Netwalker ransomware full analysis

seguranca-informatica.pt/netwalker-ransomware-full-analysis/

September 3, 2021

maxtrilha

Netwalker is a data encryption malware that represents an evolution of the well-known Kokoklock ransomware and has been active since September 2019. This article will detail the specific technical features of the Netwalker ransomware. We will analyze what Netwalker is, how it works, and how you can avoid falling victim to this threat.

In the new version of this piece of ransomware, the threat is not compiled into a PE file. Threat actors are using PowerShell scripts to load the threat into the memory (using a well-known technique called reflective loading) and make it a fileless threat, more difficult to detect and analyze. For that, malware operators achieve persistence and evade detection by abusing tools that are already in the system to initiate attacks.

Recent samples of Netwalker are not distributed via social engineering attacks. Instead, it is loaded into the memory via DLL injection during a targeted attack. Thus, it doesn't need a Windows loader to execute. This is a technique used for several PowerShell scripts, such as PowerSploit's Invoke-Mimikatz, during Red Team operations. Figure 1 shows the PowerShell script used to initiate the infection process.

Figure 1: *PowerShell script with the malicious payload encoded in Base64.*

After decoding the initial payload, a byte array is obtained. As shown in Figure 2, an XOR call is used with the key “0xA9” in order to obtain the next stage.

Figure 2: *Byte array and XOR call executed during the ransomware infection process.*

When the task is terminated, the obfuscation phase is skipped. Now, a readable form of the script is obtained. Two DLL files are coded in two-byte arrays along with the source code responsible for executing the encryption process. Notice that the code is quite obfuscated in order to make its analysis difficult.

Figure 3: *Part of the source code with two DLL files — for x64- and x86-bit OS.*

During the infection process, the script determines the system version — x64 or x86 — to execute the correct DLL version.

Figure 4: *The right DLL version is executed (x64 or x32) depending on the target OS.*

The reflective DLL injection is performed after some API addresses are resolved from kernel32.dll.

Figure 5: *API addresses resolved from kernel32.dll during the reflective DLL injection.*

Before starting the loading task, the script deletes shadow volume copies and prevents the victim from using shadow volumes to recover the encrypted files. Next, the DLL is loaded onto the memory system and the encryption process is initiated.

Figure 6: *Shadow Copies deleted during the ransomware loading process.*

Ransomware execution and config

The ransomware uses some cryptographic functions including SHA256, CRC32 and RC4 KSA to decrypt the malware config and to resolve all the needed functions dynamically.

Figure 7: *Offsets of cryptographic functions found.*

After decoding the obfuscated config hardcoded inside the DLL loaded in the memory, the config data can be analyzed (Figure 8).

***Figure 8:** Netwalking config decoded in runtime.*

This variant of Netwalker is similar to the past versions in terms of behavior. In detail, it terminates some target and hardcoded processes and services depicted in the following figures.

