Tech Note: DICOM File Format Exploit

DICOM® Part 10 File Format Exploit Vulnerability (a.k.a. PE/DICOM)

[Note: Please also see the references listed at the end of this summary.]

Background

In an article published on 16-Apr-2019, Markel Ortiz of the cybersecurity firm Cylera demonstrated that a feature in the DICOM Part 10 file format can be exploited to produce a Trojan horse DICOM image file that is both a valid DICOM file (although his actual example image file is not well-formed DICOM – more on this later) and a valid Windows "portable executable" (PE-format) program. He calls this hybrid file format "PE/DICOM".
(See article at: https://labs.cylera.com/2019/04/16/pe-dicom-medical-malware/.)

The design feature that Markel exploited in his article revolves around the support that the DICOM Part 10 file format provides for what are known as "dual-personality" files, which are files which are both valid DICOM-formatted images, while also being valid image files in some other format, such as TIFF. To enable this behavior, the DICOM Part 10 file format defines a free-form, 128-byte preamble in the file format specification. This preamble is supposed to be ignored by any DICOM-aware software that processes the file, while legacy imaging software such as a TIFF file viewer can read and parse the preamble, allowing it to correctly display the image data (assuming a valid TIFF header is contained in the preamble), while ignoring the DICOM metadata. (For an example discussion of such dual-personality files, see the paper, "Dual-Personality DICOM-TIFF for whole slide images: A migration technique for legacy software" by David Clunie.)

The heart of the flaw in this feature is that the DICOM specification never placed any restrictions on the contents of the preamble. This was likely done to allow maximum flexibility of legacy image file formats, but it also allows malicious software to turn the DICOM image file into an executable by placing an MS-DOS "new executable" (NE-format) header in the preamble. The NE-format header can then, in turn, provide an offset to a PE-format executable embedded in the DICOM metadata as a DICOM element.

While the actual example PE/DICOM file that Markel produced as a proof-of-concept is not a fully-compliant DICOM image file (it contains an out-of-order private group/element (0009,0000) containing the PE payload), it does prove the viability of the attack, as the NE-format offset value can jump far enough to skip both the group [0002] metadata elements and the group [0008] identifying elements of a compliant DICOM Part 10 file. In addition, many DICOM file readers are programmed in such a way as to be tolerant of certain encoding errors, such as out-of-order elements, as found in the example.

Also, while Markel's example used a private group/element (0009,0000) to embed the PE-format executable, there is no technical reason why this must be the case. PE payloads could be embedded in any DICOM group/element, including well-known public group/elements, provided they have a value representation (VR) capable of storing binary data (such as FD, FL, OB, OD, OF, OL, OV, OW, SL, SS, SV, UL, UN, US, and UV – see DICOM Standard, PS 3.5, Section 6.2 for more details). Of course, if a well-known, defined public element was used, it could cause processing errors in a DICOM application using the file and accessing such an element.

The PE/DICOM exploit that Markel describes in his article applies only to DICOM Part 10 files stored on a Windows system (which uses NE/PE-formatted executables, as discussed earlier). A similar exploit could be crafted in Linux using the "executable and linkable format" (ELF-format) executable header as the DICOM Part 10 file preamble. However, Linux itself provides an additional safeguard of not allowing normal data files to be directly executed (i.e., files must have special "execute" permissions set in order to be executable, and they do not inherit these permissions by default), so the Linux environment is inherently more resistant to this mode of attack.

Impact

As Markel points out in his article, there are three major impacts of this weakness in the DICOM Part 10 file format:

1. Evasion – Because the executable payload is hidden inside a seemingly-innocuous image file, it allows the Trojan horse malware to hide in plain sight. While virus scanning software can easily detect the malicious payload, virus scanners are often configured to ignore medical image files and directories, both for performance reasons (they can be both large and numerous) and because of the now proven-to-be-mistaken assumption that DICOM image files cannot be executed. This ability to hide in plain sight is likely the most serious impact of the PE/DICOM vulnerability.

2. Spread – Because better evasion can lead to better spread, a PE/DICOM file can be used as the basis of a multi-stage attack, where infected PE/DICOM files are introduced onto more vulnerable computers in a network, then copied onto less vulnerable computers via a second stage attack. While the DICOM networking (DIMSE) protocols cannot be used to propagate the infected PE/DICOM files (more on the reasons behind this later), weaknesses in the configuration of the Windows file sharing protocols can be exploited to spread these infected files.
3. Persistence – Since a PE/DICOM file successfully fuses patient imaging data with malware, removal of such malware is more problematic. Due to regulatory requirements, such malware cannot be simply deleted by deleting the whole file but must rather be surgically removed from the imaging data without harming the regulatory-protected patient health information (PHI). Fortunately, the structured nature of the DICOM file format makes it possible to remove an offending element containing a PE/DICOM payload without damaging the associated PHI.

Remediation

The fusion of malware with PHI significantly complicates the automated removal of such malware. For one thing, the somewhat common practice of virus scanning software automatically uploading suspicious files to the cloud for further analysis by the vendor must be disabled, as it may violate HIPAA or other PHI regulatory requirements. In addition, imaging files containing malware cannot be simply deleted but must rather be carefully parsed in order to extract the PHI without allowing the malware to remain (at least in its active form).

