# 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
- Founding Mentor of *xCTF* and *Blue-lotus* Team
- 2016 DARPA Cyber Grand Challenge Finalist


disekt
CTF Team


# Success of AI



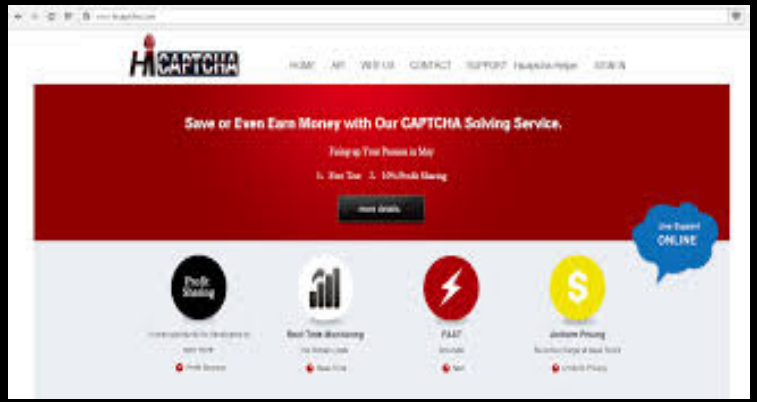
# Success of AI (on the dark side)


## 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)

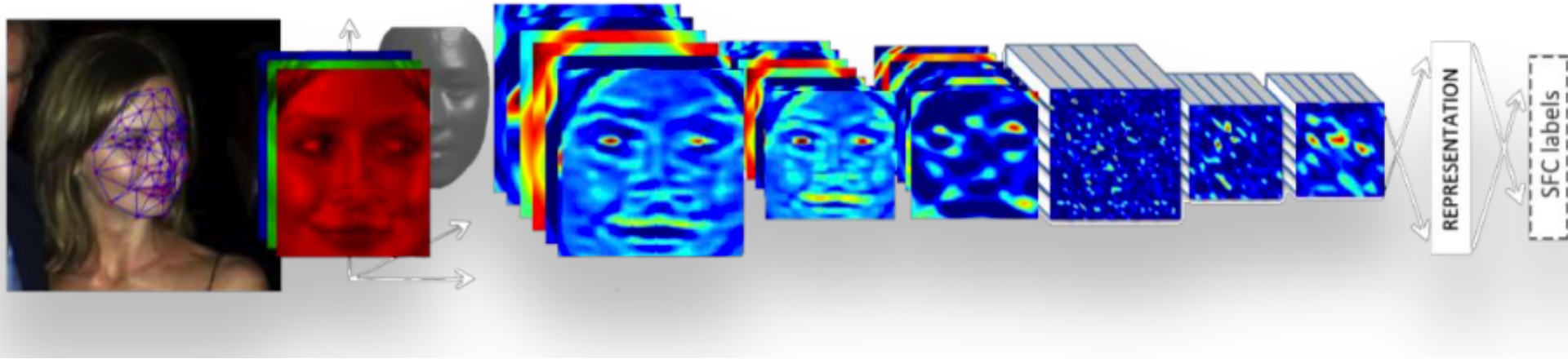




**Defeating Captcha with Learning**
https://deepmlblog.wordpress.com/2016/01/03/how-to-break-a-captcha-system/

**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

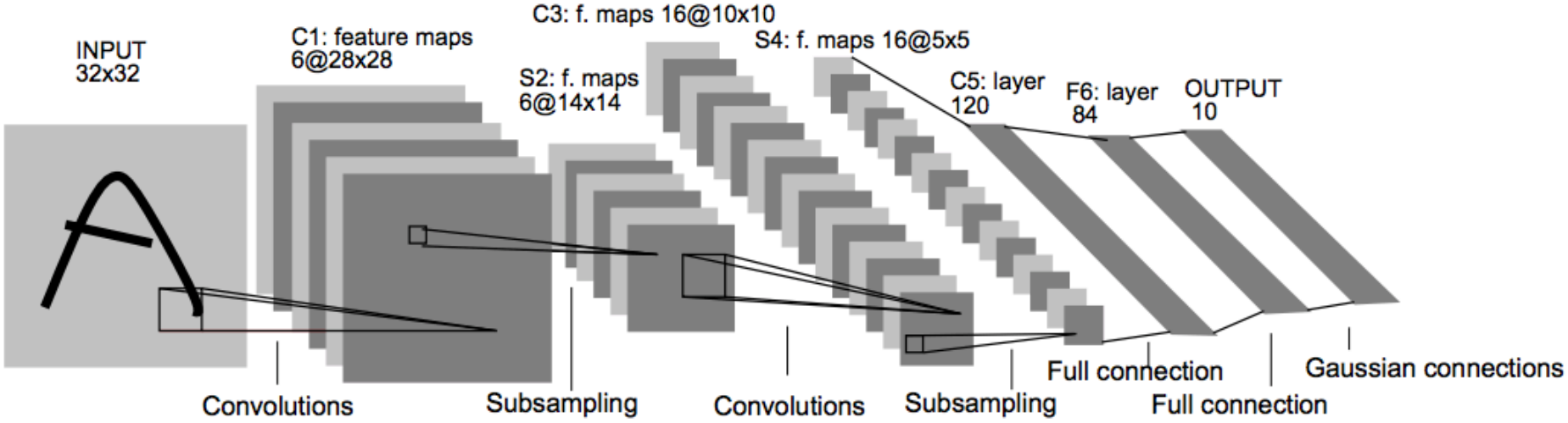



https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf

# The Implementation of Deep Learning Applications

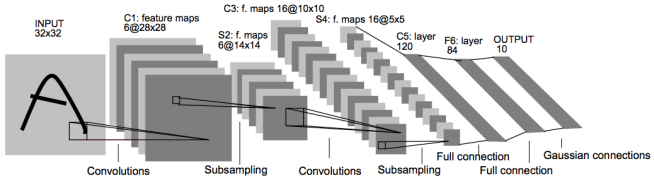
# MNIST Handwriting Digits Recognition

Sample Neural Network (LeNet-5) Architecture




http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

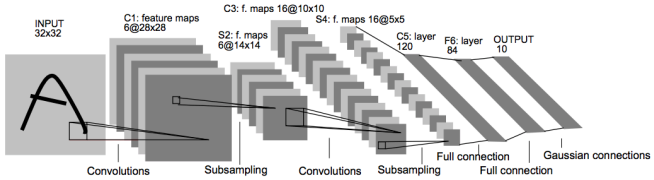
# MNIST Handwriting Digits Recognition




Solver

Net

# MNIST Handwriting Digits Recognition




**Solver**

**Net**

```
