LECTUTRE 2 MACHINE LEARNING

Win+w

Overall System Goal

- Goal: Detect and classify network attacks from real internet traffic

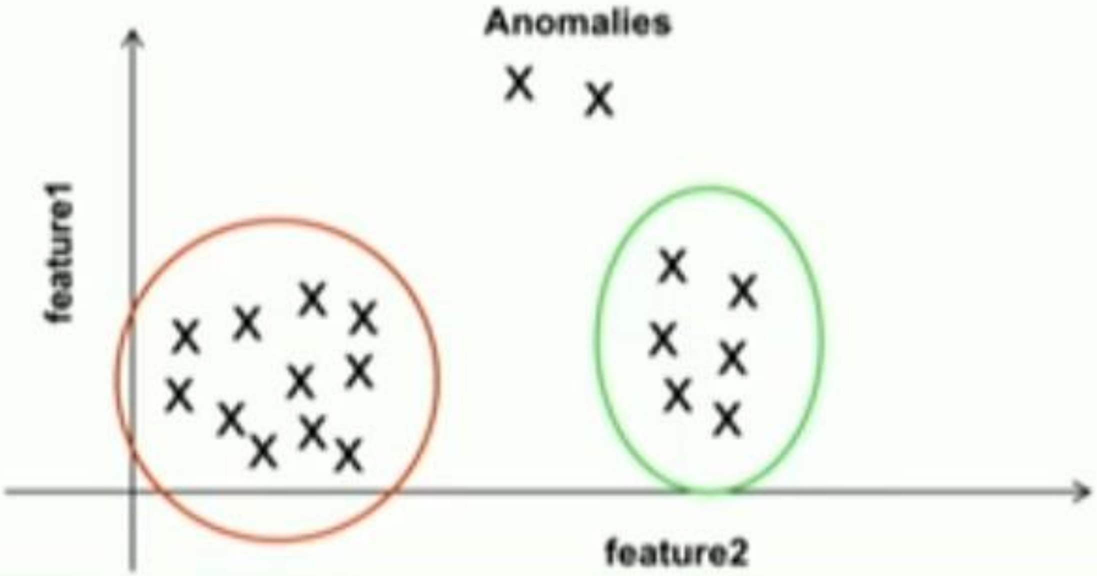
- Dataset:
  - Measurement and Analysis of Wide-area Internet (MAWI) working group
    - Day-in-the-Life of the Internet 2015 (http://mawi.wide.ad.jp/mawi/ditl/ditl2015)
    - Day-in-the-Life of the Internet 2017 (http://mawi.wide.ad.jp/mawi/ditl/ditl2017)
    - 2x48=96 hours of 1 Gigabit packet capture (PCAP) headers collected in Tokyo
    - 0.7 TB compressed; 20 TB in analysts friendly form

      Normalized, sorted, indexed, and read optimized
    - IP addressed deterministically anonymized *within* each collect

      Network analysis is still valid

Anomaly Detection

- “An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism” *
  - Outlier is sometimes referred to as anomaly, surprise, exception, …
  - Within the context of cyber networks, these “mechanisms” can be botnets, C&C servers, insider threats or other attacks such as DDOS & Port Scan attacks, …



