SANS The most trusted source for cybersecurity training, certifications, degrees, and research

Pen Test HackFest & Cyber Ranges Summit
Live Online

Nidhi Rastogi

Research Scientist, Rensselaer Polytechnic Institute, Troy, NY

iamnidhirastogi

nidhirastogi

Summit: Jun 4-5 | Training: Jun 8-13

sans.org/hackfest

Automated detection of software vulnerabilities using Deep learning

#SANSHackFest

What is this talk about

- Find Software vulnerabilities using Deep Learning
  - *both for profit and pleasure!*
- Run through the end-2-end process
- Access to github code
  - try it out for yourself

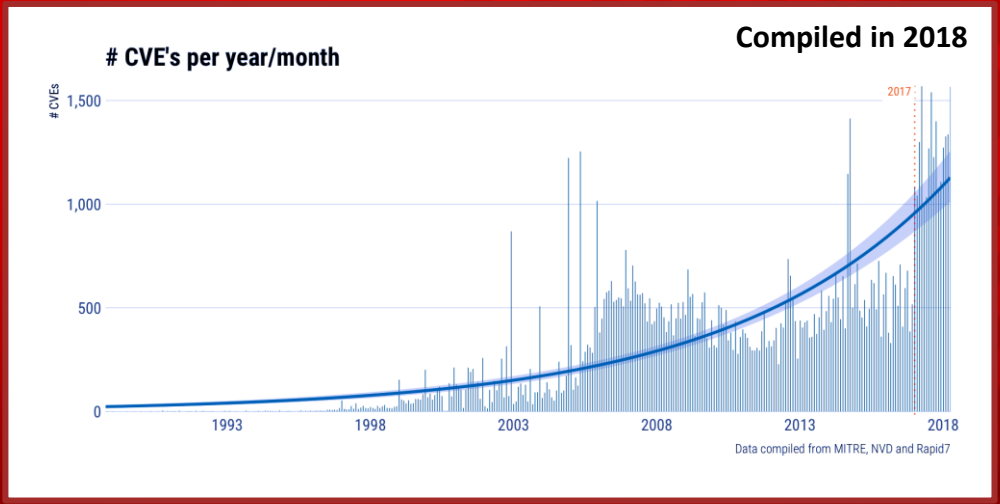
Software Vulnerability Detection

- Vulnerabilities pose serious risks of exploit
  - System compromise
  - Information leaks
  - DoS

Software Vulnerability Detection

Increasing number of software vulnerabilities

- CVE count increasing every year



#SANSHackFest

Machine Learning approaches

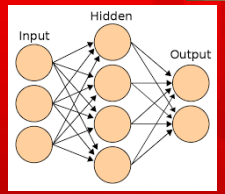
(+)detect vulnerabilities using patterns learned from analyst-defined feature representations of vulnerabilities

(-)Security experts have to define vulnerability features

- function length, nesting depth, string entropy, n-grams and suffix trees, etc.

Machine Learning approaches

Cat Classifier



ML Model

70% Cats

30% Others

Features

- *Ears*
- *Color*
- *Shape*
- *Eye shape*
- *Eye color*

Machine Learning approaches

(+)detect vulnerabilities using patterns learned from analyst-defined feature representations of vulnerabilities

(-)Security experts have to define vulnerability features

- Human labor intensive
- High False negative rates

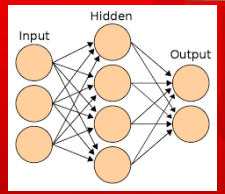
Deep Learning based approaches

- Automatic Discovery of Features
  - Not deciding what features should be selected for the deep learning model
- Alleviate human expert involvement

