# *Identification and handling of electronic evidence*

## *Toolset, Document for students*

September 2013

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 www.enisa.europa.eu.

## Authors

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

Don Stikvoort and Alan Thomas Robinson 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.

## Contact

For contacting the authors please use CERT-Relations@enisa.europa.eu

For media enquires about this paper, please use 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 Juszczyk from ING Services, Poland, Vincent Danjean from Interpol, Daniel Röthlisberger and Frank Herbert from SWITCH, Switzerland, and Dawid Osojca from ComCERT SA, Poland.

# Table of Contents

| | | |
|---|---|---|
| Main Objective | Present the trainees with the principles of evidence gathering. Establish a common knowledge of the requirements regarding evidence admissibility in a court of law. 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 dedicated for CERT staff involved in process of gathering data from customer devices during fraud investigations. It applies especially to events when there is a possibility of further legal actions to happen. | |
| Total Duration | 4.5 hours | |
| Time Schedule | Introduction to the exercise | 0.5 hour |
| | **Task1:** Cold clone a workstation | 1.0 hour |
| | **Task 2**: Gather live data | 0.5 hour |
| | **Task 3**: Evaluate gathered evidence | 2.0 hours |
| | Summary of the exercise | 0.5 hour |
| Frequency | It is advised to organise the exercise at least once a year or more frequently, when CSIRT/CERT team have information about new fraud(s) targeting the e-banking customer(s) which are not detected by currently working security monitoring. The big value of the regular exercise execution is the identification of the new signature/description of fraud. | |

# 1 General Description

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 environment. As we cannot provide complete prevention against the misuse of e-banking systems, we have to 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.

This exercise presents the trainees with basic principles of evidence gathering in a case where a bank customer agrees to collaborate on a phishing investigation. First two phases of evidence gathering are shown: capturing evidence and verifying its applicability to the case under investigation. At all points 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. Gathering evidence by cold cloning the hard disk,
2. Gathering live data from a working system,
3. Checking obtained data for forensic value.

## 2 Introduction

Legal codes are one of the rudiments of any modern civilisation. Along with the developments in law, law enforcement units were created. A court of law is the entity that 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' being 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.* [1]

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, requires 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). [2]

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. |

[1] *Encyclopedia Britannica – http://www.britannica.com/EBchecked/topic/197308/evidence*
[2] *Wikipedia – 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 of 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 require 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 identifying, acquisition, processing, analysis 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.

To provide a formal guidance there are 5 main principles provided to 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, laws regarding admissibility of evidence differ between countries, using these principles is considered appropriate as they are common internationally.[3]


### 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 commencing 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.

[3] *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 access 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.

After we know what are the basic requirements for collecting data to be valid evidence we may begin considering what sources of the evidence we may consider interesting. In the past, this data may have been collected from company or private PCs, servers, and possibly mobile phones – where text messages sent or received and the contents of phonebooks might have been interesting from an investigative point of view. We could also capture solid state data storage – hard disks, magnetic tapes, or CDs, where the latter were especially useful in computer piracy or pornography cases.

Nowadays however, almost every electronic device can be a valuable source of electronic evidence. The proliferation of high capacity consumer devices – smartphones, digital cameras, pendrives, or

even the recent innovation of electronic glasses results in a whole universe of devices where traces of activity can be collected from. With the geolocation features and many automated actions taken by the devices – automatic queries and notifications sent by the device to cloud services for nearby attractions for instance or to social media – there is a whole trail of traces to be found around the Internet.

For many years the first thing that was advised when collecting evidence was to unplug the system from the electric power source. We should remember that shutting down a system – be it by issuing the shutdown command or using a physical switch, the 'ATX power switch', modifies data in the storage, possibly wiping out valuable evidence. This can be an effect of legitimate system clean-up or a result of malicious code left in the system by an attacker just for that purpose.

To understand why we had to change our approach to a forensic investigation in a digital world let's divide electronic evidence into two categories – solid and volatile.

We assume that solid digital evidence is what we can retrieve from a computer system after power is cut off and volatile digital evidence is the evidence that would be lost along with electric power.

In modern investigations we pay much more attention to data that is volatile and would be lost if we cut the power off. Modern malware often downloads much of the data from network, and purposely does not store it. The only way to obtain this type of information is to retrieve it from a live system. Volatile information we may need for investigation resides in computer memory – memory space of processes, system dynamic tables (such as ARP information about active MAC addresses the computer was communicating with, open network ports or ports in closing state). Since encryption is increasingly common in today's computing environment, encryption keys often reside in memory only, and losing them could prevent us from reading disk contents.

# 3 Task 1 – Cold clone a workstation

We are working on the:

1) Customer workstation, possibly a laptop computer

We use tools such as:

1) Standard Unix commands: dd, nc, find
2) Specialized Linux distributions, or
3) Write blockers and Linux

Main goals of Task 1:

1) Gathering evidence while preserving its original form and contents

Let's assume that, during a phishing case investigation, one of the customers identified as hit by phising malware agreed to cooperate with your CSIRT team. You arrange a meeting during which you will be able to examine his personal computer for traces of malicious software.

One of the best ways of ensuring integrity of evidence is to take a full dump (copy) of a computer that is powered off. To do it properly we need to ensure that the tool we use does not modify the harddrive before it gets duplicated. We have to remember, that simply booting another copy of an operating system from a different disk, DVD or a pendrive does not meet this requirement. Modern operating systems tend to make modifications to harddrive partitions before we ever know about it – they perform filesystem checks and mount the drive automatically, to provide 'user friendly feel'.

Since this is not what we need, we have to use one of the following two techniques:

- Use a write blocker – a small device connected between the disk and the computer we want to copy the disk with and any tool able to perform a disk level copy. This is the preferred way, so a team expected to take such copies should have at least one such blocker.
- Use a 'forensic distribution' of Linux or other system, specially crafted with read-only mode. This method is more difficult, as the forensic expert is required to show that it really doesn't modify the disk. Special care must be taken to be sure that the system boots up from the forensic distribution, not from its harddrive.