*D. M. Hawkins.Identification of outliers. Chapman and Hall, London, 1980

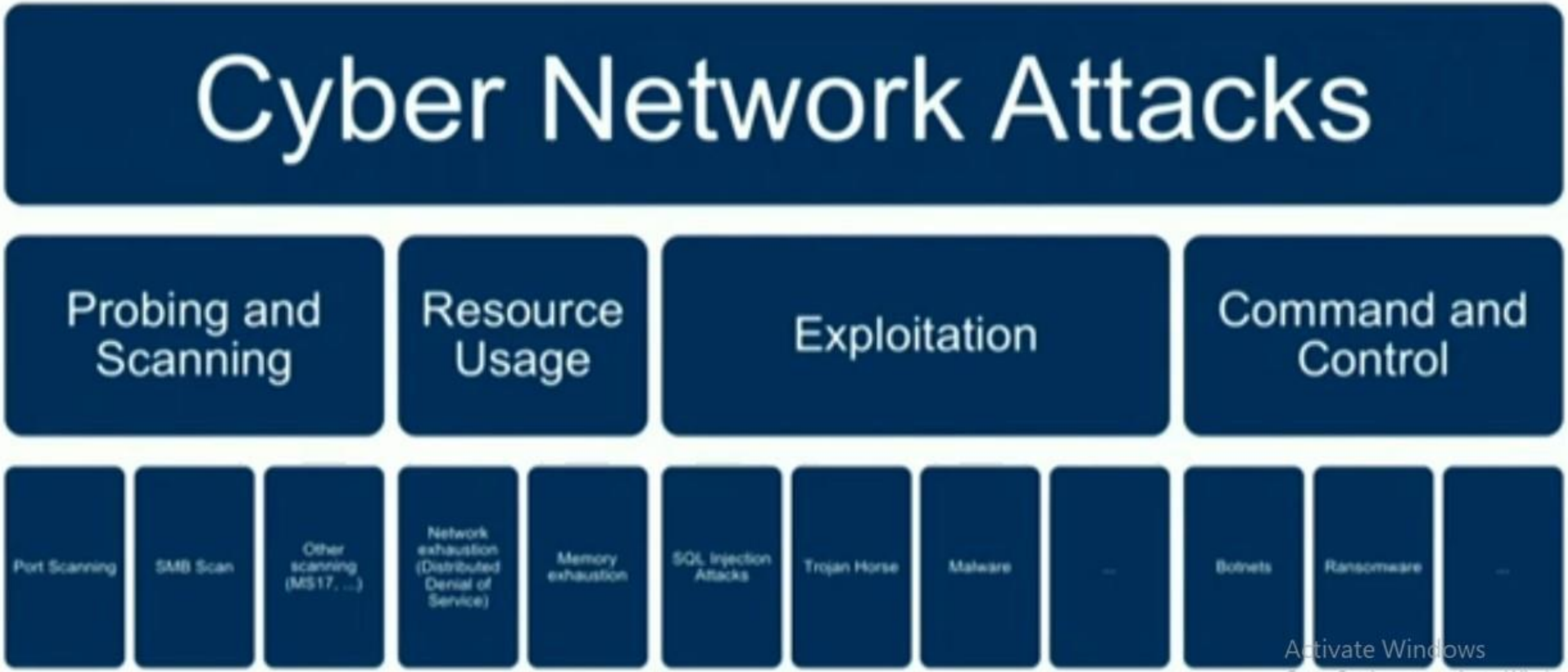
Anomaly Detection

- General techniques for outlier detection (with exemplar technique):
  - Statistics: Look for changes in patterns/distributions (e.g., dimensional analysis)
  - Clustering: cluster input data based on a set of features (e.g., k-means)
  - Distance-based: Look for observations that are very far from other observations (e.g., k-nearest neighbor)
  - Model-based techniques such as ANNs: Come up with a background model and look for deviations from the expected (e.g., replicator neural network)

Given the complexity of network traffic, we use a model based technique

Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 15.

Anomalies in Cyber Networks



Cyber Network Attacks

- Probing and Scanning
  - Port Scanning
  - SMB Scan
  - Other scanning (MS17, ...)
- Resource Usage
  - Network exhaustion (Distributed Denial of Service)
  - Memory exhaustion
- Exploitation
  - SQL Injection Attacks
  - Trojan Horse
  - Malware
  - ...
- Command and Control
  - Botnets
  - Ransomware
  - ...