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='xavie
r'))
    n.pool2 = L.Pooling(n.conv2, kernel_size=2, stride=2, pool=P.Pooling.MAX)
    n.fc1 =   L.InnerProduct(n.pool2, num_output=500, weight_filler=dict(type='xavier'))
    n.relu1 = L.ReLU(n.fc1, 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

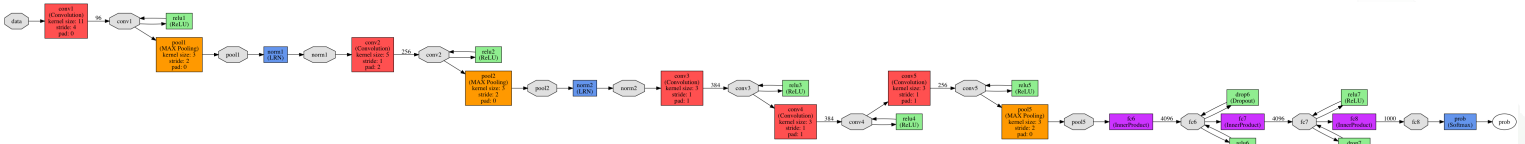
# Deep Learning Frameworks



Many words about these Frameworks:
**Functionality, Flexibility, Scientific Computing** …

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

# Standalone DL Application (Caffe Example)



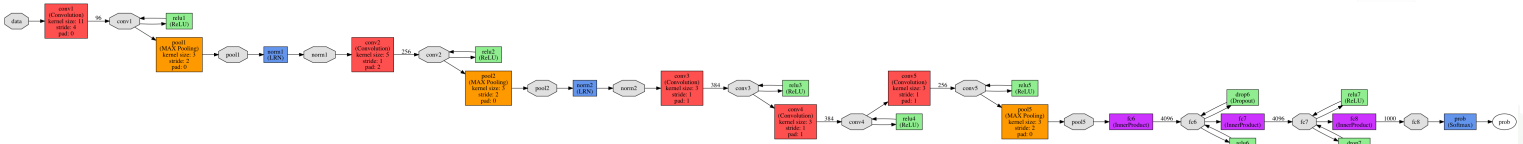
https://github.com/BVLC/caffe/tree/master/examples/cpp_classification

```
./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)



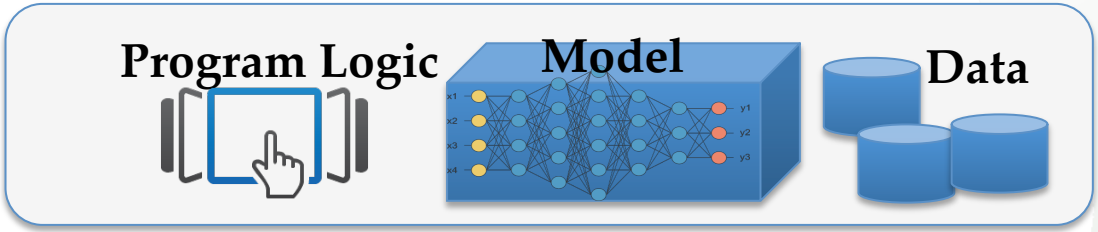
~/caffe/build/examples/cpp_classification$ **ldd classification** | more

```
linux-vdso.so.1 =>  (0x00007fffdf9fe000)
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)
libIlmImf.so.6 => /usr/lib/x86_64-linux-gnu/libIlmImf.so.6 (0x00007f3c80f2f000)
  ……
```

