo_ .9), Dst: 192	.168.0.44 (192.168.0.44)
Print Quit 0x00000100 0 Credential RC4 0x00000000 6	Cttl+Q CadmusCo_a8: Ctrl+Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol	8d:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 (172.27.128 Src Port: http (80), Dst Po	st: CadmusCo_ .9), Dst: 192	.168.0.44 (192.168.0.44)
2rint Quit Credential RC4 0x00000000 0x00000000 0x000000010 4	Curl+Q CadmusCo_a8: Ctrl+Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > SIP/NOE Protocol, unknown (0x2 00e0 74 24 52 61 66 67 65 73	8d:00 (08:00:27:a8:8d:00), D frc: 172.27.128.9 (172.27.128 , Src Port: http (80), Dst Po 2f) 3a 20 62 79 74 65 73 0d t-f	st: CadmusCo_ 8.9), Dst: 192 ort: 1027 (102 anges : bytes	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint Quit Credential RC4 0x00000000 0x00000000 0x000000010 4	Currer CadmusCo_a8: Ctrl4Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > SIP/NOE Protocol, unknown (0x2 00e0 74 2d 52 61 6e 67 65 773 3 00e0 74 2d 52 61 6e 67 65 67 4	88:00 (08:00:27:a8:84:00), D Frc: 172.27.128.9 (172.27.128 Src Port: http (80), Dst Po 2f) 3a 20 62 79 74 65 73 6d t-f 2d 4c 65 6e 67 74 68 3a .Cc	st: CadmusCo 3.9), Dst: 192 ort: 1027 (102 anges : bytes anges : bytes	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint 22uit Credential RC4 0x00000000 6 0x00000000 4 0x00000020 4 0x00000020 4 0x00000020 4	Ctri+Q CadmusCo a8: Ctri+Q Version 4, S > Transmission Control Protocol. > SIP/NOE Protocol, unknown (0x2 00e60 74 2d 52 61 6e 67 65 73 3 00e76 0a 43 6f 6e 74 55 6e 74 2 00e7 0a 43 64 69 43 6f 6 0 a 43 6f 60 44 36 76 0 a 43 66 48 48 67 0 a 43 66 48 48 68 48 68 0 a 43 66 48 48 68 48 68 0 a 43 66 48 48 68 68 0 a 43 66 68 48 48 68 0 a 43 66 68 48 68 48 68 0 a 43 66 68 48 68 48 68 0 a 43 66 68 48 68 68 0 a 43 66 68 48 68 48 68 0 a 48 66 68 48 68 48 68 0 a 48 66 68 48 68 48 68 0 a 48 66 68 48 68 68 48 0 a 48 66 68 48 68 48 68 0 a 48 66 68 48 68 48 68 0 a 48 66 68 48 68 48 68 0 a 48 66 68 48 68 68 0 a 48 66 68 48 68 68 68 0 a 48 66 68 48 68 68 68 0 a 48 66 68 48 68 68 0 a 48 66 68 68 68 68 0 a 48 66 68 68 68 68 0 a 48 66 68 68 68 68 68 68 0 a 48 66 68 68 68 68 68 68 68 0 a 48 66 68 68 68 68 68 68 68 68 68 68 68 68	88:00 (08:00:27:a8:84:00), D 5rc: 172.77.128.9 (172.27:128 , Src Port: http (88), Dst Po 2f) 3a 20 62 79 74 65 73 0d t f 2d 4c 65 6e 67 74 68 3a c 6e 63 57 469 66 33	st: CadmusCo_), Dst: 192 urt: 1027 (102 Ranges : bytes ontent -Length 64Co nnectio	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint 2uit Credential RC4 0x000000100 0x00000010 40x00000020 0x00000020 0x00000020 