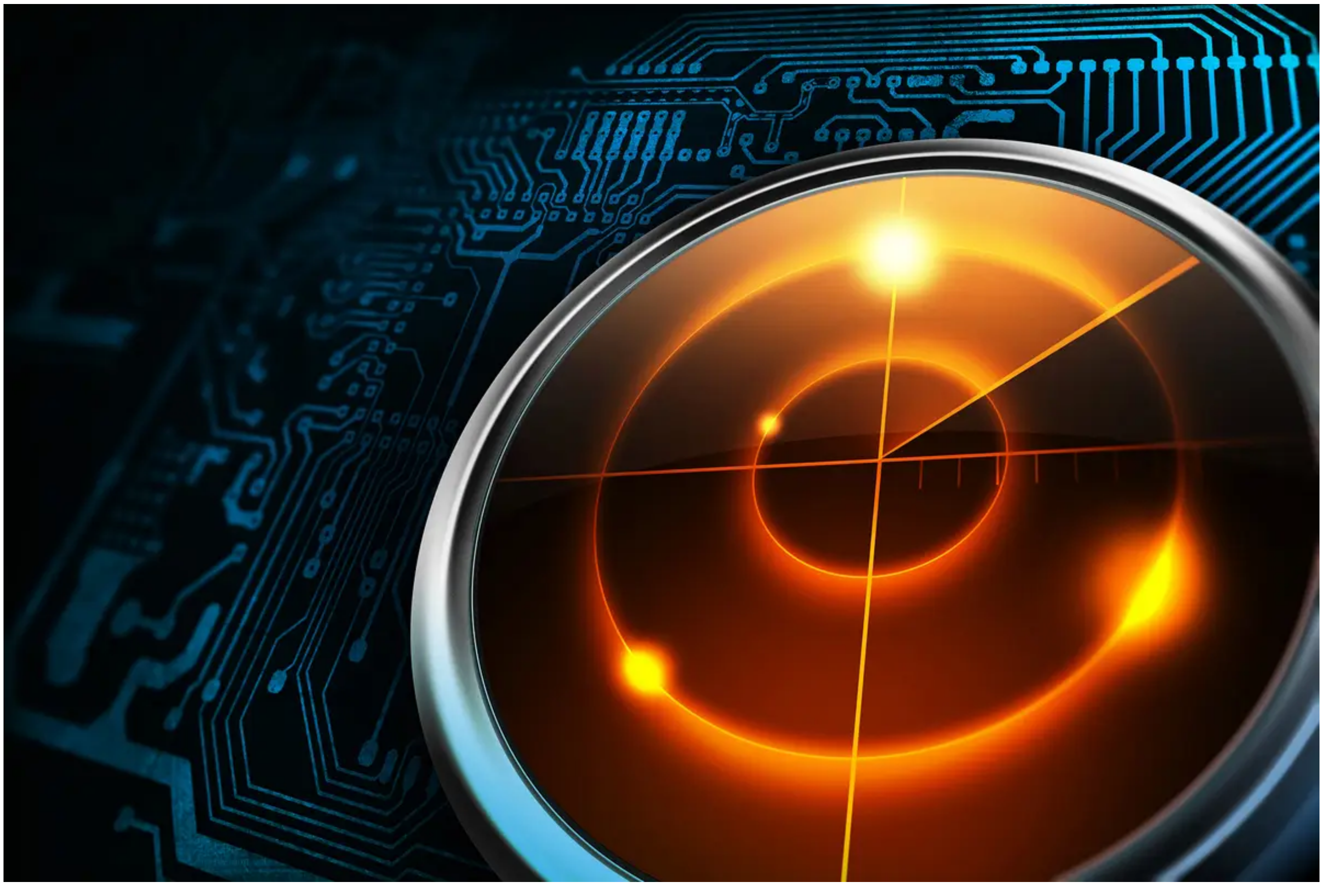
LockFile ransomware uses intermittent encryption to evade detection

csoonline.com/article/3631517/lockfile-ransomware-uses-intermittent-encryption-to-evade-detection.html

Lucian Constantin



A new ransomware threat called LockFile has been victimizing enterprises worldwide since July. Key to its success are a few new tricks that make it harder for anti-ransomware solutions to detect it.