Since setting up a proper lab is out of scope of this exercise, we assume the use of a 'forensic distribution' tool like Backtrack (a well known distribution with established credibility) or Kali-Linux (gaining in popularity – a distribution from former Backtrack creators).

Disk imaging can take several hours so it's not always possible to perform it. In our case, in the real world the only way we could take a full image of the disk is when we're allowed to take the original disk to a laboratory. As the examination is carried out, at the courtesy of our customer, we don't usually take a full snapshot.

However, for the purpose of the exercise we can image any smaller disk, a pen drive for example. To see what device is the usb thumb drive we can use the udev feature (default in both forensic Linux distributions) and list all USB disks with the following command:

```
# ls /dev/disk/by-id/usb-*
/dev/disk/by-id/usb-Vendor_Model_SID-0:0 -> ../../sdb
/dev/disk/by-id/usb-Vendor_Model_SID-0:0-part1 -> ../../sdb1
```

In this example we can see, there is one USB device: /dev/sdb containing one partition (/dev/sdb1).

You may want to refer to exercise 13[4] for a guidance with mounting a dumped disk.

The standard UNIX way of transferring bit-images is a *'dd'* command. Since we possibly run our forensic tools from a DVD or a pendrive, we don't have enough space to copy a hard disk contents to it. In our example we use another computer which we can plug into the same network as the examined one to establish network storage. We'll use a well known Linux distribution tool called *'nc'* also known as a TCP/IP swiss army knife.

After setting up the network we run the following command on the storage computer:

```
# nc –l –p 23456 > forensic_image.dd
```

It instructs the system to setup a listening nc process (-l parameter) on the TCP port 23456 (any unused port can be put there) and to redirect received the byte stream to a file called 'forensic_image.dd' in the current directory.

On the computer examined we issue the following command:

```
# dd if=/dev/sda | nc storage_IP 23456
```

It copies, byte by byte, the contents of a hard disk /dev/sda (normally the first HDD in the computer, it might be another device in your case) to STDOUT (standard output) and then to the nc tool to forward them to the IP address of the storage computer on port 23456 (this must be the port number given previously).

As we have to ensure integrity of the data, we must verify if the data copied and the original data are the same. Of course comparing the two every time is not practical, so a better approach would be to use a cryptographical sum of the data. Although it is possible for two data sets to have the same sum (sum is much shorter than the data), it is almost impossible to compute a data set for a given sum, and the sums are very sensitive to minor changes – changing only one bit in the data changes the sum completely. For these mathematical reasons we can assume that two data streams with the same sum are the same. To remove all doubt it is advised to compute two different sums. In this example we would use a standard tool 'md5sum' to compute MD-5 hash of the data (MD-5 hashes are commonly used to verify the integrity of software downloaded from Internet). To make the check bulletproof we'll compute a sha512 sum, one of the strongest hashes commonly available. We run the two commands on the source system:

```
# md5sum /dev/sda
f05eb856d20cd6309619644702fa7dce  /dev/sda
# sha512sum /dev/sda
679d85ec31f579ea6a0d8d508951599e91a62b602bc4b9285e9701d8430c6cb648a50273668835b7a
5cad44819c8cc70e01f7517094dccb6aa167483bd73acec  /dev/sda
```

Note how much longer is the newer hash.

We run the same commands on the storage system:

[4] *ENISA CERT Exercises (http://www.enisa.europa.eu/activities/cert/support/exercise)*

| |
|---|
| ***# md5sum forensic_image.dd*** |
| ***f05eb856d20cd6309619644702fa7dce forensic_image.dd*** |
| ***# sha512sum forensic_image.dd*** |
| ***679d85ec31f579ea6a0d8d508951599e91a62b602bc4b9285e9701d8430c6cb648a50273668835b7a5cad44819c8cc70e01f7517094dccb6aa167483bd73acec forensic_image.dd*** |

We compare the two to check they are identicals on both systems. Our copy is an exact copy of the original drive.

We have to save the computed sums in a safe place along with the audit trail documentation. From now on this is the only way to prove that forensic experts were working on the same evidence. If anyone can reasonably doubt the legitimacy of this evidence during judicial proceedings, it will be rejected as evidence.

As maintaining the Chain of Custody Record is essential to a success of legal proceeding it is strongly advised to accept and use a standardised form agreed by both legal and forensic bodies. Such a practice greatly reduces the risk of making errors or omissions when collecting evidence. A very good practical approach to documenting the CoC is presented in the '*Electronic evidence guide*' in Appendix F.[5]

[5] *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)*

# 4 Task 2 – Gather live data

We are working on the:

1) Raw transmission on the network card

We use tools such as:

1) Wireshark (tcpdump or tshark also possible)

Main goal of Task 1:

1) Familiarising with network data caputure as means of data gathering with minimal possible impact on examined computer.

As we now have a full system image stored safely for further examination (or for restoring the computer to its original state) we can move on to the next step. In this step we will try to gather some evidence without installing or running any external software on the computer. While it must be stressed, that powering on the computer and running any software on it will change the system state, some evidence must be gathered in the runtime for behavioural analysis. To make the case acceptable (remember, we've secured the original state) we must mimic a regular users behaviour – by performing typical actions the user would perform on a day to day basis. We make the least possible impact, trying to see if there are any anomalies in system's behaviour. In the very nature of malware we may expect in this case that there is persistence. In all cases but some very specialised and targeted attacks, malware installs a persistent part and tries to hide and stay in the system forever. To make it even worse – there is known malware that activates only when user activity is detected, mouse clicks in case of Trojan.APT.BaneChant for instance. There is virtually no way of capturing malicious activity in that case without simulating a regular user.