Deep Learning Approaches

Cat Classifier

~~Features~~
- ~~Ears~~
- ~~Color~~
- ~~Shape~~
- ~~Eye shape~~
- ~~Eye color~~



NN Model

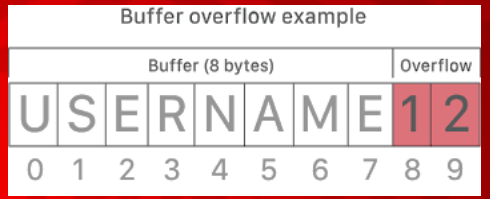
70% Cats

30% Others

Goal : Automatically detect software vulnerabilities using Deep Learning

Specifically…

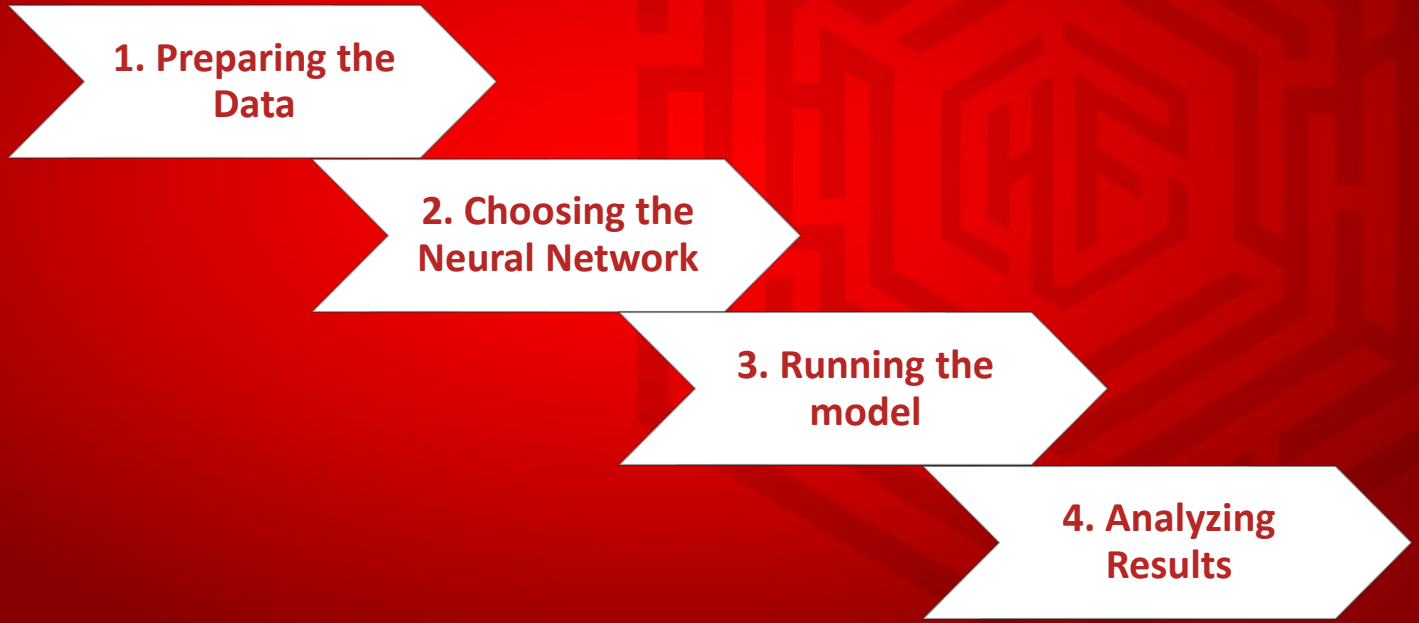
- Detect buffer overflows
- resource management errors



- Deep Learning Model (BLSTM) based on VulDeePecker*

*https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-2_Li_paper.pdf

DL Modeling entails…



Data Preparation…

1. Preparing software programs for DL model
   - Identifying level of granularity of the code or the model
   - Location can be identified in the program

Data Preparation…

1. Preparing software programs for DL model
   - Identifying level of granularity of the code or the model
   - Location can be identified in the program
2. Convert Programs to Code Gadget

   lines of code semantically related to each other

   Can be vectorized as i/p to NN model.

