

Exposing Vulnerabilities in Deep Learning Frameworks

Kang Li

Department of Computer Science
University of Georgia



About Me

- Professor at the University of Georgia
- Director of UGA Institute for Cybersecurity and Privacy (ICSP)
- Founder of the *Disekt*, *SecDawgs* CTF Teams
- 2016 DARPA Cyber Grand Challenge Finalist



Success of AI



Success of AI (on the dark side)

Buy Online
Reviews

Easily Get More Online Reviews

Buy Yelp Reviews

Buy Yelp reviews from the most trusted source in the industry. We offer 100% real reviews from aged accounts. All have real friends, activity, check-ins etc.. Using our in house Yelp experts we form high quality reviews that will not be filtered out. We analyze all aspects of your business and ensure that your reviews are realistic. Receive unlimited 5 star reviews and start attracting more customers.

Buy Reviews Find Out More

2.2+ BILLION

42

32,425

3.88

Reggie

Zagat

Kim N.

yelp UNITED STATES OF YELP

AI for Fake Review Generation

“Leverage deep learning language models (**Recurrent Neural Networks** or **RNNs**) to automate the generation of fake online reviews for products and services”

https://www.schneier.com/blog/archives/2017/09/new_techniques_.html

Success of AI (on the dark side)



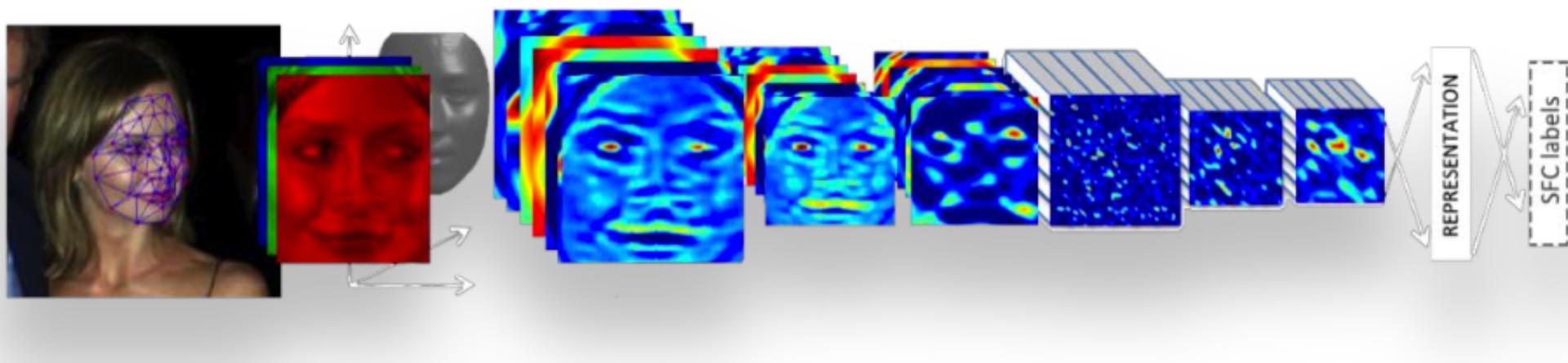
Defeating Captcha with Learning

<https://deepmlblog.wordpress.com/2016/01/03/how-to-break-a-captcha-system/>

I'm not a human: Breaking the Google reCAPTCHA

<https://www.blackhat.com/docs/asia-16/materials/asia-16-Sivakorn-Im-Not-a-Human>