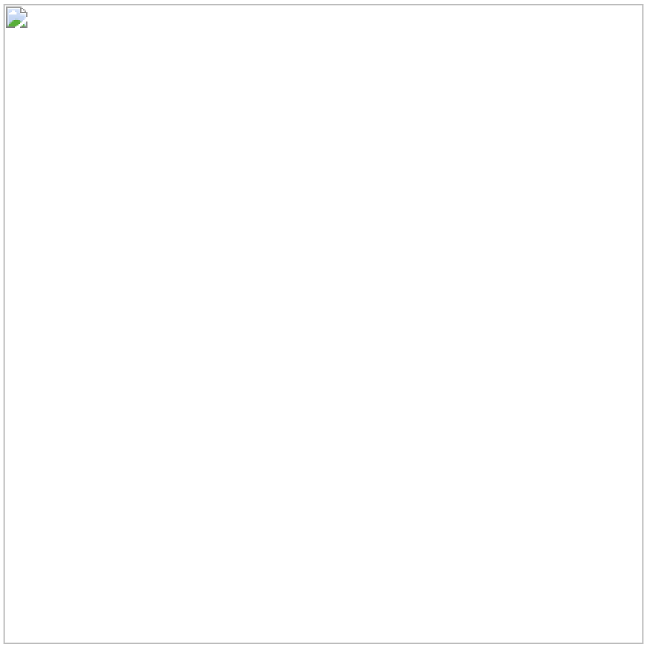


Figure 9: *Processes terminated during the Netwalker execution.*

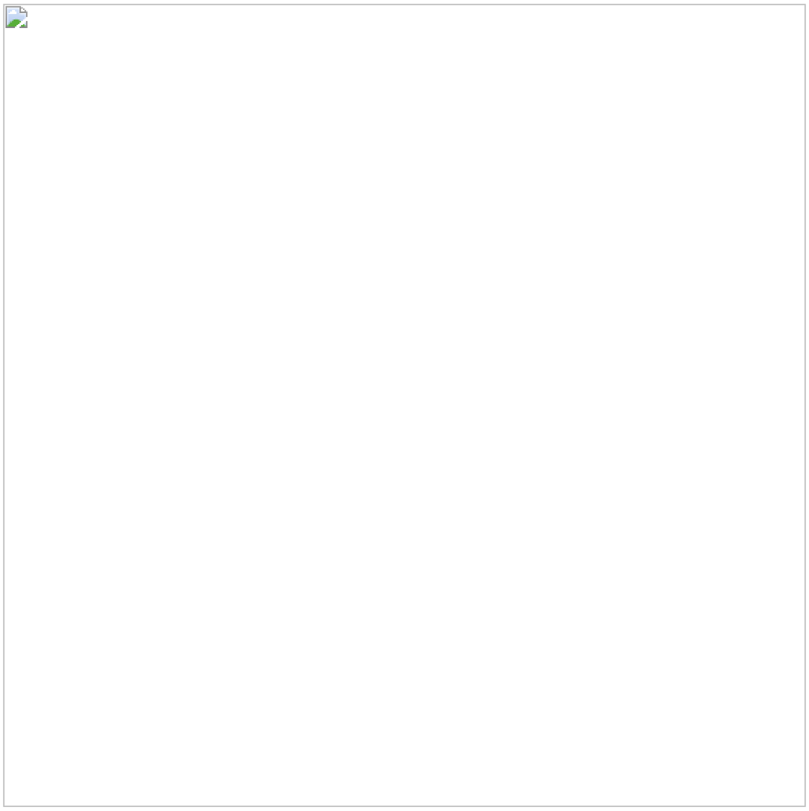
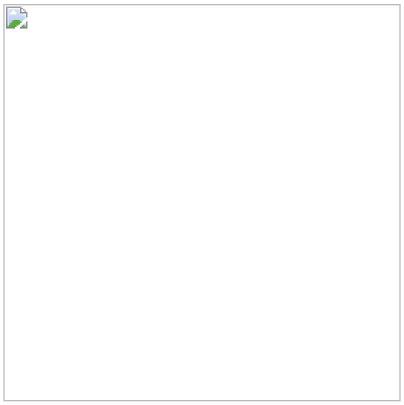


Figure 10: *Services terminated during the Netwalker execution.*

In addition, some file extensions and folders are also ignored as a way of keeping the operating system operating.



