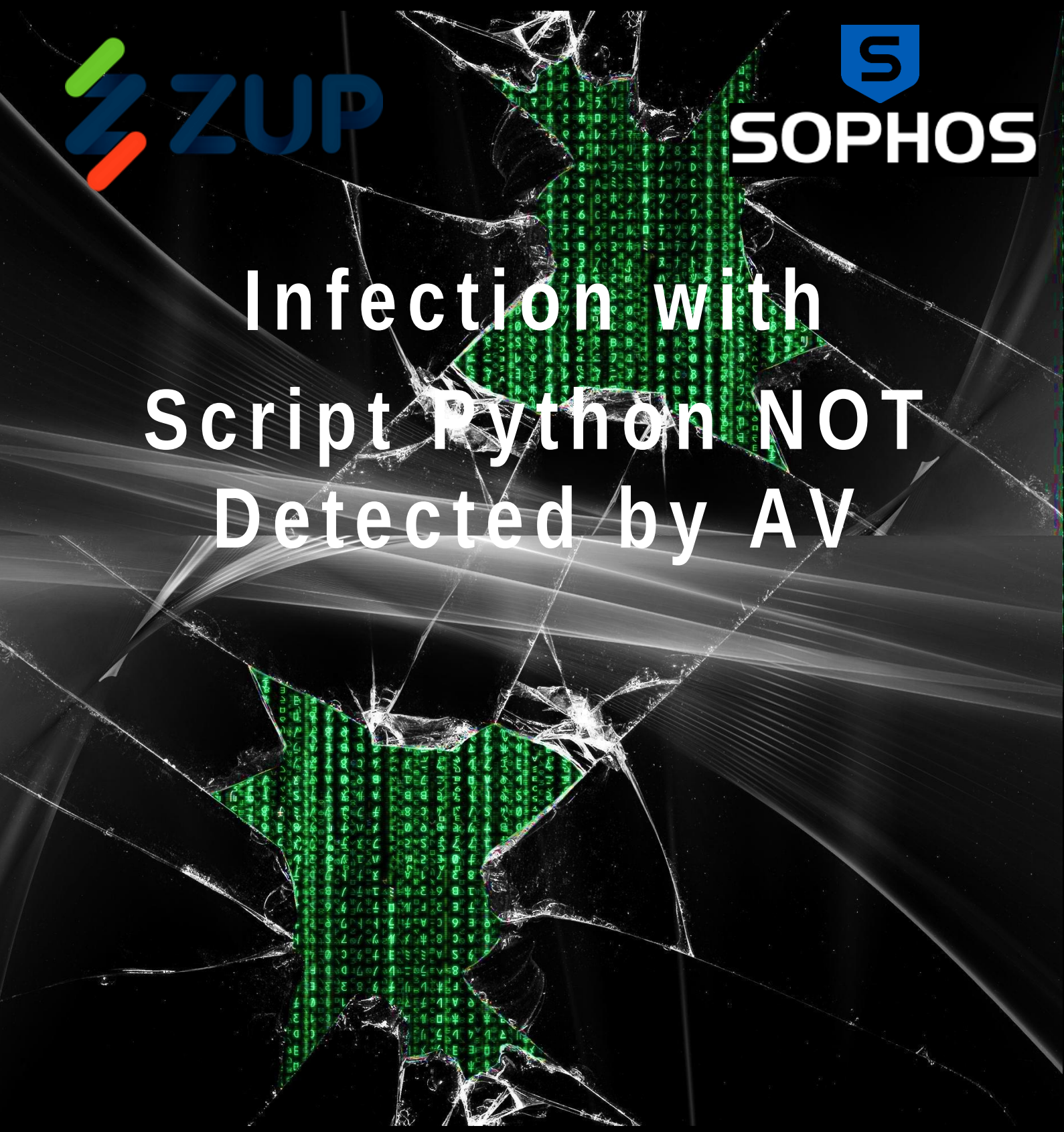

Infection with Script Python NOT Detected by AV

ZUP Security Labs at Zup Innovation

Researcher and CyberSecurity Manager (s): 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 an execution of two python scripts responsible to download some malware in our environment.

Regarding the test performed, the first objective it's to simulate targeted attacks using a python script to obtain a panoramic view of the resilience presented by the solution, with regard to the efficiency in it's detection by signatures, NGAV and Machine Learning, running this script, the idea is downloading these artifacts directly on the victim's machine. The second objective consist in running this script another python script with daily malwares, provide by ***MalwaresBazaar*** by request using API access, in the day of this test we downloaded more than **600 real Malwares** (632 Malware exactly) and the third objective consisted of analyzing the detection of those same malwares (or those not detected yet) when they were changed directories, the idea here is to work with manipulation of samples (without execution).

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:

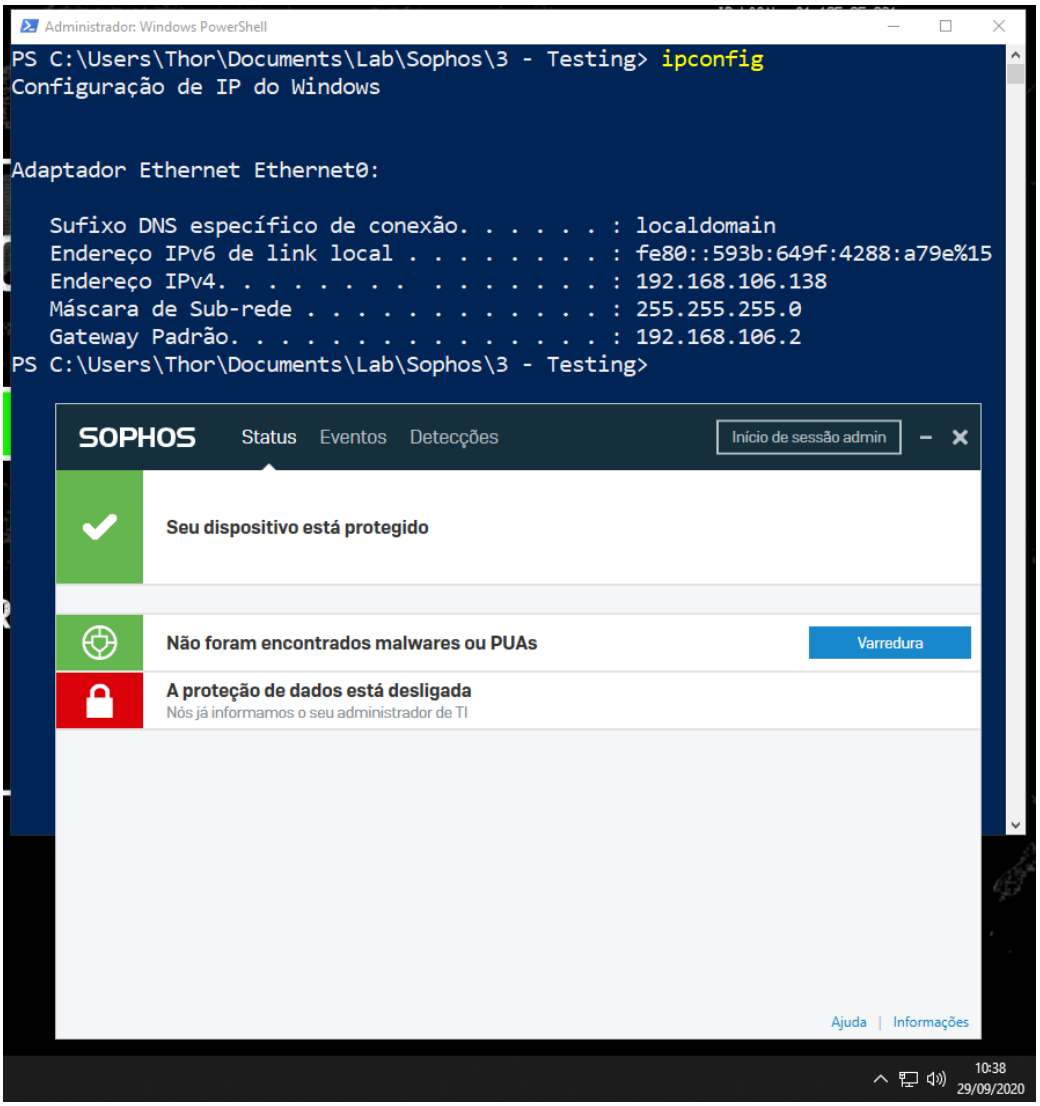

Image 1.1: Windows 10 Pro 2019 Virtual Machine

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 **3 day**, without count the weekend, along with the making of this document. The intrusion test started on the **29th of September** of the year 2020 and it was completed on the **30th of September** of the same year.

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.