Image Recognition: Flagship of AI Applications

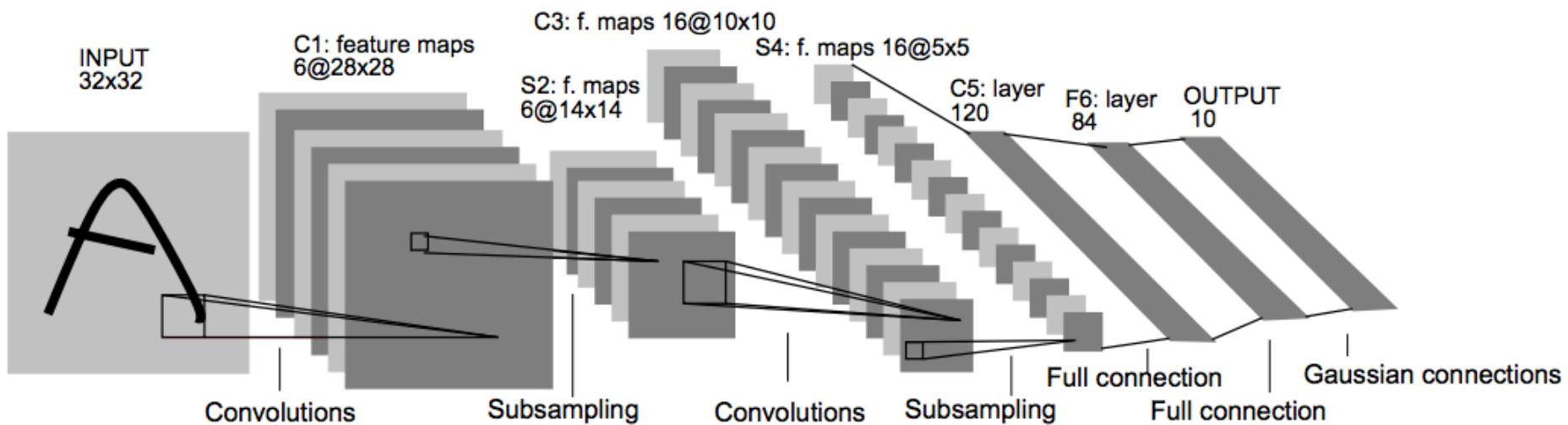


The Implementation of Deep Learning Applications

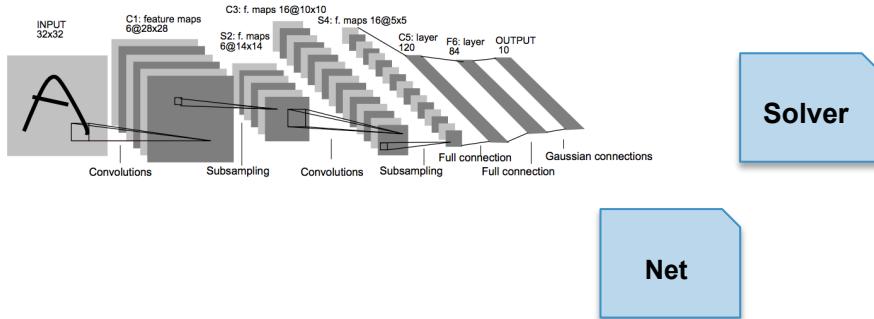


MNIST Handwriting Digits Recognition

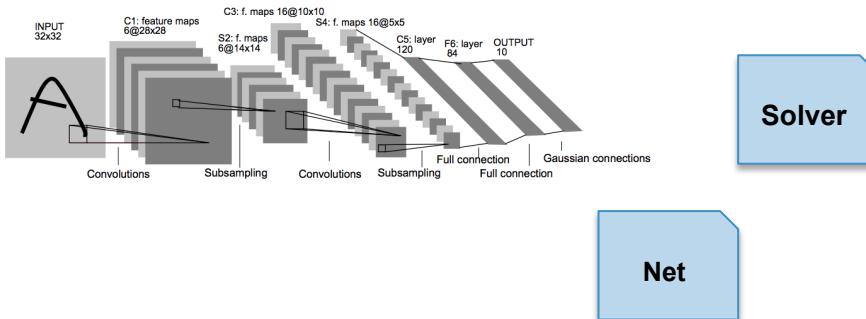
Sample Neural Network (LeNet-5) Architecture



MNIST Handwriting Digits Recognition



MNIST Handwriting Digits Recognition



```
from caffe import layers as L, params as P

def lenet(lmdb, batch_size):
    # our version of LeNet: a series of linear and simple nonlinear transformations
    n = caffe.NetSpec()

    n.data, n.label = L.Data(batch_size=batch_size, backend=P.Data.LMDB, source=lmdb,
                           transform_param=dict(scale=1./255), ntop=2)

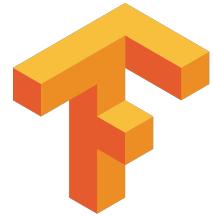
    n.conv1 = L.Convolution(n.data, kernel_size=5, num_output=20,
                           weight_filler=dict(type='xavier'))
    n.pool1 = L.Pooling(n.conv1, kernel_size=2, stride=2, pool=P.Pooling.MAX)
    n.conv2 = L.Convolution(n.pool1, kernel_size=5, num_output=50, weight_filler=dict(type='xavier'))
    n.pool2 = L.Pooling(n.conv2, kernel_size=2, stride=2, pool=P.Pooling.MAX)
    n.fcl = L.InnerProduct(n.pool2, num_output=500, weight_filler=dict(type='xavier'))
    n.relu1 = L.ReLU(n.fcl, in_place=True)
    n.score = L.InnerProduct(n.relu1, num_output=10, weight_filler=dict(type='xavier'))
    n.loss = L.SoftmaxWithLoss(n.score, n.label)

    return n.to_proto()
```

```
# The train/test net protocol buffer definition
train_net: "mnist/lenet_auto_train.prototxt"
test_net: "mnist/lenet_auto_test.prototxt"
# test_iter specifies how many forward passes the test should carry out.
# In the case of MNIST, we have test batch size 100 and 100 test iterations,
# covering the full 10,000 testing images.
test_iter: 100
# Carry out testing every 500 training iterations.
test_interval: 500
# The base learning rate, momentum and the weight decay of the network.
base_lr: 0.01
momentum: 0.9
weight_decay: 0.0005
# The learning rate policy
lr_policy: "inv"
gamma: 0.0001
power: 0.75
# Display every 100 iterations
display: 100
# The maximum number of iterations
max_iter: 10000
# snapshot intermediate results
snapshot: 5000
snapshot_prefix: "mnist/lenet"
```

Available at:
<https://github.com/BVLC/caffe/tree/master/examples/mnist>

Deep Learning Frameworks



theano



Caffe

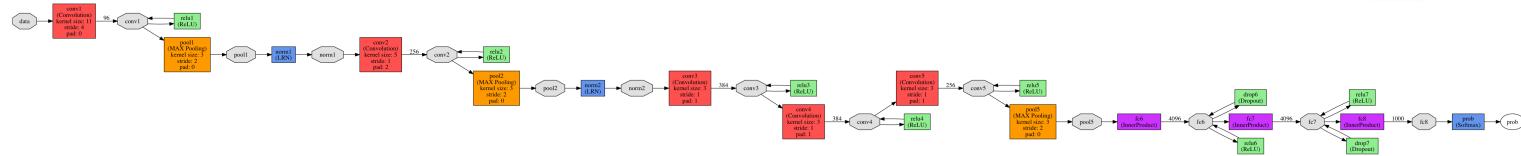


Many words about these Frameworks:

Functionality, Flexibility, Scientific Computing ...

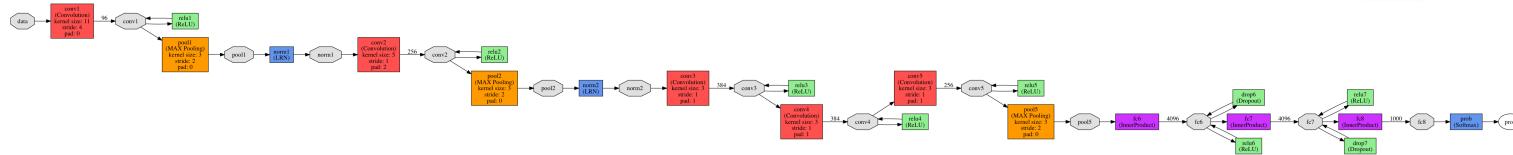
This talk is about a different aspect: **Implementation Security**

Standalone DL Application (Caffe Example)



https://github.com/BVLC/caffe/tree/master/examples/cpp_classification

Standalone DL Application (Caffe Example)

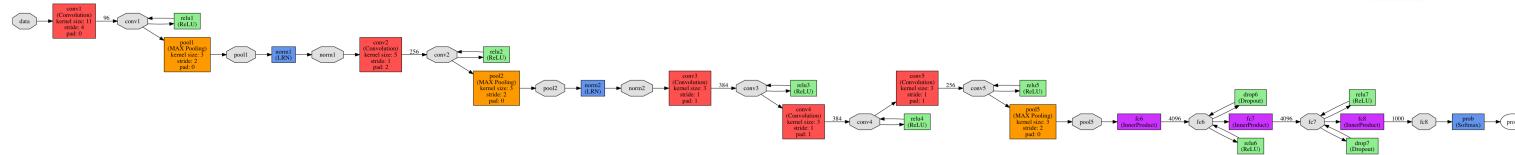


https://github.com/BVLC/caffe/tree/master/examples/cpp_classification



```
./build/examples/cpp_classification/classification.bin \
models/bvlc_reference_caffenet/deploy.prototxt \
models/bvlc_reference_caffenet/bvlc_reference_caffenet.caffemodel \
data/ilsvrc12/imagenet_mean.binaryproto \
data/ilsvrc12/synset_words.txt \
examples/images/cat.jpg
```