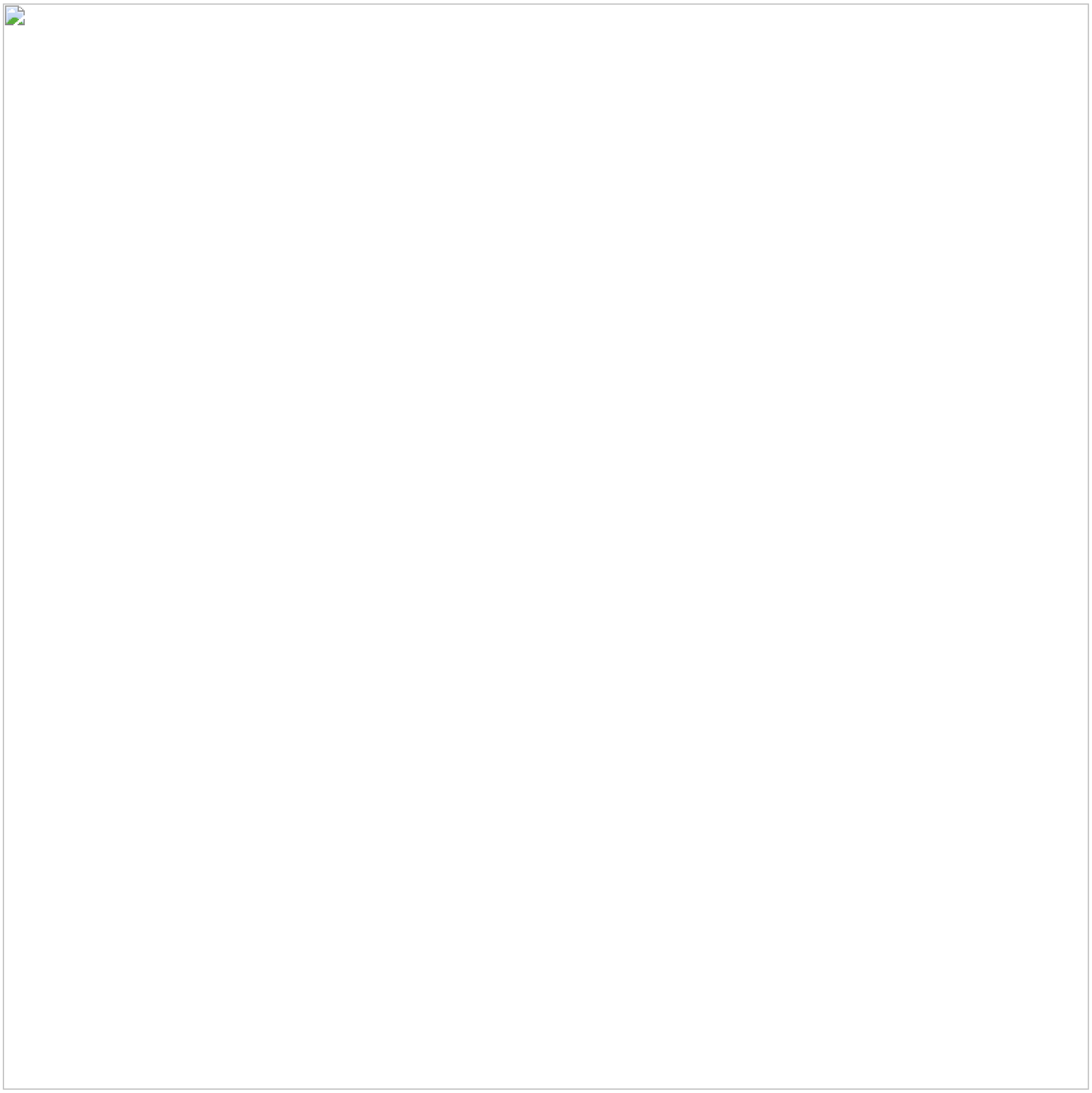


Figure 11: *File extensions and folders ignored and not encrypted during Netwalker execution.*

The JSON file with the ransomware configuration also has the ransom note available and the onion’s links to the dark web forum maintained by the operators of this threat.

Figure 12: *Ransom note and onion links available on the JSON configuration.*

The ransom note is encoded in Base64 and can be observed in Figure 13.

Figure 13: *Ransom note extracted and decoded from the config loaded in memory.*

After encrypting all the files, the text file with the ransom note is dropped by Netwalker with the fields coded in the config now populated.

Figure 14: *Ransom note dropped after Netwalker executio.*

At the end of the encryption process, it is interesting to look at the file on the left side above. It is a sample encrypted during the malware execution. Comparing this file with the original file (Figure 15), at first glance, this ransomware does not encrypt the files in their entirety.

__Figure 15:__ Block of data not encrypted by Netwalker ransomware.

Dark web forum and ransom payment

Like other modern pieces of ransomware, criminals, before executing the encryption process, exfiltrate organization data as a way of increasing the caused damage. The data is published online if the ransom request has not been paid. Netwalker operators began publishing victim data in a public blog accessible via TOR, similar to Maze, DoppelPaymer, REvil, Ragnar and others that list “non-compliant” victims along with download links to the leaked data.

Figure 16: *Ransom payment gateway and leaked data on the Netwalker blog on the dark web.*

Preventative and protective measures

The first and principal recommendation in these cases of data encryption incidents is to never pay the ransom requested by the cybercriminals. Among the preventative measures to be taken, it is important to highlight the following:

- Don’t downloaded and execute suspicious files from unknown sources
- Use PowerShell commands such as ConstrainedLanguageMode to secure systems from malicious code
- Make backups periodically and be ensure that system can be re-established quickly with the minimal loss of information
- Improve network segmentation to prevent massive propagation threats this nature
- Review and reinforce security policies
- Create awareness programs educating employees on the dangers of social engineering

---

The article was initially published by Pedro Tavares on resources.infosecinstitute.com.

Pedro Tavares

Pedro Tavares is a professional in the field of information security working as an Ethical Hacker/Pentester, Malware Researcher and also a Security Evangelist. He is also a founding member at CSIRT.UBI and Editor-in-Chief of the security computer blog seguranca-informatica.pt.

In recent years he has invested in the field of information security, exploring and analyzing a wide range of topics, such as pentesting (Kali Linux), malware, exploitation, hacking, IoT and security in Active Directory networks. He is also Freelance Writer (Infosec. Resources Institute and Cyber Defense Magazine) and developer of the 0xSI_f33d – a feed that compiles phishing and malware campaigns targeting Portuguese citizens.

Read more here.