# Malware Countermeasures

Nico Ferrari

## I. Malware mitigation research

Computer security is a field which is growing day by day, constantly enhancing and developing new countermeasures techniques to protect the systems,but still, malicious software are succeeding in their destructive objectives. In fact, as well as computer security, attacking techniques are overcoming the new challenges imposed by the enhancement of the new security techniques , becoming an endless race between the security experts and the attackers. The goal of malwares is to infect the computer system or resources, deleting the data, slow down the working system or steal the important information, affecting then confidentiality, integrity and availability of the systems and resources.

The research of countermeasures against malware attacks, trying to protect data and operations, is typically divided in:

- securing the targeted systems against the attacks,
- respond to the attacks against the target systems.

Comparing traditional malwares with the modern ones, we notice that nowdays they are harder to detect and very targeted (i.e. affecting specific versions of software), exploiting zero-day vulnerabilities, persistent and stealthy, having the ability to change their code as they propagate and capable of propagating through the network and easily bypassing the defenses [1]. When viruses are known, it is possible to track their behaviour and save it in a database as signature. Static, Dynamic and Hybrid (static+dynamic) analysis can detect malicious behavior at runtime inside a safe environment, such as a VM. But nowdays, malwares are able to change their behaviour according to the target, making the analysis of their 'pattern' and their identification becomes much more difficult. As shown in [2], there are mainly two types of mitigation techniques:

- Sandboxig: where malwares are analyzed in a safe environment
- Application hardening techniques: to help programs be more resilient against exploits.

## II. Sandboxing

Once a safe environment called sandbox is set up, the software can be analyzed, examining the files for signs of malicious behaviours. The analysis may be conducted in a manner that is static, dynamic or a hybrid of the two. Basic static analysis does not require that the code is actually running. Each file of the software is analyzed, often with the help of tools like disassemblers and network analyzers, used to observe the malware without actually running it, with the goal to collect information on how it works. With modern malwares, malicious runtime behaviour can run undetected, therefore, for a complete understanding of the malicious behaviour, dynamic analysis is used. Dynamic analysis involves the execution of malware in order to study its behaviour at runtime. To overcome to this, adversaries had to face this new challenges and become very good at detecting them. In fact, to deceive a sandbox, adversaries hide code inside them that may remain dormant until certain conditions are met and only then does the code is executed. Now days the two analysis techniques are used together and this is called hybrid analysis. Once the malware is detected in the sandbox environment, some tools such as YARA [3] can be used in order to detect them.

## III. Application hardening techniques

There are several techniques used to make the systems more resilient against malware [4]. A common technique is Data Execution Prevention, an host based system level memory protection feature designed to prevent the local copy of malicious code into a foreign process space and subsequently executing it [5]. This techinque can be implemented via hardware and software. Complementary to the data execution prevetion, Address Space Layout Randomization creates high randomness in memory addresses of a target process, helping to reduce the access of determined addresses of memory which could be know for their vulnerabilities.

Some malwares aim to infect the heap memory with malicious code then exploits a different vulnerability to cause the exploit to call the commands in the heap memory.To reduce this kind of attacks, the heap is filled with *NOP* (no operation) instructions.

Other hardening techniques try instead to prevent that malicious libraries such as DLL are preloaded by the malware. In fact, if an application dinamically load DLL without a full qualified path, OSs like Windows try to find it in specific directories. In order to limit this attacks, could be necessary to always specify the full path of the DLL and have total control of those directories.

## IV. General Countermeasures

In order to mitigate the risk of being affected by malicious software, one of the most powerful countermeasures is the user awareness. Given that users of non-technical backgrounds might not be sufficiently aware of the cybersecurity risks they face, they can be the easy target of the attacker. Therefore, educating these personnel to be cautious of malware and unknown hardware with an introduction to cybersecurity, could greatly reduce the risks of institutions. Due to the low costs and availability of malicious software and how easy and fast is the diffusion process of the

malicious code, attackers of any level can cause several problems on the targeted systems. Therefore, being able to recognize the different malware categories, their features and countermeasures could help the users to understand the consequences of careless operations.

Often the attacker writes malicious code exploiting vulnerabilities in the software loaded in systems. For this reason, a Patch Management process becomes vital in order to keep the system functioning. The patch management process is usually divided in the following phases:

- scan of the system for missing security patches;
- determine the severity of the issue(s) addressed by the patch, evaluating the threat to the current environment. In fact, sometimes, patches could interfere with the current processes.
- if the decision goes to mitigate the threat using the patch, this patch must be downloaded;
- the patch is installed and tested, in order to evaluate how it affects the environment and actual configuration/workflow;
- the system is backed up and then the patch will be deployed.

In order to find malwares in the system, anti-virus becomes necessary. In fact, they will look for patterns matching with the known malicious code and will try to prevent the attacks.

Keeping controlled log files, could help to recognize the activity of malicious software. For example, a keylogger will probably try to sent the data acquired to an external entity, and observing the established connections from the logs, it could be possible to find weird behaviours. Firewalls will help to restrict the interaction of a remote control program with system. In fact, some malwares try to communicate with the target systems through some special ports, and blocking the unused ones using a firewall will reduce these kind of risks.

Another countermeasure consists in the implementation of system access controls, and the policy of running applications with least privilege, trying to minimize the damage caused by malicious software [6].

## REFERENCES


[1] E. Gandotra, D. Bansal, and S. Sofat, "Malware Analysis and Classification: A Survey," *Journal of Information Security*, vol. 05, no. 02, pp. 56–64, 2014.

