Classifying Malicious Websites With Image Recognition Models

Akbar Qureshi
About Me

• Over 15 years of experience in cyber security
• Provided security advisory and technical services to both private and government sectors
• Interests – Cyber Counter Intelligence, Offensive Security, Artificial Intelligence, ICS \ SCADA Security, Cyber Network Defense
Agenda

• Keep it simple
• Try and not bore you
• Not impress you with scientific formulas
• Keep the presentation interactive
Malicious Websites – They Come In All Shapes & Flavors

- Phishing
- Botnet C2 Panels
- Drive-by-Downloads
- Malware Distribution
- Watering Hole Attacks
- Exploit Kits
Legit Today Malicious Tomorrow

- Legitimate websites are attacked and compromised on a daily basis
- Unsuspecting users visit these trusted “legit” sites and get infected with malware
Who is Getting Compromised?

• Both “big” and “small” name company websites are compromised
  Example - Online payment systems of British Airways, Feedify, and Newegg
  compromised by the Magecart threat group

• According to Trustwave 100% of web applications remain vulnerable to attack –
  2018 Global Security Report
Time To Fine Tune the Risk Equation

New Way  Risk = Assume we are already compromised x Now what
Life of a Cyber Warrior – With All The Maliciousness

- Continuous barrage of alerts
- Overworked
- Alert fatigue

Insufficient Incident Analysis and Reporting

YOU HAVE BEEN HACKED!

Did you update the ticket?
How Do We Optimize Analysis?

In most cases we are:

- Hiring skilled staff
- Buying expensive security tools
- Increasing the cyber security budget and again buying more tools

What we should focus on:

- Effective use of cutting-edge technology, especially the ones that can outperform humans
- In our case we are using image recognition to classify & label malicious websites
How Does Image Recognition Help Us?

• Malicious website classification can be automated with the power of deep learning
• Incident response analysis can be optimized
• Malicious website threat label data can be enhanced
  Example - badsite.com is categorized as botnet – that’s good but which botnet?
What is Deep learning?

- Deep learning is a subset of machine learning.
- Algorithms in deep learning mimic the structure and function of the human brain as artificial neural networks (ANN).
  
  Example – *Convolutional neural networks (CNN)* can classify images similar to how the brain identifies objects.

- Deep learning algorithms automatically learn and extract relevant features from raw data.

Automatic Feature Engineering
Automatic Feature Engineering Makes Life Easy

Traditional Machine Learning
• In traditional machine learning the programmer has to manually select the features from the raw data
• In other words, the programmer has to be very specific in telling the “computer” what to look for when deciding if the image contains a car or an airplane
• Manual feature extraction is time consuming, difficult and requires expert knowledge

Deep learning
• No need to manually extract features
Why is Deep Learning Gaining Popularity?

• We now have access to large amounts of high-quality labelled data
• Complex deep learning models can be trained in less time using GPUs (graphics processing unit)
• Fairly easy to implement using open source frameworks such as Tensorflow, Caffe, Keras, PyTorch
• Deep learning breakthroughs are driving Artificial Intelligence boom
Applications of Deep Learning

• Autonomous driving (e.g. detecting traffic lights, stop signs)
• Robots (e.g. vision)
• Aerospace and Defense (e.g. identification of objects from satellites)
• Video Surveillance (e.g. facial recognition, behavior recognition)
• Image Recognition
• Natural Language Processing (NLP)
• Speech Recognition
The Robot Apocalypse – Autonomous Weaponry

• The manufacturer of the famous “AK-47” rifle Kalashnikov has developed a range of products based on artificial neural networks
• Kalashnikov’s fully automated “combat module” has the capacity to make decisions and identify targets on its own
• The “combat module” analyzes image data to identify targets

Deep Neural Networks (DNN) – Why Deep?

DNN is simply an artificial neural network with **multiple** hidden layers
Deep Neural Networks - Image Recognition

Hidden Layer 1
Algorithm learns to first recognize pixels and then edges and shapes

Hidden Layer 2
Learns to identify more complex features and shapes such as headlights and grill

Hidden Layer 3
Learns which shapes and objects can be used to identify a car

How objects are recognized in a hierarchical structure
Image Recognition – The Human Way

Let's see how we classify this image
Image Recognition – The Human Way

You see a dog, broken plate, tissue paper = Brain tells you “Yup the dog did it”
Deep Learning – The Brain’s Visual Assessment

The point is:

• We are constantly evaluating things in the world around us
• Subconsciously making thousands of predictions everyday
• Recognizing patterns and labelling objects based on what we have learned or experienced in the past
Training machines for image classification
Convolutional Neural Networks (CNN or Convnets)

• CNNs have proven to be extremely effective in performing image recognition and classification tasks

• They perform feature identification and classification of images, text, sound, and video

• CNN takes an input image, processes it and outputs a class or a probability of classes that best describes the image (e.g. car, truck, plane)
Convolutional Neural Networks – Why So Popular?