Example: Probing & Scanning

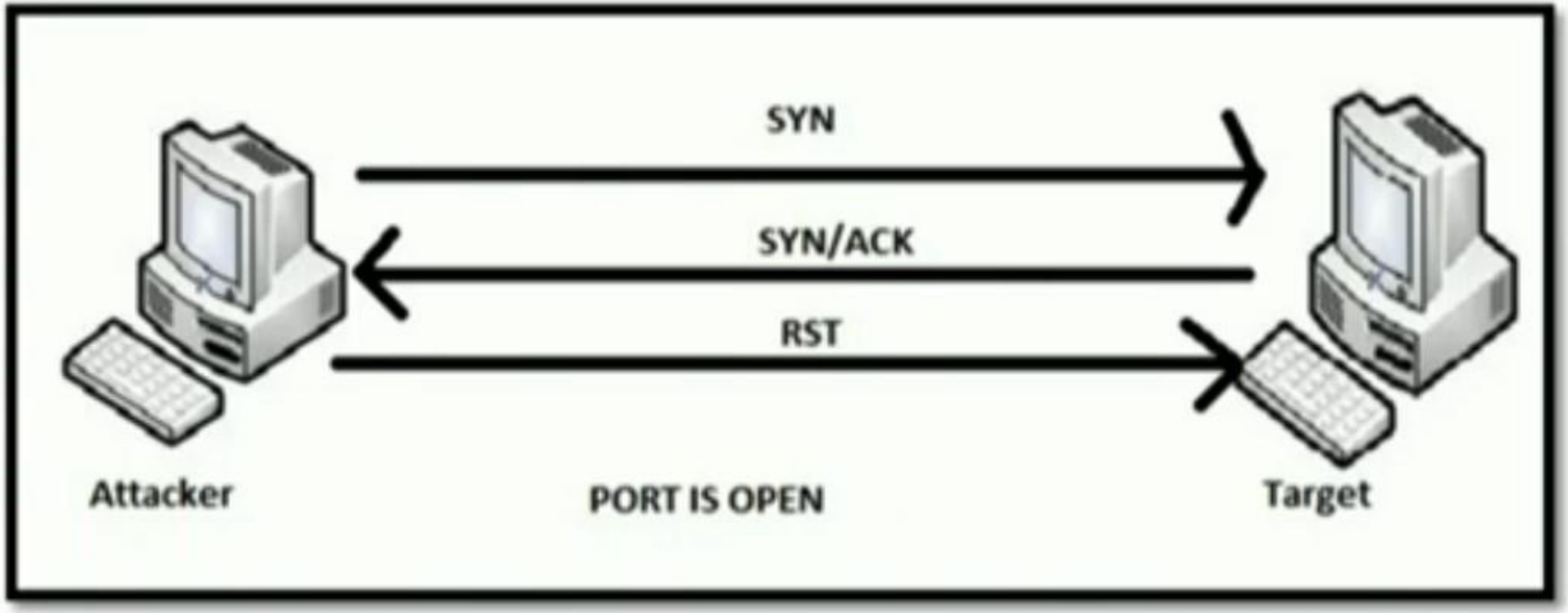
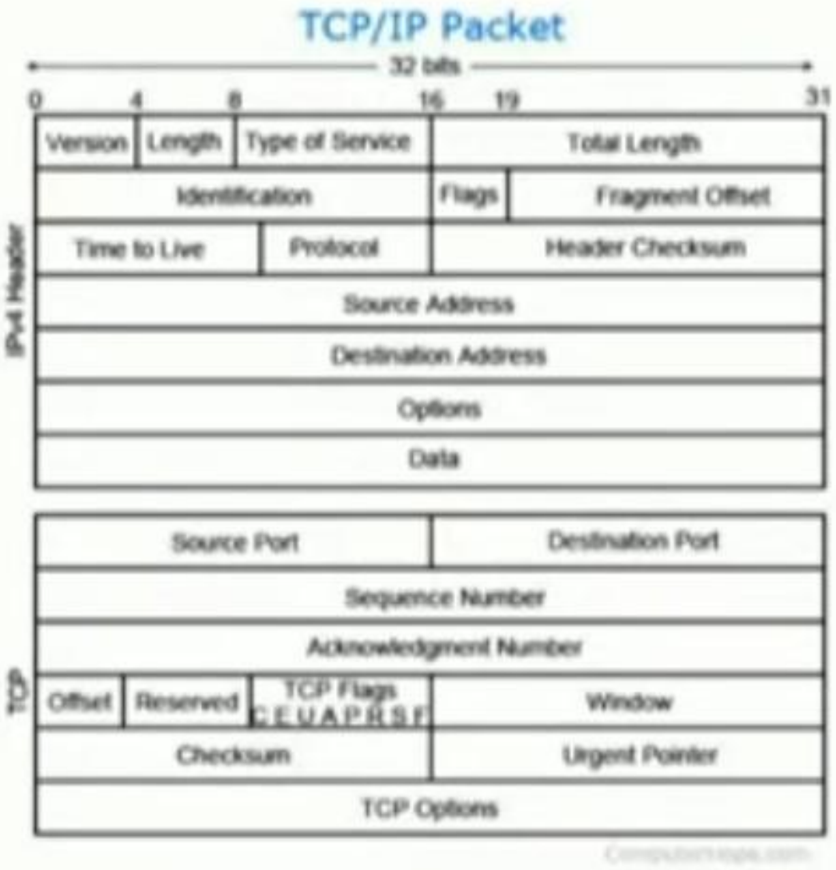