[2] S. Sibi Chakkaravarthy, D. Sangeetha, and V. Vaidehi, "A Survey on malware analysis and mitigation techniques," pp. 1–23, 5 2019.

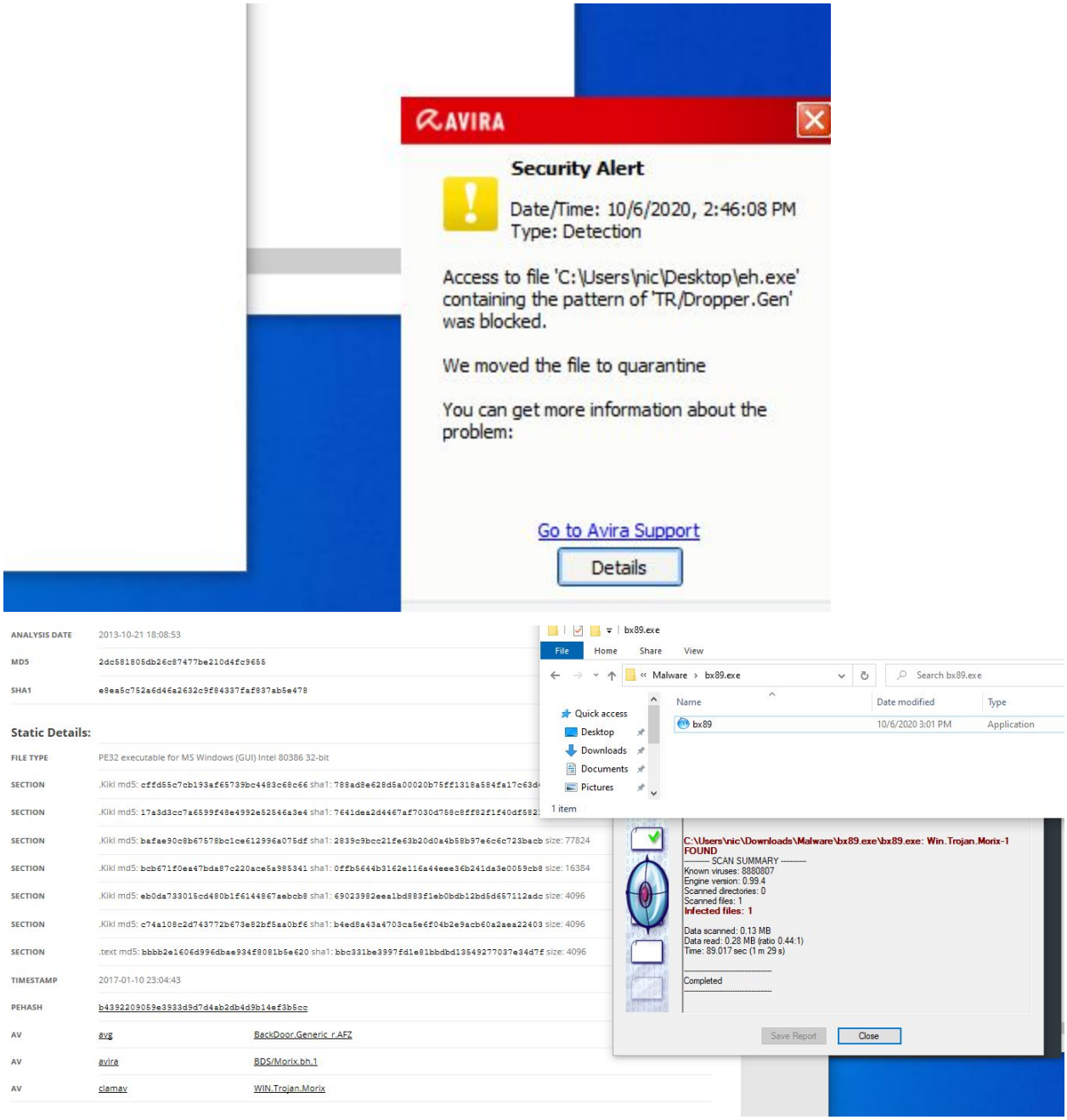
[3] VirusTotal, " YARA in a nutshell. [Online]," 2019. [Online]. Available: https://virustotal.github.io/yara

[4] K. Jørgen, "Toward Anti-fragility: A Malware-Halting Technique," Tech. Rep. [Online]. Available: www.computer.org/security