Standalone DL Application (Caffe Example)



https://github.com/BVLC/caffe/tree/master/examples/cpp_classification

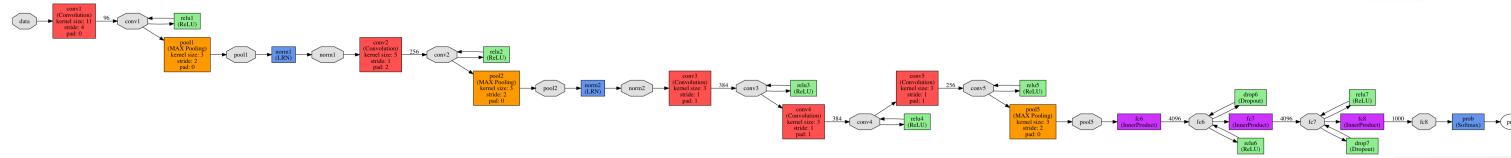


```
./build/examples/cpp_classification/classification.bin \
models/bvlc_reference_caffenet/deploy.prototxt \
models/bvlc_reference_caffenet/bvlc_reference_caffenet.caffemodel \
data/ilsvrc12/imagenet_mean.binaryproto \
data/ilsvrc12/synset_words.txt \
examples/images/cat.jpg
```

Output:

```
----- Prediction for examples/images/cat.jpg -----
0.3134 - "n02123045 tabby, tabby cat"
0.2380 - "n02123159 tiger cat"
0.1235 - "n02124075 Egyptian cat"
0.1003 - "n02119022 red fox, Vulpes vulpes"
0.0715 - "n02127052 lynx, catamount"
```

Checking Application Dependencies (Caffe)



```
~/caffe/build/examples/cpp_classification$ ldd classification | more
```

```
linux-vdso.so.1 => (0x00007ffffdf9fe000)
libcaffe.so.1.0.0 => /home/kodos/caffe/build/lib/libcaffe.so.1.0.0 (0x00007f3c86781000)
libopencv_imgproc.so.2.4 => /usr/lib/x86_64-linux-gnu/libopencv_imgproc.so.2.4 (0x00007f3c85c54000)
libopencv_core.so.2.4 => /usr/lib/x86_64-linux-gnu/libopencv_core.so.2.4 (0x00007f3c8581c000)
libprotobuf.so.8 => /usr/lib/x86_64-linux-gnu/libprotobuf.so.8 (0x00007f3c845e1000)
libopenblas.so.0 => /usr/lib/libopenblas.so.0 (0x00007f3c822ab000)
libGL.so.1 => /usr/lib/x86_64-linux-gnu/mesa/libGL.so.1 (0x00007f3c81b23000)
libpng12.so.0 => /lib/x86_64-linux-gnu/libpng12.so.0 (0x00007f3c816a8000)
libjasper.so.1 => /usr/lib/x86_64-linux-gnu/libjasper.so.1 (0x00007f3c811de000)
libIImImf.so.6 => /usr/lib/x86_64-linux-gnu/libIImImf.so.6 (0x00007f3c80f2f000)
.....
```

137 lib*.so

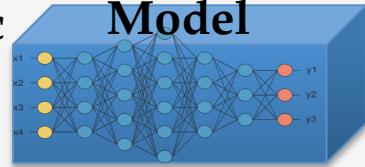
Software Layers in Deep Learning Apps

DL
Applications

Program Logic



Model



Data



DL
Frameworks

Torch theano
TensorFlow Caffe

Framework
Dependencies



GNU LibC



protobuf
Protocol Buffers



OpenCV



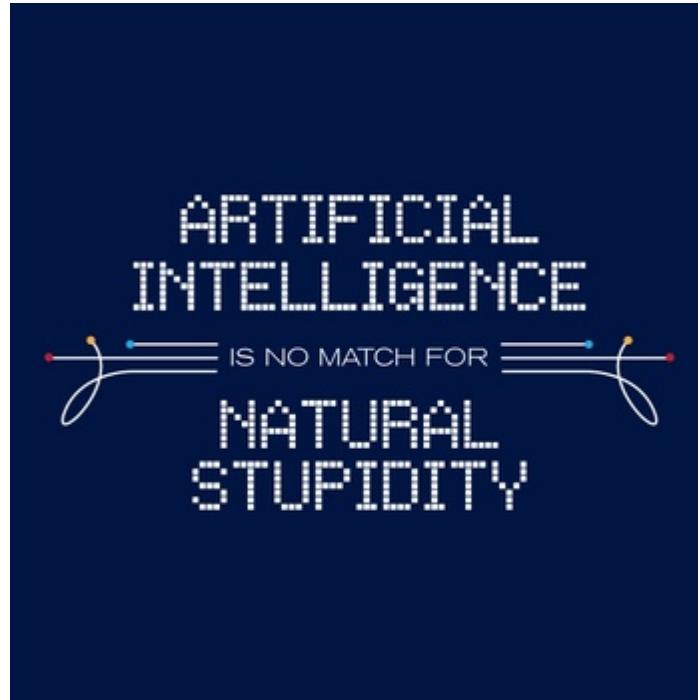
DL Framework Complexity and Dependencies

DL Framework	Lines of Code	Number of Dep. Packages	Sample Packages
Caffe	127K+	137	Libprotobuf, libz, opencv, libopenblas
 TensorFlow	887K+	97	numpy, librosa
 torch	590K+	48	xlua, qtsvg, opencv

