LOCKBIT 2.0 RANSOMWARE BECOMES LOCKFILE RANSOMWARE WITH A NEVER-BEFORE-SEEN ENCRYPTION METHOD

SEP 21, 2021 | MOSHE HAYUN

Threat actors are constantly improving their attack mechanisms to gain an advantage over cybersecurity defenses. Sometimes this involves new malware, other times this means making iterative adjustments to previously successful malware to exploit new vulnerabilities or use new attack techniques to evade and breach underprepared network environments.

We have seen ransomware evolve in this way previously, perhaps most famously with Petya and NotPetya (https://www.csoonline.com/article/3233210/petya-ransomware-and-notpetya-malware-what-you-need-to-know-now.html). Ransomware is one of the top threats (https://www.deepinstinct.com/2021/07/22/2021-threscape-cyber-threat-landscape-report/) to businesses, with DarkSide (https://www.deepinstinct.com/2021/06/04/the-ransomware-conundrum-a-look-into-darkside/), WastedLocker (https://www.youtube.com/watch?v=qQLdVXb4hOU), and many others leaving their mark on the international business landscape.

In this post we provide analysis on an emerging ransomware variant called LockFile (https://threatpost.com/lockfile-ransomware-avoid-detection/169042/) which evolved from Lockbit 2.0 (https://threatpost.com/lockbit-ransomware-proliferates-globally/168746/) and has breached security defenses using new attack techniques.

LockFile's Unique Encryption

Most ransomware operates in a similar way. It generates an encryption key and uses it to encrypt the entire binary of the file, corrupting all data.

Lockfile works differently. Rather than encrypting the entire file, it intermittently encrypts 16 bytes at a time. For textual data, this means that part of the file will remain readable. For files where the structure is important (such as a pdf), it will corrupt the file and make it unusable.

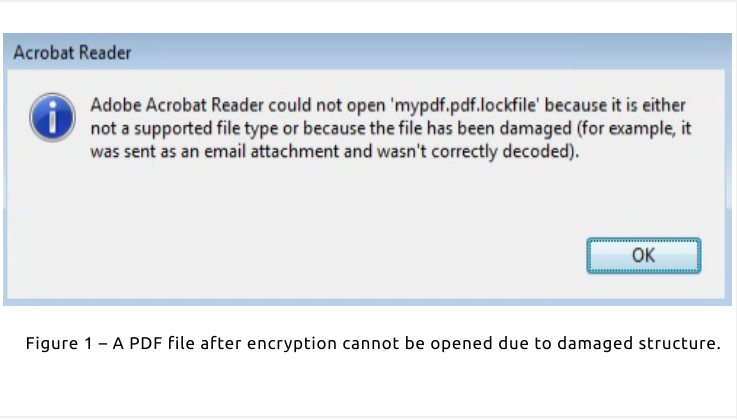

Figure 1 – A PDF file after encryption cannot be opened due to damaged structure.

Additionally, LockFile ransomware does not encrypt certain file extensions (for example .exe and .dll), a common behavior amongst a variety of ransomware. The purpose of this encryption technique is to allow the operating system to still function for the victim, albeit only by using corrupted data, ensuring the infected organization pays the ransom.

One aspect of LockFile that makes it different from other ransomware is that it does not attack image files (jpeg, bmp, giff, jpg). This is a curious approach. Does LockFile do this to preserve a victim's animal pictures, or is this simply coincidental?

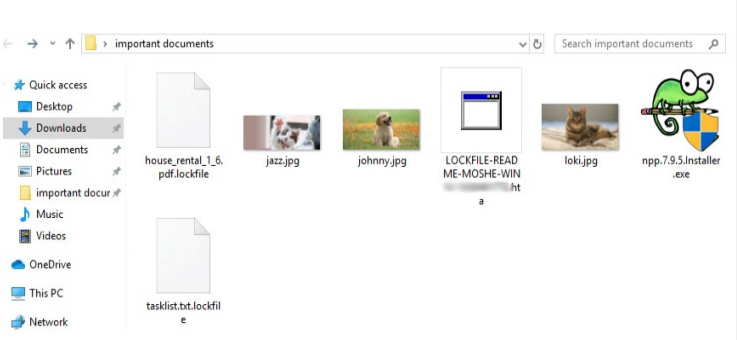

Figure 2 – After running LockFile, images remained unharmed.