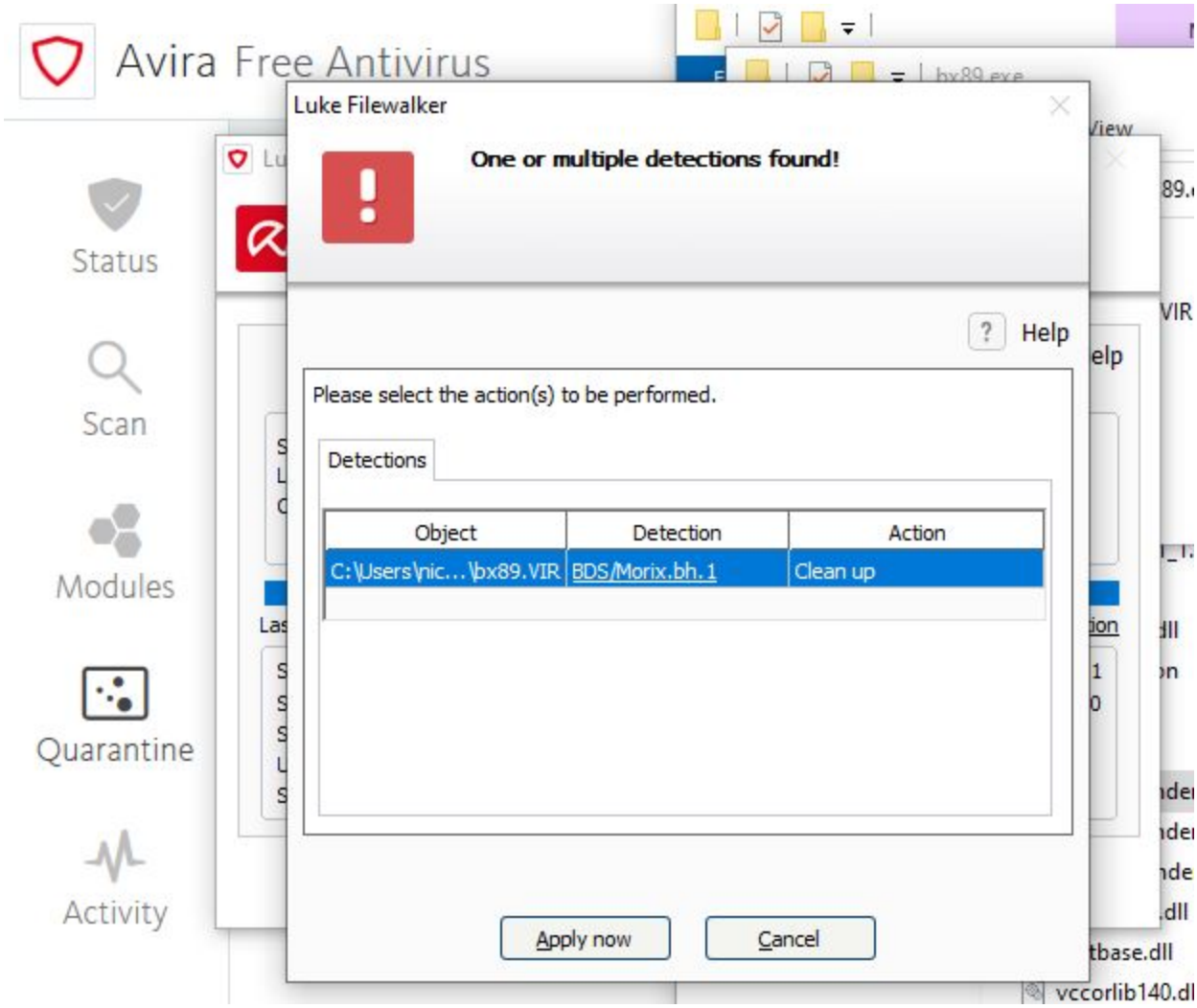
[5] F. C. Colon Osorio, IEEE Computer Society, Wireless Systems Security Research Laboratory, Microsoft Corporation, and Institute of Electrical and Electronics Engineers, "Host-Based Code Injection Attacks: A Popular Technique Used By Malware."

[6] M. Souppaya and K. Scarfone, "Guide to Malware Incident Prevention and Handling for Desktops and Laptops," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., 7 2013. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-83r1.pdf

Since Metadefender was not available, i used as first attempt avira and ClamWin as antivirus. The first malwares i tried to analyze were eh.exe and bx89.exe. Both malwares have been recognized by the AVs and it has been possible to get more informations regarding them with Avira. In fact, avira redirected to an online page explaining which kind of malwares they were and then, searching on internet, it was possible to find their signatures. This would have been possible also with clamWin, querying the name of the malware in his database.




As mitigation strategy, avira immediately asked to remove the malwares.



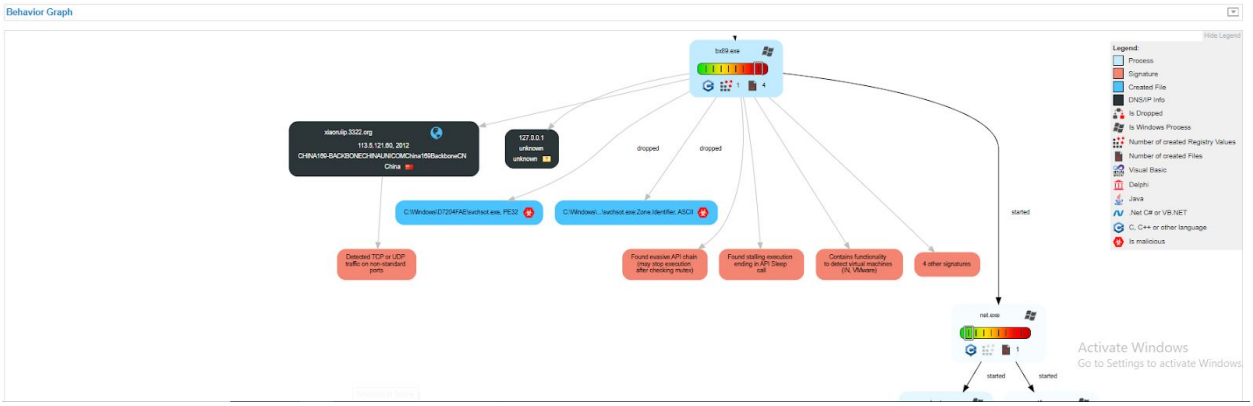

The attention focused on bx89.exe, a spyware classified as shown below.