Implementation Security of DL Applications

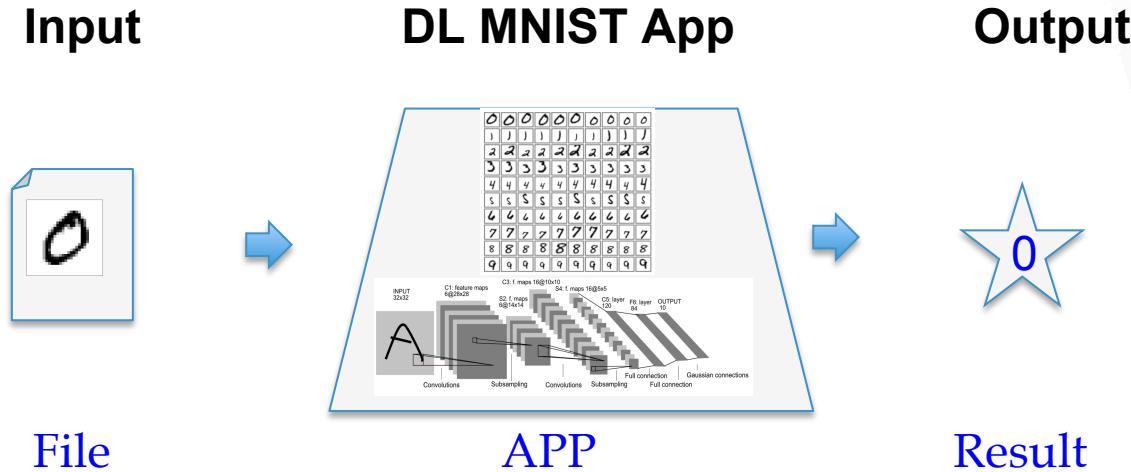


Why Implementation Security Matters?

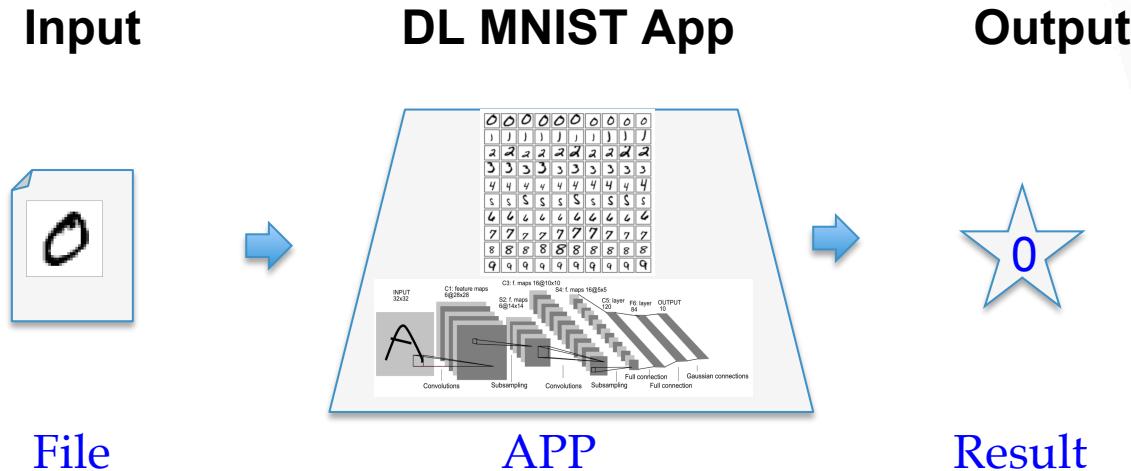


* The term was initially used by Drew McDermott in his 1976 paper “Artificial Intelligence meets Natural Stupidity”, <http://dl.acm.org/citation.cfm?id=1045340>

Why Implementation Security Matters?



DL Researcher/Developer Considers ...

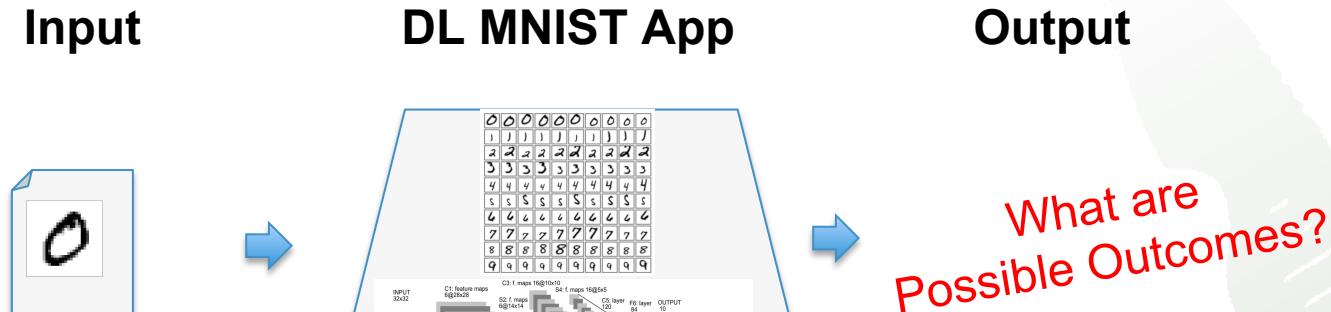


0 1 9 2 1 3 1 4 3
5 3 6 1 7 2 8 6 9 4
0 9 1 1 2 4 3 2 2 3
8 6 9 0 5 6 0 7 6 1
8 7 9 3 9 8 5 1 3 3
0 7 5 9 8 0 9 4 1 4
4 6 0 4 5 6 1 0 0 1
2 1 6 3 0 2 7 1 7 9
0 2 6 7 8 3 9 0 4 6
7 4 6 8 0 7 8 3 1 5



Error Rate: 0.4% ...

DL Researcher/Developer Considers ...



File

APP

Result

0 1 1 9 2 1 3 1 4 3
5 3 6 1 7 2 8 6 9 4
0 9 1 1 2 4 3 2 2 3
8 6 9 0 5 6 0 7 6 1
8 7 9 3 9 8 5 1 3 3
0 7 5 9 8 0 9 4 1 4
4 6 0 4 5 6 1 0 0 1
2 1 6 3 0 2 7 1 7 9
0 2 6 7 8 3 9 0 4 6
7 4 6 8 0 7 8 3 1 5



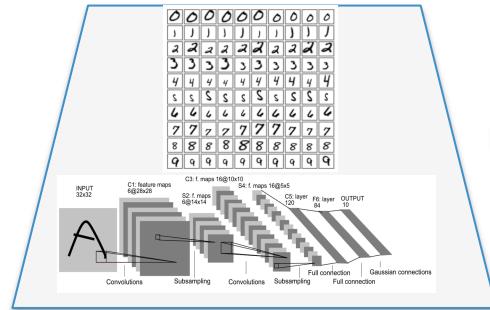
Error Rate: 0.4% ...