There are several methods of capturing live traffic from network:

- Using a network hub. A hub behaves just as a network cable – input on any port is broadcasted to all other ports.
- Using a switch with port monitoring (or port mirroring) feature. This feature allows copying all the traffic present on one port to another, where the capturing device can be connected to.
- Setting up the sniffing computer to be a transparent proxy – copying traffic from the computer on one network interface card to the network on the other card and back.
- Using a specialised device performing one of the above.

Now, open wireshark and watch the traffic when you're surfing the net.

As we cannot provide a copy of an infected Windows operating system, we've prepared a PCAP file with traffic from the actual examined computer. You may find it in /home/enisa/enisa/forensic2/network.pcap file. We did not use the computer during network dump in this case, all the traffic logged is comes from system components or from installed software communicating without user interaction, possibly malicious software.

Load the file to Wireshark:

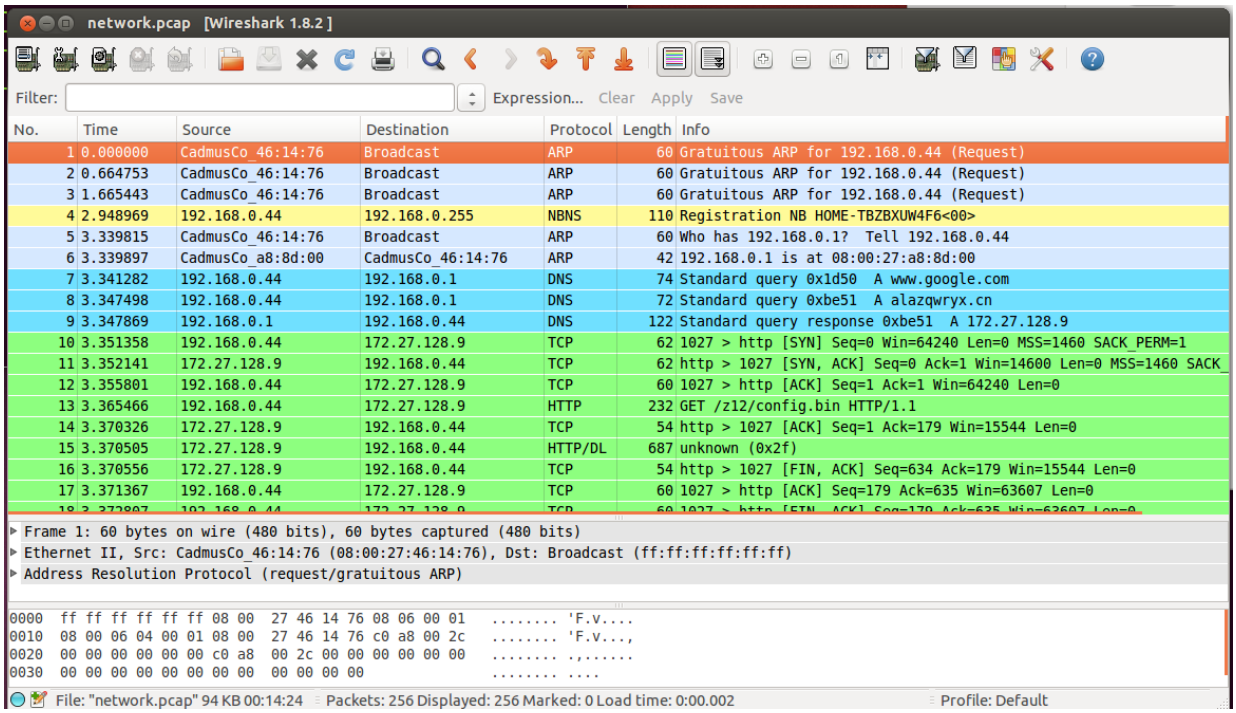


*Figure 1: Network traffic loaded into Wireshark*

This capture contains data from the moments just after the computer was booted. We can see some network activity, so the computer tried to communicate with the external world. After a few ARP packets (the protocol that performs IP address to MAC address translation) some DNS queries were made.

We decide to filter the traffic to DNS only to see what addresses the computer tries to find automatically. We type 'dns' in the 'Filter' field and accept the input by pressing ENTER:

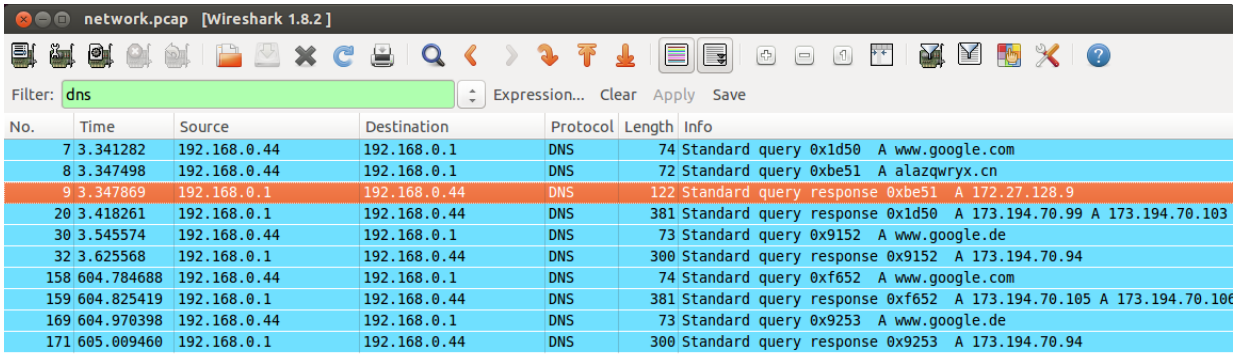


*Figure 2: Possible malware trace in network traffic*

There are Google servers prevailing in the log, but one query catches our attention: The computer queried about the IP address of 'alazqwryx.cn' and got a response with IP address '172.27.128.9', highlighted in this picture. This is one of common patterns used by malware – asking for 'unidentified addresses' which are the most probably botnet C&C servers. C&C servers are contacted by malware for configuration or to send harvested data.