Figure Source: https://resources.infosecinstitute.com/port-scanners/

The Network Packet



Headers converted to human readable form

```
(1.201704132329.pcap..A.mat,,frame.time_relative|0.000000000)
1 (1.201704132329.pcap..A.mat,,frame.time|2017 Apr 13
10:29:59.938632000 EDT) 1
(1.201704132329.pcap..A.mat,,ip.dst|17.114.183.195) 1
(1.201704132329.pcap..A.mat,,ip.len|52) 1
(1.201704132329.pcap..A.mat,,ip.proto|6) 1
(1.201704132329.pcap..A.mat,,ip.src|163.35.157.212) 1
(1.201704132329.pcap..A.mat,,tcp.dstport|80) 1
(1.201704132329.pcap..A.mat,,tcp.flags|0x00000010) 1
(1.201704132329.pcap..A.mat,,tcp.srcport|47438) 1
```

Network packets are relatively easy to collect and often form the lowest common denominator across cyber network processing pipelines

Figure Source:
https://www.computerhope.com/jargon/p/packet.htm

Data: Public Internet Packet Capture (PCAP)

- Measurement and Analysis of Wide-area Internet (MAWI) working group
  - Day-in-the-Life of the Internet 2015 (http://mawi.wide.ad.jp/mawi/ditl/ditl2015)
  - Day-in-the-Life of the Internet 2017 (http://mawi.wide.ad.jp/mawi/ditl/ditl2017)
  - 2x48=96 hours of 1 Gigabit packet capture (PCAP) headers collected in Tokyo
  - 0.7 TB compressed; 20 TB in analysts friendly form
    - Normalized, sorted, indexed, and read optimized
  - IP addressed deterministically anonymized *within* each collect
    - Network analysis is still valid

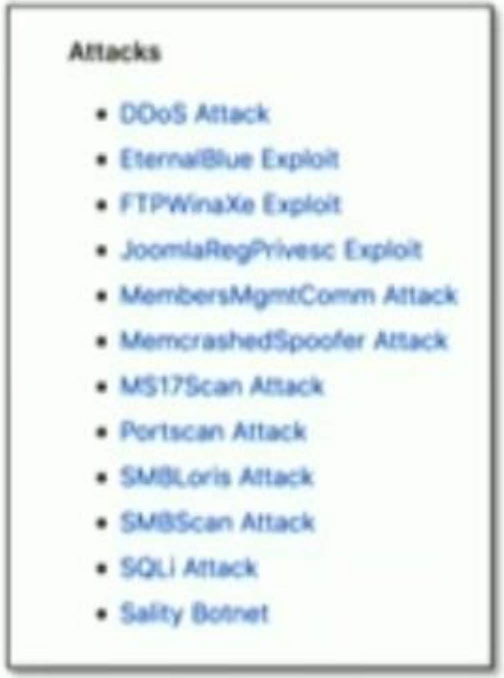
Labelled Data – Using Synthetic Attacks

Synthetic Attack Generation

- One of the major challenges associated with cyber network analysis is the lack of data with known attacks
- ID2T is a toolkit developed at TU Darmstadt that allows users to inject synthetic attacks directly into PCAP data
  - Still difficult to use but it is one of the best (open source) tools we have come across
  - Supports a number of attacks
- For our pipeline:
  - We use ID2T to generate a series of synthetic attacks that are embedded into PCAP data
  - We focus on DDOS and Port Scan attacks

ID2T

the intrusion detection dataset toolkit.

