4th Lab

Juan Esteban Murcia y Rodrigo Castillo

September 3, 2020

1 Static Analysis

1.1 Get the hash of the malware and search for it in Virus Total. Is is recognized as a malware for some antivirus?

The library outlook.dll is recognized by virustotal as a malware by 59/70 antiviruses.

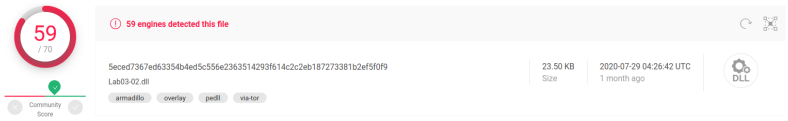


Figure 1: Hash

1.2 Analyze the strings using the command line tool "Strings". Which are the strings more suspicious and why?

inside the list we can find the library *kernel*32.*dll* which is really powefull, also, we can see that is importing *sleep* library, that means that the malware probably is not going to execute inmediatly, sometimes, malwares are programmed with delay with the prupose of being sneaky. Also, it seems to be exporting functions for installing purposes like *installA* , *install* ...

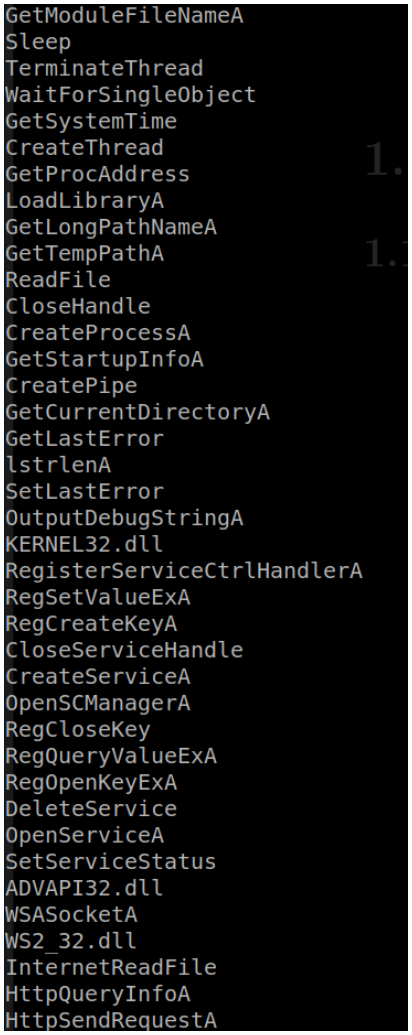


Figure 2: Strings

etc, It's also importing libraries for networking , this can be used by the malware for remotely installing processes later. There is a library that is interesting not because of its potential but it talks about the way that the program was made, the library *malloc* its a *C* library, that means that the pogram was made in C or C++.
The malware is also connecting to an attacker's localhost in *practicalmalwareanalysis.com* .

1.3 Is the malware packed? Analyze it with PEiD. Unpack if possible.

The malware is not packed but its a c++ program .

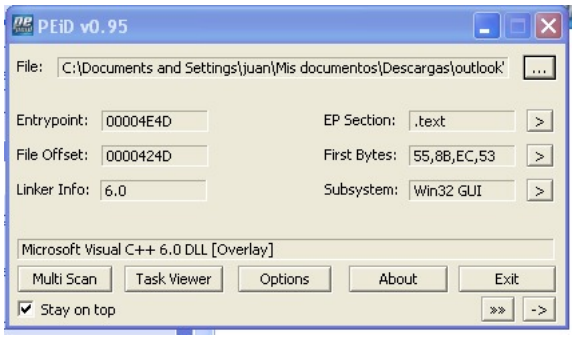


Figure 3: PEid

1.4 Analyze the PE with PEView and review detailed the "text", "data", "rdata" and "resource" sections (Use "Resource Hacker" to access to the resource section). What information is useful from theses sections?