Actually, we can see in previous picture, that the computer was contacting this IP address over HTTP protocol. We modify the filter to see what the communications looked like:

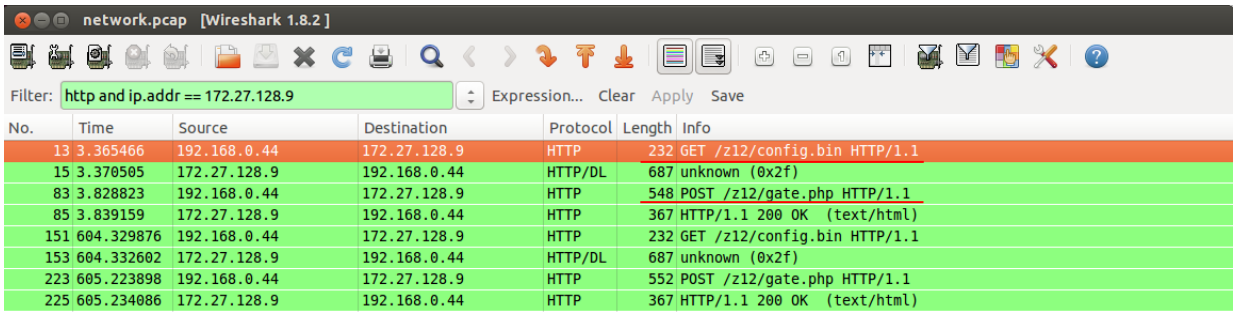


*Figure 3: Suspected malware traffic filtered out*

New filter shows only the HTTP protocol where the IP address in question is used.

We notice two HTTP requests:

**GET /z12/config.bin**

**POST /z12/gate.php**

If we look in the Time column we see that the pattern repeated after roughly 600 seconds (10 minutes).

All this information combined suggests that we have a computer infected with malware and the HTTP requests format suggests this is some ZeuS variant. We have also acquired probable botnet's C&C server IP address.

Now that we know what we're looking for, we have to gather more traces. First, we make a dump of physical memory inside the workstation:

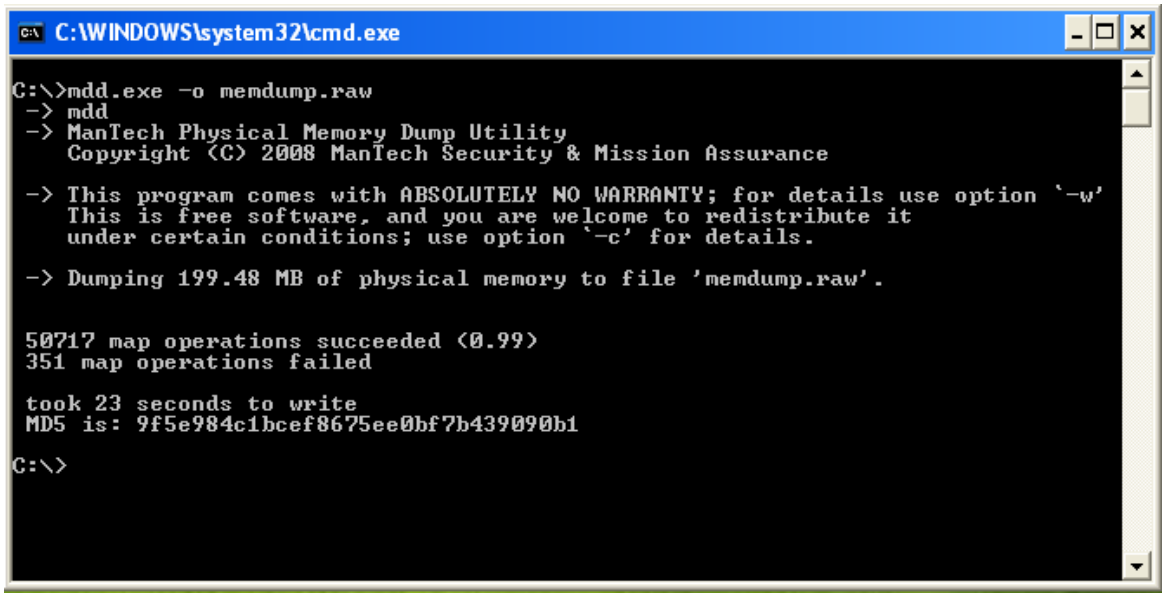


*Figure 4: Acquiring memory*

We use a small and simple tool called mdd[6] This tiny utility copies all the contents of computer's physical memory to a file, memdump.raw in this case. Memory dump from infected workstation is available in file /home/enisa/enisa/forensic2/memdump.raw

[6] *mdd tool can be found at http://sourceforge.net/projects/mdd/*

# 5 Task 3 – Evaluate gathered evidence

We are working on the:

1) Results from Task 1 and Task 2

We use tools such as:

1) Standard command lines utils (bash, grep, sort, cut, etc.)
2) Volatility Framework ver. 2.1[7]

Main goal of the Task 3:

To verify that information gathered from the user computer contains all the information necessary to prove malicious activity and to connect it with information gathered at the bank side.

After we have completed the network traffic analysis, we would like to verify the existence of malicious code from the information gathered. Although we expect to find ZeuS code based on traffic pattern experienced, we must obtain a proof, a code sample in this case. Some malware might be injected to a running system from the network without writing its code to the disk. This is not the way Zeus works, but we've not yet proved it was Zeus, this is our guess.