0x00000020 0x00000030 0x00000030 0x00000030 0x00000050 2	Carl+Q CadmusCo_a8: Crt+Q Version 4, S > Transmission Control Protocol > SIP/NOE Protocol, unknown (0x2 00660 74 2d 52 61 6e 67 65 73 3 00670 0a 43 6f 6e 74 55 6e 74 2 0060 20 33 63 49 40 6a 43 6f 0 0100 20 33 63 49 40 6a 43 6f 0 0102 2d 54 79 70 65 3a 20 61 7 0122 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0122 8 2d 54 79 70 65 3a 20 61 7 0128 8 2d 56 8 2d 5	88:00 (08:00:27:a8:84:00), D 5rc: 172.27.128.9 (172.27.128 , Src Port: http (80), Dst Po 2f) 3a 20 62 79 74 65 73 0d t-f 2d 4c 65 66 67 74 66 3a 6e 6e 65 63 74 69 6f 6e 3a 43 6f 6e 74 65 66 74 ; c	st: CadmusCo 3.9), Dst: 192 ort: 1027 (102 anges : bytes anges : bytes	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint 2vit 0x000000100 0 Credential RC4 0x000000010 4 0x000000010 4 0x000000010 4 0x000000010 4 0x000000010 6 0x00000050 2 0x00000050 2	Curl+Q CadmusCo_a8: Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > SJP/NOE Protocol, unknown (0x2 00e0 74 2d 52 61 6e 67 65 73 3 00f0 0a 43 6f 6e 74 65 6e 74 2 0100 20 33 63 48 04 6a 43 6f 0110 3a 20 63 6c 6f 73 65 04 6 0120 2d 54 79 70 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 2 0130 6f 6e 2f 6f 65 74 65 74 57 7 000 6f	88:00 (08:00:27:a8:8d:00), D src: 172.27.128.9 (172.27.128.9 src Port: http (80), DSt Pc 2fj 3a 20 62 79 74 65 73 0d t-f 2d 4c 65 6e 67 74 68 3a cc c 2d 4c 65 6e 77 4 68 3a cc c 3a 20 62 79 74 65 73 0d t-f f 2d 4c 65 6e 67 74 68 3a cc c 2d 4c 65 6e 74 69 67 a a 2d 42 67 66 74 69 67 a a 2d 42 67 66 74 65 6e 74 c a 2d 42 67 66 77 66 66 74 c a 2d 42 67 66 77 66 66 74 c a 2d 42 75 74 72 75 61 66 04 67 c a	st: CadmusCo_ .9), Dst: 192 ort: 1027 (102 anges : bytes ontent -Length b4Co nnectio closeConten ype: a pplicat 'octet -stream	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
Print Quit Credential RC4 0x000000100 Credential RC4 0x00000010 40x00000020 0x00000050 0x000000050 0x000000050 0x000000050 0x000000050 0x00000050 0x00000000	Curti-Q CadmusCo_a8: Ctrl-Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > SIP/NOE Protocol, unknown (0x2 00e0 74 2d 52 61 6e 67 65 73 3 00e0 74 2d 52 61 6e 67 65 73 3 00e0 74 2d 52 61 6e 67 65 73 3 00e0 0 63 40 64 67 65 674 2 0100 3a 20 63 6c 6f 73 65 0d 6 0120 2d 54 79 70 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0140 0a 0d 60 2f ed 04 0a 13	88:00 (08:00:27:a8:8d:00), D Src: 172.27.128.9 Src Port: http (80), Dst Pc 2fj 3a 20 62 79 74 65 73 0d t-f 2d 4c 65 6c 67 74 68 3a .cc 5e 6c 65 63 74 69 6f 6e 3a 43 6f 6c 74 65 66 47 45 65 .cc 2d 70 6c 69 63 61 74 65 74 50 .cc .cc 