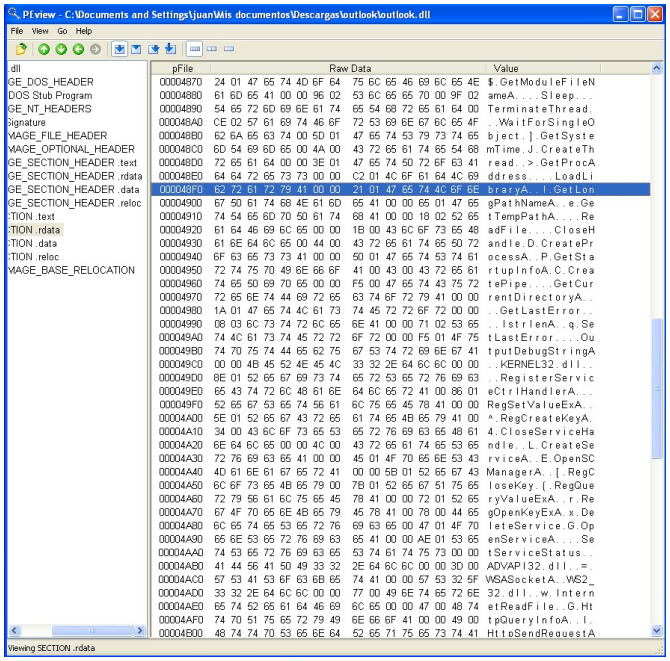


Figure 4: Rdata

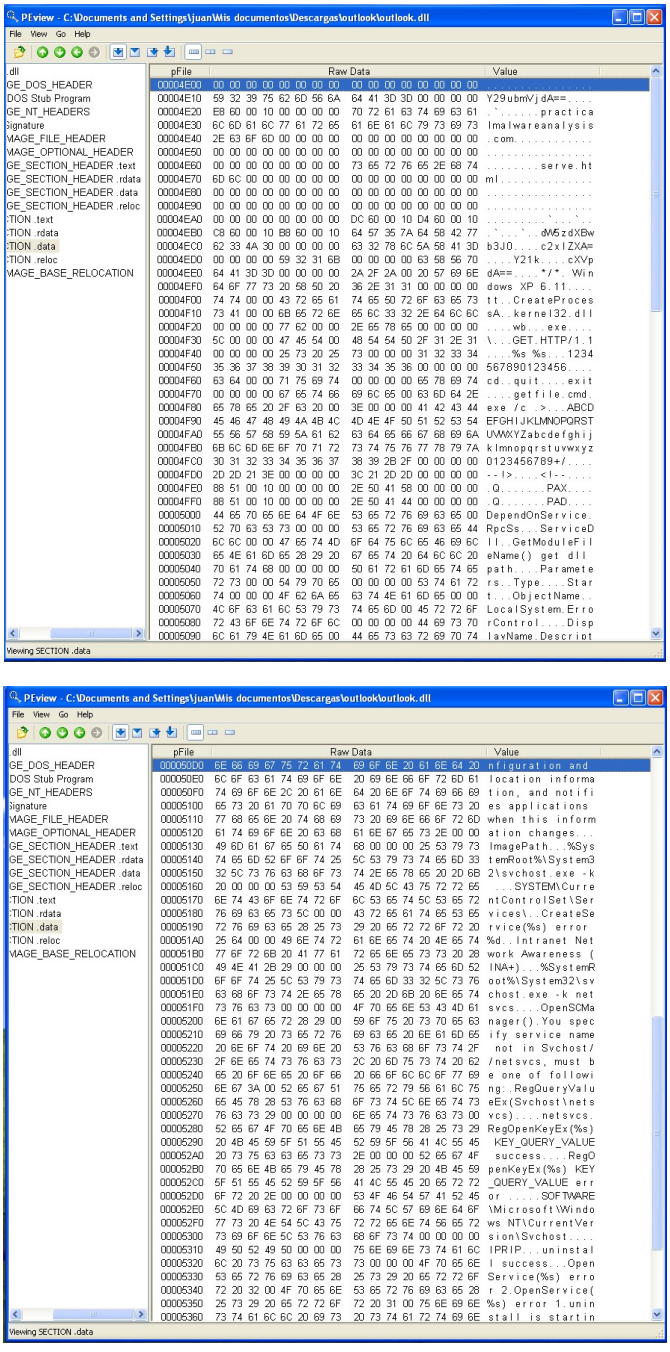
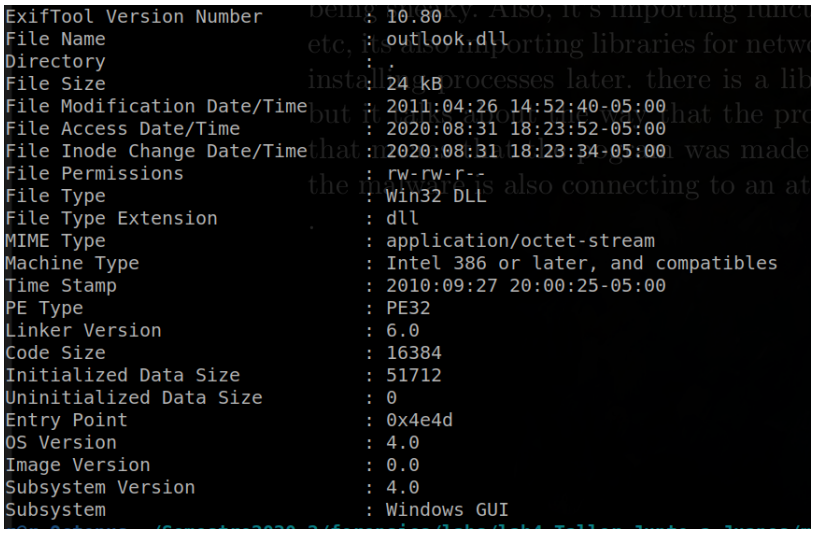