In the first step we should find the process which was communicating with C&C server. We will use VF (Volatility Framework) commands: connections, connscan, sockets and sockscan.

```
enisa@enisa-VirtualBox: ~/enisa/forensic2
enisa@enisa-VirtualBox:~/enisa/forensic2$ vol.py -f memdump.raw connections
Volatile Systems Volatility Framework 2.1
Offset(V)  Local Address             Remote Address               Pid
---------- ------------------------ ------------------------ ------
0x811d0680 192.168.0.44:1044        172.27.128.9:80              236
enisa@enisa-VirtualBox:~/enisa/forensic2$
```

*Figure 5: Identification of active connections*

```
enisa@enisa-VirtualBox: ~/enisa/forensic2
enisa@enisa-VirtualBox:~/enisa/forensic2$ vol.py -f memdump.raw connscan
Volatile Systems Volatility Framework 2.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x010c4680 192.168.0.44:1044        172.27.128.9:80          236
0x058c3b48 192.168.0.44:1043        173.194.70.94:80         236
0x0a5b1b48 192.168.0.44:1043        173.194.70.94:80         236
0x0c06add8 192.168.0.44:1030        172.27.128.9:80          236
enisa@enisa-VirtualBox:~/enisa/forensic2$
```

*Figure 6: Identification of all connections*

We can see that the computer had an active connection to the IP address 172.27.128.9 (suspected to be a C&C server) on port 80 in the very moment of performing memory dump. Additionally, a connection scan found information about past connections, but still not overwritten by other data. Again there were connections to the suspect, along with the IP 173.194.70.94 port 80 – which is Google.com server. You can check back with the Wireshark output in Task 2 to compare.

Another piece of information we extracted is the windows process ID (PID) that opened the connections: 236. We use the Volatility Framework command pslist:

[7] *See: https://www.volatilesystems.com/default/volatility and https://code.google.com/p/volatility/wiki/CommandReference22*

```
enisa@enisa-VirtualBox: ~/enisa/forensic2
enisa@enisa-VirtualBox:~/enisa/forensic2$ vol.py -f memdump.raw pslist
Volatile Systems Volatility Framework 2.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                  Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------- -------------------
0x8132b020 System                    4      0     54      523 ------      0
0x81231c60 smss.exe                408      4      3       21 ------      0 2013-08-26 15:32:08
0x8119b698 csrss.exe               564    408     10      341      0      0 2013-08-26 15:32:09
0x811f4b00 winlogon.exe            588    408     18      503      0      0 2013-08-26 15:32:09
0x81232020 services.exe            792    588     15      253      0      0 2013-08-26 15:32:09
0x81235020 lsass.exe               804    588     20      331      0      0 2013-08-26 15:32:09
0xffbd5530 VBoxService.exe         988    792      8      107      0      0 2013-08-26 15:32:09
0x81230020 svchost.exe            1060    792     22      215      0      0 2013-08-26 15:32:09
0x81195790 svchost.exe            1172    792      9      239      0      0 2013-08-26 15:32:10
0xffac11d8 svchost.exe            1368    792     59     1139      0      0 2013-08-26 15:32:10
0xffab8688 svchost.exe            1424    792      6       76      0      0 2013-08-26 15:32:10
0x811d7760 svchost.exe            1456    792     14      206      0      0 2013-08-26 15:32:10
0xffa92c08 spoolsv.exe            2008    792     10      106      0      0 2013-08-26 15:32:11
0x811c62f0 explorer.exe            236    216     18      365      0      0 2013-08-26 15:32:12
0xffa7b8c0 VBoxTray.exe            288    236      7       71      0      0 2013-08-26 15:32:12
0xffa7b280 msmsgs.exe              296    236      3      176      0      0 2013-08-26 15:32:12
0xffa7a660 emneo.exe               304    236      0 --------      0      0 2013-08-26 15:32:12  2013-08-26 15:32:13
0xff9d6d08 alg.exe                 868    792      6      104      0      0 2013-08-26 15:32:29
0xffbb7228 wscntfy.exe             200   1368      3       48      0      0 2013-08-26 15:32:30
0x811d5c08 taskmgr.exe             340    588      3       76      0      0 2013-08-26 15:32:35
0x8119a130 cmd.exe                1432    236      1       53      0      0 2013-08-26 15:32:56
0x811e3a58 wpabaln.exe            1356    588      1       77      0      0 2013-08-26 15:34:08
0xffa9f800 mdd.exe                 744   1432      1       41      0      0 2013-08-26 15:42:21
enisa@enisa-VirtualBox:~/enisa/forensic2$
```

*Figure 7: Listing processes*

In the PID column we quickly find that process number 236 is explorer.exe. This is the name of a Windows Explorer executable, the main process of the Windows user interface. Under normal conditions the process should not make any external connections, so we might expect that the binary was modified or some other process injected malicious code into one of explorer's threads. Such an injection is a popular technique used by malicious software to hide its existence.

Another finding is easy to spot here, a process 'emneo.exe' which started and quit almost immediately (this is the only finished process, which makes it easy to find).

Next, we want to find so called 'API hooks'. Hooking is method for modifying a program at runtime. An API hook is basically a way of asking the operating system to call a specific function every time something happens in the system (for example a specific API function is called). One good example is an antivirus registering to be called every time the system or any application wants to open a file from disk. Given the power hooking gives to applications and its creators it's understandable that malicious software uses this technique often.

```
enisa@enisa-VirtualBox: ~/enisa/forensic2
enisa@enisa-VirtualBox:~/enisa/forensic2$ vol.py -f memdump.raw apihooks > data/apihooks.out
```

*Figure 8: Finding hooks*

This command can take several minutes to execute and can produce large amount of output, so we redirected the output to a file 'apihooks.out'.