RELATED POSTS

Healthcare vs. HIVE Ransomware: How to Protect Yourself (https://www.deepinstinct.com/2021/08/31/healthcare-vs-hive-ransomware-how-to-protect-yourself/)

READ MORE (HTTPS://WWW.DEEPINSTINCT.COM/2021/08/31/HEALTHCARE-VS-HIVE-RANSOMWARE-HOW-TO-PROTECT-YOURSELF/)

Re-Emphasizing Prevention in the Face of Sobering Security Realities (https://www.deepinstinct.com/2021/08/31/re-emphasizing-prevention-in-the-face-of-sobering-security-realities/)

READ MORE (HTTPS://WWW.DEEPINSTINCT.COM/2021/08/31/RE-EMPHASIZING-PREVENTION-IN-THE-FACE-OF-SOBERING-SECURITY-REALITIES/)

Black Hat 2021 + Def Con 29 – New Research on Excel 4.0 Macros (https://www.deepinstinct.com/2021/08/31/black-hat-2021-def-con-29-new-research-on-excel-4-0-macros/)

READ MORE (HTTPS://WWW.DEEPINSTINCT.COM/2021/08/31/BLACK-HAT-2021-DEF-CON-29-NEW-RESEARCH-ON-EXCEL-4-0-MACROS/)

What is the Strategy Behind Encrypting Only Part of the File?

To understand the thinking behind encrypting only part of the file rather than the entire thing, think of a file like an enormous puzzle. When you encrypt the file, you scramble the puzzle image to the point where you can't recognize the original. At that point no one could be tricked into thinking that this is a legitimate puzzle.

But what if a large portion of the puzzle remained untouched? Without careful examination, you might not notice. As such, it may well be that only part of these files are encrypted by design to obfuscate the threat. When a full file is encrypted, it is quite easy to determine that the file has been tampered with. However, using intermittent encryption may be a new tactic to avoid detection.

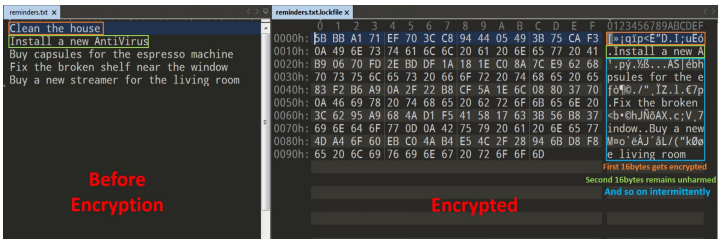

Figure 3 – Plaintext file being partially encrypted.

Old Dog, New Tricks

So, who is behind this new ransomware?

Some speculate that the Conti ransomware gang is responsible based on the email address in the ransom note (contact@contipauper.com).

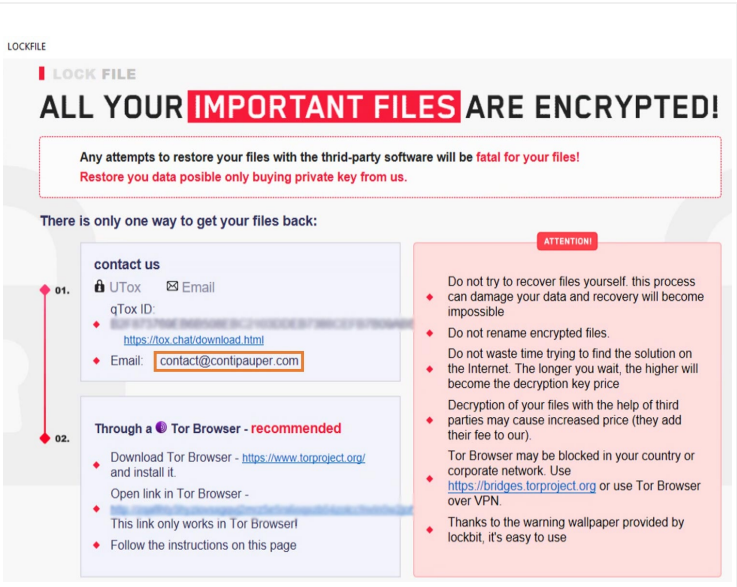

Figure 4 – New LockFile ransomware.