137 lib*.so

# Software Layers in Deep Learning Apps

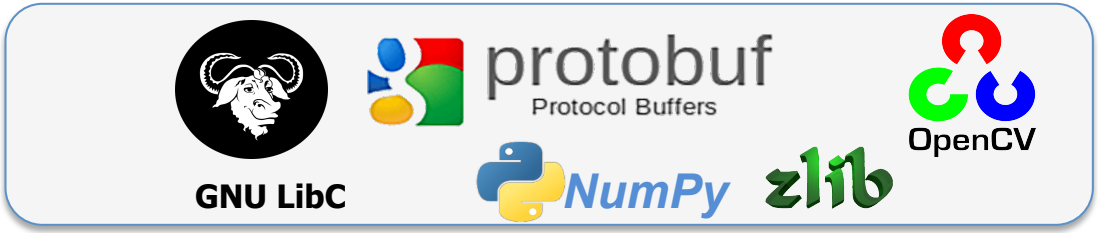
DL Applications




DL Frameworks

Torch theano TensorFlow Caffe

Framework Dependencies

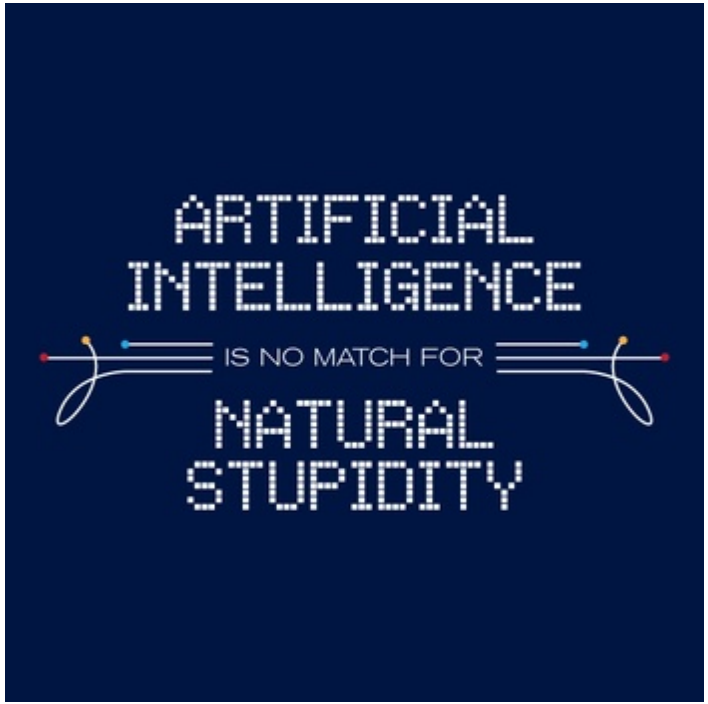


# DL Framework Complexity and Dependencies

| 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

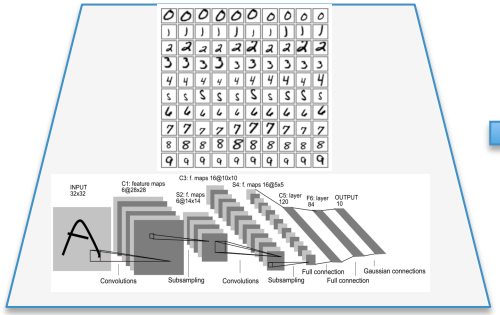
# Why Implementation Security Matters?




* 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?