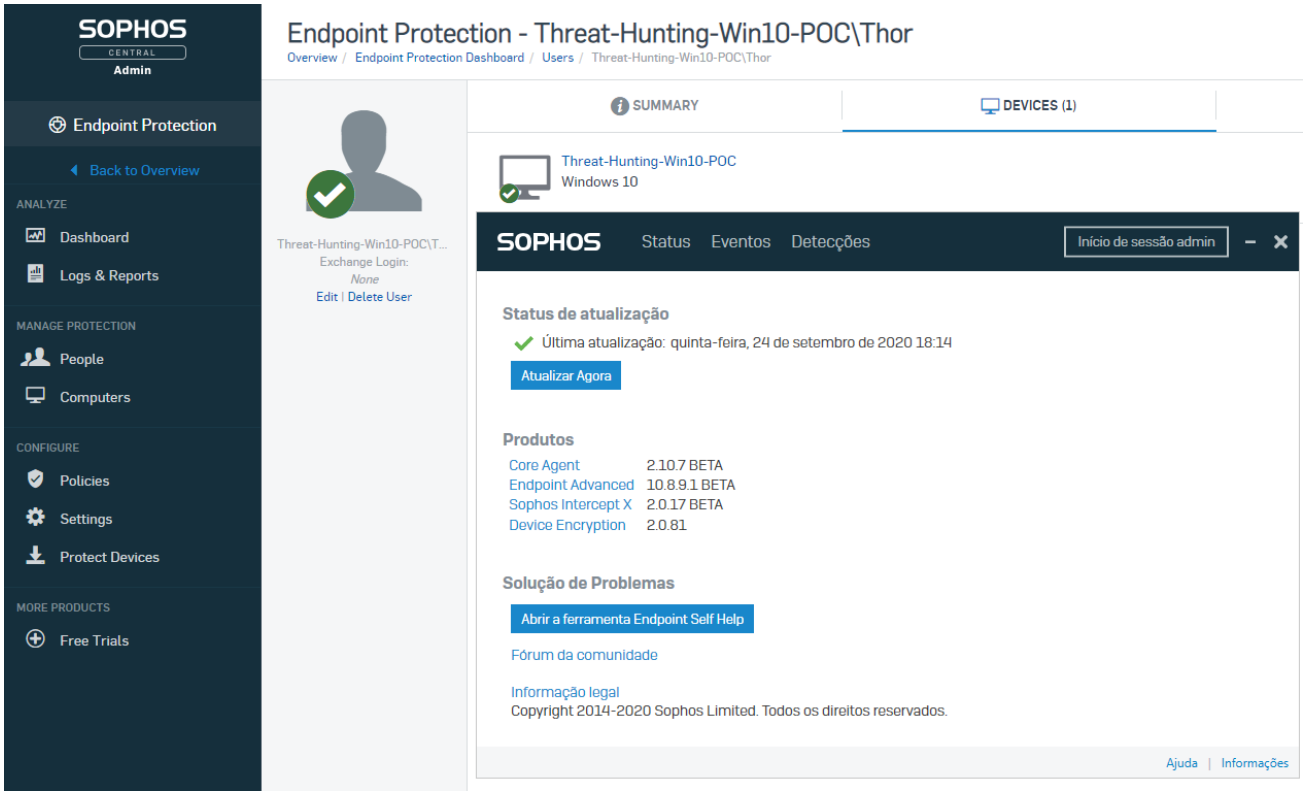

Image 1.2: Virtual Machine with Policy applied

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.

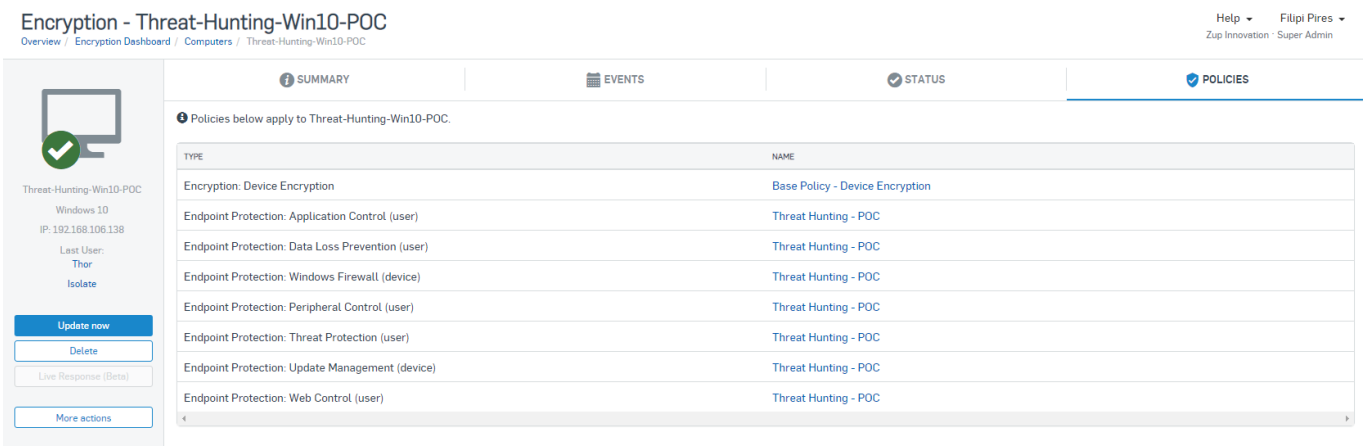

Image 1.3: Policy created by Sophos Central

Attacking validation

Before starting the detection tests, we need to validate if all those scripts are functional.

We used the Policy applied: **- THEATH HUNTING – NO Policy**

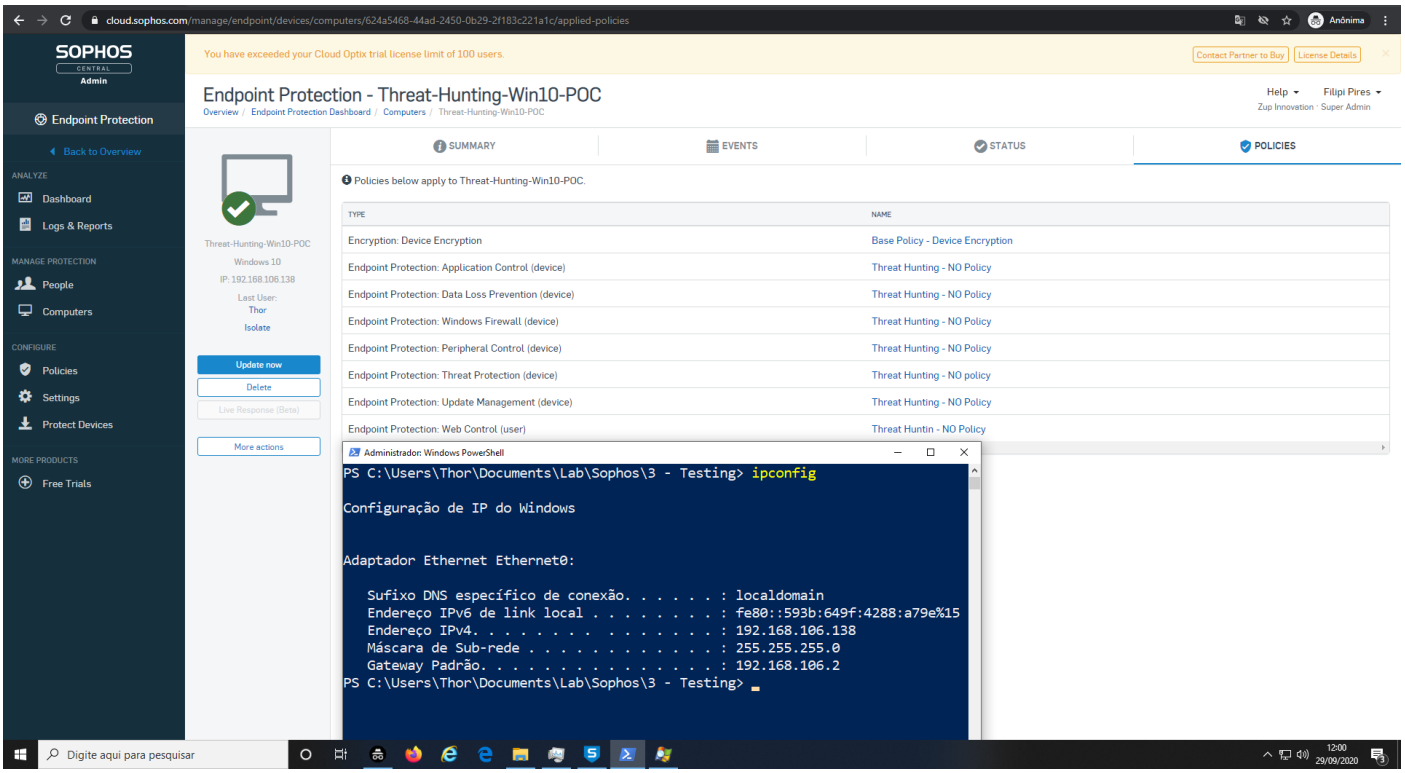

Image 1.4: NO Policies

We used a Bazzar.py (based in MalwareBazzar Documentation)