70 70 6c 69 63 61 74 65 .cc .cc 78 43 67 74 55 61 6d 0d .cc .cc 78 78 65 74 65 65 66 77 46 83 a .cc .cc 78 74 76 56 74 65 65 61 74 69 .cc .cc 78 74 75 65 61 6d 0d .cc .cc 78 74 72 65 61 6d 0d .cc .cc 78 74 75 65 74 74 60 74 65 .cc .cc 78 74 72 65 74 65 61 6d 0d .cc .cc 78 74 72 65 74 74 74 75 75 74 74 75 75 74 75 75 74 75 75 74 75 75 75 74 75 75 75 74 75 75 75 74 75 75 75 75 75 75 75 75 75 75 75 75 75	ist: CadmusCo_ .9), Dst: 192 rt: 1027 (102 Canada and and and and and and and and an	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint 2viit 0x00000100 0 Credential RC4 0x00000010 4 0x00000010 4 0x00000010 4 0x00000010 4 0x00000010 6 0x00000010 5 0x00000050 2 0x00000050 2 0x00000050 2	Curl+Q CadmusCo_a8: ∨ersion 4, S ∨ersion 1, S version 1, S version 2, S version 2, S version 2, S version 2, S version 4, S version	88:00 (08:00:27:a8:80:00), D src: 172.27.128.9 (172.27.128.9 src Port: http (80), Dst Pc 2f) 3a 20 62 79 74 65 73 0d t-F 3a 20 62 79 74 65 73 0d t-G 3a 2f d 65 66 77 46 83 aa cc 70 70 65 69 74 65 66 74 46 83 ab cc 3a 43 6f 66 74 65 66 74 46 74 69 2d 73 74 72 65 61 66 0d on, 78 4a 36 72 bf f1 ad 65 on,	st: CadmusCo_ .9), Dst: 192 rrt: 1027 (102 canges : bytes rotent -Length 54Co nnectio closeConten rope: a pplicat foctet -stream /	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
zint 2uit Credential RC4 0x00000000 6 0x00000000 6 0x00000020 4 0x00000020 4 0x00000020 4 0x00000050 2 0x00000050 2 0x00000050 5 0x00000050 5 0x000000000 5 0x000000000 5 0x000000000 5 0x00000000 5 0x000000000 5 0x000000000 5 0x00000000 5 0x000000000 5 0x000000000 5 0x0000000000000000 5 0x0000000000 5 0x000000000 5 0x000000000 5 0x0000000000 5 0x000000000000000 5 0x00000000000000000000000000000000000	Curti-Q CadmusCo_a8: Crti-Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > STP/NOE Protocol, unknown (0x2 000e0 74 2d 52 61 6e 67 65 73 3 00f0 0a 43 6f 6e 74 65 6e 74 2 0100 20 33 36 34 6d 0a 43 6f 6 0110 3a 26 63 6c 6f 73 65 0d 0 0120 2d 54 79 70 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0140 0a 0d 0a 2f ed 04 0a 13 7 0150 01 e6 98 66 92 ff 81 9c f 0160 91 df e5 8b 57 14 e5 73 35	88:00 (08:00:27:a8:8d:00), D 5rc: 172.27.128.9 172.27.128.9 (172.27.128 5rc: 172.27.128 2f) 3a 3a 20.62 2d 46.56 65.66 67.74 2d 46.56 66.69 74.65 9.43 66.67 9.43 67.465 70 70.67 65.61 70.70 76.56 16.40 78.4a 36.72 65.11 9a 34.96 64.93 78.4a 36.72 65.61 74.60 9a 34.96 64.92 76.73 74.72 75.73 76.73 74.75 75.61 64.00 9a 34.04 2d 16.80 16.018	ist: CadmusCo_ .9), Dst: 192 rt: 1027 (102 Canada and and and and and and and and an	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