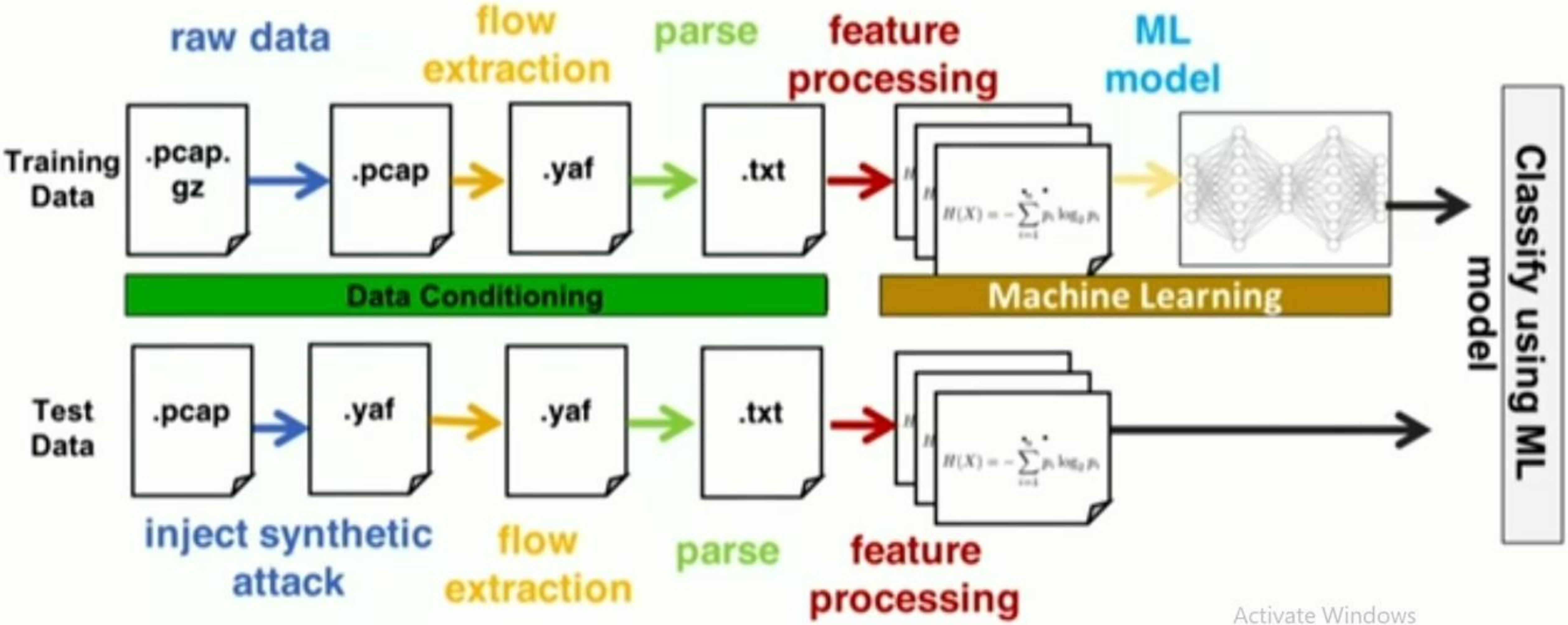


Attacks

- DDoS Attack
- EternalBlue Exploit
- FTPWinaXe Exploit
- JoomlaRegPrivesc Exploit
- MembersMgmtComm Attack
- MemcrashedSpoofer Attack
- MS17Scan Attack
- Portscan Attack
- SMBLoris Attack
- SMBScan Attack
- SQLi Attack
- Sality Botnet

Garcia Cordero et al. (2015) ID2T: a DIY Dataset Creation Toolkit for Intrusion Detection System