Classification

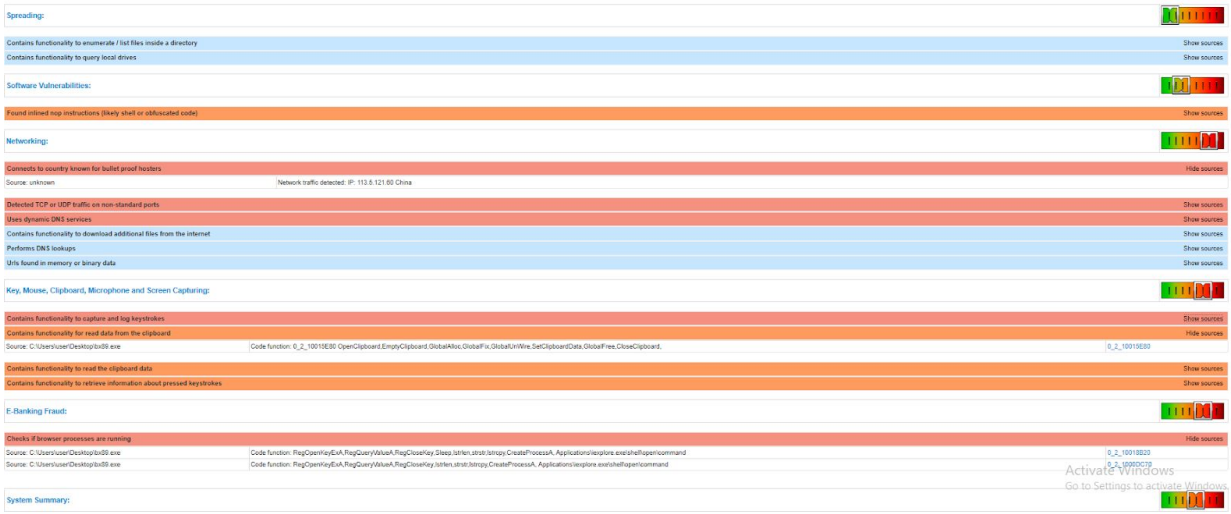

Analysis Advice

Sample has functionality to log and monitor keystrokes, analyze it with the 'Simulates keyboard and window changes' cookbook
Sample may inject into Firefox, Chrome or IE. Choose the 'Browse malicious URL' cookbook for furher analysis
Sample may offer command line options, please run it with the 'Execute binary with arguments' cookbook (it's possible that the command line switches require additional characters like: "-", "/", "--")
Sample monitors Window changes (e.g. starting applications), analyze the sample with the 'Simulates keyboard and window changes' cookbook
Sample tries to load a library which is not present or installed on the analysis machine, adding the library might reveal more behavior

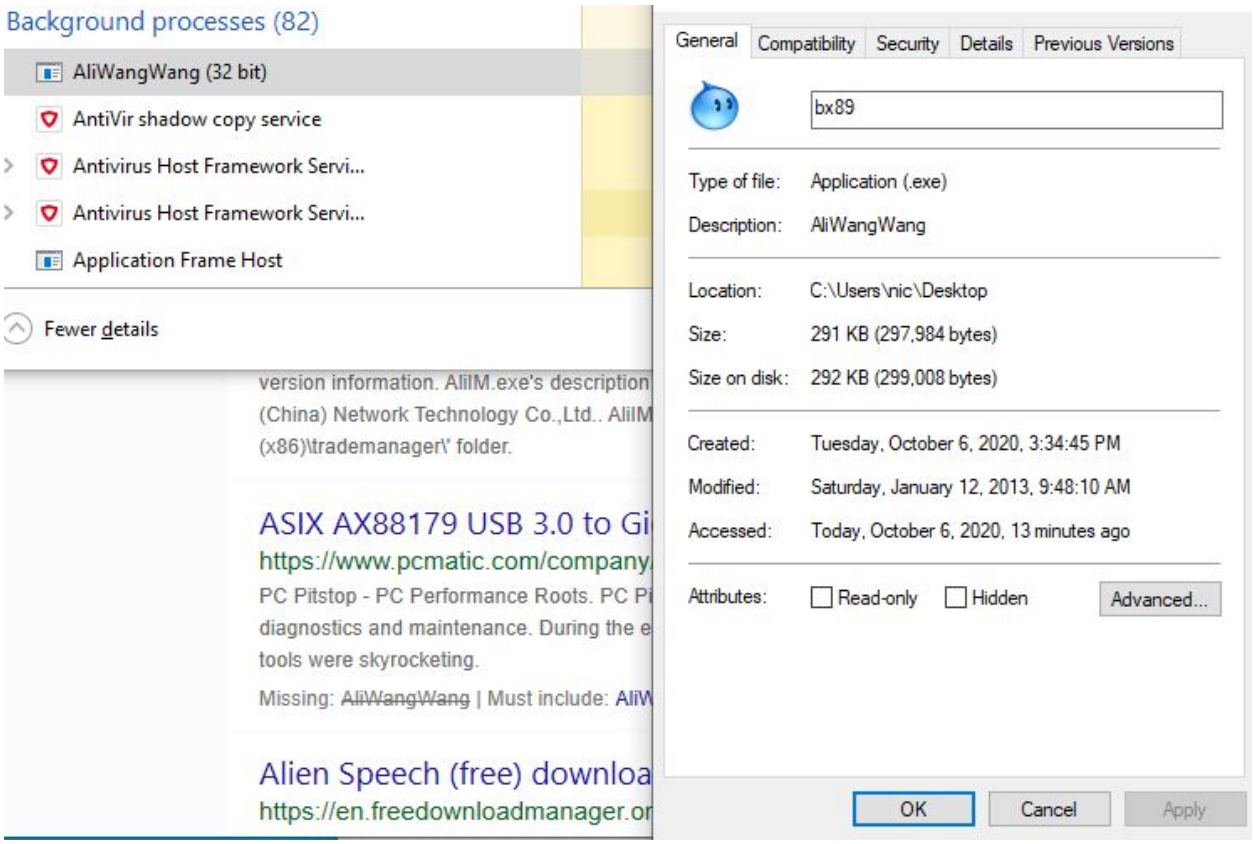
and with the following behaviour:



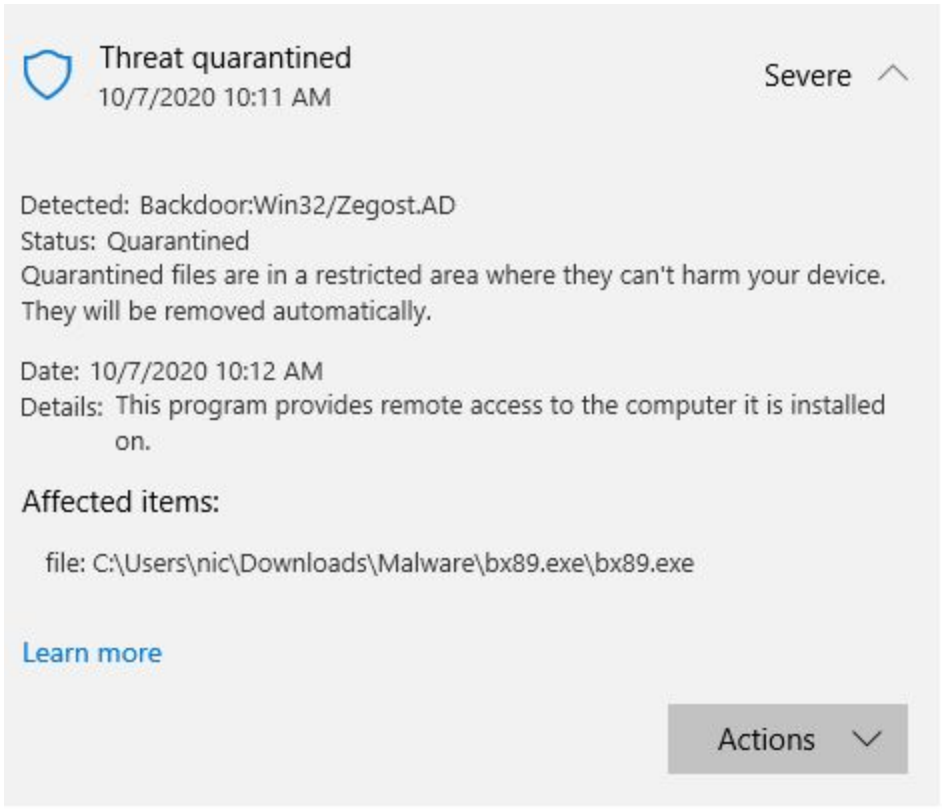

Several other informations have been found regarding this malware, such as the files affected and the network traffic established during the infection.




Running bx89.exe, doesn't shows any particular behaviour, but some background processes starts immediately to run using different names.




In order to protect the system mitigating the thread, it was possible to turn on again the real time scan of the windows' AV



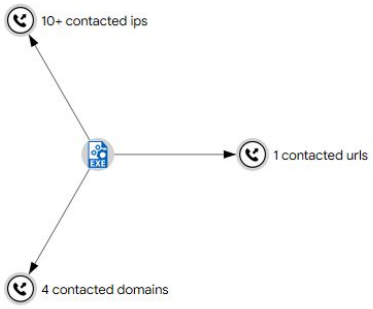

In order to obtain better information regarding the viruses, VirusTotal has been used. VirusTotal is an online scanner which scans the software that we send and tells us if it is a malicious one and then give us more informations regarding his behavior. This information could be important in order to manually remove or recognize malwares. During the execution of the lab, i tried to execute also a ransomware, which blocked all my VM. in order to remove the thread I reinstalled the whole system, an approach which could be dangerous in a production system. In fact, files could be infected and backing them up after the infection would mean spreading the virus. For this reason regular backups are fundamental also to help to recover infected machines. In order to avoid to infect my VM machine another time, I discovered any.run , an online platform which allows the user to test malwares and share the results, all contained in a sanboxed environment. i

ae2086e8789ec946f5ed43bf09cc86f407836707169f10d17a3aa8beec05bea2

| | | | |
|---|---|---|---|
| 107.165.236.233 | 1 / 87 | 18779 | US |
| 184.168.221.89 | 1 / 87 | 26496 | US |
| 208.91.197.46 | 1 / 94 | 40034 | VG |
| 154.198.33.75 | 0 / 75 | 26484 | US |
| 127.0.0.1 | 0 / 94 | - | - |
| 156.232.24.75 | 0 / 87 | 26484 | US |
| 23.104.77.225 | 0 / 87 | 395954 | US |

• • •

Graph Summary



**ae2086e8789ec946f5ed43bf09cc86f407836707169f10d17a3aa8beec05bea2**


bx89.exe

armadillo | direct-cpu-clock-access | long-sleeps | peexe | persistence | runtime-modules

DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 4

### Contacted URLs

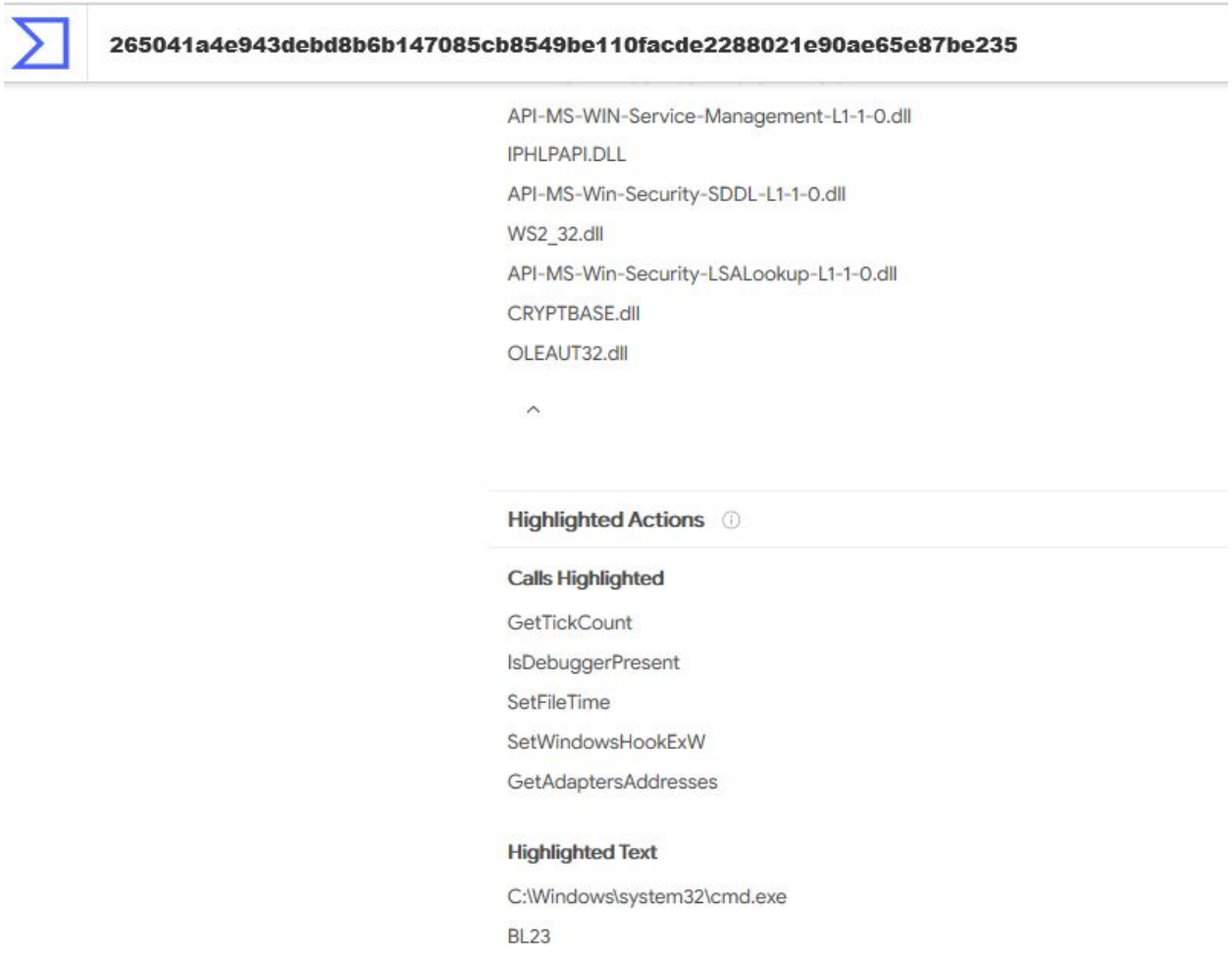
| Scanned | Detections | URL |
|---|---|---|
| 2013-04-26 | 3 / 37 | http://xiaoruiip.3322.org/ |