**Input** **DL MNIST App** **Output**

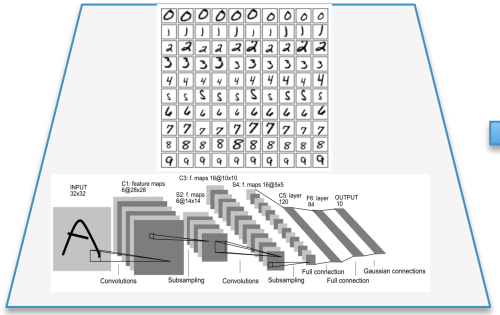




File APP Result

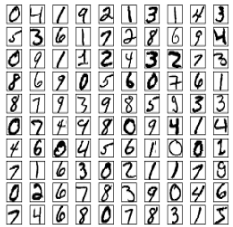
# DL Researcher/Developer Considers …

**Input** | **DL MNIST App** | **Output**





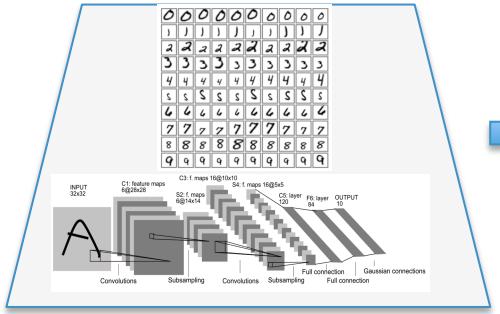
File | APP | Result



Error Rate: 0.4% …

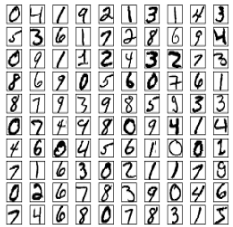
# DL Researcher/Developer Considers …

**Input** | **DL MNIST App** | **Output**



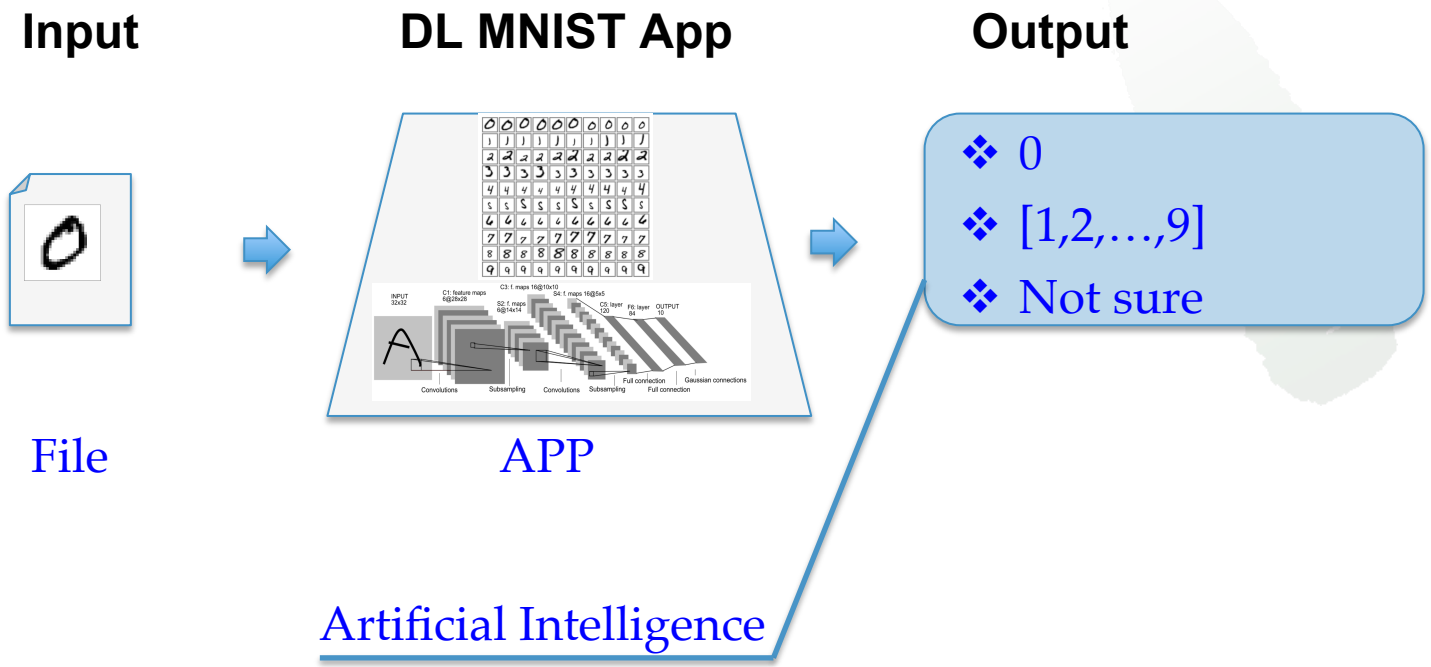

What are Possible Outcomes?