2rint Quit Credential RC4 6x000000100 0 Credential RC4 6x00000000 4 6x00000000 4 6x000000030 6 6x00000040 0 6x00000040 6 0x00000060 a 6x00000060 a 6x00000060 7 50x00000080 0 0x000000030 7 0x000000030 3	Curti-Q CadmusCo_a8: Crti-Q Version 4, S > Transmission Control Protocol, > Hypertext Transfer Protocol > STP/NOE Protocol, unknown (0x2 000e0 74 2d 52 61 6e 67 65 73 3 00f0 0a 43 6f 6e 74 65 6e 74 2 0100 20 33 36 34 6d 0a 43 6f 6 0110 3a 26 63 6c 6f 73 65 0d 0 0120 2d 54 79 70 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0140 0a 0d 0a 2f ed 04 0a 13 7 0150 01 e6 98 66 92 ff 81 9c f 0160 91 df e5 8b 57 14 e5 73 35	88:00 (08:00:27:a8:84:00), D Src: 172.27.128.9 (172.27.128 Src Port: http (80), Dst Po Str 20 62 79 74 65 73 0d Str 20 62 79 74 65 73 0d t-f Str 20 62 79 74 65 73 0d t-f Str 65 66 74 63 64 64 34 Str 66 65 67 74 63 64 64 34 Str 74 72 65 67 74 63 64 74 70 74 72 65 64 74 74 72 75 61 60 74 74 72 75 61 64 74 74 72 75 75 74 74 72 75 75 74 74 74	st: CadmusCo_ .9), Dst: 192 rrt: 1027 (102 canges : bytes rotent -Length 54Co nnectio closeConten rope: a pplicat foctet -stream /	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
Print Quit Credential R44 0x000000100 0 Credential R44 0x00000020 4 0x00000020 4 0x00000020 4 0x00000020 4 0x00000050 2 0x00000050 2 0x00000050 2 0x00000050 3 0x00000050 3 0x00000050 3 0x00000050 3	CadmusCo a8: Ctrl+Q Version 4, S > Transmission Control Protocol. > SIP/NOE Protocol, unknown (0x2 Obee 74 2d 52 61 6e 67 65 73 3 Obee 0 a43 6f 6e 74 65 6e 74 2 0000 20 33 63 49 40 84 36 f 0110 3a 20 63 6c 6f 73 65 00 4 0120 2d 54 79 79 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0140 8a 00 6a 2f de 40 9a 13 7 0150 6f 6e 2f 6f 63 74 65 74 2 0140 8a 00 6a 2f de 40 9a 13 7 0150 6f 8e 2f 6f 63 74 65 73 5 0160 91 d7 25 8b 57 14 65 73 5 0176 41 88 79 87 ab 80 12 1C 0180 66 90 15 67 8a d5 az 39 c 0180 66 90 15 67 8a d5 az 39 c 0190 68 d4 7d ad 01 3d 74 es	88:00 (08:00:27:a8:84:00), D Src: 172.27.128.9 (172.27.128 Src Port: http (88), Dst Po Std Sc Sc Port: http://sc Std Sc Sc Sc Fort: Sc Std Sc Sc Sc Fort: Sc Sc Std Sc Sc Sc Sc Sc Sc Sc Std Sc Sc	st: CadmusCo_ .9), Dst: 192 rt: 1927 (192 	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