### Contacted Domains

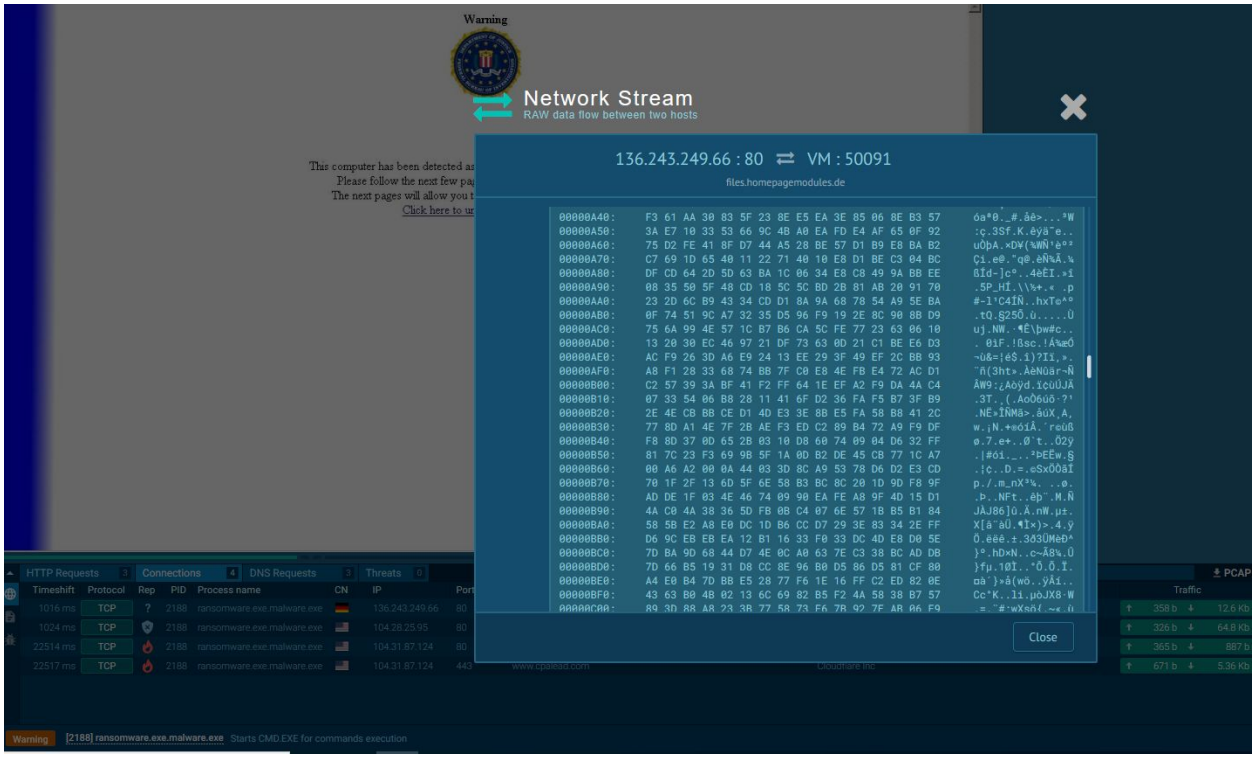
| Domain | Detections | Created | Registrar |
|---|---|---|---|
| xiaoruiip.3322.org | 1 / 76 | - | - |
| www.fz0575.com | 2 / 94 | 2018-09-23 | Network Solutions, LLC |
| www.wk1888.com | 8 / 94 | 2019-10-30 | Fujian Domains, Inc. |
| www.af0575.com | 5 / 94 | 2019-10-11 | DYNADOT, LLC |

### Contacted IPs

| IP | Detections | Autonomous System | Country |
|---|---|---|---|
| 113.5.121.60 | 0 / 75 | 4837 | CN |
| 185.245.180.55 | 0 / 75 | 46261 | US |
| 154.90.68.52 | 1 / 87 | 134548 | US |
| 107.165.236.233 | 1 / 87 | 18779 | US |
| 184.168.221.89 | 1 / 87 | 26496 | US |
| 208.91.197.46 | 1 / 94 | 40034 | VG |
| 154.198.33.75 | 0 / 75 | 26484 | US |
| 127.0.0.1 | 0 / 94 | - | - |
| 156.232.24.75 | 0 / 87 | 26484 | US |




ransomware run in the sendboxed environment.


Warning
Network Stream
RAW data flow between two hosts
136.243.249.66 : 80
VM : 50091
files.homepagemodules.de
Close
HTTP Requests
Connections
DNS Requests
Threats
Timeshift
Protocol
Rep
PID
Process name
CN
IP
Traffic
PCAP
Warning
[2188] ransomware.exe.malware.exe
Starts CMD.EXE for commands execution

## General Info

| | |
|---|---|
| File name | ransomware.exe.malware |
| Full analysis | https://app.any.run/tasks/6700ecfa-3fc7-435f-9423-f6d5c75965d6 |
| Verdict | Suspicious activity |
| Analysis date | 7/23/2018, 04:50:29 |
| OS: | Windows 7 Professional Service Pack 1 (build: 7601, 32 bit) |
| Indicators: | |
| MIME: | application/x-dosexec |
| File info: | PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed |



| | |
|---|---|
| MD5 | 63BAB74409C514ED8548B1F33D0ACEDC |
| SHA1 | ABC6BB8DD01FA83D7FD92182601B868D2B6DD1EA |
| SHA256 | 265041A4E943DEBD8B6B147085CB8549BE110FACDE2288021E90AE65E87BE235 |
| SSDEEP | 12288:36WQ4AAE6KWYF5L0Y2D1PQLV5V7UIKFGD9OFNU:QTHEVAPQLILUBGDYU |



Software environment set and analysis options

## Behavior activities

| MALICIOUS | SUSPICIOUS | INFO |
|---|---|---|
| **Changes the autorun value in the registry**<br>• ransomware.exe.malware.exe (PID: 2188) | **Reads Internet explorer settings**<br>• ransomware.exe.malware.exe (PID: 2188)<br>**Creates files in the user directory**<br>• ransomware.exe.malware.exe (PID: 2188)<br>**Starts CMD.EXE for commands execution**<br>• ransomware.exe.malware.exe (PID: 2188) | **Dropped object may contain URL's**<br>• ransomware.exe.malware.exe (PID: 2188) |

Find more information about signature artifacts and mapping to MITRE ATT&CK™ MATRIX at the full report

## Static information

### TRiD

.exe | AutoIt3 compiled script executable (88.1%)
.exe | UPX compressed Win32 Executable (4.6%)
.exe | Win32 EXE Yoda's Crypter (4.5%)
.dll | Win32 Dynamic Link Library (generic) (1.1%)
.exe | Win32 Executable (generic) (0.7%)