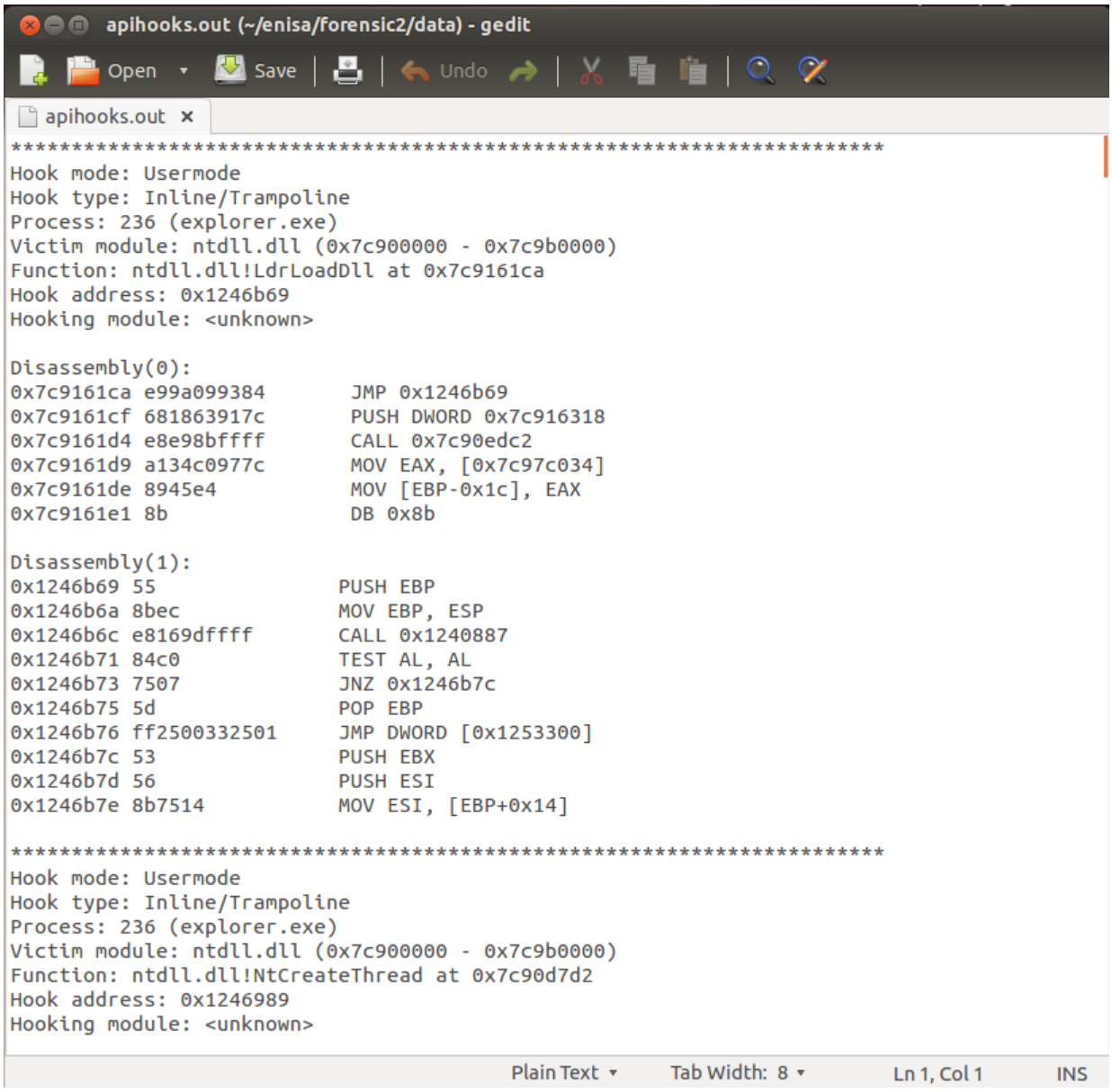


*Figure 9: Investigating hooks*

As we can see after opening the file there are many hooks created at miscellaneous API functions calls.

Let's find what processes memory spaces contained API hooks with something registered:

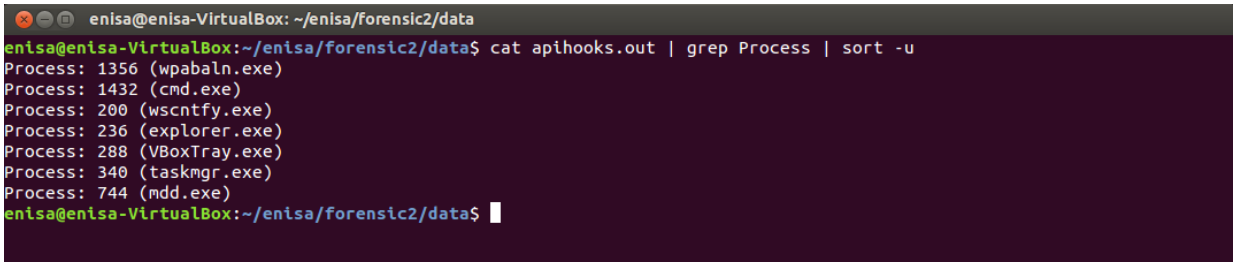


*Figure 10: Listing hooks with registered processes*

As we expected explorer.exe is in that list. We will now extract from the computer memory image the memory space of explorer.exe:

```
enisa@enisa-VirtualBox: ~/enisa/forensic2
enisa@enisa-VirtualBox:~/enisa/forensic2$ vol.py -f memdump.raw memdump -p 236 -D data/
Volatile Systems Volatility Framework 2.1
************************************************************************
Writing explorer.exe [   236] to 236.dmp
enisa@enisa-VirtualBox:~/enisa/forensic2$
```