DL Researcher/Developer Considers ...

Input



DL MNIST App



Output

- ❖ 0
- ❖ [1,2,...,9]
- ❖ Not sure

File

APP

Artificial Intelligence

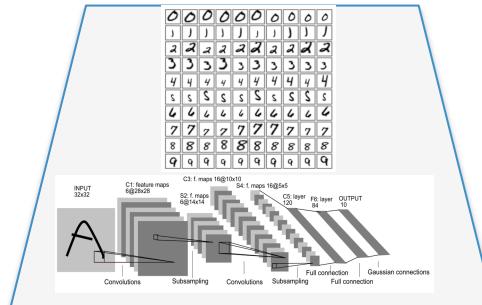
What Security Researchers Consider?

Input



File

DL MNIST App



APP

Output

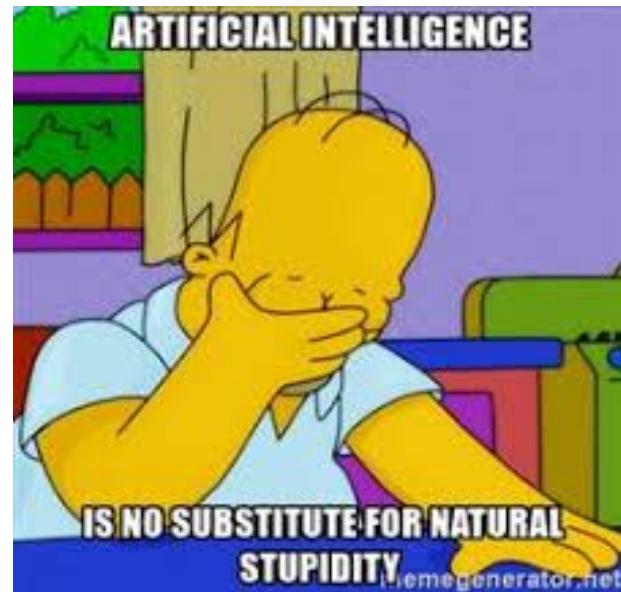
- ❖ 0
- ❖ [1,2,...,9]
- ❖ Not sure
- ❖ App hangs
- ❖ App gets owned

Artificial Intelligence

Natural Stupidity

Bugs Found in DL Frameworks and Dependencies

DL Framework	dep. packages	CVE-ID	Potential Threats
Tensorflow	numpy	CVE-2017-12852	DOS
Tensorflow	wave.py	CVE-2017-14144	DOS
Caffe	libjasper	CVE-2017-9782	heap overflow
Caffe	openEXR	CVE-2017-12596	crash
Caffe/Torch	opencv	CVE-2017-12597	heap overflow
Caffe/Torch	opencv	CVE-2017-12598	crash
Caffe/Torch	opencv	CVE-2017-12599	crash
Caffe/Torch	opencv	CVE-2017-12600	DOS
Caffe/Torch	opencv	CVE-2017-12601	crash
Caffe/Torch	opencv	CVE-2017-12602	DOS
Caffe/Torch	opencv	CVE-2017-12603	crash
Caffe/Torch	opencv	CVE-2017-12604	crash
Caffe/Torch	opencv	CVE-2017-12605	crash
Caffe/Torch	opencv	CVE-2017-12606	crash
Caffe/Torch	opencv	CVE-2017-14136	integer overflow



Joint effort with Qihoo360 Team Seri0us members: Qixue Xiao, Deyue Zhang

Security Risks Caused by DL Implementation Vulnerabilities

CVE 2017-12603: Heap Overflow

```
***** BMP decoder *****

bool BmpDecoder::readHeader()
{
    ...
    if( size >= 36 )
    {
        m_width = m_strm.getDWord();
        m_height = m_strm.getDWord();
        m_bpp = m_strm.getDWord() >> 16;
        m_rle_code = (BmpCompression)m_strm.getDWord();
        m_strm.skip(12);
        int clrused = m_strm.getDWord();
        m_strm.skip( size - 36 );

        if( m_width > 0 && m_height != 0 && ....
            (m_bpp == 8 && m_rle_code == BMP_RLE8) )
        {
            iscolor = true;
            result = true;

            if( m_bpp <= 8 )
            {
                memset(m_palette, 0, sizeof(m_palette));
                m_strm.getBytes(m_palette,
                    !clrused == 0? 1<*4 );
                iscolor = IsColorPalette( m_palette, m_bpp );
            }
            else if ...
        }
    }
}
```

grfmt_bmp.cpp

```
int RLByteStream::getBytes( void* buffer, int count )
{
    uchar* data = (uchar*)buffer;
    int readed = 0;
    assert( count >= 0 );

    while( count > 0 )
    {
        int l;
        for(;;)
        {
            l = (int)(m_end - m_current);
            if( l > count ) l = count;
            if( l > 0 ) break;
            readBlock();
        }
        memcpy( data, m_current, l );
        m_current += l;
        data += l;
        count -= l;
        readed += l;
    }
    return readed;
}
```

bit_strm.cpp

CVE 2017-12603: Heap Overflow

```
***** BMP decoder *****

bool BmpDecoder::readHeader()
{
    ...
    if( size >= 36 )
    {
        m_width = m_strm.getDWord();
        m_height = m_strm.getDWord();
        m_bpp = m_strm.getDWord() >> 16;
        m_rle_code = (BmpCompression)m_strm.getDWord();
        m_strm.skip(12);
        int clrused = m_strm.getDWord();
        m_strm.skip( size - 36 );

        if( m_width > 0 && m_height != 0 && ....
            (m_bpp == 8 && m_rle_code == BMP_RLE8) )
        {
            iscolor = true;
            result = true;

            if( m_bpp <= 8 )
            {
                memset(m_palette, 0, sizeof(m_palette));
                m_strm.getBytes(m_palette,
                    !clrused == 0? 1<*4 );
                iscolor = IsColorPalette( m_palette, m_bpp );
            }
            else if ...
        }
    }
}
```