File | APP | Result

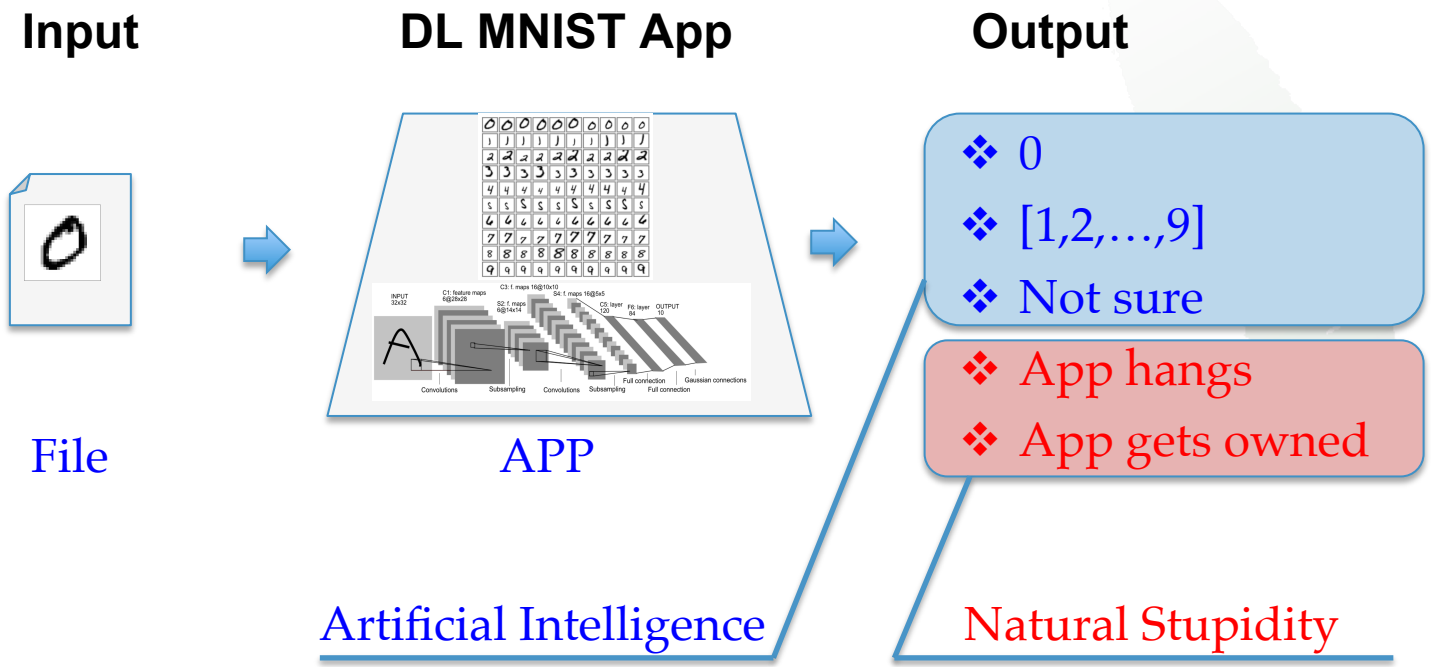


Error Rate: 0.4% …

# DL Researcher/Developer Considers …

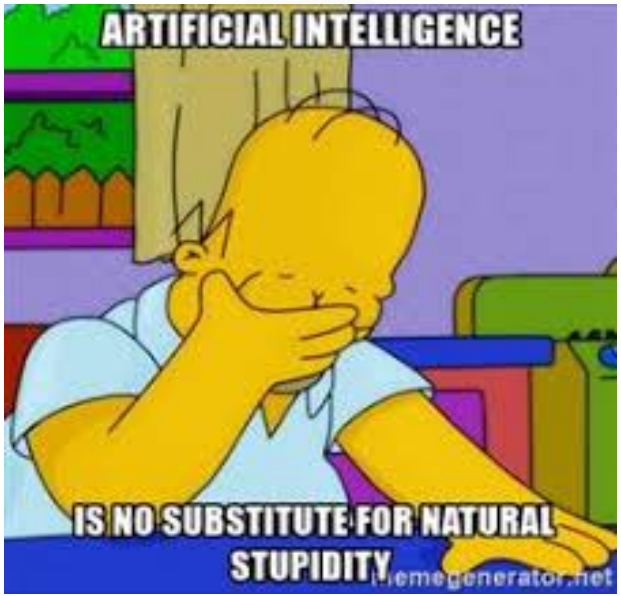


# What Security Researchers Consider?



# 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<<m_bpp : clrused)*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<<m_bpp : clrused)*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,
                     (clrused == 0? 1<<m_bpp : clrused)*4 );
                  iscolor = IsColorPalette( m_palette, m_bpp );
              }
              else if  …
```

**Controlled by External Input**

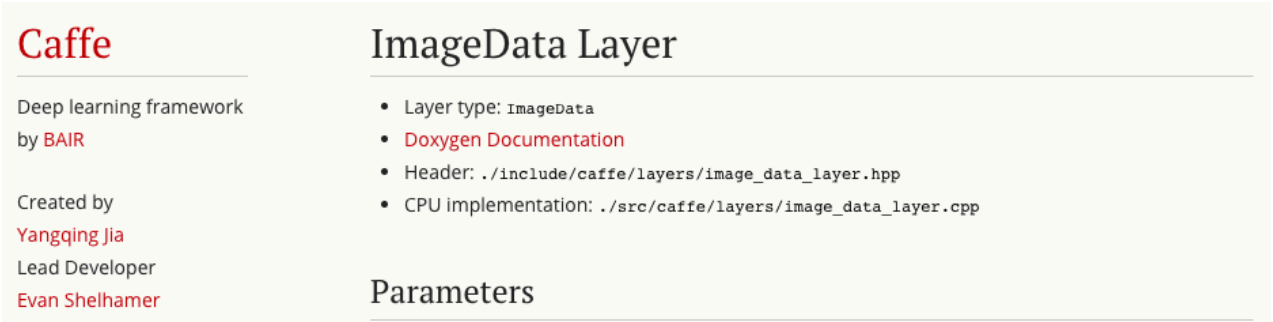
**grfmt_bmp.cpp**

```
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;
}
```

**count > buffer length**

**bit_strm.cpp**

# 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
Yangqing Jia

## ImageData Layer

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

Features Business Explore Marketplace Pricing This repository Search

BVLC / caffe Watch 2,087

Code Issues 585 Pull requests 259 Projects 0 Wiki Insights

Branch: master caffe / src / caffe / layers / image_data_layer.cpp

cypof Switched multi-GPU to NCCL

16 contributors

180 lines (161 sloc) 6.78 KB

```
#ifdef USE_OPENCV
#include <opencv2/core/core.hpp>

#include <fstream>  // NOLINT(readability/streams)
#include <iostream>  // NOLINT(readability/streams)
#include <string>
#include <utility>
#include <vector>

#include "caffe/data_transformer.hpp"
#include "caffe/layers/base_data_layer.hpp"
#include "caffe/layers/image_data_layer.hpp"
#include "caffe/util/benchmark.hpp"
#include "caffe/util/io.hpp"
#include "caffe/util/math_functions.hpp"
#include "caffe/util/rng.hpp"

namespace caffe {
```

# Caffe ImageData Layer Implementation

## Caffe

Deep learning framework
by BAIR

Created by
Yangqing Jia

## ImageData Layer

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

Features Business Explore Marketplace Pricing This repository Search

BVLC / caffe Watch 2,087

Code Issues 585 Pull requests 259 Projects 0 Wiki Insights

Branch: master caffe / src / caffe / layers / image_data_layer.cpp

cypof Switched multi-GPU to NCCL

16 contributors

180 lines (161 sloc) 6.78 KB

```
#ifdef USE_OPENCV
#include <opencv2/core/core.hpp>

#include <fstream>  // NOLINT(readability/streams)
#include <iostream>  // NOLINT(readability/streams)
#include <string>
#include <utility>
#include <vector>

#include "caffe/data_transformer.hpp"
#include "caffe/layers/base_data_layer.hpp"
#include "caffe/layers/image_data_layer.hpp"
#include "caffe/util/benchmark.hpp"
#include "caffe/util/io.hpp"
#include "caffe/util/math_functions.hpp"
#include "caffe/util/rng.hpp"

namespace caffe {
```

```
1  #ifdef USE_OPENCV
2  #include <opencv2/core/core.hpp>
3
4  #include <fstream>  // NOLINT(readability/stre
5  #include <iostream>  // NOLINT(readability/str
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 threat 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”, line149, 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/

Screenshot: courtesy from Qixue Xiao @ Qihoo 360 Team Seri0s

```
top - 19:12:39 up 30 days,  6:33,  1 user,  load average: 4.86, 4.30, 3.74
Tasks: 401 total,   4 running, 397 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.8 us, 12.0 sy,  0.0 ni, 83.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:  13200669+total, 12699846+used,  5008228 free,   432120 buffers
KiB Swap: 67092476 total,    94612 used, 66997864 free. 14962400 cac

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+
25875 xiaoqix+  20   0  0.109t 0.102t  15964 R 100.0 83.2   2:18.36
```