• Automatic feature Engineering
• State-of-the-art performance in image recognition tasks e.g. accurately identifying objects such as cars in pictures
• Pre-existing networks can be retrained for new recognition tasks
Convolutional Neural Networks – Layers

Three main types of Layers

**Convolution Layer** - Main task is to extract features from the input image

**Sub-sampling (Pooling) Layer** - Reduces the amount of parameters and computation in the network

**Fully-Connected (Output) Layer** - Classifies the input image
You Must Train Them Right

Accuracy and performance of the classifier depends on the quality of the training data
Training the classifier for malicious website detection
Tasks

My tasks were simple:

• Use image recognition algorithms to classify malicious websites
• Enrich malicious website threat label data with more than just phishing, suspicious or malicious
Training A Neural Network From Scratch

Requires:

• Processing power (multiple graphics processing units)
• Large amounts of labelled data (millions of images)
• Lots of time (several days or weeks)
Solution - Transfer Learning

- Transfer Learning is the process of applying a trained model to a different, but related task.
- Transfer learning is typically faster, since we only retrain the last layer (fully connected layer) of an existing model.
- Retraining requires less resources and data.
- Transfer learning has become the go-to method in many real world use cases, such as cancer detection.
Transfer Learning - Retraining Inception v3 CNN Model

• Inception-v3 is a convolutional neural network that is trained on more than a million images from the ImageNet database

• The model achieves state-of-the-art accuracy for recognizing general objects with 1000 classes, such as "Laptop", "Cheeseburger", "Beer" etc.

• The model extracts general features from input images and classifies them based on those features

My Goal = Retrain Inception-v3 to classify images of malicious websites
Inception v3 Architecture
Transfer Learning With TensorFlow

• TensorFlow is an opensource software library for high performance numerical computation
• Originally developed by researchers and engineers from the Google Brain team
• Strong support for machine learning and deep learning
• Flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs)
• URL: https://www.tensorflow.org
My TensorFlow Setup

classify.py does the following:

- Takes a screenshot of the website using Selenium WebDriver
- Saves the screenshot as "out.png"
- Predicts whether "out.png" is a phishing or botnet site

The “retrain.py” script in TensorFlow re trains the image classifier

- This script loads the pre-trained Inception v3 model
- Removes the old final layer and trains a new one on the downloaded images

The command `aq@u1:~/TF$ ls` shows the files:

- classify.py
- inception
- retrain.py
- tf_files
- training_dataset
- train.sh

- Contains training data
  - E.g. Images of phishing sites etc.
Training Data

• Used opensource data to download images of phishing and botnet websites

• Moved downloaded images to the appropriate sub folders in the `training_dataset` directory

• Sub folder names define what label is applied to each image during the inception retraining process
Retraining Inception v3

• Once “retrain.py” script is executed with all the parameters, it starts the retraining process

• The script downloads the previously trained Inception-v3 model (if not already installed)

• Looks for images in the “training_dataset” folder

• Proceeds to generate bottleneck files
Retraining Inception v3 – cont’d

• Once the bottleneck files are generated for all the training data, the script begins training of the new top layer
• ‘Bottleneck’ is an informal term used for the layer just before the final output layer that actually does the classification
Retraining Inception v3 – cont’d

In each step three values are calculated:

**Training Accuracy** - Percentage of training images that were classified correctly

**Cross Entropy** - A Loss function which tells us how well the learning process is progressing

**Validation Accuracy** - True measure of how well the model is performing

```
2018-08-24 13:46:29.904036: Step 10: Train accuracy = 91.0%
2018-08-24 13:46:29.904143: Step 10: Cross entropy = 0.925858
2018-08-24 13:46:30.025306: Step 10: Validation accuracy = 82.0% (N=100)
```
Retraining Inception v3 – cont’d

• In each step 10 images are randomly chosen from the training set
• Bottlenecks are fed into the final classification layer to perform predictions
• Predictions are checked for accuracy by comparing them to the actual labels
• Depending on the accuracy of the predictions, the final layer's weights are updated through the back-propagation process
Retraining Inception v3 – cont’d

Once the training process is complete, the script outputs the final test accuracy for the model

Final test accuracy = 94.5% (N=55)
Time To Predict
Select your email provider

Now, you can sign in to Dropbox with your email.
Questions?
Thank you