*Figure 11: Creating process explorer.exe memory dump*

With the extracted memory space and information gathered previously we search the process memory for phrases of interest. First, is there any reference to the bank we are working for, to the C&C server or domain and if we're able to find the mysterious 'emneo.exe'? We will use an internal tool in UNIX system – 'strings', to extract information from a binary file:

```
enisa@enisa-VirtualBox: ~/enisa/forensic2/data
enisa@enisa-VirtualBox:~/enisa/forensic2/data$ strings 236.dmp | grep bank.pl
http://bank.pl/*
enisa@enisa-VirtualBox:~/enisa/forensic2/data$ strings 236.dmp | grep alazqwryx.cn
: alazqwryx.cn
://alazqwryx.cn/z12/config.bin
alazqwryx.cn
alazqwryx.cn
://alazqwryx.cn/z12/gate.php
://alazqwryx.cn/z12/gate.php
http://alazqwryx.cn/z12/config.bin
http://alazqwryx.cn/z12/config.bin
http://alazqwryx.cn/z12/bot.exe#N
http://alazqwryx.cn/z12/gate.php%N
://alazqwryx.cn/z12/gate.php
http://alazqwryx.cn/z12/config.bin
alazqwryx.cn
http://alazqwryx.cn/z12/config.bin
|alazqwryx.cn
http://alazqwryx.cn/z12/config.bin
http://alazqwryx.cn/z12/config.bin
http://alazqwryx.cn/z12/config.bin
http://alazqwryx.cn/z12/config.bin
Host: alazqwryx.cn
Host: alazqwryx.cn
Host: alazqwryx.cn
Host: alazqwryx.cn
Host: alazqwryx.cn
enisa@enisa-VirtualBox:~/enisa/forensic2/data$ strings 236.dmp | grep emneo.exe
emneo.exe
Myyw\emneo.exe
Myyw\emneo.exe
Myyw\emneo.exe
C:\Documents and Settings\Peter Kowalski\Application Data\Myyw\emneo.exe
C:\Documents and Settings\Peter Kowalski\Application Data\Myyw\emneo.exe
C:\Documents and Settings\Peter Kowalski\Application Data\Myyw\emneo.exee
emneo.exe
enisa@enisa-VirtualBox:~/enisa/forensic2/data$
```

*Figure 12: Investigating explorer.exe memory dump*

In all cases the strings we were looking for are present in the memory dump. There's our bank website, there are numerous references to alazqwryx.cn domain and URLs within it. Finally there is a reference to 'emneo.exe' file. We suspect that this is a process run just after powering up the system and then used to inject malicious code into explorer.exe process.

We can now look into the file with hex editor and search for our bank website (we know there is exactly one reference):

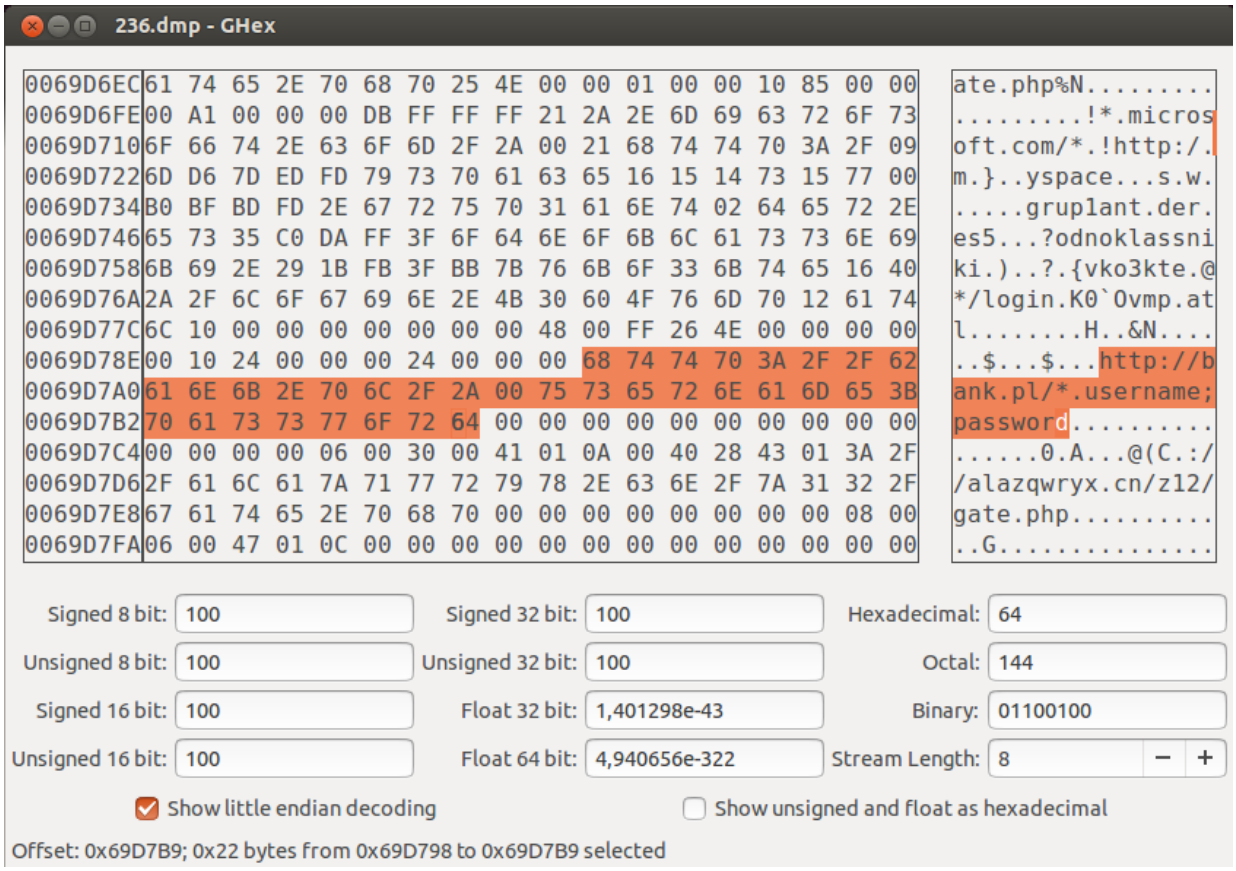


*Figure 13: Using hex-editor*

We found a suspicious line, which we expect to be e-banking URL along with part of the input form instructing the malicious code where to inject its scripts.