(https://bazaar.abuse.ch/api/#download)

```
#!/usr/bin/env python3
import requests
import sys
import argparse
import json
import pyzipper

def check_sha256(s):
    if s == "":
        return
    if len(s) != 64:
        raise argparse.ArgumentTypeError("Please use sha256 value instead of '" + s
+ "'")
    return str(s)

parser = argparse.ArgumentParser(description='Download a malware sample from Malware
 Bazaar by abuse.ch')
parser.add_argument('-s', '--
hash', help='File hash (sha256) to download', metavar="HASH", required=True, type=ch
eck_sha256)
parser.add_argument('-u', '--
unzip', help='Unzip the downloaded file', required=False, default=False, action='sto
re_true')
parser.add_argument('-i', '--
info', help='Get information on a hash (do not download file)', required=False, defa
ult=False, action='store_true')

args = parser.parse_args()

if(args.unzip == True and args.info == True):
    print("Sorry, please select unzip or information display.")
    sys.exit(1)

ZIP_PASSWORD = b'infected'
#ZIP_PASSWORD = "infected"
headers = { 'HERE API provided by MalwareBazaar': '' }

if(args.info == False):
    data = {
        'query': 'get_file',
        'sha256_hash': args.hash,
    }

    response = requests.post('https://mb-
api.abuse.ch/api/v1/', data=data, timeout=15, headers=headers, allow_redirects=True)
    open(args.hash+'.zip', 'wb').write(response.content)

    if(args.unzip == True):
        with pyzipper.AESZipFile(args.hash+".zip") as zf:
            zf.pwd = ZIP_PASSWORD
```

```
            my_secrets = zf.extractall(".")

            print("Sample \""+args.hash+"\" downloaded and unpacked.")
    else:
        print("Sample \""+args.hash+"\" downloaded.")
else:
    data = {
        'query': 'get_info',
        'hash': args.hash,
    }
    print(data)
    response = requests.post('https://mb-
api.abuse.ch/api/v1/', data=data, timeout=15, headers=headers)
    print(response.content.decode("utf-8", "ignore"))
```

After to perform this script as we can see below, the hash chosen it was downloaded and extracted in virtual machine.

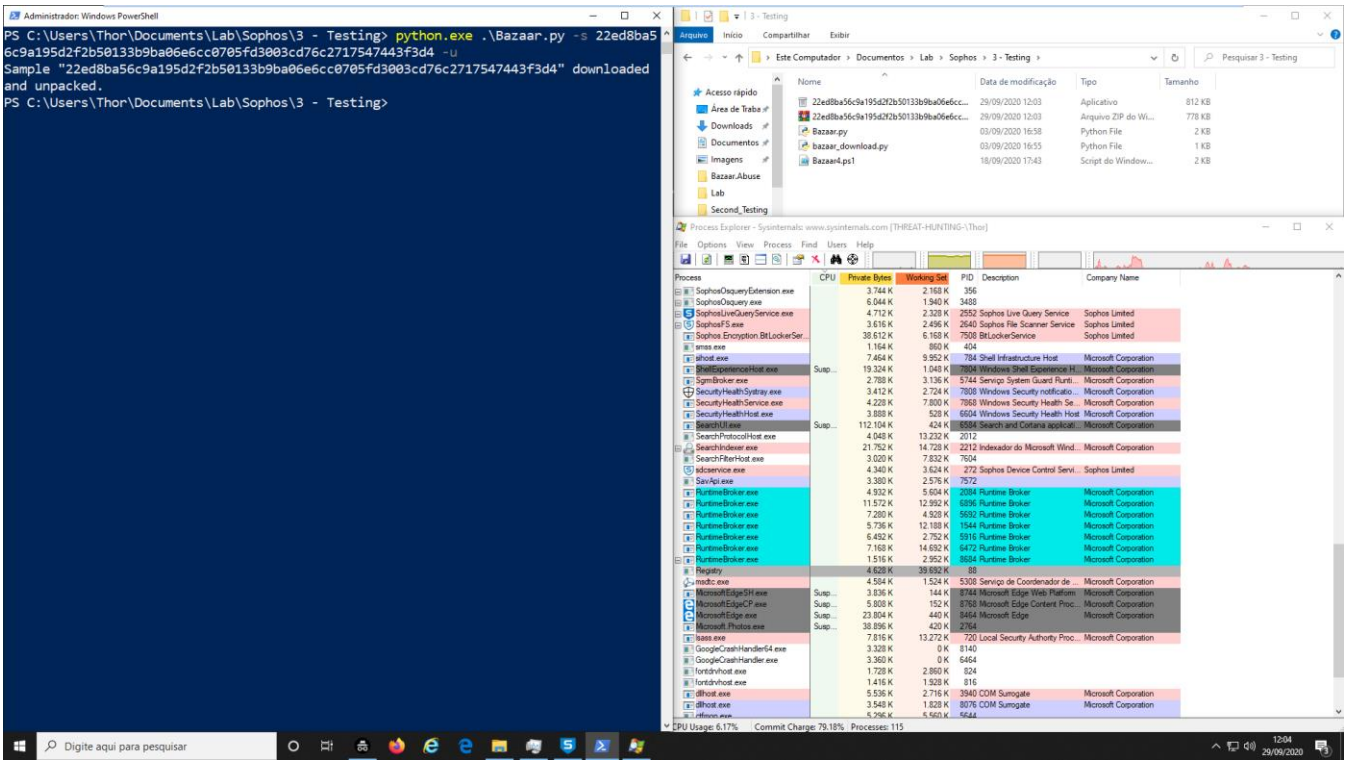

Image 1.5: Python Script execution without policy

As we expected, the file is downloaded from the Malware Bazaar using python script utilizing an API called and then its extraction is carried out in automatize way.

Malicious Ransomware Copa | Ransomware Stop provided by **Malware Bazaar**

22ed8ba56c9a195d2f2b50133b9ba06e6cc0705fd3003cd76c2717547443f3d4

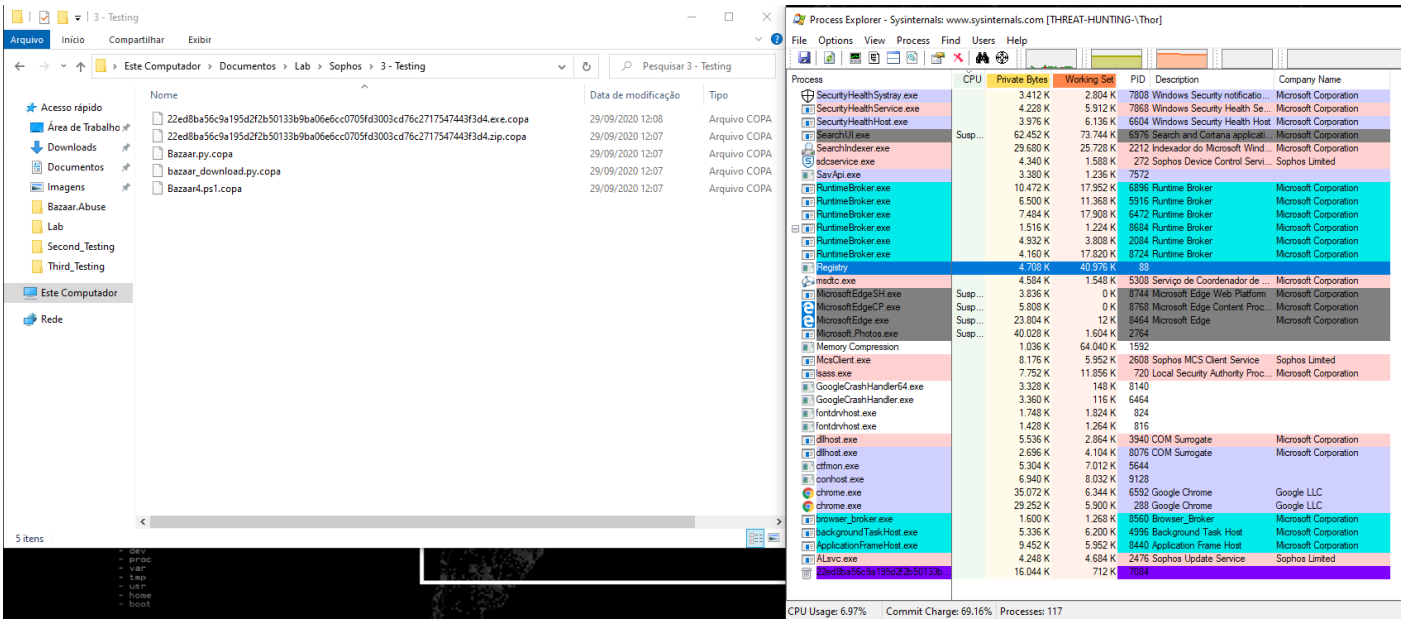

Image 1.6: Python Script Infection Complete (NO Policy)