### EXIF

EXE

| | |
|---|---|
| MachineType: | Intel 386 or later, and compatibles |
| TimeStamp: | 2012:01:29 22:32:28+01:00 |
| PEType: | PE32 |
| LinkerVersion: | 10 |
| CodeSize: | 274432 |
| InitializedDataSize: | 122880 |
| UninitializedDataSize: | 573440 |
| EntryPoint: | 0xcee90 |
| OSVersion: | 5 |
| ImageVersion: | *null* |
| SubsystemVersion: | 5 |
| Subsystem: | Windows GUI |
| FileVersionNumber: | 3.3.8.1 |
| ProductVersionNumber: | 3.3.8.1 |
| FileFlagsMask: | 0x0017 |
| FileFlags: | (none) |
| FileOS: | Win32 |
| ObjectFileType: | Unknown |
| FileSubtype: | *null* |
| LanguageCode: | English (British) |
| CharacterSet: | Unicode |
| FileDescription: | *null* |
| FileVersion: | 3, 3, 8, 1 |
| CompiledScript: | AutoIt v3 Script: 3, 3, 8, 1 |

### Summary

| | |
|---|---|
| Architecture: | IMAGE_FILE_MACHINE_I386 |
| Subsystem: | IMAGE_SUBSYSTEM_WINDOWS_GUI |
| Compilation Date: | 29-Jan-2012 21:32:28 |
| Detected languages | English - United Kingdom<br>English - United States |
| FileDescription: | *null* |
| FileVersion: | 3, 3, 8, 1 |
| CompiledScript: | AutoIt v3 Script: 3, 3, 8, 1 |

| PID | Process | Operation | Key | Name | Value |
|---|---|---|---|---|---|
| 2188 | ransomware.exe.malware.exe | write | HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run | ransomware.exe.malware.exe | |
| | C:\Users\admin\AppData\Local\Temp\ransomware.exe.malware.exe | | | | |
| 2188 | ransomware.exe.malware.exe | write | HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap | UNCAsIntranet | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap | AutoDetect | 1 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | EnableFileTracing | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | EnableConsoleTracing | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | FileTracingMask | 4294901760 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | ConsoleTracingMask | 4294901760 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | MaxFileSize | 1048576 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASAPI32 | FileDirectory | %windir%\tracing |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | EnableFileTracing | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | EnableConsoleTracing | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | FileTracingMask | 4294901760 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | ConsoleTracingMask | 4294901760 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | MaxFileSize | 1048576 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\ransomware_RASMANCS | FileDirectory | %windir%\tracing |
| 2188 | ransomware.exe.malware.exe | write | HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings | ProxyEnable | 0 |
| 2188 | ransomware.exe.malware.exe | write | HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings\Connections | SavedLegacySettings | |
| | 460000004700000001000000000000000000000000000000000000000000000000B096B68868EBD3010000000000000000000000020000001700000000000000FE800000000000007D6CB050D9C | | | | |
| 2188 | ransomware.exe.malware.exe | write | HKEY_CLASSES_ROOT\Local Settings\MuiCache\59\52C64B7E | LanguageList | en-US |

## Files activity

| Executable files | Suspicious files | Text files | Unknown types |
|---|---|---|---|
| 0 | 1 | 4 | 1 |

### Dropped files

| PID | Process | Filename | Type |
|---|---|---|---|
| 2188 | ransomware.exe.malware.exe | C:\Users\admin\AppData\Roaming\Microsoft\Windows\IETldCache\index.dat | dat |
| | | **MD5:** D7A950FEFD60DBAA01DF2D85FEFB3862 **SHA256:** 75D0B1743F61B76A35B1FEDD323788378O5DE58D79FA950CB6E8164BFA72073A | |
| 2188 | ransomware.exe.malware.exe | C:\Users\admin\AppData\Local\Tempscratch.cmd | — |
| | | **MD5:** — **SHA256:** — | |
| 2188 | ransomware.exe.malware.exe | C:\Users\admin\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\GUFVP8I9\Glene[1].png | Image |
| | | **MD5:** 2A21843D535267F62BA3D318C1FCC884 **SHA256:** 08A45DE65DD737899F4B167AB9E6BA502D96E0E10A23C2728789A9A444059D9E | |
| 2188 | ransomware.exe.malware.exe | C:\Users\admin\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\XB3OCR2W\avatar-f6cfa349-11[1].png | Image |

With all these informations would be possible to check all the registers and files created, removed and edited and manually trying to restore the normal conditions.

In the beginning clamWin was not detecting the malwares, and, if that happens, it is possible to load your own signatures from the .exe files and reperform the scan.
I tries to generate a simple signature following the commands suggestd in the documentation and from online forums, but this didn't bring any success under windows environment. Here the commands executed:
*.\sigtool.exe --md5  C:\Users\nic\Downloads\Malware\bx89.exe\bx89.exe > custom.hdb*

To generate the signature from a malware.
*.\clamscan.exe -d 'C:\Program Files (x86)\ClamWin\bin\custom.hdb'*

*C:\Users\nic\Downloads\Malware\bx89.exe\bx89.exe*
*LibClamAV Error: cli_loadhash: Problem parsing database at line 1*
*LibClamAV Error: Can't load C:\Program Files (x86)\ClamWin\bin\custom.hdb: Malformed database*
*ERROR: Malformed database*

The .hdb file has also been edited using the following format as suggested in the official documentation:

*Name:Type:Offset:malware hex output*

But still not working.