The threat uses what researchers from antivirus vendor Sophos call “intermittent encryption,” meaning it only encrypts chunks of data inside a file instead of its complete contents. This speeds the encryption process, or better said data corruption process, significantly but also tricks ransomware protection systems that rely on statistical analysis to detect potentially unauthorized file encryption.

LockFile built with evasion in mind

LockFile uses multiple techniques designed to evade detection, starting with its own executable file which is both packed and malformed. The first section of the file is full of zeroes and is followed by a second section that contains encoded data. Three functions located at the end decode the data from the second section, place it into the first section, and then jump to that code to execute it. The goal of this routine is to throw off endpoint protection software that monitors file execution.

The malware then leverages the Windows Management Interface (WMI) to scan for and kill important processes associated with business applications including Hyper-V virtual machines, Oracle VM Virtual Box manager, Oracle VM Virtual Box services, Microsoft SQL Server, MySQL database, Oracle MTS Recovery Service, Oracle RDBMS Kernel, Oracle TNS Listener and VMware virtual machines.

The goal of killing these processes is to remove any system locks put on databases, virtual machines, or configuration files put by those applications so that the ransomware can encrypt them. By leveraging the WMI, the processes will appear to be terminated by the system itself, not by the ransomware executable. This is another detection evasion technique that is also designed to complicate incident response.

Another noteworthy trick is the way in which LockFile performs operations on files. The malware doesn't directly modify files on disk, but maps them into the system's RAM memory first, performs the modifications there and then relies on the Windows System process to commit the modifications to disk.

To a behavior monitoring product, this will appear as input/output (I/O) operations performed by the OS itself, not by a potentially suspicious process. It will also happen with a delay that can range from seconds to minutes, depending on how busy the disk is.

LockFile is not the first ransomware threat to use memory mapped I/O. Maze and WastedLocker have also used this technique, but it is not very common, Mark Loman, Sophos's director of engineering for Next-Gen Technologies, said in a blog post.

Intermittent encryption

The use of intermittent encryption, however, is a new development that the Sophos researchers have not seen before in ransomware. Other threats like LockBit 2.0, DarkSide and BlackMatter have used partial encryption, encrypting only the beginning of documents to speed the process, but LockFile's approach is different and significant.

From a security perspective, incomplete encryption is bad because it leaves data exposed, but the goal of ransomware is not data privacy. It is controlled and reversible data corruption that just uses encryption as a tool. Therefore, ransomware doesn't need to encrypt the full contents of files but just enough to make them unusable to the user, which is what LockBit 2.0, DarkSide and BlackMatter achieve by encrypting the starting portion of files.

LockFile's approach, however, is to encrypt every other 16 bytes of a file. So, the resulting files will contain 16 bytes of scrambled data, followed by 16 bytes of untouched original data, followed by another 16 bytes of scrambled data and so on. This process is not as fast as encrypting just the starting portion but has another benefit: It skews statistical analysis.

Some ransomware detection programs use statistical analysis tests to detect if a file modification is the result of file encryption. If the test indicates that a file has been encrypted, the program will block the process from modifying additional files.

This works because encrypted files, which are made up of random data, look very different from an unencrypted file to statistical analysis. One of the tests commonly used to detect statistically significant differences in data is called the chi-squared (chi^2) test.

"An unencrypted text file of 481KB (say, a book) has a chi^2 score of 3850061. If the document was encrypted by DarkSide ransomware, it would have a chi^2 score of 334 – which is a clear indication that the document has been encrypted," Loman explained. "If the same document is encrypted by LockFile ransomware, it would still have a significantly high chi^2 score of 1789811." In other words, if a detection program is calibrated by its creators to only detect and act on very big statistical differences to avoid false positives, it could miss the encryption performed by LockFile.

The last trick in LockFile's playbook is to delete itself after finishing the encryption process. This can frustrate incident response because responders will search for a ransomware binary to analyze and clean off the system.

LockFile distribution

The LockFile ransomware has been distributed by exploiting a series of vulnerabilities in Microsoft Exchange servers known collectively as ProxyShell (CVE-2021-34473, CVE-2021-34523 and CVE-2021-31207). Patches for these vulnerabilities have been available since April and May, but despite being more serious and easier to exploit than the ProxyLogon vulnerability exploited to install web shells on Exchange servers, they have not received the same level of attention. As a result, many organizations have not patched their servers.

The group behind the ransomware is also leveraging an NTLM relay attack known as PetitPotam to gain access to domain controllers inside corporate networks.