3.2 First Test

So now, we going to execute the first stage of the tests where we perform a python script with malicious hash provide by **M*alwaresBazaar***, but know we applied the policy named **`Threat-Hunting-POC`**

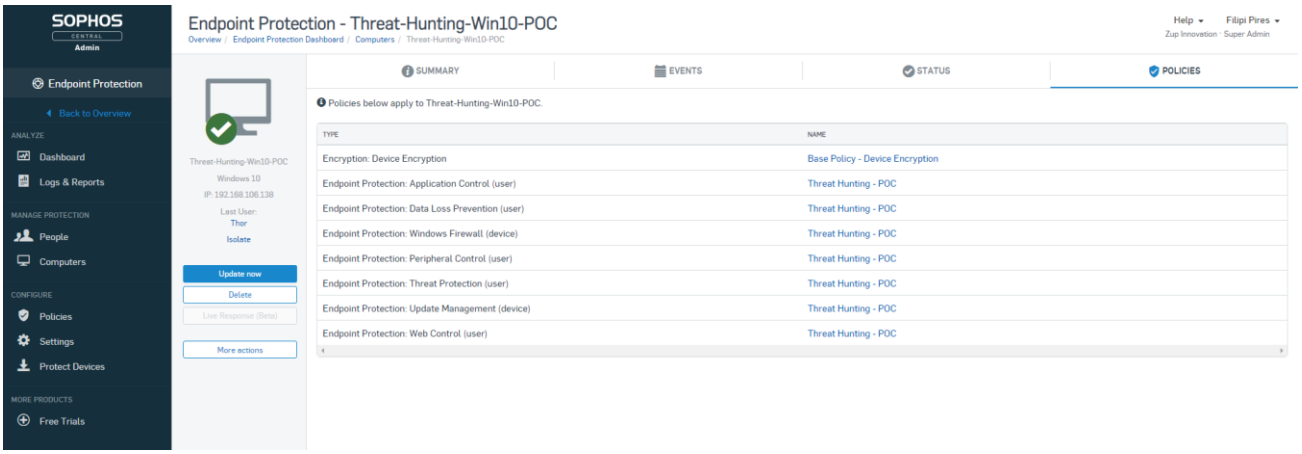

Image 1.7: SOPHOS Policy applied

As you can see, in this first test, performing **Ransomware Copa**, the sample it was block by Sophos platform.

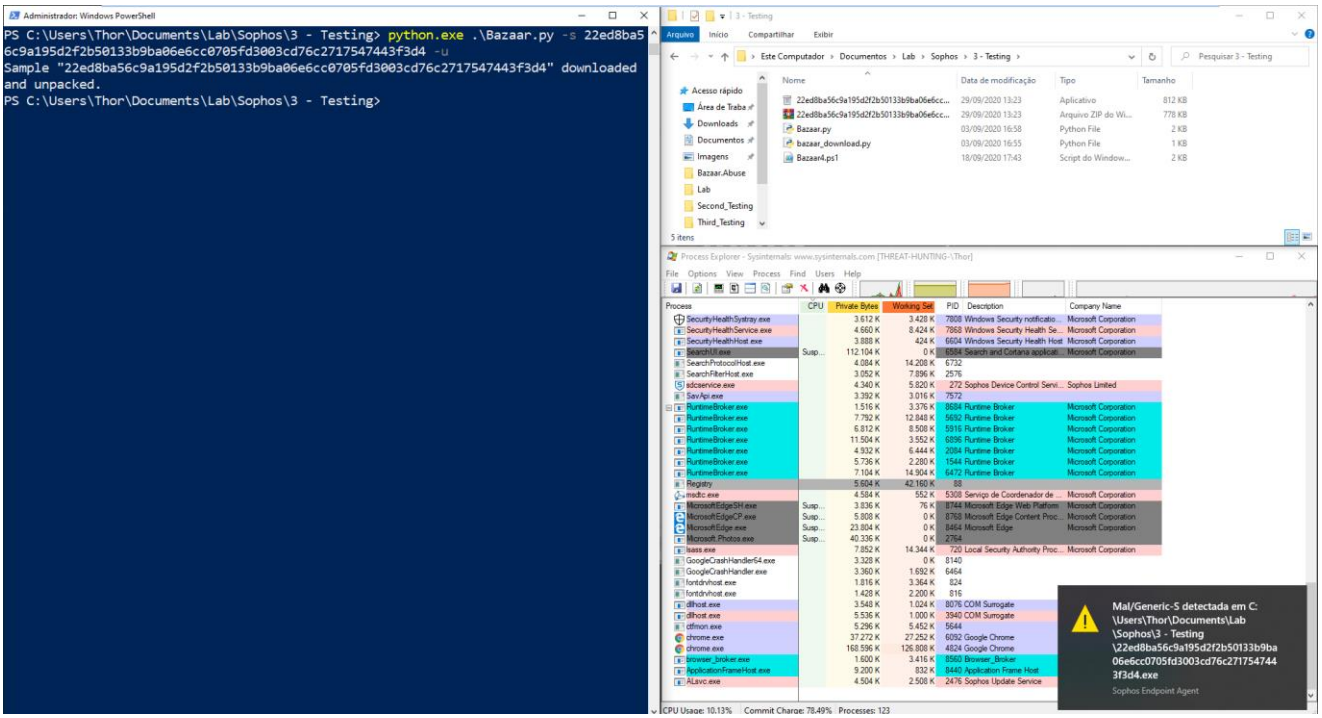

Image 1.8: Python Script with SOPHOS Policy applied

3.3 Second Test

The second test we used another **python script**, responsible to download the daily Malwares collected by Malware Bazaar as you can see below

```
#!/usr/bin/env python3
from datetime import date, timedelta
import urllib.request
import sys
import pyzipper

ZIP_PASSWORD = b'infected'
print(sys.argv)
if len(sys.argv) == 2:
    datefile = sys.argv[1]
else:
    datefile = (date.today() - timedelta(days=1)).strftime("%Y-%m-%d")

print("Using date: %s" % datefile)
print("Downloading https://mb-api.abuse.ch/downloads/%s.zip dataset..." % datefile)
response = urllib.request.urlopen('https://mb-
api.abuse.ch/downloads/%s.zip' % datefile)
print("Download complete!")
open('%s.zip' % datefile, 'wb').write(response.read())
print("Saving dataset... complete!")

with pyzipper.AESZipFile("%s.zip" % datefile) as zf:
```

```
    zf.pwd = ZIP_PASSWORD
    my_secrets = zf.extractall(".")
    print("Dataset unpacked!")
```

All this files that it was uploaded from public repository known and maintained by the security community in this web (`https://bazaar.abuse.ch/`).

> *MalwareBazaar is a project from abuse.ch with the goal of sharing malware samples with the infosec community, AV vendors and threat intelligence providers.*

MalwareBazaar creates daily batches of malware sample). The daily batches are created once a day at midnight (00:00 UTC). Please consider that it takes a few minutes to create the batch. So, I kindly ask you to not fetch the daily batch before 00:15 UTC.

