Threat Spotlight: Defray Ransomware Hits Healthcare and Education

threatvector.cylance.com/en_us/home/threat-spotlight-defray-ransomware-hits-healthcare-and-education.html

The BlackBerry Cylance Threat Research Team

Defray is a sophisticated, high-price ransomware attack aimed at very specific victims in the Healthcare and Education sectors.

Impact

One could be forgiven for having not heard of Defray in recent news. With the worlds' attention on WannaCry, Petya/Not-Petya/Petya-Like, and the return of Locky and Globe Imposter, the focus has been squarely on widespread chaos campaigns. The focus of these large attacks is garnering headlines and making money.

Defray differentiates itself by moving on specific targets in the healthcare industry, and doing so in such a way that data destruction may be its most important goal. Especially in healthcare, there is a need to be ever vigilant, as things like patient records, monitoring machines, and ultimately entire hospital operations could be affected.

We're seeing more targeted attacks aimed at certain industries, and we expect this trend to continue in the years to come.

Tech Analysis

The community was made aware of the ransomware termed Defray when our colleagues over at Proofpoint published an article on it.

In our analysis, we will discuss the Word document sample 71089d862e3bb4c3a351252fcd6d9018866c265707508ed397f3efcdf3702723. This sample drops an executable file with hash

08cf8ed94cc1ef6ae23133f3e506a50d8aad9047c6fa74568a0373d991261aa4

This infection document is tailor-made to act as a kind of sophisticated phishing attack. In this type of social engineering, the goal is to know ones' target well enough as to provide them with reassurance so that they are never suspicious as to the source.

Defrays' author has even gone so far as to do extensive research of the fake document origination organization. The name, job title, and organization details included in the document are all legitimate in an effort to make the victim feel like this is a document they should be expecting to receive.

Figure 1: Infection Document

The YouTube-style play button may seem out of place, but the other details included are there to reassure the victim that it came from a legitimate source. The executable payload is not a script or dropped by a script, but rather simply appended to the end of the OLE.

This attack vector is interesting as it does not cause Word to ask the user to enable macros, which many users are trained to be aware of.

Figure 2: Actual Ransomware Executable

The file is named ‘explorer.exe’ possibly in an attempt to show the user some name familiarity tie-in to Internet Explorer. The user may believe that by activating this, they will be taken to a site or portal where the needed document is available or that it may simply download the document for them.

The process starts quietly. It immediately calls ShellExecute, a command to open or launch something else which can be passed parameters to run as another (more privileged) user. In this case, it is cmd.exe that is chosen in order to set the system up to be even more vulnerable to attack. We will analyze the result of the shell command below.

Figure 3: Shell Command to Disable Protection

cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete

First, it deletes all the ‘shadow copies,’ or restore points, as they are commonly known. Upon the realization that something has gone wrong, the average user will try to affect repairs to their system via the last known good restore point. By anticipating this reaction, and removing that option, the malware author has ensured that most victims will start to feel out of their depths.

Figure 4: System Restore Before Infection

Figure 5: System Restore After Infection

& bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no

These commands set the system up to ignore failures on future boots and disable any automatic repair effort. This continues the process of taking away restoration options and ensures that the victim makes it back to the desktop to see that the system is still ransomed.

& wbadmin delete catalog -quiet

Delete backup catalog on local system as a final surety that the victim cannot simply recover to an earlier point. At this point, any online backups have become unavailable to the user as an option.

& wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application

These event logs related to Setup, System, Security, and Application are cleared in order to make an Incident Response (IR) effort more difficult.

& fsutil usn deletejournal /D C:

Disables Update Sequence Number journal, the journal that provides a log of all changes made to files on the volume. File creation, deletion, modification, etc. are all wiped here, again making IR more difficult.

& powercfg.exe -x -standby-timeout-ac 0 & powercfg.exe -x -standby-timeout-dc 0 & powercfg.exe -x -hibernate-timeout-ac 0 & powercfg.exe -x -hibernate-timeout-dc 0

Ensures computer will not standby or hibernate at all, regardless of plug or battery power, allowing the malware time to complete its task uninterrupted.