Now we may consider the evidence verification complete, we found traces of malicious software, we found proof of connections being made.

However, we mentioned in Task 1 that taking a full image of user computer's hard disk might not be possible or practical. We may want to do one more thing in that case – find and copy files related to the malware found.

There is a prepared list of files on the examined system in '/home/enisa/enisa/forensic2/fs.list'. We verify the location of emneo.exe file:

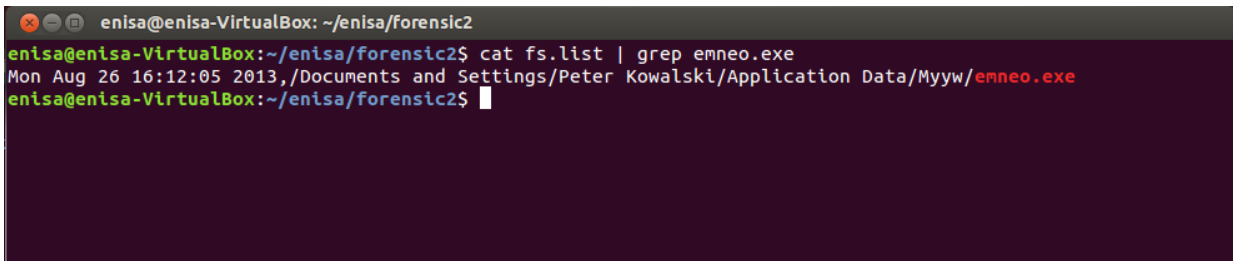


*Figure 14: Finding suspected malware locations*

This is exactly the location we have found in the explorer.exe memory dump. We notice the date and time of file creation: 26th August 16:12:05 2013. This is probably the moment this computer got infected, or the moment malware was last updated. We look for other files created around the same time:

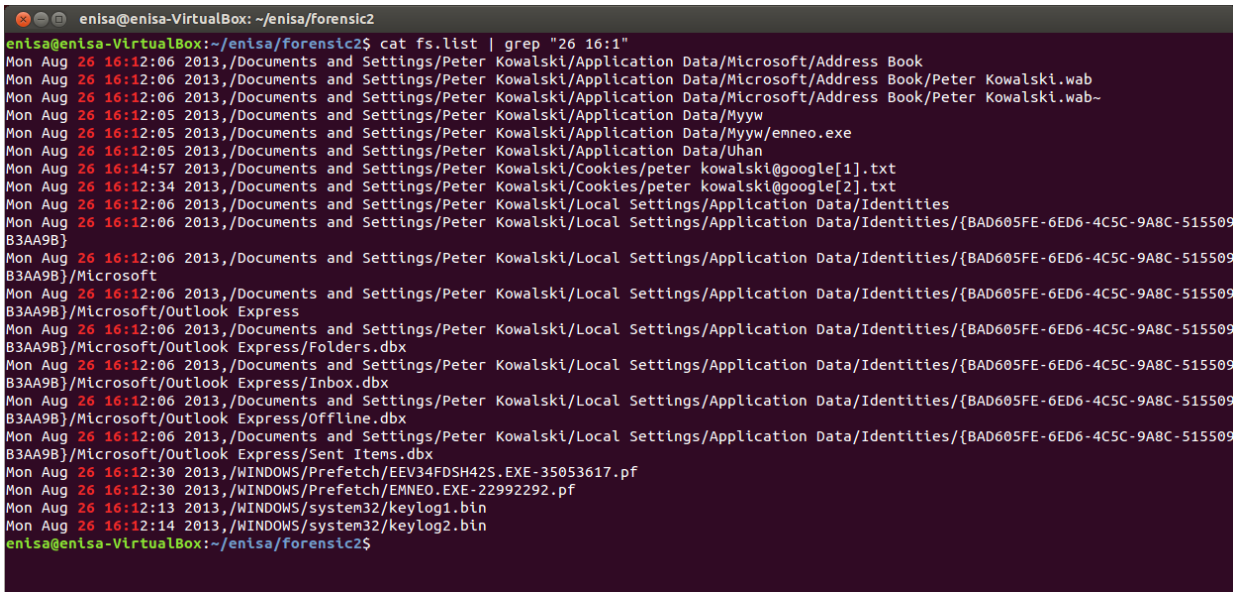


*Figure 15: Listing all files suspected to be malware related*

There are more suspicious files found, there are two files in \WINDOWS\system32 – keylog1.bin and keylog2.bin. Their names suggest they may contain information about keys pressed by the user. All the files found should be secured for further investigation.

# 6 Summary of the exercise

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

You should be now aware of the complexity of forensic proceedings and understand the legal aspects that drive the requirements. Also you should know the basics of computer imaging, network sniffing and network traffic capturing.

The most important outcome from this exercise however, should be the understanding of the forensic process leading from capturing evidence in unaltered form, through the examination of suspicious activities as seen from the outside, possibly from a distance to gathering pieces of evidence from the inside of an operating system. During all the steps the principles were followed, keeping impact on data as low as possible, saving the state of data before altering it and documenting all the alterations, tools and methods used.

Now you are familiar with basic operating system concepts – system memory, process memory, the existence of system structures regarding network connections, file handles. Exploiting these concepts led you to final conclusions including malware location and possible infection time.

# 7 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)
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)
4. ENISA – CERT Exercises – Exercise 13, 2012 (http://www.enisa.europa.eu/activities/cert/support/exercise)


**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