The day chosen for this test it was **2020-09-28**

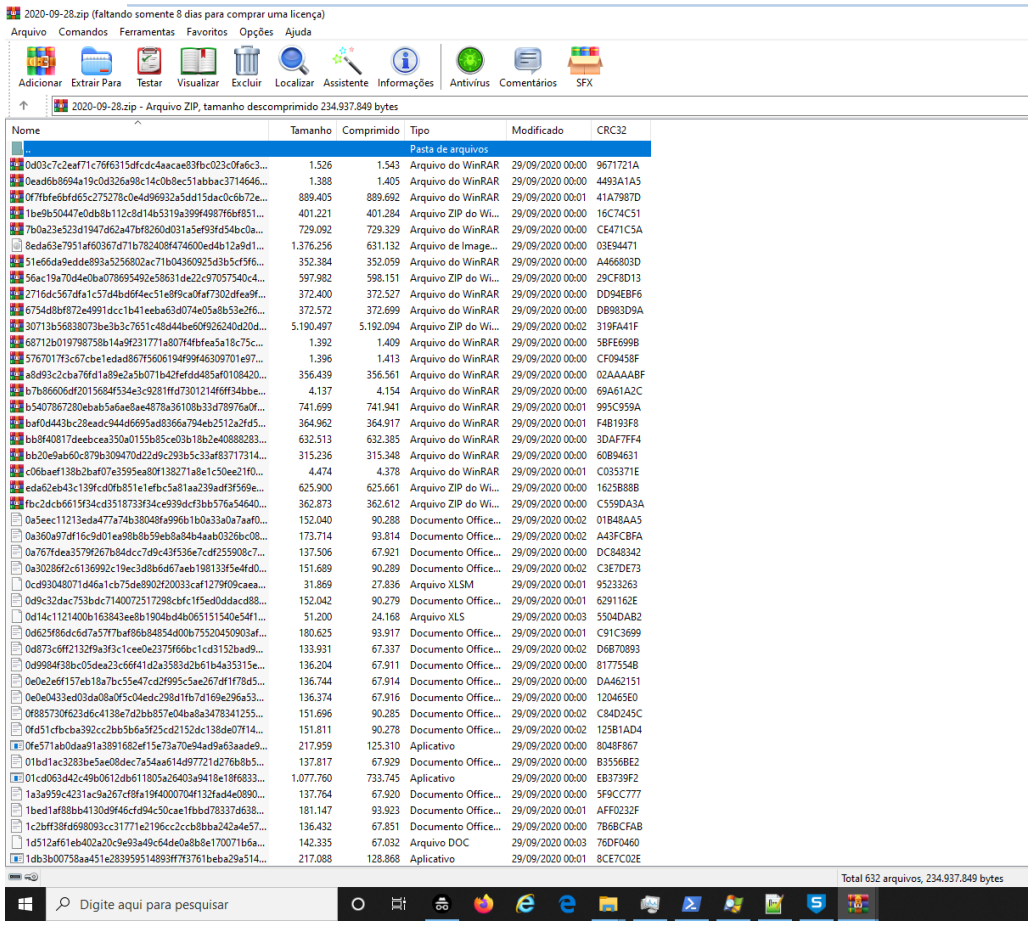

Image 1.9: Python Script works well

The purpose of this test, it's to simulate the same process when the user receives any email with a python script and after that he could even click in this script and he (user) will be downloading a zipped file (`.zip`) and will perform the extraction of these artifacts in their own environment.

During this test, one thing called my attention:

- **First Detection** happened on **September 29, 2020 at 13:44 | GMT-3**

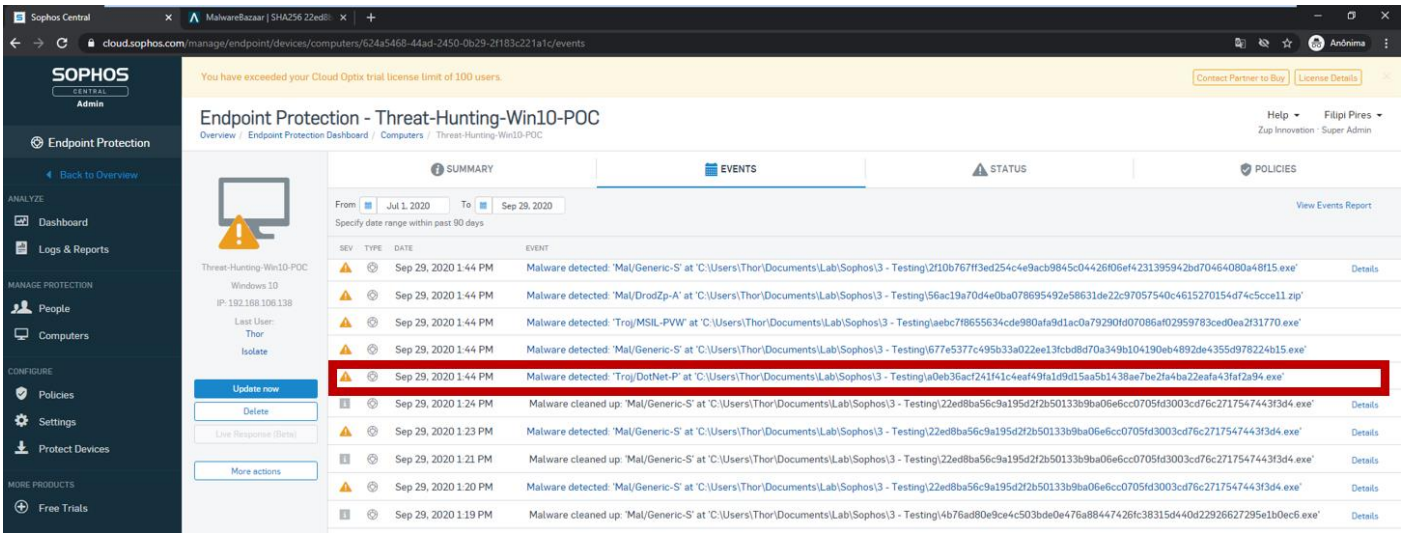

Image 1.10: Start Detection

- **Last Detection** happened on **September 29, 2020 at 16:21 | GMT-3**

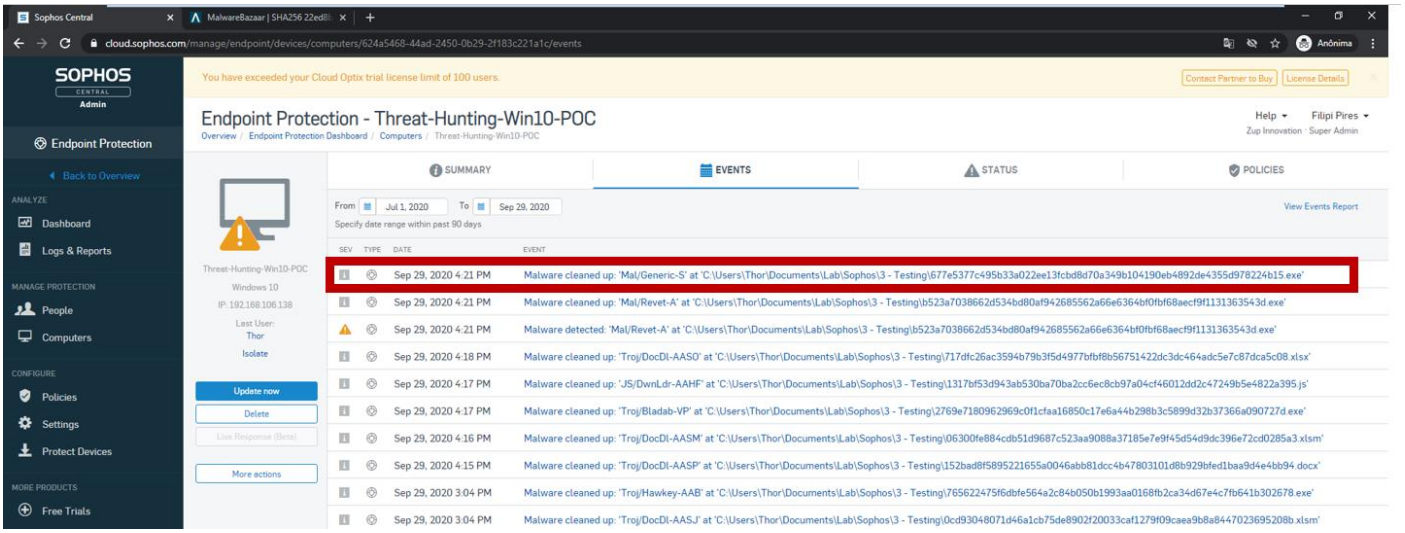

Image 1.11: Finish Detection

That is, we have a time gap with almost **three hours** between the first and the last detection, that was the time it took for malwares to be detected.

During this test, one of the things that brought us concerns was the somewhat **CPU** and **Memory** consumption.

As we can see in this image below, Sophos solution has a lot binaries consuming resources from machine.