Pipeline for Cyber Network Anomaly Detection



raw data

flow extraction

parse

feature processing

ML model

Training Data

.pcap.gz

.pcap

.yaf

.txt

$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$

Data Conditioning

Machine Learning

Classify using ML model

Test Data

.pcap

.yaf

.yaf

.txt

$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$

inject synthetic attack

flow extraction

parse

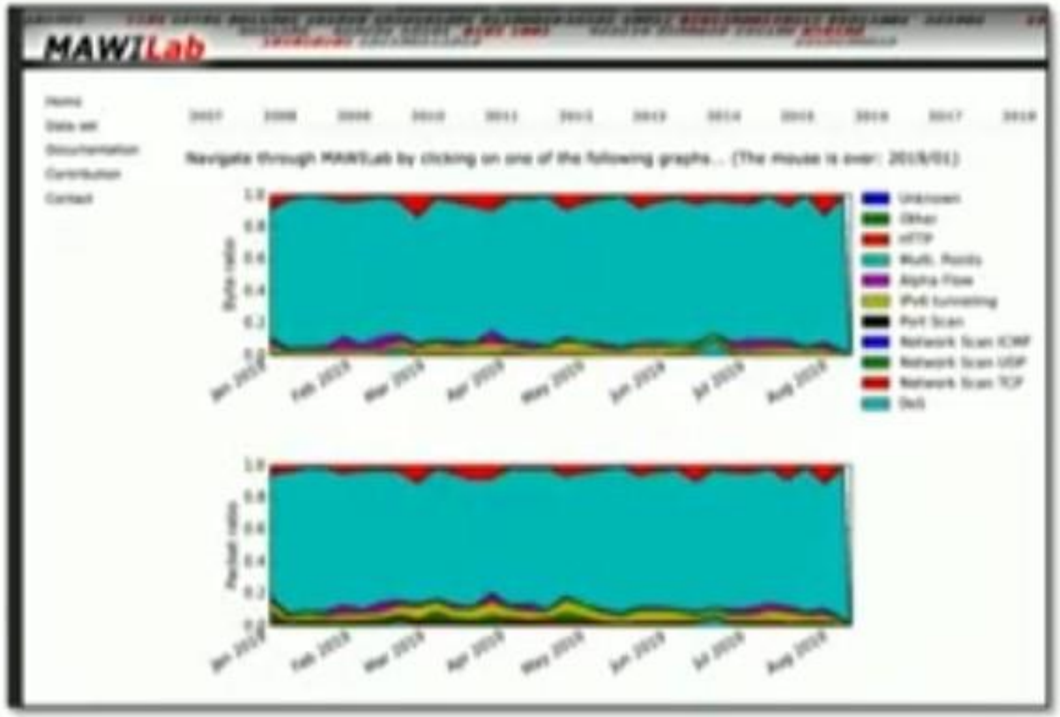
feature processing

Data Conditioning (1)

Raw Data

- Packet Capture data is typically generated from a network traffic analyzer or a utility such as tcpdump

- For our pipeline:
  - Data is downloaded from the MAWI Lab website in .tar.gz
  - Data is uncompressed into the .pcap files
  - Typical size of a single .tar.gz: 2.5GB
  - Typical size of uncompressed .pcap: 10 GB
  - A single .pcap file corresponds to 900 seconds (15 minutes)
  - Corresponds ~150,000 packets/second



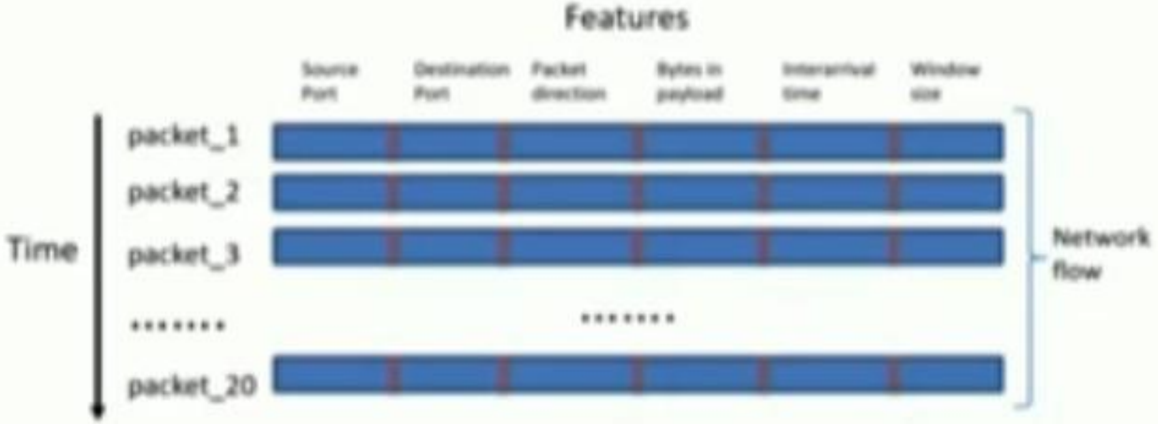
Navigate through MAWILab by clicking on one of the following graphs... (The mouse is over: 2018/01)

Image Source MAWILAB: http://www.fukuda-lab.org/mawilab/data.html

Data Conditioning (2)

Flow Extraction

- A network flow is defined as a sequence of packets from a source to a destination.
- RFC 2722 describes flows as “an artificial logical equivalent to a call or connection”
- For our pipeline:
  - We convert the 15 minute .pcap files into network flow representation using YAF (yet another flowmeter)
  - .yaf output is a binary format
  - Size of 15 minutes worth of flows: 2GB
  - Defined between a source ip, destination ip with the same port. Flow timeout is set to a default 5 min



Software Engineering Institute | Carnegie Mellon University

CERT NetSA Security Suite

Monitoring for Large-Scale Networks

YAF

Documentation

Downloads

YAF is Yet Another Flowmeter. It p
by tcpdump(1) or via live capture
then exports those flows to IPFIX C
output can be used with the SiLK t
IPFIX compliant toolchain.