Neural Network Model

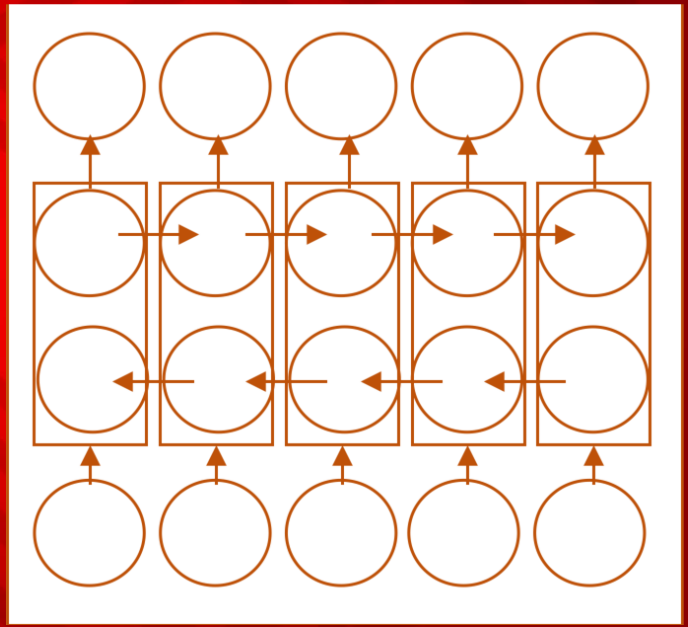
Bidirectional LSTM Model, because…

- Supervised Model
- used where the learning problem is sequential
- you feed the learning algorithm with the original data
  - once from beginning to the end AND
  - once from end to beginning
- Training: Randomly choose 80% of programs
- Testing : Rest of the 20% programs

bilstm rnn model

bilstm has two networks:

1. Past information in forward direction
2. Future in the reverse direction



Running bilstm rnn model

1. Input -> Code gadgets
2. Model bilstm
3. Output -> 1(vulnerable) / 0 (not vulnerable)

Analyzing Results

1. Confusion Matrix - TP, FP, TN, FN
2. Accuracy, Precision, Recall
3. RoC, AUC
4. …

		Predicted / Classified		
		- (Negative)	+ (Positive)	
True	- (Negative)	True Negative (TN) 9,000	False Positive (FP) 700	Overall True Negative: 9,700
	+ (Positive)	False Negative (FN) 200	True Positive (TP) 100	Overall True Positive: 300
		Overall Predicted Negative: 9,100	Overall Predicted Positive: 900	

$$Precision = \frac{TP}{TP+FP} = \frac{100}{100+700} = 0.125$$

$$Accuracy = \frac{True}{True+False} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{100+9,000}{100+9,000+700+200} = \frac{9,100}{10,000} = 0.91$$

$$Recall(TruePositiveRate) = \frac{TP}{TP+FN} = \frac{100}{100+200} \approx 0.333$$

$$F1 = 2 * \frac{Recall*Precision}{Recall+Precision} = 2 * \frac{0.333*0.125}{0.333+0.125} \approx 0.182$$

Experiments

Preparing the data (input)

2 datasets are available(extracted from NVD)

- Buffer Error Vulnerability (CWE-119)
  - 520 Open source software programs
  - 8,122 test cases
- Resource Management Vulnerability (CWE-399)
  - 320 Open Source Software Programs
    - 1,729 test cases

Prepare the Environment (macOS)

1. pip install pandas gensim keras tensorflow sklearn
2. Install git
3. Run the following command in terminal:

git clone https://github.com/nidhirastogi/Deep-Learning-Based-System-for -Automatic-Detection-of-Software-Vulnerabilities.git

4. Extract the 2 Datasets : cwe119 and cwe399

Generating code gadgets

```
1 CVE-2010-1444/vlc_media_player_1.1.0_CVE-2010-1444_zipstream.c cfunc 449
ZIP_FILENAME_LEN, NULL, 0, NULL, 0 )
char *psz_fileName = calloc( ZIP_FILENAME_LEN, 1 );
if( unzGetCurrentFileInfo( file, p_fileInfo, psz_fileName,
vlc_array_append( p_filenames, strdup( psz_fileName ) );
free( psz_fileName );
0
---------------------------------
2 CVE-2010-1444/vlc_media_player_1.1.0_CVE-2010-1444_zipstream.c cppfunc 449
char *psz_fileName = calloc( ZIP_FILENAME_LEN, 1 );
ZIP_FILENAME_LEN, NULL, 0, NULL, 0 )
if( unzGetCurrentFileInfo( file, p_fileInfo, psz_fileName,
vlc_array_append( p_filenames, strdup( psz_fileName ) );
free( psz_fileName );
0
---------------------------------
3 CVE-2011-2896/cups_1.4.2_CVE-2011-2896_image-gif.c inputfunc 100
fread(buf, 13, 1, fp);
img->xsize = (buf[7] << 8) | buf[6];
img->ysize = (buf[9] << 8) | buf[8];
ncolors    = 2 << (buf[10] & 0x07);
if (buf[10] & GIF_COLORMAP)
if (gif_read_cmap(fp, ncolors, cmap, &gray))
switch (getc(fp))
fclose(fp);
buf[0] = getc(fp);
if (buf[0] == 0xf9)
gif_get_block(fp, buf);
fread(buf, 9, 1, fp);
if (buf[8] & GIF_COLORMAP)
ncolors = 2 << (buf[8] & 0x07);
if (gif_read_cmap(fp, ncolors, cmap, &gray))
img->xsize = (buf[5] << 8) | buf[4];
img->ysize = (buf[7] << 8) | buf[6];
if (img->xsize == 0 || img->ysize == 0)
img->xsize, img->ysize);
fprintf(stderr, "DEBUG: Bad GIF image dimensions: %dx%d\n",
fclose(fp);
i = gif_read_image(fp, img, cmap, buf[8] & GIF_INTERLACE);
int interlace);
i = gif_read_image(fp, img, cmap, buf[8] & GIF_INTERLACE);
static int      gif_read_cmap(FILE *fp, int ncolors, gif_cmap_t cmap,
fclose(fp);
```