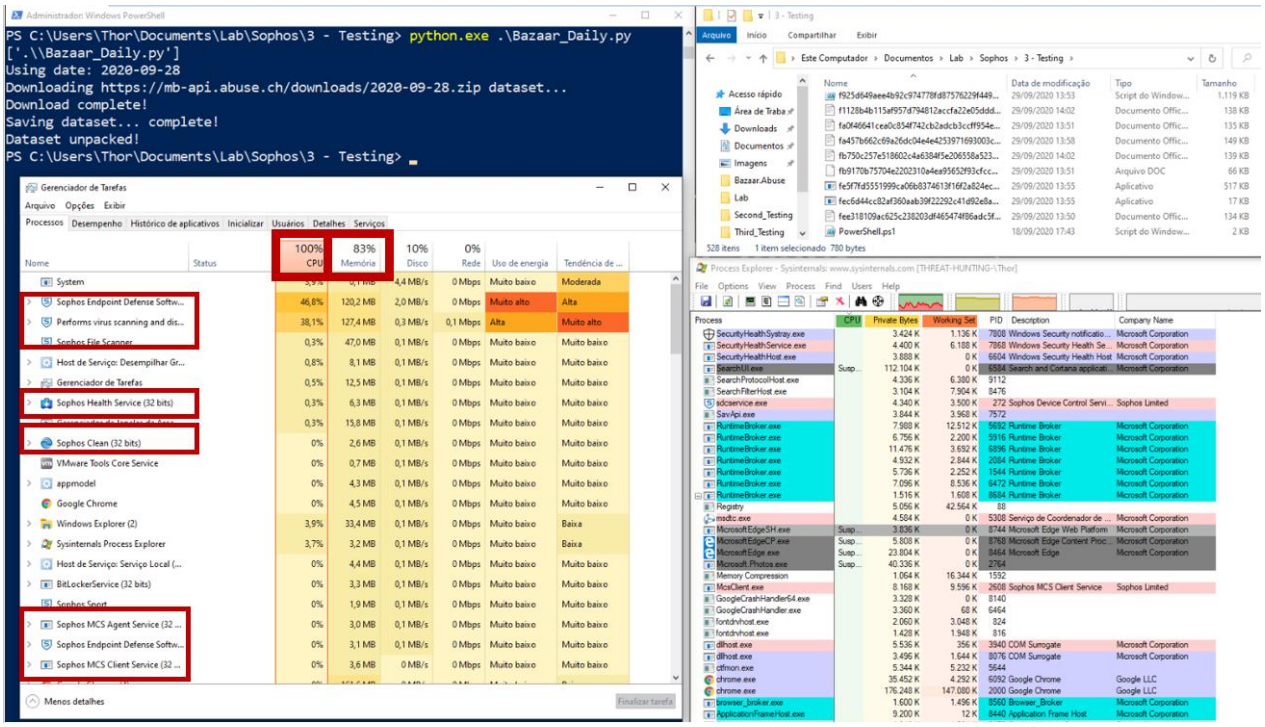

Image 1.12: High consumption from Sophos's agent

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 **103 (One hundred and three) Malwares** that, when executed inside the environment, could perform an infection.

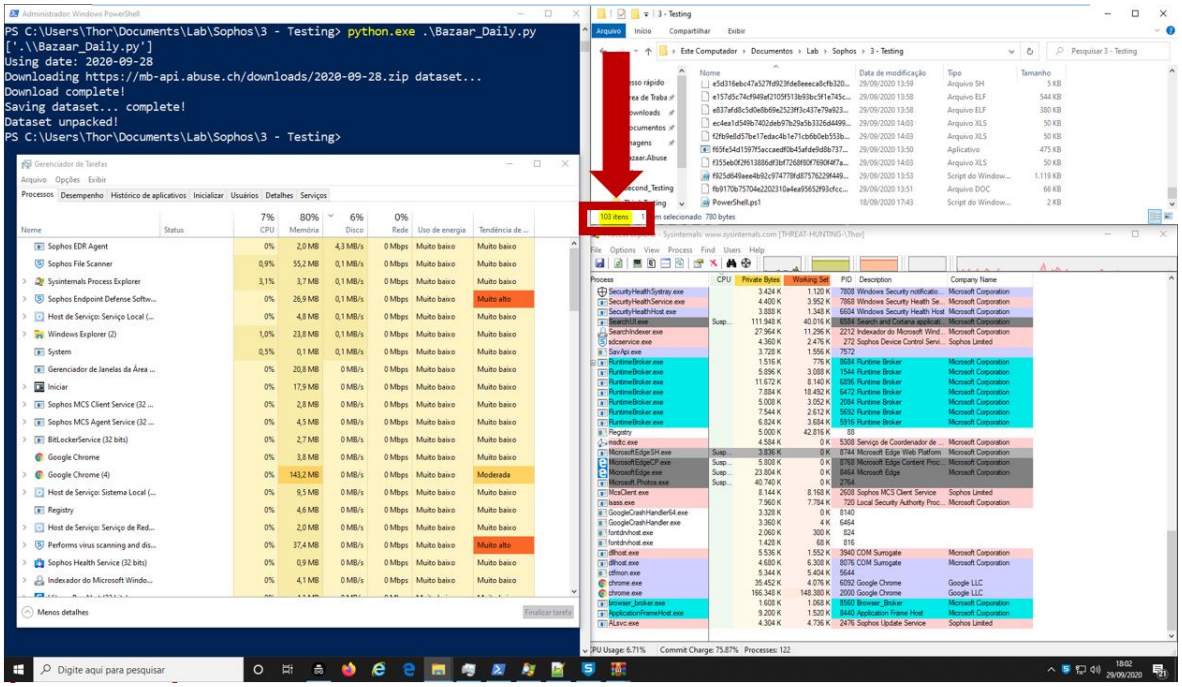

Image 1.13: **103** Malware NOT detected

3.4 Third Test

In this third stage of the tests was through the transfer of folders to another directory within the same machine, the purpose of this test was to simulate a transfer of files within the same environment, during this action, 2 more malware's it was detected and deleted.

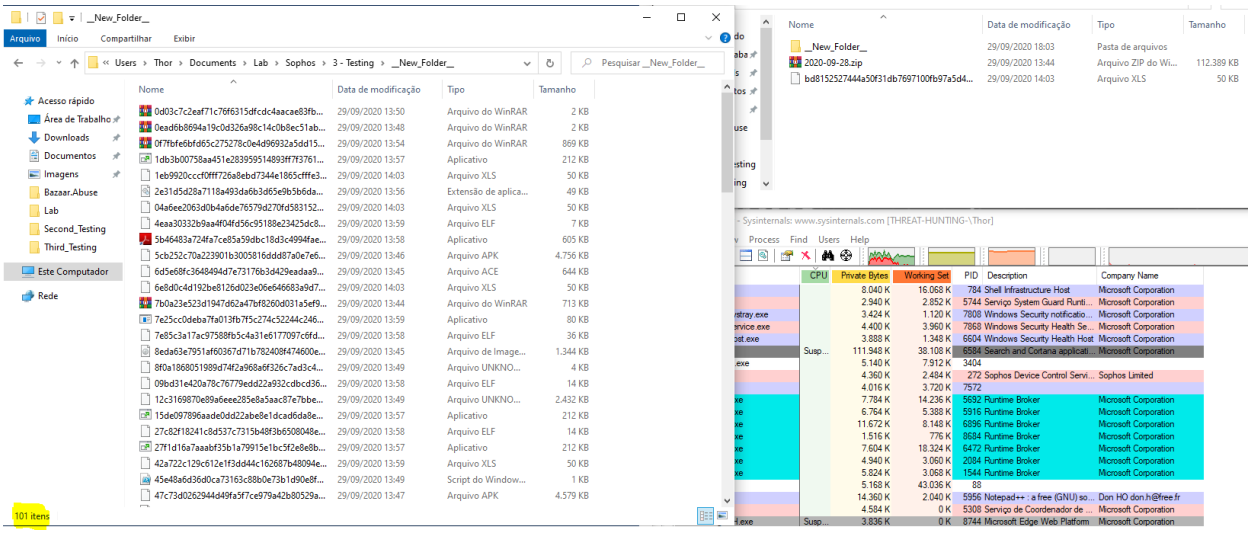

Image 1.13: __NEW_FOLDER__(Sophos) – Coping another folder

When a new file is generated on the disk, soon we should have a new entry in a block of that disk and in theory the antivirus should take some action (considering that it has the real time enabled), we could define it as a file manipulation (still not running) where the endpoint protection is already necessary, considering that a new directory was created, soon we would have a new repository with several hashes inside to be examined.