Controlled by External Input



grfmt_bmp.cpp

```
int RLByteStream::getBytes( void* buffer, int count )
{
    uchar* data = (uchar*)buffer;
    int readed = 0;
    assert( count >= 0 );

    while( count > 0 )
    {
        int l;
        for(;;)
        {
            l = (int)(m_end - m_current);
            if( l > count ) l = count;
            if( l > 0 ) break;
            readBlock();
        }
        memcpy( data, m_current, l );
        m_current += l;
        data += l;
        count -= l;
        readed += l;
    }
    return readed;
}
```

bit_strm.cpp

CVE 2017-12603: Heap Overflow

```
***** BMP decoder *****  
bool BmpDecoder::readHeader()  
{  
    ...  
  
    if( size >= 36 )  
    {  
        m_width = m_strm.getDWord();  
        m_height = m_strm.getDWord();  
        m_bpp = m_strm.getDWord() >> 16;  
        m_rle_code = (BmpCompression)m_strm.getDWord();  
        m_strm.skip(12);  
        int clrused = m_strm.getDWord();  
        m_strm.skip( size - 36 );  
  
        if( m_width > 0 && m_height != 0 && .....  
            (m_bpp == 8 && m_rle_code == BMP_RLE8))  
        {  
            iscolor = true;  
            result = true;  
  
            if( m_bpp <= 8 )  
            {  
                memset(m_palette, 0, sizeof(m_palette));  
                m_strm.getBytes(m_palette,  
                    7clrused == 0? 1 < m_bpp : clrused)*4 );  
                iscolor = IsColorPalette( m_palette, m_bpp );  
            }  
            else if ...  
        }  
    }  
}
```

Controlled by
External Input

```
int RLByteStream::getBytes( void* buffer, int count )  
{  
    uchar* data = (uchar*)buffer;  
    int readed = 0;  
    assert( count >= 0 );  
  
    while( count > 0 )  
    {  
        int l;  
  
        for(;;)  
        {  
            l = (int)(m_end - m_current);  
            if( l > count ) l = count;  
            if( l > 0 ) break;  
            readBlock();  
        }  
        memcpy( data, m_current, l );  
        m_current += l;  
        data += l;  
        count -= l;  
        readed += l;  
    }  
    return readed;  
}
```

count > buffer length

Caffe ImageData Layer Implementation

Caffe

Deep learning framework
by BAIR

Created by
[Yangqing Jia](#)
Lead Developer
[Evan Shelhamer](#)

ImageData Layer

- Layer type: `ImageData`
- [Doxygen Documentation](#)
- Header: `./include/caffe/layers/image_data_layer.hpp`
- CPU implementation: `./src/caffe/layers/image_data_layer.cpp`

Parameters

<http://caffe.berkeleyvision.org/tutorial/layers/imagedata.html>



Caffe ImageData Layer Implementation

Caffe

Deep learning framework
by BAIR

Created by
[Yangqing Jia](#)
Lead Developer
[Evan Shelhamer](#)

ImageData Layer

- Layer type: `ImageData`
- [Doxygen Documentation](#)
- Header: `./include/caffe/layers/image_data_layer.hpp`
- CPU implementation: [`./src/caffe/layers/image_data_layer.cpp`](#)

Parameters



Caffe ImageData Layer Implementation

Caffe

Deep learning framework
by BAIR

Created by

Xiangang Li

A screenshot of a GitHub repository page for 'BVLC / caffe'. The repository has 2,087 stars. The 'Code' tab is selected, showing the file 'caffe/src/caffe/layers/image_data_layer.cpp'. A blue arrow points from the text 'CPU implementation: ./src/caffe/layers/image_data_layer.cpp' in the main content area to the file link in the code listing.

ImageData Layer

- Layer type: `ImageData`
- [Doxygen Documentation](#)
- Header: `./include/caffe/layers/image_data_layer.hpp`
- CPU implementation: `./src/caffe/layers/image_data_layer.cpp`

Caffe ImageData Layer Implementation

Caffe

Deep learning framework
by BAIR

Created by

Xiangang Li



ImageData Layer

- Layer type: `ImageData`
- [Doxygen Documentation](#)
- Header: `./include/caffe/layers/image_data_layer.hpp`
- CPU implementation: `./src/caffe/layers/image_data_layer.cpp`

A screenshot of the GitHub code editor for `image_data_layer.cpp`. The code includes #ifdef USE_OPENCV and #include <opencv2/core/core.hpp>. A blue arrow points from the 'Header' link in the 'ImageData Layer' section above to this code block.

```
1 #ifndef USE_OPENCV
2 #include <opencv2/core/core.hpp>
3
4 #include <fstream> // NOLINT(readabilitystreams)
5 #include <iostream> // NOLINT(readabilitystreams)
6 #include <string>
7 #include <utility>
8 #include <vector>
9
10 #include "caffe/data_transformer.hpp"
11 #include "caffe/layers/base_data_layer.hpp"
12 #include "caffe/layers/image_data_layer.hpp"
13 #include "caffe/util/benchmark.hpp"
14 #include "caffe/util/io.hpp"
15 #include "caffe/util/math_functions.hpp"
16 #include "caffe/util/rng.hpp"
17
18 namespace caffe {
```

```
1 #ifdef USE_OPENCV
2 #include <opencv2/core/core.hpp>
3
4 #include <fstream> // NOLINT(readabilitystreams)
5 #include <iostream> // NOLINT(readabilitystreams)
6 #include <string>
7 #include <utility>
8 #include <vector>
```

Example #1 DoS Attack Caffe CPPClassification

- Standard Example in Caffe Framework

Trained based on ImageNet data

Both Net and trained model available to download



```
./build/examples/cpp_classification/classification.bin \
models/bvlc_reference_caffenet/deploy.prototxt \
models/bvlc_reference_caffenet/bvlc_reference_caffenet.caffemodel \
data/ilsvrc12/imagenet_mean.binaryproto \
data/ilsvrc12/synset_words.txt \
examples/images/cat.jpg
```

Normal Output:

```
----- Prediction for examples/images/cat.jpg -----
0.3134 - "n02123045 tabby, tabby cat"
0.2380 - "n02123159 tiger cat"
0.1235 - "n02124075 Egyptian cat"
0.1003 - "n02119022 red fox, Vulpes vulpes"
0.0715 - "n02127052 lynx, catamount"
```

Example #1 DoS Attack Caffe CPPClassification

- Standard Example in Caffe Framework

Trained based on ImageNet data

Both Net and trained model available to download



```
./build/examples/cpp_classification/classification.bin \
models/bvlc_reference_caffenet/deploy.prototxt \
models/bvlc_reference_caffenet/bvlc_reference_caffenet.caffemodel \
data/ilsvrc12/imagenet_mean.binaryproto \
data/ilsvrc12/synset_words.txt \
examples/images/evil.bmp
```

Output:

```
----- Prediction for examples/images/evil.bmp -----
Segmentation fault (core dumped)
```

Example #2: NumPy Bug (TensorFlow)



- NumPy is a fundamental package for scientific computing with python
- TensorFlow Apps commonly use numpy for data representation and conversion.