Next, the malware enumerates open processes and starts a copy of those processes in suspended mode. After opening the process suspended, it writes a completely new PE (including MZ header) to the memory space. This is done using ZwUnmapViewOfSection, a function that releases all memory pointed to by a section. This allows for a WriteProcessMemory call to add the malicious code in to the process. This procedure is known as process hollowing and is effective for ‘borrowing’ a legitimate process name in order to do nefarious activities.

Figure 6: Process Hollowing

Figure 7: Spawned Processes

These processes are the ones used to do the heavy lifting. They perform the actual encryption, communication with a C&C, and the writing of the ransom notes.

The code injected in to these processes is packed in the original executable, and as such we dumped the memory only once these injected processes had been initiated. Upon dumping of the hollowed process, we can find the kinaesthetic-electr[dot]000webhostapp[dot]com listing hardcoded preceding a call to form a HTTP request.

Figure 8: Hardcoded URL

We noticed the same sequence Proofpoint did, appended to the end of the file "3082 04A4 0201 0002 8201 0100 9FCF 5284". This string acts as a flag to the encryption, letting it know that a given file has already been processed. This is the only flag required in order to skip encryption; even a file appended with just these bytes will not be touched by the encrypting routine at all.

Figure 9: End of File Encrypted Flag

We should also note that unlike many other ransomware variants out today, the files are simply encrypted as is; there is no change of type or extension.

The ransom notes are written via the sub-processes as both ‘FILES.TXT’ and ‘HELP.TXT’ and are coded to drop a file in every folder.

The ransom note is written in an almost friendly tone, asking the victim to contact IT and suggesting that questions and negotiation for a lower price may be possible. Provided are two encrypted email sources: a mail.ru address, and a BitMessage identity for "fastest response."

The message to IT spells out the specific algorithms used and claims the development is advanced enough to deter local decryption attempts. It ends with a suggestion to use offline backups in the future.

Figure 10: The Ransom Note

Of particular note is that this piece of ransomware has no dependence on a network connection or the C&C to perform its basic duties. The entire process can be executed without a NIC, and even with a connection, our sample did not execute the code reaching out to "kinaesthetic." The ransom note does not provide a bitcoin wallet address up front, requiring the victim to reach out in some way. As of this writing, there had been no response to email or BitMessage, seemingly suggesting that they are abandoned because the campaign is over.

Conclusion

Defray is a highly targeted, highly sophisticated attack against specific targeted organizations in the Healthcare and Education industries. The malware author thoroughly researched his/her intended victims and carefully crafted the attack to look legitimate.

Threat behavior found shows much activity with the purpose of putting recovery out of the grasp of the average user. The ransom note is written in a friendly tone, with an almost educational piece of advice to perform offline backups in the future. We do not expect to see this variant become widespread, and instead believe that the attack on its target has been completed.

This campaign should serve as a reminder to be vigilant as to the methods an organization uses to exchange documents, and never to trust anything that violates those methods.

If you use our endpoint protection product, CylancePROTECT®, you are protected from this attack.

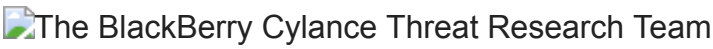
Indicators of Compromise (IoCs)

08cf8ed94cc1ef6ae23133f3e506a50d8aad9047c6fa74568a0373d991261aa4 – Binary file
2861159ee2de902bca0bde831f7194fccf3b5b6342fed486df4f3c912c4398da – Binary file
71089d862e3bb4c3a351252fcd6d9018866c265707508ed397f3efcdf3702723 – OLE document - patient_report.doc
kinaesthetic-electr[dot]000webhostapp[dot]com – C2 Domain

References

https://www.cylance.com/en_us/blog/https://threatvector.cylance.com/en_us/home/https://threatvector.cylance.com/en_us/home/threat-spotlight-petya-like-ransomware-is-nasty-wiper.html
https://www.cylance.com/en_us/blog/https://threatvector.cylance.com/en_us/home/https://thr

eatvector.cylance.com/en_us/home/threat-spotlight-wannacry-ransomware.html
https://www.proofpoint.com/us/threat-insight/post/defray-new-ransomware-targeting-education-and-healthcare-verticals



About The BlackBerry Cylance Threat Research Team

The BlackBerry Cylance Threat Research team examines malware and suspected malware to better identify its abilities, function and attack vectors. Threat Research is on the frontline of information security and often deeply examines malicious software, which puts us in a unique position to discuss never-seen-before threats.