Figure 5: data and text sections

Clearly in this images we can find really important information, like how the malware is opening and using registry keys, create a host service with the name of IPRIP, adding all the dynamic libraries and functions the malware seems to be using.

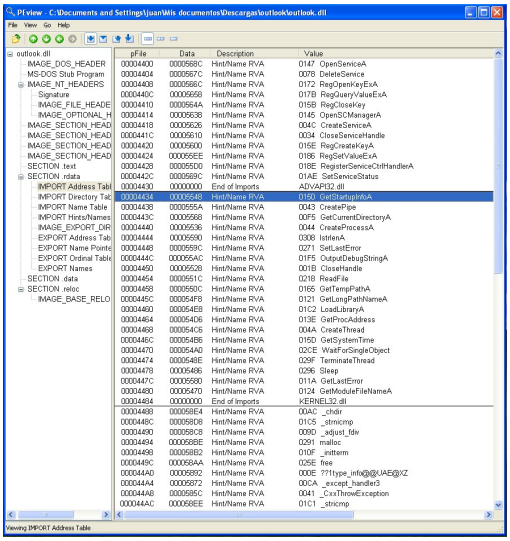
1.5 When the file was compiled?

The file was created on $2010:09:2720:00:25-05:00$



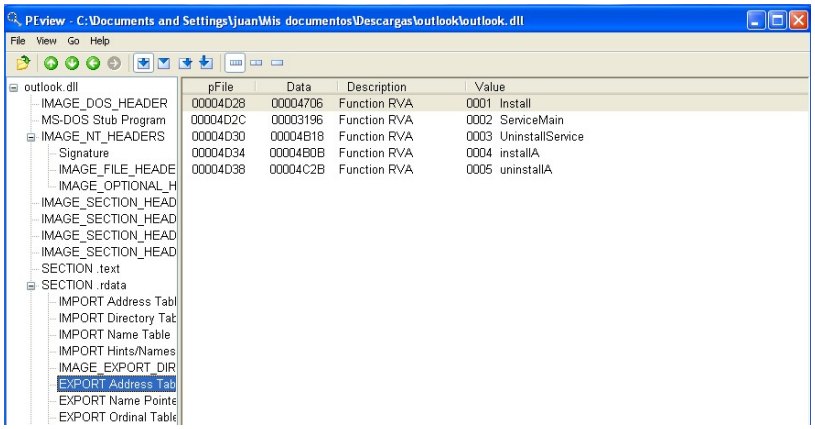
1.6 Which imported libraries and functions can be seen from the static analysis?

Those are the libraries and functions that the program is importing.
From here, we can see very important information such as the keys that the malware is creating, here there are the imported libraries:



1.7 There is some exports?

Yes, there are exports, it exports several functions that seems to have installing and uninstalling purposes.