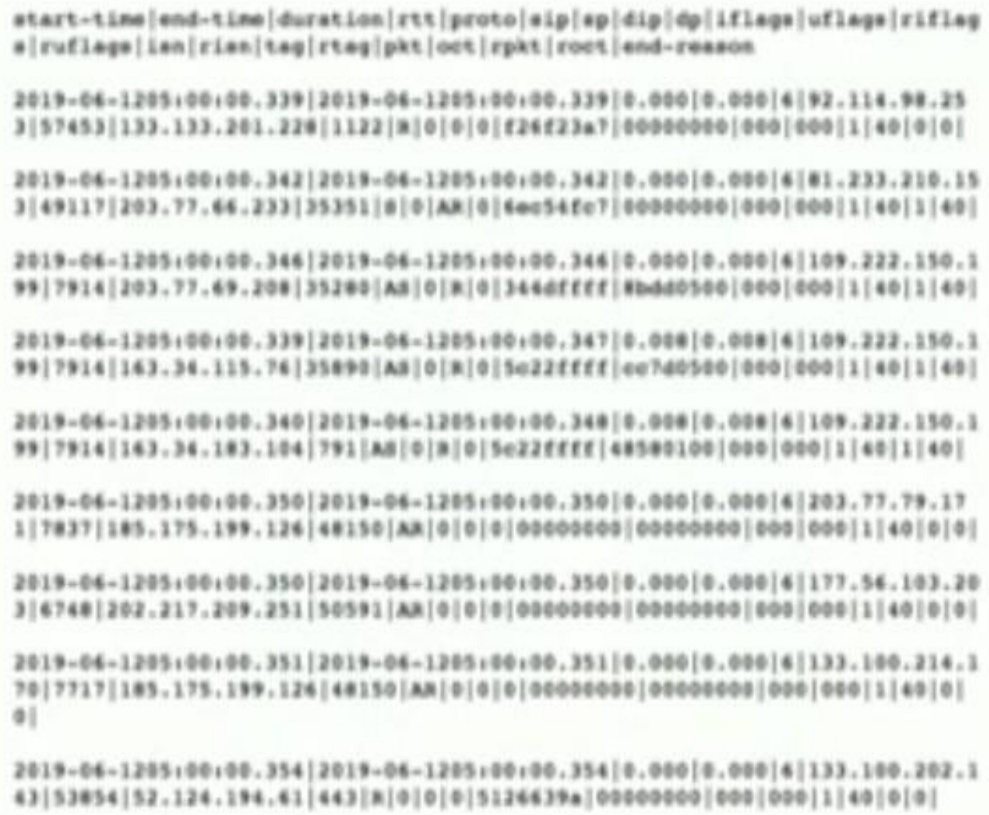
Flow image source:
Lopez-Martin, Manuel, et al. "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things." *IEEE Access* 5 (2017): 18042-18050.

Yaf: https://tools.netsa.cert.org/yaf/yaf.html

Data Conditioning (3)

Parse Flows

- Machine learning models require conversion of binary flow format into some tabular form
- YAF comes with a tool: yafscii to convert binary flows into a human readable form
- For our pipeline:
  - We convert each of the .yaf files into a .txt file using yafscii
  - Typical size of this ascii table is: 8GB
  - Each line of the output text file corresponds to a single flow
  - Following fields are recorded for each flow:



```
start-time|end-time|duration|rtt|proto|sip|sp|dip|dp|iflags|uflags|riflag
s|ruflags|isn|risn|tag|rtag|pkt|oct|rpkt|roct|end-reason

2019-06-1205:00:00.339|2019-06-1205:00:00.339|0.000|0.000|6|92.114.98.25
3|57453|133.133.201.228|1122|S|0|0|0|f26f23a7|00000000|000|000|1|40|0|0|

2019-06-1205:00:00.342|2019-06-1205:00:00.342|0.000|0.000|6|81.233.210.15
3|49117|203.77.66.233|35351|S|0|AR|0|6ec54fc7|00000000|000|000|1|40|1|40|

2019-06-1205:00:00.346|2019-06-1205:00:00.346|0.000|0.000|6|109.222.150.1
99|7914|203.77.69.208|35280|AS|0|R|0|344dffff|8bdd0500|000|000|1|40|1|40|

2019-06-1205:00:00.339|2019-06-1205:00:00.347|0.008|0.008|6|109.222.150.1
99|7914|163.36.115.76|35890|AS|0|R|0|5e22ffff|ee7d0500|000|000|1|40|1|40|

2019-06-1205:00:00.340|2019-06-1205:00:00.348|0.008|0.008|6|109.222.150.1
99|7914|163.36.183.104|791|AS|0|R|0|5e22ffff|48580100|000|000|1|40|1|40|

2019-06-1205:00:00.350|2019-06-1205:00:00.350|0.000|0.000|6|203.77.79.17
1|7837|185.175.199.126|48150|AR|0|0|0|00000000|00000000|000|000|1|40|0|0|

2019-06-1205:00:00.350|2019-06-1205:00:00.350|0.000|0.000|6|177.56.103.20
3|6748|202.217.209.251|50591|AR|0|0|0|00000000|00000000|000|000|1|40|0|0|

2019-06-1205:00:00.351|2019-06-1205:00:00.351|0.000|0.000|6|133.100.214.1
70|7717|185.175.199.126|48150|AR|0|0|0|00000000|00000000|000|000|1|40|0|
0|

2019-06-1205:00:00.354|2019-06-1205:00:00.354|0.000|0.000|6|133.100.202.1
43|53854|52.124.194.61|443|S|0|0|0|5126639a|00000000|000|000|1|40|0|0|
```