Caffe

torch

```
450
451         jpeg_start_decompress( cinfo );
452
453         buffer = (*cinfo->mem->alloc_sarray)((j_common_ptr)cinfo,
454                                              JPOOL_IMAGE, m_width*4, 1 );
455
456         uchar* data = img.ptr();
457         for( ; m_height--; data += step )
458         {
459             jpeg_read_scanlines( cinfo, buffer, 1 );
460             if( color )
461             {
462                 if( cinfo->out_color_components == 3 )
463                     icvCvt_RGB2BGR_8u_C3R( buffer[0], 0, data, 0, cvSize(m_width,1) );
464                 else
465                     icvCvt_CMYK2BGR_8u_C4C3R( buffer[0], 0, data, 0, cvSize(m_width,1) );
466             }
467             else
468             {
469                 if( cinfo->out_color_components == 1 )
470                     memcpy( data, buffer[0], m_width );
471                 else
472                     icvCvt_CMYK2Gray_8u_C4C1R( buffer[0], 0, data, 0, cvSize(m_width,1) );
473             }
474         }
475
476
477         jpeg_finish_decompress( cinfo );
478     }
479 }
```

CVE-2017-12602

# Example #3: DoS caused by Memory Exhaustion

# 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

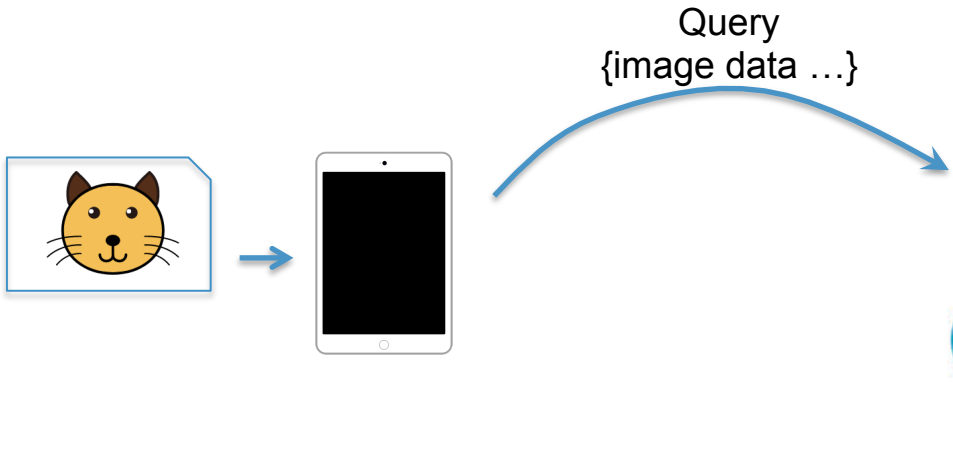
## Patch for CVE-2017-12603

```