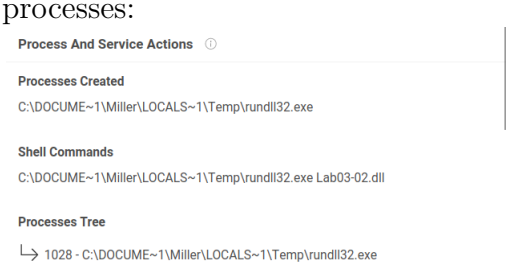


1.8 What may be the functionalities of the malware?

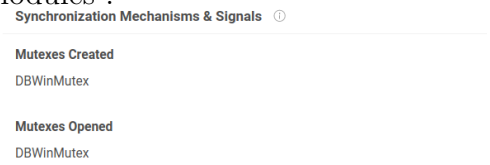
My partner and I think that the malware its a dropper, because it has the capability of install processes that it's going to use later. it also has the capability of create network services using the *srvhost* process. this can mean that an attacker (*practicalmalwareanalysis*) can install other programs in the victim's machine.

1.9 From all previous answers, identify all host-based signatures for this malware.

The next information is all the host based signatures virustotal.com identified, but that we also identified with the static analysis, all these signatures are based on how the malware behaves inside an infected machine, which libraries it import, which functions it exports, among other characteristics.
processes:



modules :



runtime modules :

Modules Loaded ⓘ

Runtime Modules

c:\windows\system32\wininet.dll

c:\windows\system32\imm32.dll

c:\documents and settings\miller\local settings\temp\lab03-02.dll

c:\windows\system32\mpr.dll

c:\windows\system32\comctl32.dll

c:\windows\system32\ws2help.dll

c:\windows\system32\rsaenh.dll

c:\windows\system32\ole32.dll

c:\windows\system32\secur32.dll

c:\windows\system32\msvcrt.dll

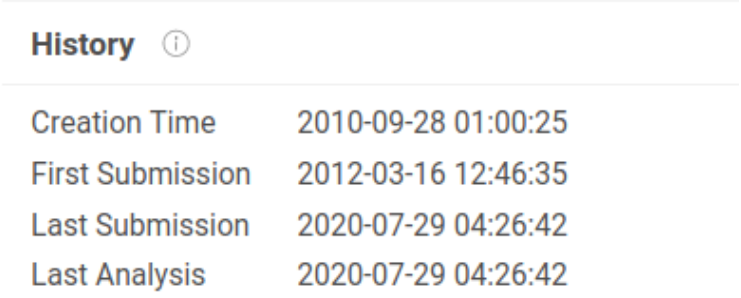
1.10 From all previous answers, identify all antivirus signatures for this malware.

As virustotal.com and we identified, the following information related to the presentation of the malware, however these characteristics can be easily modified by an attacker. Information like its hash values, its size, the names it might have, etc. hashes :

MD5	84882c9d43e23d63b82004fae74ebb61
SHA-1	c6fb3b50d946bec6f391aefa4e54478cf8607211
SHA-256	5eced7367ed63354b4ed5c556e2363514293f614c2c2eb187273381b2ef5f0f9
Vhash	124046655d5550c8z142qz71ze6z5
Authentihash	b76700f50d6f09408958f9e40f562908cd4050e0f992efaec0ca63e0fc9638e0
Imphash	3167552ee0bbbd4f5f440adf5f65bab8
SSDEEP	384:NcTA0TAKHWYvVvUYGXFgeJGjHwTACLPkldSgbl/xAlrWdhoQsxRiAHz:NcTA0TAK2y2oBCbH4gtxrWd5sxRL
File type	Win32 DLL
Magic	PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit
File size	23.50 KB (24065 bytes)
PEiD packer	Microsoft Visual C++ v6.0 DLL

names

history



1.11 From all previous answers, identify all network-based signatures for this malware.

Finally, as we can check in the first image if the figure **??**, the malware has a domain within its data setion, `practicalmalwareanalysis.com`, meaning that the malware is probably going to connect to that url, perform a DNS resolution and communicate with a command&control server.