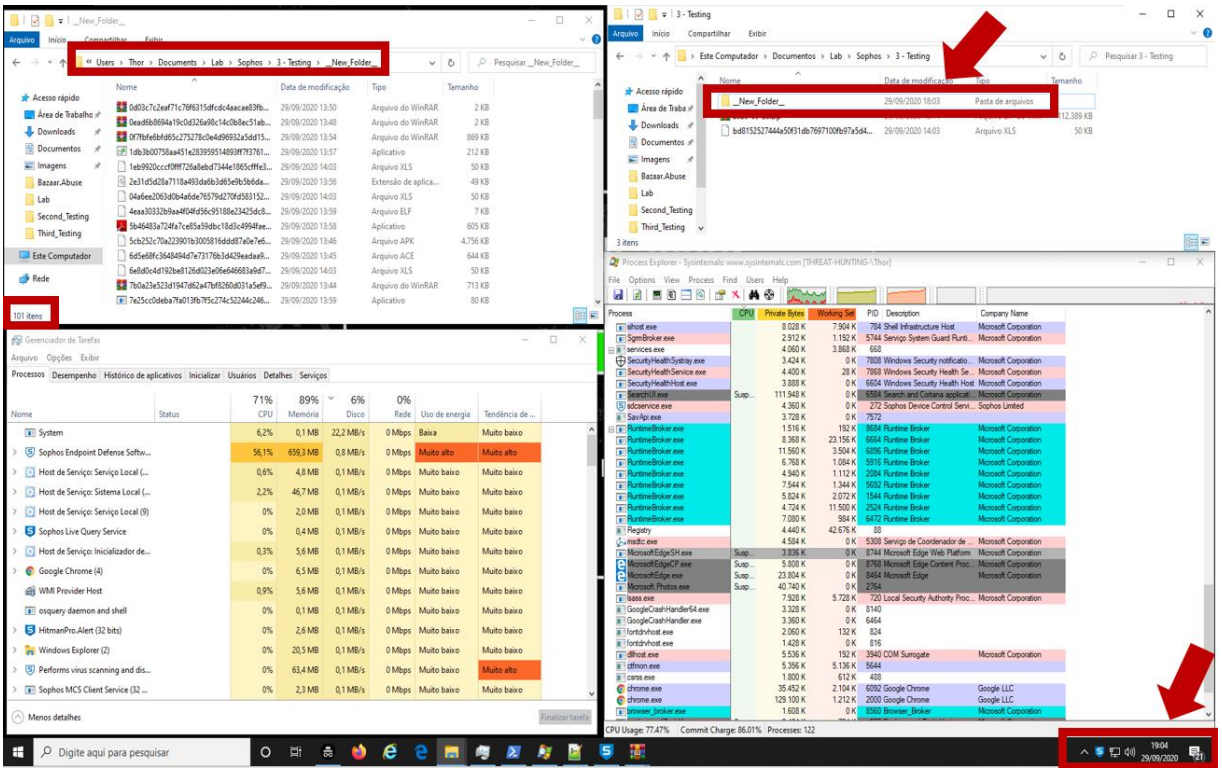

Image 1.14: __NEW_FOLDER__(Sophos) – Malware manipulation

During this stage a very strange situation caught our attention, all **101 malwares** were moved to new folder, after almost one hour waiting, no one of them malware were detected, which brought us great surprise.

All surprises forced us to perform an unscheduled test for this stage.

3.4 Fourth Test

The fourth stage of the tests (**unscheduled**) using "*Varredura/SCAN*" action by local agent scan, to perform a complete scan on the machine, manually, in this way, all malware should be eliminated, as they are already known malware as mentioned earlier

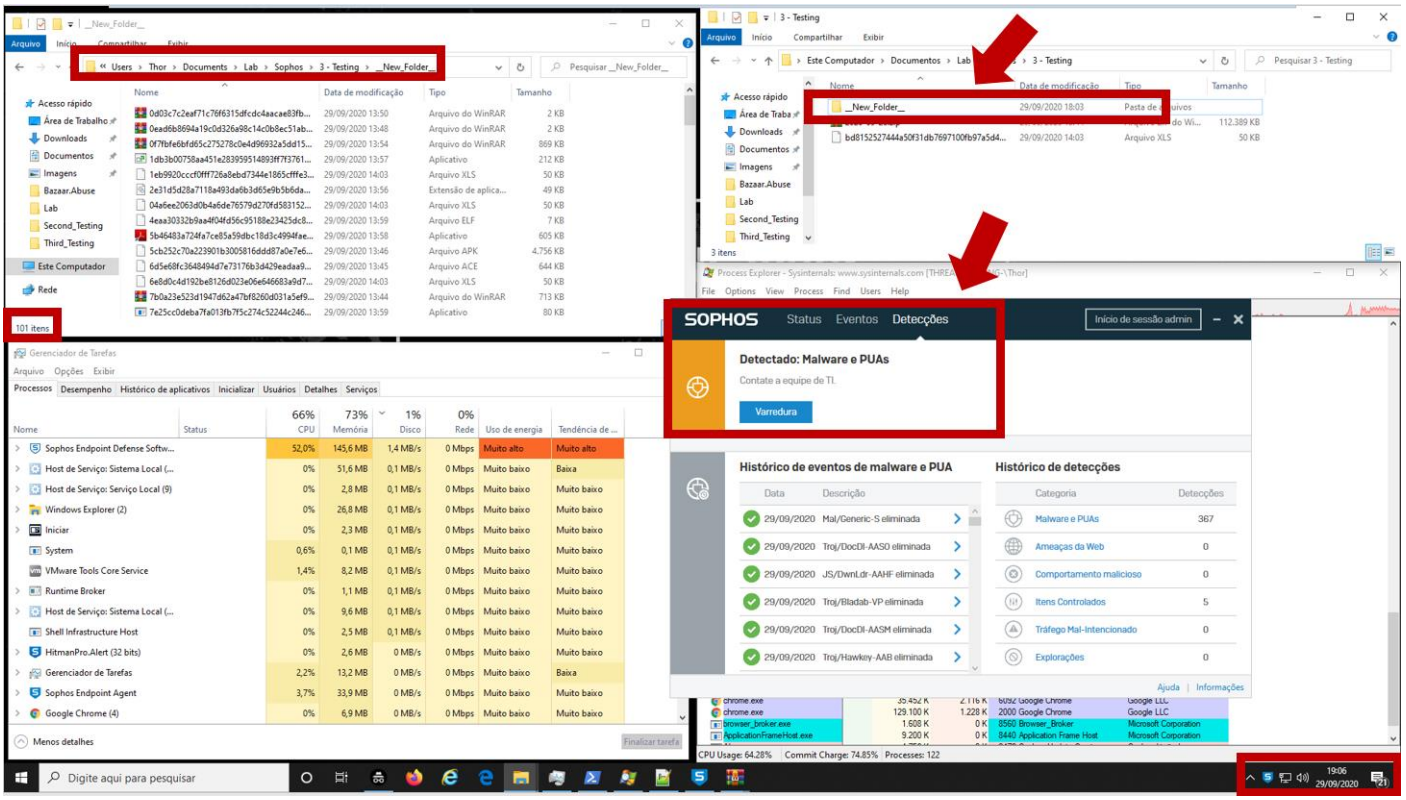

Image 1.15: Scan by agent

After this action, we are waiting the results of this scan in our machine, unfortunately it was not clear something about the product, like a detection time, flow of detection engines, machine performance and mainly inefficient of detection, when we look inside of the Sophos Central in the Windows 10 Machine, we found that the console was very alert with detections

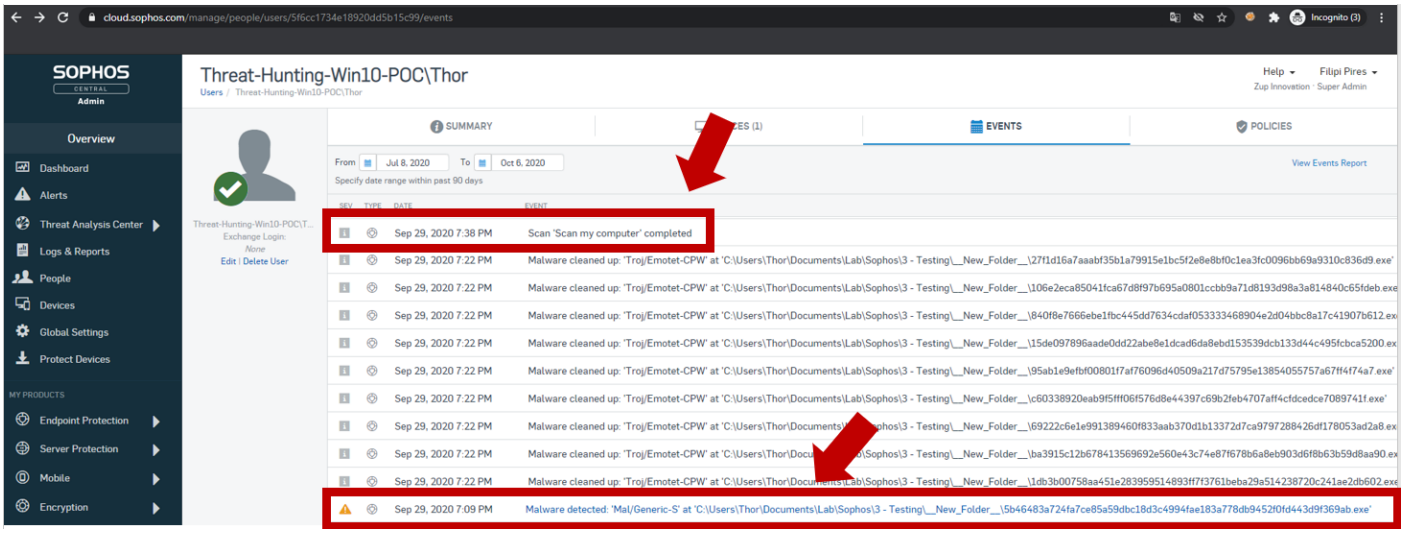

Image 1.16: Sophos Central – *Detection by Threat-Hunthing Machine*

So, to finalize this test we deleted all files manually, because the Sophos Endpoint Security didn't detect more than 80 malwares