%diff --git a/modules/imgcodecs/src/grfmt_bmp.cpp b/modules/imgcodecs/src/grfmt_bmp.cpp
%index 86cacd3..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

# Demo Setup

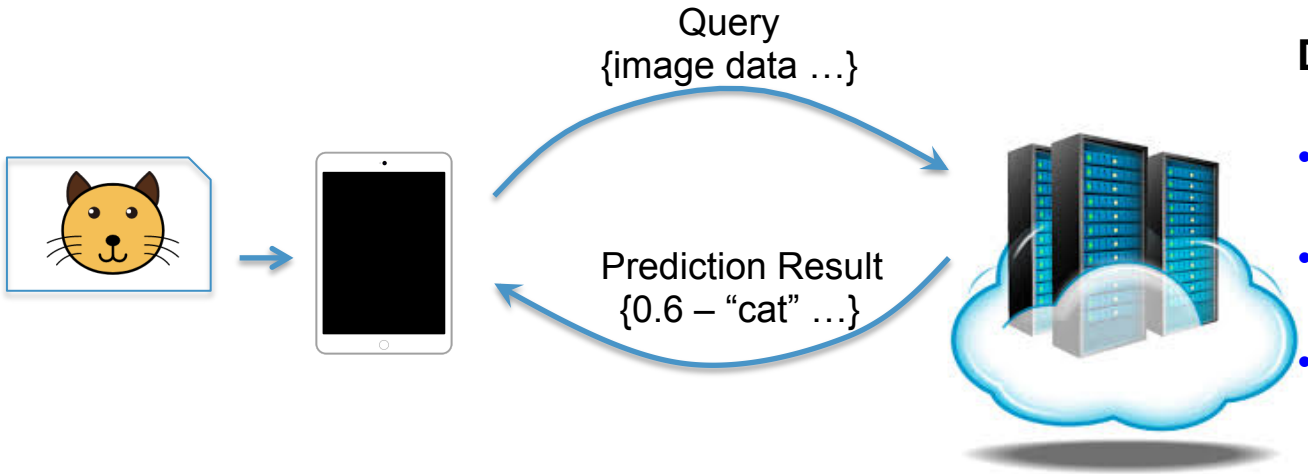



**DL Software on Cloud**

- **Caffe**
- **CPPClassification**
- **Model: BAIR/BVLC CaffeNet Model**

All software packages are the latest versions from github, pulled on Oct 25, 2017

# Demo Setup

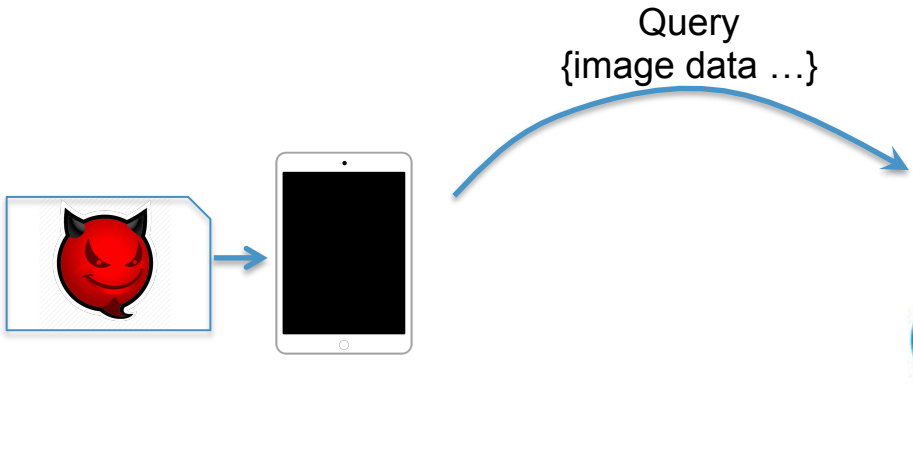



**DL Software on Cloud**

- **Caffe**
- **CPPClassification**
- **Model: BAIR/BVLC CaffeNet Model**

All software packages are the latest versions from github, pulled on Oct 25, 2017

# Demo Setup

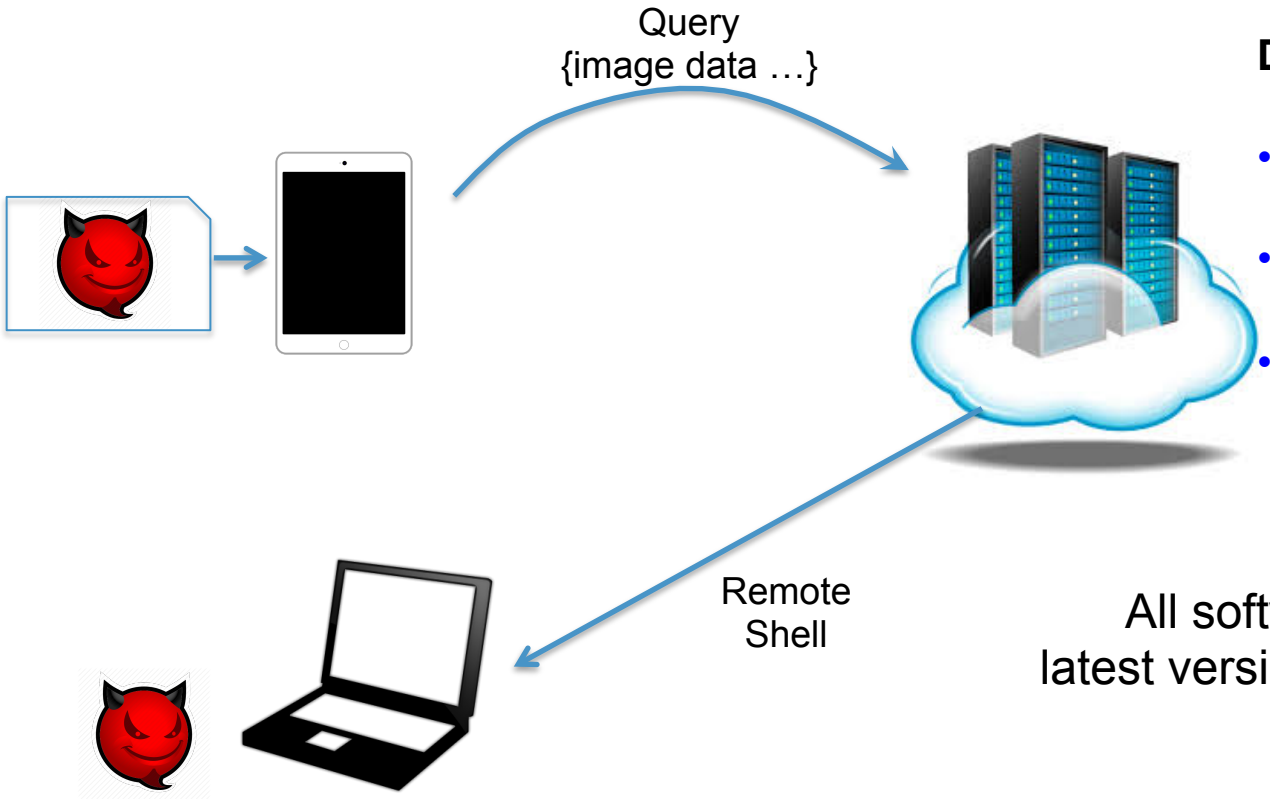



**DL Software on Cloud**

- **Caffe**
- **CPPClassification**
- **Model: BAIR/BVLC CaffeNet Model**

All software packages are the latest versions from github, pulled on Oct 25, 2017

# Demo Setup




**DL Software on Cloud**

- **Caffe**
- **CPPClassification**
- **Model: BAIR/BVLC CaffeNet Model**

All software packages are the latest versions from github, pulled on Oct 25, 2017

# DL Image Classifier Server Setup

```
# Steps to Build Caffe Deep Learning Image Classifier
# Instruction: http://caffe.berkeleyvision.org/installation.html
# Dependencies: OpenCV, OpenBlas