But a close examination of the note reveals a striking similarity to the note used in LockBit 2.0.

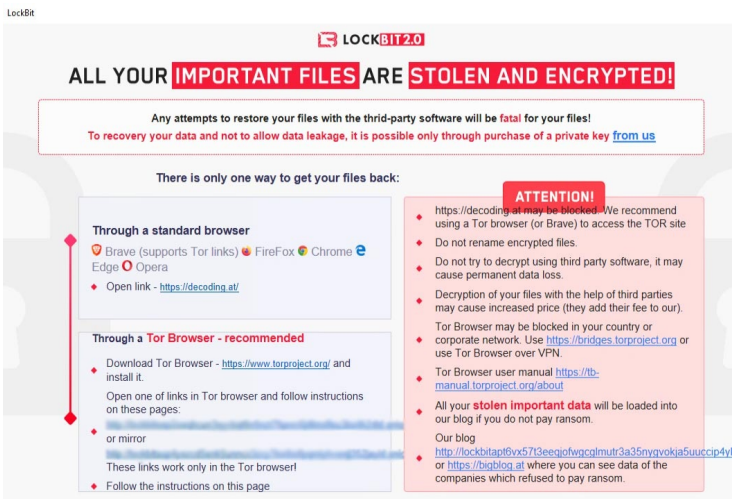

Figure 5 – LockBit 2.0 ransomware note from earlier this year.

Although the ransom note is nearly identical (the font, colors, and formatting are the same) LockBit 2.0 uses 32bit files while all the LockFile samples we found were 64bit. The 64bit payload Lockfile narrows the target range of potential victims since 64bit won't work on machines with older operating systems or 32bit operating systems.

This is interesting because threat actors typically prefer to keep a wider scope, not limiting their targets to specific OS architectures. Perhaps Lockfile's developers had performance aspects in mind during the development process which led them to this decision?

LockBit 2.0 and LockFile: What are the Differences?

There are three more noticeable 'features' which differentiate LockBit 2.0 and LockFile:

1. LockBit 2.0 did not use the unique intermittent encryption we saw in LockFile.
2. LockBit 2.0 used wmic.exe only for deleting shadow copies, while in LockFile we saw their extensive use to terminate any process related to virtualization or databases the machine is connected to.
3. LockBit 2.0 encrypted victim's images, whereas LockFile did not do image encryption.

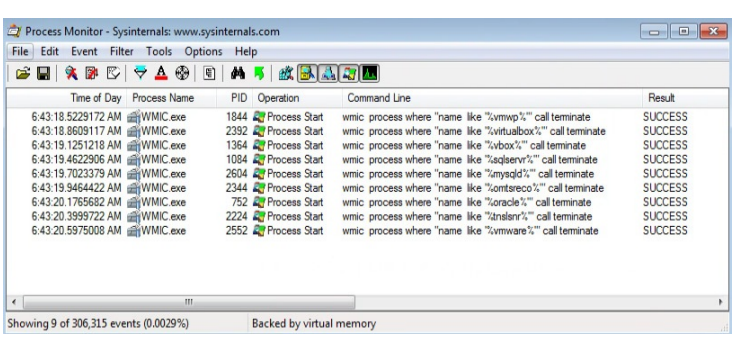

Figure 6 – LockFile extensive wmic.exe usage.