Patch for CVE-2017-12852

```
--- a/numpy/lib/arraypad.py
+++ b/numpy/lib/arraypad.py
@@ -1406,7 +1406,10 @@ def pad(array, pad_width, mode, **kwargs):
     newmat = _append_min(newmat, pad_after, chunk_after, axis)

     elif mode == 'reflect':
-        for axis, (pad_before, pad_after) in enumerate(pad_width):
+        if narray.size == 0:
+            raise ValueError("There aren't any elements to reflect in 'array'!")
+
+        for axis, (pad_before, pad_after) in enumerate(pad_width):
            ...
            ...
            method = kwargs['reflect_type']
            safe_pad = newmat.shape[axis] - 1
            while ((pad_before > safe_pad) or (pad_after > safe_pad)):
```



Infinite Loop when input array is zero length (`safe_pad == -1`)



Example #2: NumPy Bug (TensorFlow)

```
audio-classify/audio-classification$ python result.py audio-min-samples/dogbark.wav
audio file: audio-min-samples/dogbark.wav
softmax output: [[ 9.82184019e-07 1.81138901e-07 2.68021075e-04 9.97506797e-01
 3.25933332e-04 4.26165315e-07 1.18322554e-03 4.01796569e-08
 2.90570169e-05 6.85345207e-04]]
The audio is dog_bark!
```

- Application: Urban Sound Classification (by Aaqib Saeed),
available at <https://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/>





Example #2: NumPy Bug (TensorFlow)

```
audio-classify/audio-classification$ python result.py /tmp/dogbark-mod.wav
audio file: /tmp/dogbark-mod.wav
^CException in thread Thread-1:
Traceback (most recent call last):
  File "/usr/lib/python3.4/threading.py", line 920, in _bootstrap_inner
    self.run()
  File "/usr/local/lib/python3.4/dist-packages/audioread/gstdec.py", line 149, in run
    self.loop.run()
  File "/usr/lib/python3/dist-packages/gi/overrides/GLib.py", line 526, in run
    raise KeyboardInterrupt
KeyboardInterrupt

^C^C^C^C^C
^Z
```

- Application: Urban Sound Classification (by Aaqib Saeed),
available at <https://aqibsaeed.github.io/2016-09-03-urban-sound-classification-part-1/>



Risks of Evasion Attacks and System Compromises



Example #4 Exploit Caffe CPPClassification

- Standard Example in Caffe Framework

Trained based on ImageNet data

Both Net and trained model available to download



```
./build/examples/cpp_classification/classification.bin \
models/bvlc_reference_caffenet/deploy.prototxt \
models/bvlc_reference_caffenet/bvlc_reference_caffenet.caffemodel \
data/ilsvrc12/imagenet_mean.binaryproto \
data/ilsvrc12/synset_words.txt \
examples/images/cat.jpg
```

Output:

```
----- Prediction for examples/images/cat.jpg -----
0.3134 - "n02123045 tabby, tabby cat"
0.2380 - "n02123159 tiger cat"
0.1235 - "n02124075 Egyptian cat"
0.1003 - "n02119022 red fox, Vulpes vulpes"
0.0715 - "n02127052 lynx, catamount"
```

Responses to the Earlier Bug in Data Layer

Patch for CVE-2017-12603

```
%diff --git a/modules/imgcodecs/src/grfmt_bmp.cpp b/modules/imgcodecs/src/grfmt_bmp.cpp
%index 86cad3..257f97c 100644
--- a/modules/imgcodecs/src/grfmt_bmp.cpp
+++ b/modules/imgcodecs/src/grfmt_bmp.cpp
@@ -118,8 +118,9 @@ bool BmpDecoder::readHeader()
    m_strm.skip(12);
    int clrused = m_strm.getDWord();
    ...
+
+   if ( m_bpp <= 8 )
+   {
+       memset( m_palette, 0, sizeof(m_palette));
+       m_strm.getBytes( m_palette, (clrused == 0? 1<<m_bpp : clrused)*4 );
+       CV_Assert(clrused < 256);
+       memset(m_palette, 0, sizeof(m_palette));
+       m_strm.getBytes(m_palette, (clrused == 0? 1<<m_bpp : clrused)*4 );
+       iscolor = IsColorPalette( m_palette, m_bpp );
+   }
+   else if ( m_bpp == 16 && m_rle_code == BMP_BITFIELDS )
```

grfmt_bmp.cpp

Live Demo of Compromising a Caffe DL application



Details of Caffe CPPClassification Exploitation

```
***** BMP decoder *****/
bool BmpDecoder::readHeader()
{
    ...
    if( size >= 36 )
    {
        m_width = m_strm.getDWord();
        m_height = m_strm.getDWord();
        m_bpp = m_strm.getDWord() >> 16;
        m_rle_code = (BmpCompression)m_strm.getDWord();
        m_strm.skip(12);
        int clrused = m_strm.getDWord();
        m_strm.skip( size - 36 );

        if( m_width > 0 && m_height != 0 && ....
            (m_bpp == 8 && m_rle_code == BMP_RLE8))
        {
            iscolor = true;
            result = true;

            if( m_bpp <= 8 )
            {
                CV_Assert(clrused <= 256);
                memset(m_palette, 0, sizeof(m_palette));
                m_strm.getBytes(m_palette,
                    (clrused == 0? 1<
```

grfmt_bmp.cpp

```
int RLByteStream::getBytes( void* buffer, int count )
{
    uchar* data = (uchar*)buffer;
    int readed = 0;
    assert( count >= 0 );

    while( count > 0 )
    {
        int l;
        for(;)
        {
            l = (int)(m_end - m_current);
            if( l > count ) l = count;
            if( l > 0 ) break;
            readBlock();
        }
        memcpy( data, m_current, l );
        m_current += l;
        data += l;
        count -= l;
        readed += l;
    }
    return readed;
}
```