2 Dynamic Analysis

2.1 Take a snapshot of the register keys before the infection using the application Regshot.

eHre is the snapchot of the register keys

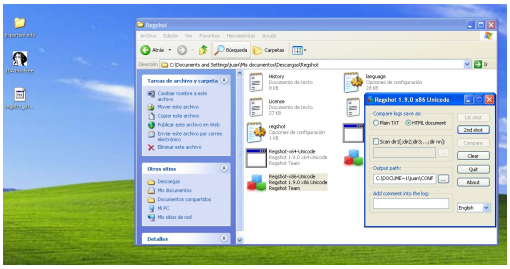


Figure 6: snapchot of the register keys

2.2 Take a snapshot of the virtual machine before the infection

here is the snapchot of the virtual machine before the infection :

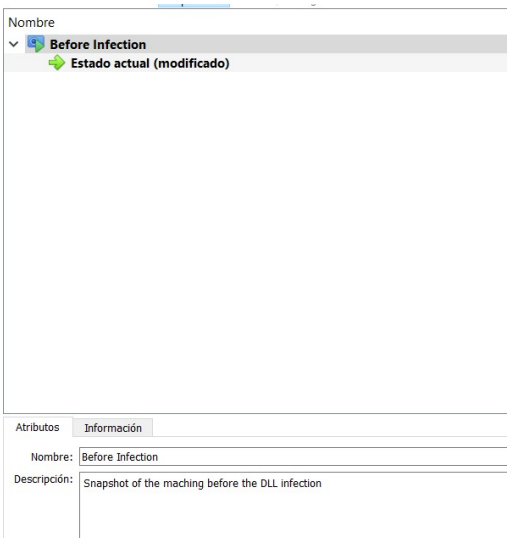


Figure 7: snapchot

2.3 Install the malware using the command: C:¿rundll32.exe outlook.dll, installA

We proceed to install the malware using the command $C :> rundll32.exeoutlook.dll$

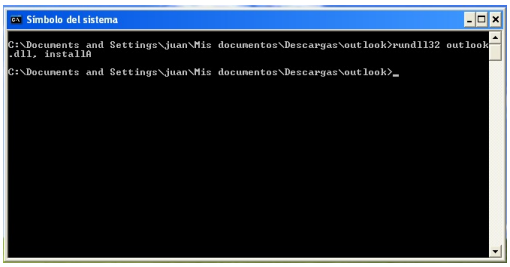


Figure 8: command

2.4 installA was a function identified previously in the static analysis?

Yes, installA was a function we identified with the static analysis inside the strings and in the exports.

2.5 Take a second snapshot of the register keys with Regshot. Identify the changes before and after the infection (keys created, modified, delete, etc)?

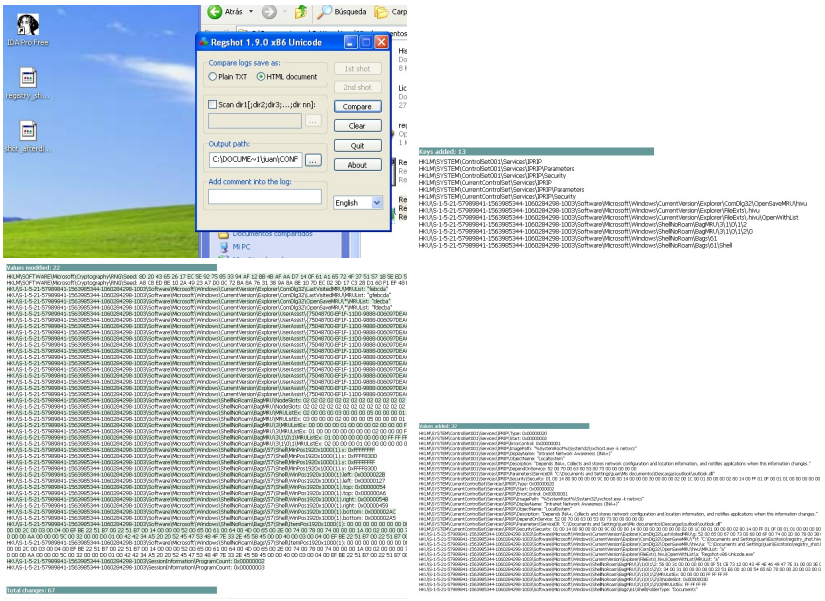


Figure 9: snapchots

- We can see that it included 13 new keys, and a good amount of them involve the new service IPRIP .
- A total of 22 keys values were modified, however the purpose of the modification can't be understood.
- And finally the infected dll added a total of 32 values to exsisting or new keys, most of these new values are for the IPRIP service.