Print Quit Credential RC4 0x000000100 0x00000010 0x00000010 0x00000010 0x00000010 0x00000010 0x00000010 0x00000010 0x00000000	Curt+Q CadmusCo_a8: Crt+Q Version 4, S Transmission Control Protocol. Hypertext Transfer Protocol SJP/NOE Protocol, unknown (0x2 OB0e0 74 2d 52 61 6e 67 65 73 3 OB0f0 0a 43 6f 6e 74 65 6e 74 2 O100 20 33 63 40 40 a+3 6f 6 O110 3a 20 63 6c 6f 73 65 0d 2 O120 2d 54 79 70 65 3a 20 61 7 O130 6f 6e 2f 6f 63 74 65 74 2 O140 0a 0d 0a 2f ed 04 0a 13 7 O150 01 e0 98 66 92 f 68 19 c f O160 91 d7 e5 8b 57 14 e5 73 5 O170 41 88 79 87 ab 80 1e 1c c O180 66 00 15 67 8a d5 a2 39 0 O190 48 dd 7d ad 01 3d 47 4e e O180 46 74 af 86 ac 86 a7 85 2	88:00 (08:00:27:a8:80:00), D src: 172.27.128.9 (172.27.128.9 s Src Port: http (80), Dst Po 2f1 3a 20 62 79 74 65 73 0d t-F 2d 4c 65 6e 67 74 68 3a 6 3a 20 62 79 74 65 73 0d t-F 2d 4c 65 6e 67 74 68 3a 6 3a 3a 66 6e 74 65 6e 74 c 3a 20 73 74 72 65 61 6d 0d on, 20 73 74 72 65 61 6d 0d on, 78 4a 36 72 65 61 6d 0d on, 78 4a 36 72 65 61 6d 0d on, 30 a 49 04 2d e1 6a b0. 5d 01 8b 9e 5a 03 55 1a 31 99 1e 16 77 9d 0d 35 5, A, y 3b 1b 47 45 e4 9a 1b 59 7, f. 2e 37 72 c2 35 a2 c7 8d A, A,	st: CadmusCo_ .9), Dst: 192 rrt: 1927 (182 ntent - Length s4connectio close. Conten r/pe: a pplicat roctet - stream /	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633
Print Quit 0x00000100 0 Credential R44 0x00000010 0x00000010 0 0x00000020 4 0x00000010 0 0x00000010 0 0x00000020 4 0x00000020 4 0x00000010 0 0x00000010 0 0x00000000 0 0x00000000 0 0x00000000 0 0x00000000 3 0x000000000 3 0x000000000 3 0x000000000 3 0x000000000 3 0x0000000000 3 0x0000000000 <	CadmusCo a8: Ctrl+Q Version 4, S > Transmission Control Protocol. > SIP/NOE Protocol, unknown (0x2 Obee 74 2d 52 61 6e 67 65 73 3 Obee 0 a43 6f 6e 74 65 6e 74 2 0000 20 33 63 49 40 84 36 f 0110 3a 20 63 6c 6f 73 65 00 4 0120 2d 54 79 79 65 3a 20 61 7 0130 6f 6e 2f 6f 63 74 65 74 2 0140 8a 00 6a 2f de 40 9a 13 7 0150 6f 6e 2f 6f 63 74 65 74 2 0140 8a 00 6a 2f de 40 9a 13 7 0150 6f 8e 2f 6f 63 74 65 73 5 0160 91 d7 25 8b 57 14 65 73 5 0176 41 88 79 87 ab 80 12 1C 0180 66 90 15 67 8a d5 az 39 c 0180 66 90 15 67 8a d5 az 39 c 0190 68 d4 7d ad 01 3d 74 es	88:00 (08:00:27:a8:84:00), D Src: 172.27.128.9 (172.27.128.9 Src Port: http (88), Dst Po Sf1 Strong State Strong State Sa 20 62 79 74 65 73 6d t f Sa 20 62 79 74 65 73 6d t f Sa 20 62 79 74 65 73 6d t f Sa 43 6f 6e 74 65 6e 74 63 6d 70 70 6c 69 63 61 74 67 6e 74 state 70 70 6c 69 63 61 74 65 64 65 74 73 65 61 6d 6d 90 79 34 69 74 2d 61 6a 56 54 50 88 96 95 16 14 6d 55 56 51 98 82 74 65 74 69 63 61 74 65 56 50 18 89 96 53 63 55 16 4d 6d 54 31 99 1e 16 57 49 63 63 55 54 31 99 1e 16 57 49 66 35 63 52 bc 36 c6 96 1 32 bb 15 47 42 52 52 ca 6c 96 1 32 bb 16 57 98 82 27 68 56 60 dc 55 55 79 88 c2 72 2	st: CadmusCo_ .9), Dst: 192 rt: 1927 (192 	.168.0.44 (192.168.0.44) 7), Seq: 1, Ack: 179, Len: 633