bitstrm.cpp

Details of Caffe CPPClassification Exploitation

```
***** BMP decoder *****/
bool BmpDecoder::readHeader()
{
    ...
    if( size >= 36 )
    {
        m_width = m_strm.getDWord();
        m_height = m_strm.getDWord();
        m_bpp = m_strm.getDWord() >> 16;
        m_rle_code = (BmpCompression)m_strm.getDWord();
        m_strm.skip(12);
        int clrused = m_strm.getDWord(); ←
        m_strm.skip( size - 36 );
    }

    if( m_width > 0 && m_height != 0 && .....
        (m_bpp == 8 && m_rle_code == BMP_RLE8))
    {
        iscolor = true;
        result = true;

        if( m_bpp <= 8 )
        {
            CV_Assert(clrused <= 256); ←
            memset(m_palette, 0, sizeof(m_palette));
            m_strm.getBytes(m_palette,
                (clrused == 0? 1<
```

grfmt_bmp.cpp

1. Integer Overflow

```
int RLByteStream::getBytes( void* buffer, int count )
{
    uchar* data = (uchar*)buffer;
    int readed = 0;
    assert( count >= 0 );

    while( count > 0 )
    {
        int l;
        for(;)
        {
            l = (int)(m_end - m_current);
            if( l > count ) l = count;
            if( l > 0 ) break;
            readBlock(); ←
        }
        memcpy( data, m_current, l ); ←
        m_current += l;
        data += l;
        count -= l;
        readed += l;
    }
    return readed;
}
```

bitstrm.cpp

Details of Caffe CPPClassification Exploitation

```
***** BMP decoder *****/
bool BmpDecoder::readHeader()
{
    ...
    if( size >= 36 )
    {
        m_width = m_strm.getDWord();
        m_height = m_strm.getDWord();
        m_bpp = m_strm.getDWord() >> 16;
        m_rle_code = (BmpCompression)m_strm.getDWord();
        m_strm.skip(12);
        int clrused = m_strm.getDWord(); -----
        m_strm.skip( size - 36 );
    }

    if( m_width > 0 && m_height != 0 && .....
        (m_bpp == 8 && m_rle_code == BMP_RLE8))
    {
        iscolor = true;
        result = true;

        if( m_bpp <= 8 )
        {
            CV_Assert(clrused <= 256); -----
            memset(m_palette, 0, sizeof(m_palette));
            m_strm.getBytes(m_palette,
                (clrused == 0? 1<
```

grfmt_bmp.cpp

1. Integer Overflow

```
int RLByteStream::getBytes( void* buffer, int count )
{
    uchar* data = (uchar*)buffer;
    int readed = 0;
    assert( count >= 0 );

    while( count > 0 )
    {
        int l;
        for(;)
        {
            l = (int)(m_end - m_current);
            if( l > count ) l = count;
            if( l > 0 ) break;
            readBlock(); -----
        }
        memcpy( data, m_current, l ); -----
        m_current += l;
        data += l;
        count -= l;
        readed += l;
    }
    return readed;
}
```

bitstrm.cpp

2. Heap Overflow

Details of Caffe CPPClassification Exploitation

```
***** BMP decoder *****
```

```
bool BmpDecoder::readHeader()
```

```
{  
    ...  
}
```

```
if( size >= 36 )  
{  
    m_width = m_strm.getDWord();  
    m_height = m_strm.getDWord();  
    m_bpp = m_strm.getDWord() >> 16;  
    m_rle_code = (BmpCompression)m_strm.getDWord();  
    m_strm.skip(12);  
    int clrused = m_strm.getDWord();  
    m_strm.skip( size - 36 );
```

```
if( m_width > 0 && m_height != 0 && .....  
    (m_bpp == 8 && m_rle_code == BMP_RLE8))
```

```
{  
    iscolor = true;  
    result = true;
```

```
    if( m_bpp <= 8 )  
    {  
        CV_Assert(clrused <= 256);  
        memset(m_palette, 0, sizeof(m_palette));  
        m_strm.getBytes(m_palette,  
                        (clrused == 0? 1<  
                        m_bpp : clrused)*4 );  
        iscolor = IsColorPalette(m_palette, m_bpp );  
    }  
    else if ...
```

grfmt_bmp.cpp

1. Integer Overflow

bitstrm.cpp

2. Heap Overflow

```
int RLByteStream::getBytes( void* buffer, int count )
```

```
{  
    uchar* data = (uchar*)buffer;  
    int readed = 0;  
    assert( count >= 0 );
```

```
while( count > 0 )
```

```
{  
    int l;
```

```
for();
```

```
{  
    l = (int)(m_end - m_current);  
    if( l > count ) l = count;  
    if( l > 0 ) break;  
    readBlock();
```

```
}
```

```
memcpy( data, m_current, l );
```

```
m_current += l;  
data += l;  
count -= l;  
readed += l;
```

```
}
```

```
return readed;
```

3. Control Flow Hijack

Final Patch for CVE-2017-12603 ...



Screenshot of a GitHub repository page for opencv / opencv. The repository has 1,792 stars and 14,246 forks. The master branch is at 3.3.1. A commit by blendifin was made 9 days ago. The commit message is "Fix out of bounds write". The commit hash is 08a5fe3661b4cab8758e289927cfcd96c10458da. The diff shows one changed file, grfmt_bmp.cpp, with 1 addition and 1 deletion.

Fix out of bounds write

blendifin committed 9 days ago

1 parent 9ae86a9 commit 08a5fe3661b4cab8758e289927cfcd96c10458da

Showing 1 changed file with 1 addition and 1 deletion.

modules/imgcodecs/src/grfmt_bmp.cpp

```
@@ -118,7 +118,7 @@ bool BmpDecoder::readHeader()
118     118
119     119         if( m_bpp <= 8 )
120     120     {
121     -     CV_Assert(clrused <= 256);
121     +     CV_Assert(clrused >= 0 && clrused <= 256);
122     122         memset(m_palette, 0, sizeof(m_palette));
123     123         m_strm.getBytes(m_palette, (clrused == 0? 1<<m_bpp : clrused)*4 );
124     124         iscolor = IsColorPalette( m_palette, m_bpp );
```

Accessed on Oct 30,2017

Summary

- Deep learning frameworks heavily depend on 3rd party packages
- Complexity leads to Vulnerabilities:
 - We found 15+ vulnerabilities in popular DL platforms
 - The threats to DL apps include **DoS**, **Evasion**, **System compromise**
- This talk presents a **PoC** that demonstrates the danger of remote system compromise to cloud services running original deep learning applications.

Questions

kangli.ctf@gmail.com