2.6 Analyze in detail the register keys and identify the name of the service that was installed by the malware?

As we identified in the last point, the service the malware created is IPRIP, we know it is a service because it was located in the service path and it seems to uses the svchost.exe executable, that hosts the services of the machine

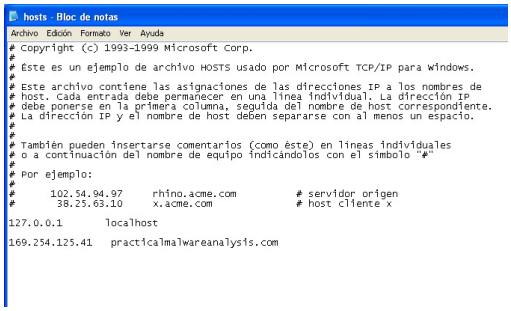
2.7 Analyze the register keys and identify the name of the executable that apparently consume (import) the DLL?

It is importing service host , It includes processes including Windows Auto Update and many required system services would be running in it. this cat let the attacker control processes and windows versions in the victims machine.

2.8 Create a virtual network of 2 Virtual machines. One of them will be the infected machine, and the other will be the server of Command & Control. Install Netcat on the last one to emulate the behavior of a web server.

On the command and control machine ...

on the victim's machine we configured the DNS Spoof , so the malware is going to connect to our command and control server and not to the atacker server...



now, we are going to test the connection between the machines with the DNS spoofed...

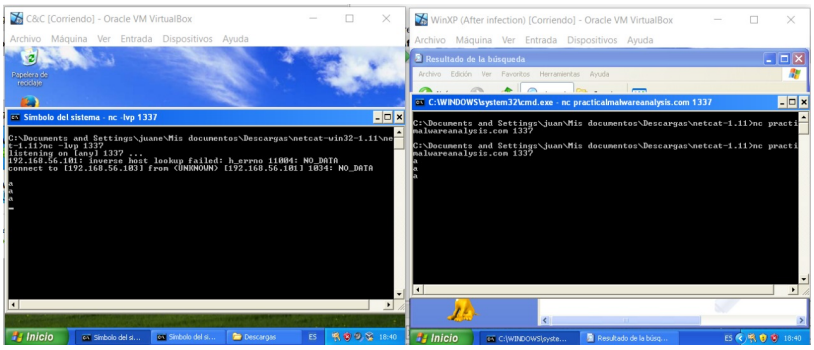


Figure 10: Connections

2.9 Start the malicious service in the infected machine using the command: net start IPRIP

now we initialized the service IPRIP. notice that the service was already initialized because we already found this service in previous section of this analysis.

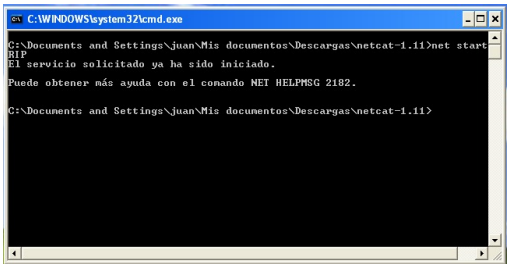


Figure 11: Service Initialized

2.10 Why IPRIP? Did you find the word (IPRIP) in some of the previous steps?

We found it interesting because it was consuming svchost.exe, this service allow the attacker to take control of the processes in the victim's machine.

2.11 Execute Process Explorer. Find the process that is running the malware using the "Find DLL" functionality of Process Explorer. Identify the Process Id (PID)

Using tool FindDll we identified the process that the library is running the process id is ...

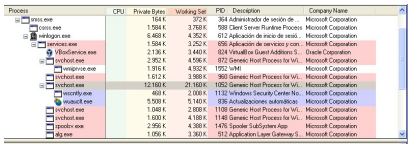


Figure 12: 1052

2.12 Execute Process Monitor and search the process using the PID

Having the process id, we found the process in the process monitor —

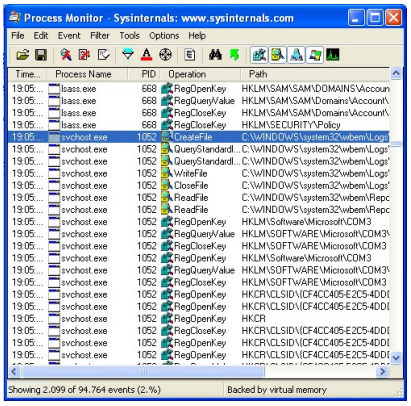


Figure 13: process of process id

2.13 Create filters to reduce the events to fewer than 10 events in the Process Monitor. Use filters that allow to identify keys and files modified or created.