Mitigating the Threat

According to sources familiar with the threat infection chain, the latest version of the ransomware is using two very new attack vectors:

- ProxyShell (https://www.bleepingcomputer.com/news/microsoft/microsoft-exchange-servers-scanned-for-proxyshell-vulnerability-patch-now/) – Microsoft Exchange servers remote code execution vulnerability

- PetitPotam NTLM relay attack (https://www.bleepingcomputer.com/news/microsoft/new-petitpotam-attack-allows-take-over-of-windows-domains/) – new technique used to perform an NTLM relay attack

To mitigate the ProxyShell vulnerabilities, one must simply patch the exchange servers. (CVE-2021-31207 (https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2021-31207), 2021-33473 (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34473), 2021-34523 (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34523), 2021-31206 (https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-31206))

Mitigating the PetitPotam requires more than just downloading a patch – it necessitates following a guide Microsoft released (https://support.microsoft.com/en-us/topic/kb5005413-mitigating-ntlm-relay-attacks-on-active-directory-certificate-services-ad-cs-3612b773-4043-4aa9-b23d-b87910cd3429), which first informs a user about whether they are affected and includes details on how to configure one's domain controllers to mitigate this attack vector.

How Deep Instinct Stopped LockFile

Threat actors will continue to find flaws and create sophisticated ways to attack your environment. The onus is on all of us to stay one step ahead.

When it comes to preventing LockFile ransomware, Deep Instinct is the answer. We detect it pre-execution without any updates or modifications to our product and stop it in its tracks.

If you'd like to learn more about our ransomware prevention capabilities – including our industry best $3M no-ransomware guarantee – we'd be delighted to give you a demo (https://www.deepinstinct.com/request-a-demo/).

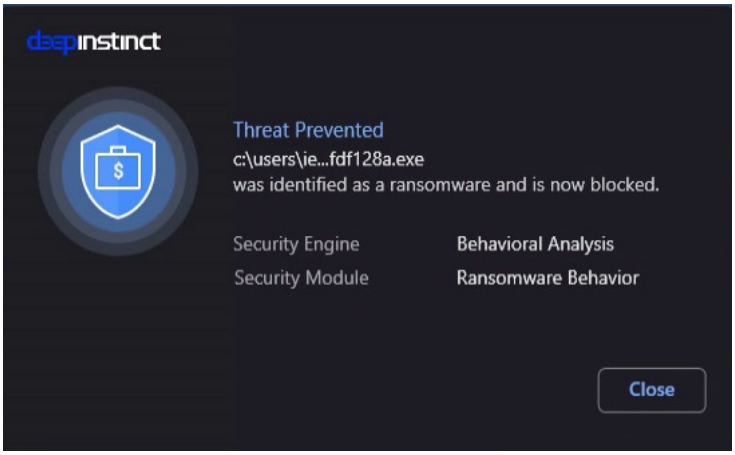

Figure 7 – Deep Instinct prevents LockFile threat.

IOCs:

Lockfile's SHA256 Referred to in this Blog Post:

SHA256	File Type
2a23fac4cfa697cc738d633ec00f3fbe93ba22d2498f14dea08983026fdf128a	64-bit executable
cafe54e85c539671c94abdeb4b8adbef3bde8655006003088760d04a86b5f915	64-bit executable
bf315c9c064b887ee3276e1342d43637d8c0e067260946db45942f39b970d7ce	64-bit executable
a926fe9fc32e645bdde9656470c7cd005b21590cda222f72daf854de9ffc4fe0	64-bit executable

Company

About Deep Instinct (https://www.deepinstinct.com/about-us/)
Careers (https://www.deepinstinct.com/career-opportunities/)
Board of Directors (https://www.deepinstinct.com/board-of-directors/)
Leadership Team (https://www.deepinstinct.com/leadership-team/)
Our Customers (https://www.deepinstinct.com/deep-instinct-customers/)
In the News (https://www.deepinstinct.com/news/)
Integrations and Compliance (https://www.deepinstinct.com/compliance-certification-evaluation/)
Contact Us (https://www.deepinstinct.com/contact-us/)

REQUEST A DEMO (HTTPS://WWW.DEEPINSTINCT.COM/REQUEST-A-DEMO/)

Privacy Policy (https://www.deepinstinct.com/privacy-policy/) | Terms of use (https://www.deepinstinct.com/terms-of-use/)

(https://twitter.com/DeepInstinctSec)
(https://www.linkedin.com/company/deep-instinct)
(https://www.youtube.com/channel/UCYerfisJf3hc9QOWmic1G9Q)
(https://www.instagram.com/deepinstinct_/)
(https://www.facebook.com/DeepInstinctInc)