At this point, it is worth clarifying that there are three different states that a DICOM Part 10 file can be in with regards to PE/DICOM malware infection:

1. Clean – The Part 10 file contains no malware whatsoever.
2. PE-Active – The Part 10 file contains a valid NE-format file preamble pointing to a valid PE-format payload hidden somewhere in the DICOM data. This file is executable and therefore capable of having its malware triggered. The PE/DICOM malware can be considered activated.
3. PE-Inactive – The Part 10 file contains a valid PE-format payload hidden somewhere in the DICOM data but does not contain the NE-format file preamble necessary for execution. This file is not executable, and the PE/DICOM malware can be considered inactivated.

There are some mitigating factors which should help healthcare organizations control the spread of malware such as PE/DICOM:

- In order to trigger the malicious payload of a PE/DICOM Trojan horse, a user would need to run (execute) the file. However, the default behavior of Windows Explorer is to only execute files for which a file type association of “exefile” has been configured. This is unlikely to be the case for DICOM files with an extension of “.dcm” or, more typically, no extension. *However, any PE file, including a PE/DICOM file, can be executed directly from a Windows command prompt, simply by typing its name as the command.*
- The DICOM standard instructs Part 10 file format readers and updaters to ignore the contents of the preamble when reading in the file (i.e., to not propagate its contents) and to zero-out the contents of the preamble when writing a file (unless specifically creating files for consumption by legacy imaging applications). Thus, for properly implemented DICOM applications, importing and exporting a DICOM Part 10 file should have the side effect of inactivating any embedded PE/DICOM malware.
- The DICOM networking (DIMSE) protocols specify that applications should not transmit the preamble or the file meta elements (i.e., group [0002] elements), which will make the PE offset in any PE/DICOM preamble invalid, even if the preamble contents were somehow to be preserved. Unfortunately, the DICOM web services (such as WADO, STOW, and QIDO – see DICOM Standard, PS 3.18 for more details) do not explicitly specify that the file preamble should be ignored on read and zeroed-out on write, so there is a possibility that PE/DICOM malware-infected DICOM files could be propagated via DICOM web services.

Laurel Bridge Software’s Response to PE/DICOM

Laurel Bridge Software DICOM libraries and applications ignore the file preamble when parsing a DICOM Part 10 file (although they do, however, look for the 4-character prefix, “DICM”, which must follow the 128-byte preamble). In addition, when writing the preamble of a Part 10 file, as specified by the DICOM Standard, PS 3.10, Section 7.1, LBS DICOM libraries and applications always zero-out the preamble, thus preventing any embedded PE payload from being activated. This behavior has the side effect of inactivating any embedded PE payload that a DICOM Part 10 file might be carrying.

LBS DICOM software also does not, by default, transmit any of the preamble or file meta elements (i.e., group [0002] elements) over the network, which will further disrupt any attempt to transmit a PE-infected DICOM file over the network using DICOM network protocols, since the encoded PE offset would now be unlikely to be correct, even if the preamble was somehow preserved or re-added to the file. LBS DICOM web services also do not transmit the file preamble across the network.

Summary

LBS DICOM software will ignore the file preamble when reading a Part 10 file, and it will clear (zero-out) the file preamble when writing a Part 10 file. Thus, LBS DICOM software will not propagate any active PE/DICOM malware-infected DICOM Part 10 file. It may possibly propagate the inactive PE payloads (masquerading as legitimate DICOM metadata, either as public or private group/elements), but without the PE/DICOM file preamble; such DICOM files cannot be executed, and the PE/DICOM malware can be considered inactivated.

Planned Actions

LBS has demonstrated and is planning to release a standalone PE/DICOM scanner (detector) application for the exclusive use of our customers. It will be able to detect the presence of a weaponized Part 10 file preamble (as identified by the presence of the hexadecimal prefix **4D 5A** ["MZ"], at offset 0 in the file preamble) and the presence of an embedded PE payload within a DICOM element (as identified by the hexadecimal PE prefix **50 45 00 00 4C 01** ["PE" + <0 as a 16-bit word> + <hexadecimal **014C**, byte-swapped, indicating x86/x64 architecture>]). Such detection and the related manual or automatic cleanup capabilities may also be added to other LBS applications as executable filters or actions.

The LBS PowerTools™ DICOM File Editor will also be enhanced to provide an indication of the presence of such suspicious DICOM data elements. The File Editor already allows individual data elements to be deleted from the DICOM data set, and it already zeros-out the file preamble whenever it writes a DICOM Part 10 file.

In addition, LBS is considering adding an option to Compass™ to automatically scan input file system hot folder(s), as this is the primary point of contact where Compass could encounter PE/DICOM malware-infected DICOM Part 10 files.

References

DICOM® is the registered trademark of the National Electrical Manufacturers Association (NEMA) for its standards publications relating to digital communications of medical information. See https://www.dicomstandard.org/

DICOM Standard: https://www.dicomstandard.org/current/

DICOM Part 10 file format: http://dicom.nema.org/medical/dicom/current/output/html/part10.html#chapter_7

Clunie DA. "Dual-Personality DICOM-TIFF for whole slide images: A migration technique for legacy software." J Pathol Inform [serial online] 2019 [cited 2019 Apr 23];10:12. Available from: http://www.jpathinformatics.org/text.asp?2019/10/1/12/255397

"HIPAA-Protected Malware? Exploiting DICOM Flaw to Embed Malware in CT/MRI Imagery," Ortiz, Markel Picado; Cylera Labs, https://labs.cylera.com/2019/04/16/pe-dicom-medical-malware/

Laurel Bridge will update and re-post this document as new information becomes available.

Please contact support@laurelbridge.com if you have specific concerns regarding LBS products.