some_gadgets - Notepad

File Edit Format View Help

```
4 CVE-2013-1706/Firefox_22.0b6_CVE_2013_1706_toolkit_components_maintenanceservice_workmonitor.cpp cppfunc 111
WCHAR installDir[MAX_PATH + 1] = {L'\0'};
if (!GetInstallationDir(argc, argv, installDir)) {
GetInstallationDir(int argcTmp, LPWSTR *argvTmp, WCHAR aResultDir[MAX_PATH + 1])
wcsncpy(aResultDir, argvTmp[2], MAX_PATH);
WCHAR* backSlash = wcsrchr(aResultDir, L'\\');
0
---------------------------------
5 CVE-2013-1732/Firefox_20.0.1_CVE_2013_1732_layout_generic_nsBlockFrame.cpp cfunc 196
DumpStyleGeneaology(nsIFrame* aFrame, const char* gap)
nsFrame::ListTag(stdout, aFrame);
nsStyleContext* sc = aFrame->GetStyleContext();
printf("%p ", sc);
psc = sc->GetParent();
sc = psc;
printf("%p ", sc);
0
---------------------------------
```

Training the model

1. python vuldeepecker_train.py /dataset/cwe119.txt

```
Using TensorFlow backend.
Found 2697 forward slices and 37056 backward slices

Training model...
Processing gadgets... 39753
WARNING:tensorflow:From C:\Users\ETsukerman\AppData\Local\Programs\Python\Python37\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is
recated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\ETsukerman\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
WARNING:tensorflow:From C:\Users\ETsukerman\AppData\Local\Programs\Python\Python37\lib\site-packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will
be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/4
2019-05-09 12:09:31.071435: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
16704/16704 [==============================] - 54s 3ms/step - loss: 0.6001 - acc: 0.6640
Epoch 2/4
16704/16704 [==============================] - 54s 3ms/step - loss: 0.5315 - acc: 0.7296
Epoch 3/4
16704/16704 [==============================] - 55s 3ms/step - loss: 0.5083 - acc: 0.7466
Epoch 4/4
16704/16704 [==============================] - 56s 3ms/step - loss: 0.4849 - acc: 0.7597
4176/4176 [==============================] - 5s 1ms/step
Accuracy is... 0.7708333333333334
False positive rate is... 0.3309386973180077
False negative rate is... 0.12739463601532566
True positive rate is... 0.8726053639846744
Precision is... 0.7250298448070036
F1-score is... 0.7920017387524451
```

Model Prediction

- python vuldeepecker_predict.py cwe119 cwe119_cgd_gadget_vectors.pkl

```
gadgets.txt cwe119_cgd_model.h5
Using TensorFlow backend.
Found 0 forward slices and 2 backward slices

Training model...
Processing gadgets... 2
WARNING:tensorflow:From C:\Users\ETsukerman\AppData\Local\Programs\Python\Python37\lib\site-packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framework.ops) is dep
recated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Users\ETsukerman\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
2019-05-09 12:21:57.663411: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
[[1. 0.]
 [1. 0.]]
```

Run implementation video

Thank You! Questions?