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



😣 🗖 🗊 config	.bin	- GH	ex															
000000D822	0D	ΘA	41	63	63	65	70	74	2D	52	61	6F	67	65	73	34	20	"Accept-Ranges:
000000EA62		74	65	73	0D		43		6E			6E	74			65		bytesContent-Len
000000FC67	74	68	ЗA	20	33	36	34	0D	0A	43	6F	6E	6E	65	63	74		gth: 364Connecti
0000010E6F	6E	ЗA	20	63	6C	6F	73	65	0D	0A	43	6F	6E	74	65	6E	74	on: closeContent
000001202D	54	79	70	65	ЗA	20	61	70	70	6C	69	63	61	74	69	6F	6E	-Type: application
000001322F	6F	63	74	65	74	2D	73	74	72	65	61	6D	0D	0A	0D	0A	2F	/octet-stream/
00000144 <mark>ED</mark>	04	0A	13	78	4A	36	72	BF	F1	AD	05	01	E0	98	C6	92	F6	xJ6r
00000156 <mark>81</mark>	9C	F9	Α3	40	04	2D	E1	6A	B0	91	D7	E5	8B	57	14	E5		@jWs
00000168 <mark>5D</mark>	01	8B	9E	5A	03	55	1A	41	88	79	87	AB	80	1E	10	01]Z.U.A.y
0000017A1E	16	B7	D9	D6	35	66	00	15	67	8A	D5	A2	39	DB	1B	47	45	5fg9GE
0000018CE4		1B	59	D8	DD	7D	AD	01	3D	47	4E	ED	B0	15	38	2B	CA	Y}=GN8+.
0000019E6C	90	41	A2	8F	8A	C8	C8	A7	85	2E	E3	77	E2	A5	A2	CF		l.Aw
000001B0D5	47	22	26	BB	56	C7	4C	~~	DD	C5	B5	F9	E8	C2	F2	E5	32	.G"&.V.L2
000001C29F	09	27	AA	70	F9	7D	FC	EB	B2	15	92	FB	20	03	39	8E	50	'. .},.9.P
000001D40E	11	/E	3F	E9	54	16	23		AC	F8	DC	04	29	CE	F1	1 17 A8~?.T.#)		
000001E6 <mark>0A</mark>	10	/٢	41	90	CF	A9	RT	0B	90	0	20	AA	00	FA	59	DD	55	. .0kY.U
Signed 8 bit:	-61						Sign	ed 3	2 bit:	19	195						kadecim	nal: C3
Unsigned 8 bit:	195					U	nsign	ed 3	2 bit:	19	195						Oc	tal: 303
Signed 16 bit:	195						Flo	oat 3	2 bit:	2,7	7325	32e-4	43				Bina	ry: 11000011
Unsigned 16 bit:	195						Flo	oat 6	4 bit:	9,0	5342	80e-3	322			Strea	am Leng	th: 8 – +
🕑 S	how	little	end	ian d	leco	ding							Sho	wur	nsign	ed a	nd float	as hexadecimal
Offset: 0x2AE; 0x	16C	byte	s fro	m 0x	143	to Ox	2AE	sele	cted									

Figure 7: Editing config file in hex-editor

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.

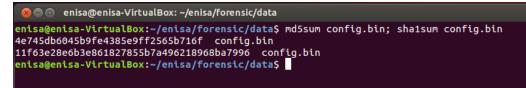


Figure 8: Calculating hashes sum

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:



September 2013

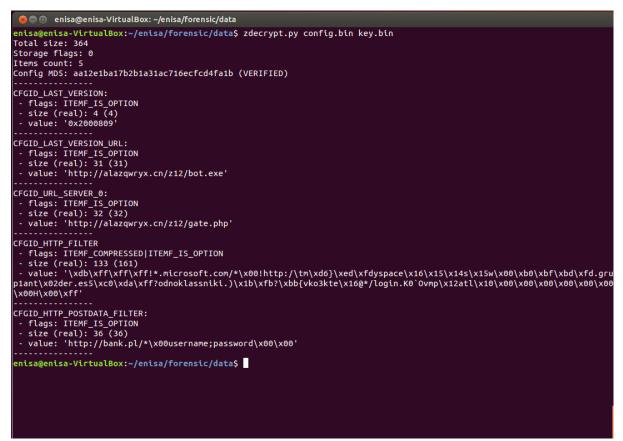


Figure 9: Decrypting Zeus config file

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 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. Trainees learned basic principles of evidence collection. At the end you should discuss the Chain of Custody documented by the trainees.

The trainees 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.

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

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

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





8 References

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



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 www.enisa.europa.eu