We applied the following filters ...

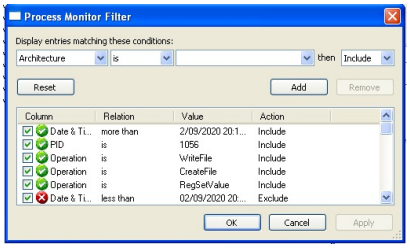


Figure 14: Filters

and we obtained these results ...

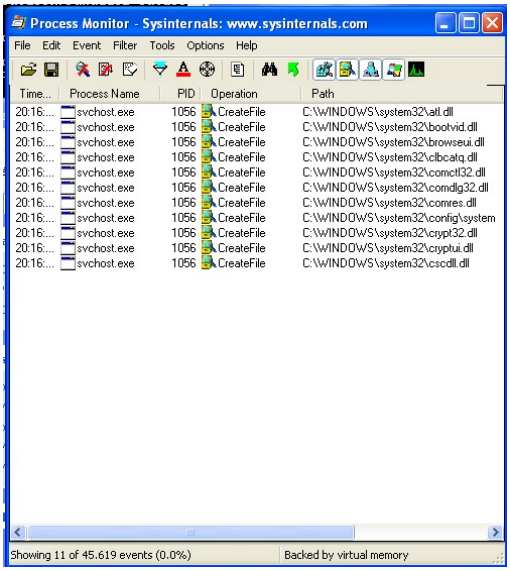


Figure 15: Filtered

2.14 Does the malware resolve some domain?

Yes, it resolving to www.practicalmalwareanalysis.com

2.15 Configure the nc tool with port 443, 8000 and 80. Which port is contacted on that domain? Capture the request done by the malware

configuration ...

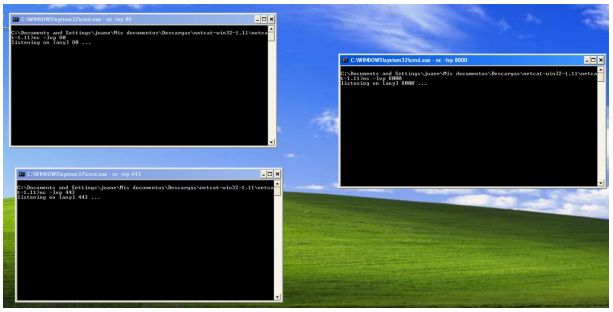


Figure 16: Configuration

we recived communication at port 80 . and we get this

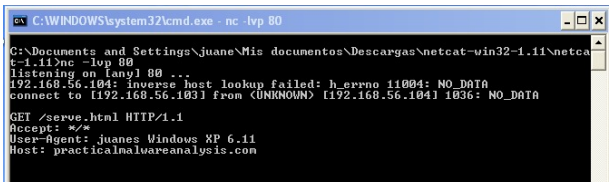


Figure 17: Port80

2.16 From all previous answers, identify all host-based signatures for this malware.

Through all this work we found several host based signatures, the most noticeable are that it imports important dynamic libraries such as *kernel32.dll*, it exports functions such as installA, install, etc. It modifies and creates files and registry keys alike as seen in figure **??**, and one of the most interesting ones is the creation of the service IPRIP that is hosted using the svchost.exe process of windows.

2.17 From all previous answers, identify all antivirus signatures for this malware.

This sections don't change much compared to the last status of the report, all due to that the dynamic analysis performed in this laboratory provide plenty of host-based signatures, that is the reason why we stick to our previous answers, the hash, creation time, the metadata, the size, etc.

2.18 From all previous answers, identify all network-based signatures for this malware.

We could confirm that the malware communicates with a command and control server using the practicalmalwareanalysis.com domain, we were able to verify that with the DNS spoof performed.