# OpenCV (latest stable version as of Sep 28, 2017)
# https://github.com/opencv/opencv/archive/2.4.13.4.zip
cmake -DCMAKE_BUILD_TYPE=RelWithDebInfo ..
sudo make install

# Openblas
git clone https://github.com/xianyi/OpenBLAS.git
make; sudo install

# Caffe
git clone https://github.com/BVLC/caffe.git
sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler
sudo apt-get install --no-install-recommends libboost-all-dev
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
make all
```

# DL Image Classifier Model and Cmdline

# Model Info
Name: BAIR/BVLC CaffeNet Model
Caffemodel: bvlc_reference_caffenet.caffemodel
Caffemodel_url: http://dl.caffe.berkeleyvision.org/bvlc_reference_caffenet.caffemodel
Caffe_commit: 709dc15af4a06bebda027c1eb2b3f3e3375d5077

# Sample cmd line of using the caffe image classifier with
~/Caffe_classification/caffe$**./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 test.jpg

# Additional Demo Info

# Web Service based on Django
pip install -U Django (1.10, 1.11 has been tested)
python manage.py runserver 0.0.0.0:8000

# reverse-binding shell
https://www.exploit-db.com/exploits/39185/

# Live Demo of Compromising a Caffe DL application

# Details of Caffe CPPClassification Exploitation

**grfmt_bmp.cpp**

```
/****************** 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  …
```

**bitstrm.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;
}
```

# Details of Caffe CPPClassification Exploitation

**grfmt_bmp.cpp**

```
/****************** 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  …
```

1. Integer Overflow

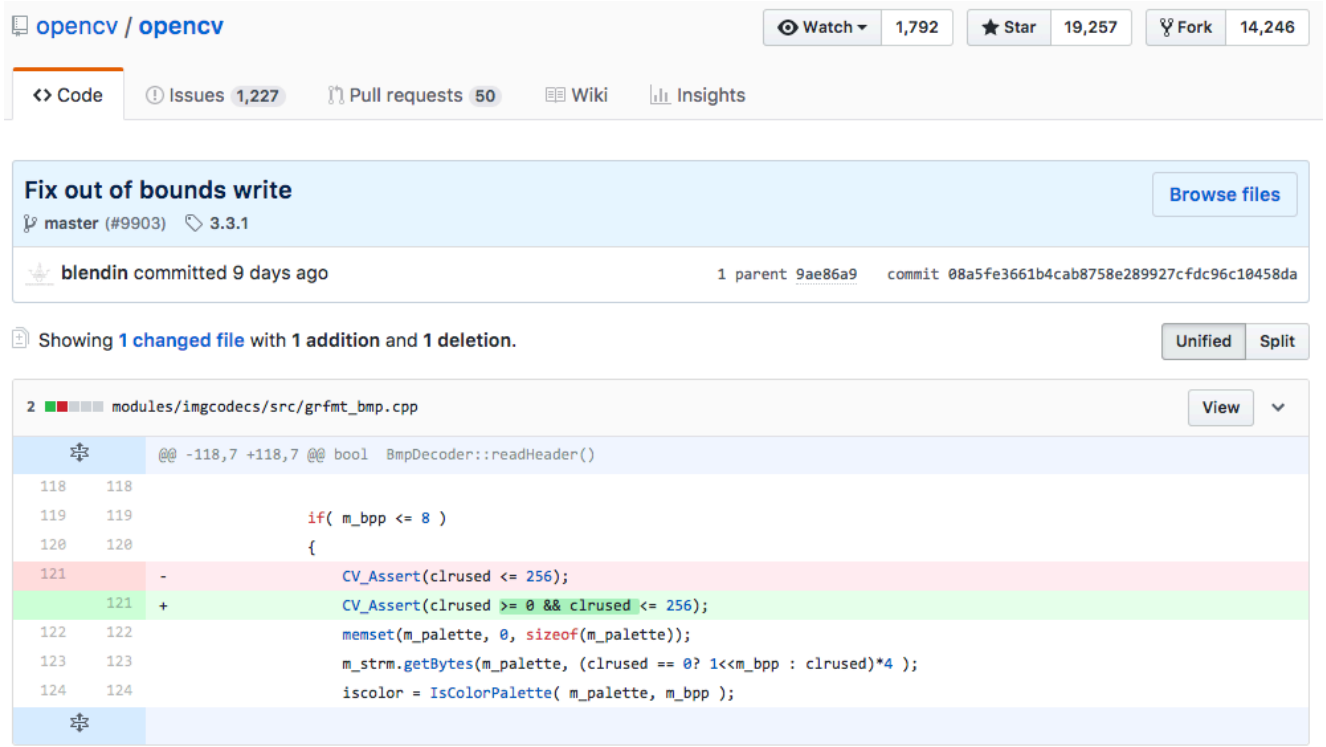
3. Control Flow Hijack

**bitstrm.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;
}
```

2. Heap Overflow

# Final Patch for CVE-2017-12603 …



opencv / opencv

Watch 1,792 | Star 19,257 | Fork 14,246

Code | Issues 1,227 | Pull requests 50 | Wiki | Insights

**Fix out of bounds write**

master (#9903) 3.3.1

Browse files

blendin committed 9 days ago — 1 parent 9ae86a9 — commit 08a5fe3661b4cab8758e289927cfdc96c10458da

Showing 1 changed file with 1 addition and 1 deletion.

Unified | Split

2 modules/imgcodecs/src/grfmt_bmp.cpp — View

```
@@ -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