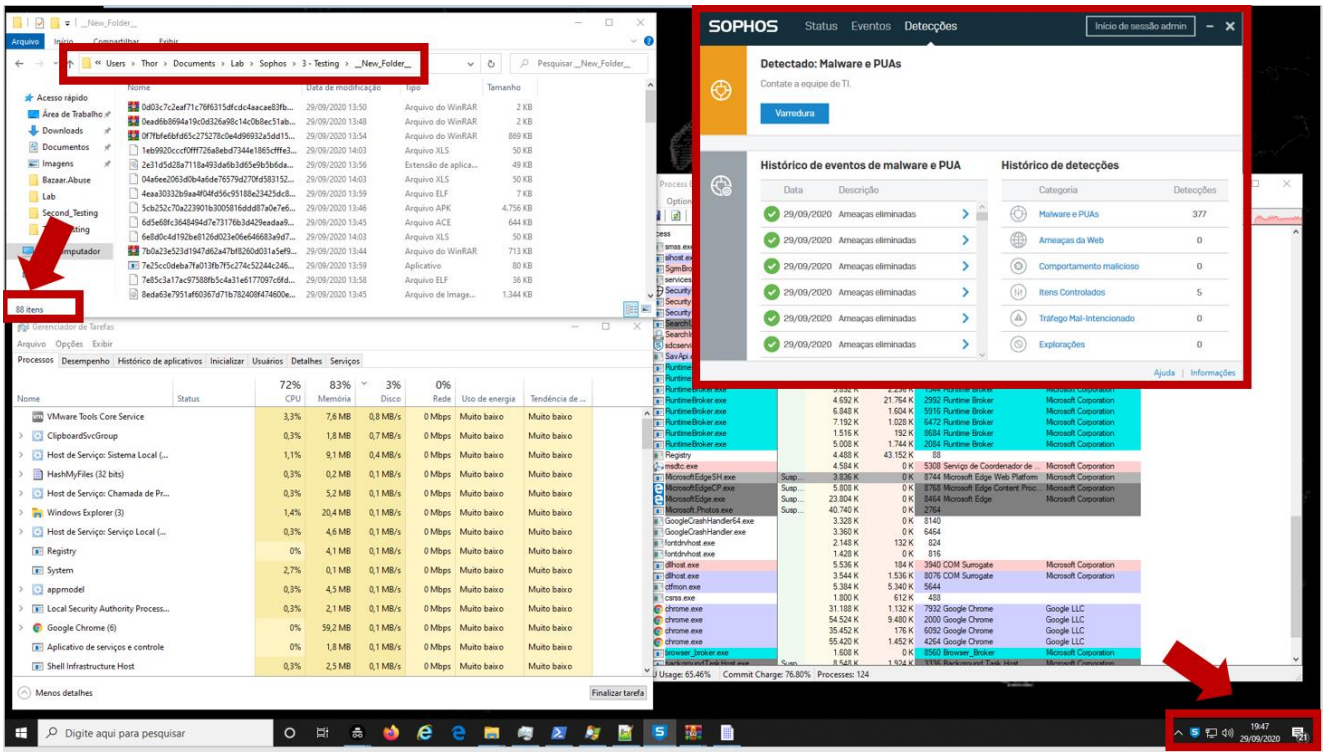

Image 1.17: Sophos Central – *Scan Finished by Threat-Hunthing Machine*

All those files you can find and download from **Malware Bazaar in the *url* below**.

hxxps://mb-api.abuse.ch/downloads/2020-09-28.zip

So, the question here is: How is it works? Detection by pattern? Signature? NGAV? ML?

Below You can see the access in Sophos Console with all logs.

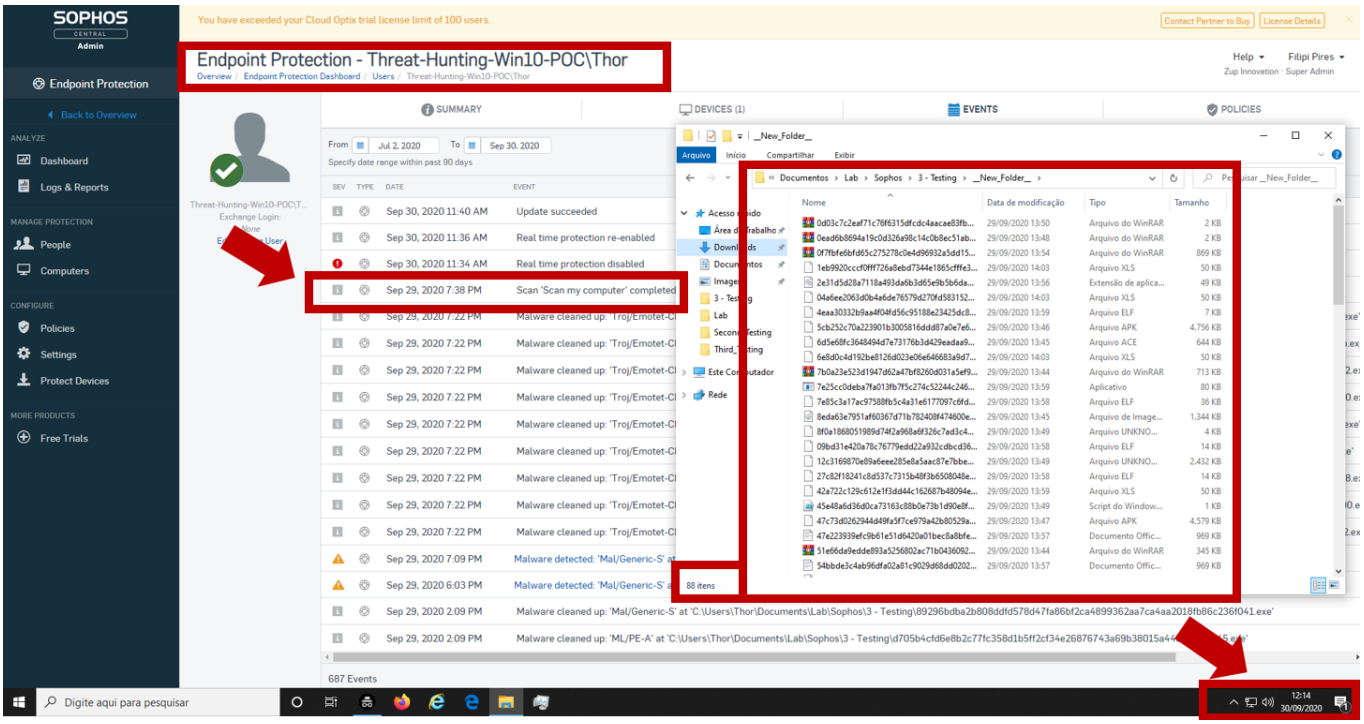

Image 1.18: Sophos Central – Next Day LOGs

3 Impact

At the end of this test, it was possible to verify that there were **88 (Eighty-eight)** 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.
- **Problem during the first test - unzipping ZIP file (detection time)**
    - During this test it was possible to see that the Sophos Endpoint Solution took almost 3 hours to realize all detections in our environment test, that is, if the attack happened in the same time in the victim, this user could click in anyone of the samples and could be infected, because it's not clear how works the prevalence, maybe priority of the engine in the detection flow.
- **High CPU and memory consumption during the detection process;**
    - Very high load on CPU and Memory used by many processes requested by AV, which directly impacts the user experience.
- **Malicious .RAR, .7z,.Zip in others compress files NOT Detected**
    - As we can see in many samples (.RAR, .7z, .Zip, etc) it's not detected like a

Malicious, we used many different sources to prove that is sample it's malicious.

- **After second test more than 101 Malwares not detected when moved to another folder;**
    - This is the big surprise, after this moviment, no one malware it was detected.
- **Malicious Files (.RAR, .7z,.Zip) without necessity of password can be executed NOT Detected**
    - As we can see in many samples (.RAR, .7z, .Zip, etc) 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 DLL files Not Detected**
    - *DLL* 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.
- **After the ScanNow and the end of this test, we had more than 88 Malwares not Detected, that is, almost 14% in unsuccess detection**
    - More than 88 Malwares inside of Windows 10 machine, not detected by Sophos Endpoint Solution, totally dangerous.

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;
- Validate why we have high consumption in the machine with CPU and Memory during the any detection process.
- Validate why we have high consumption in the machine with CPU and Memory during *ScanNow* and Scan provide by local agent.
- The best practices of the configurations will be revalidated with the Sophos team;