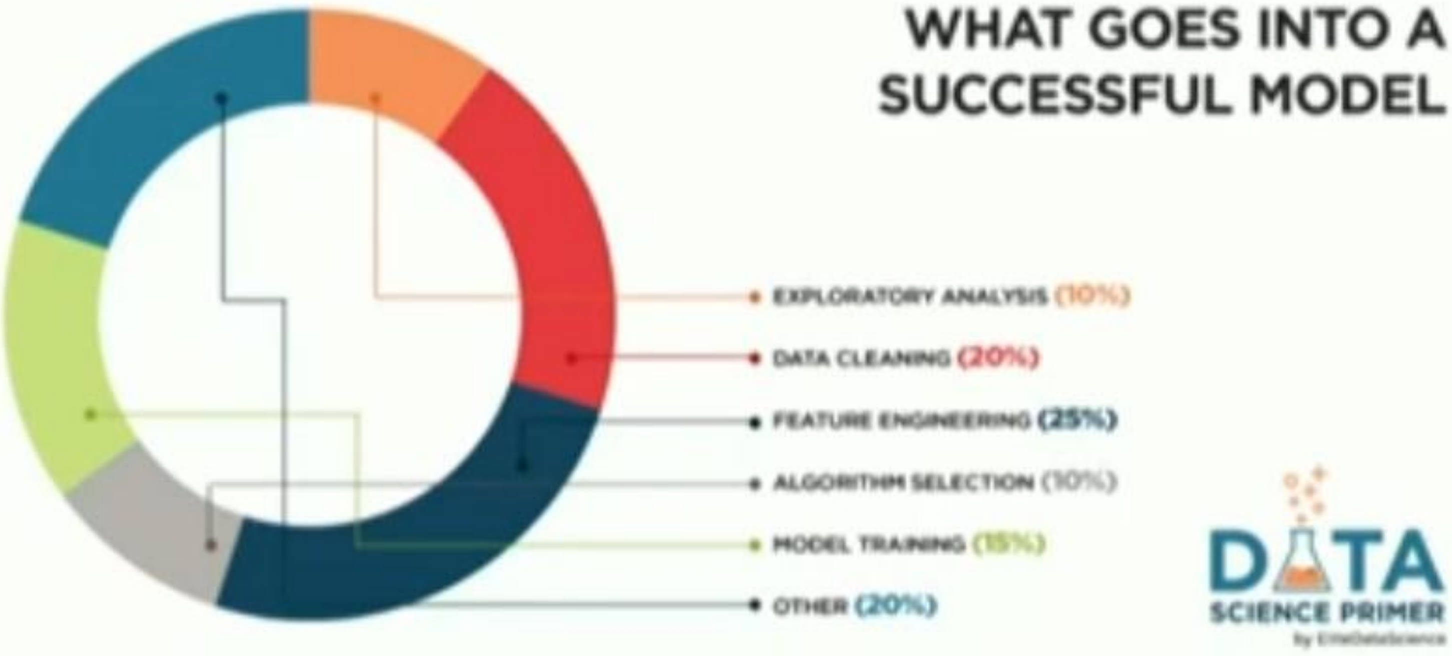
start-time|end-time|duration|rtt|proto|sip|sp|dip|dp|iflags|uflags| riflags|ruflags|isn|risn|tag|rtag|pkt|oct|rpkt|roct|end-reason

Yaf: https://tools.netsa.cert.org/yaf/yaf.html

Tabular Flow Fields

Features of interest	Explanation
Source IP	Source IP address
Source Port	Source port
Destination IP	Destination IP address
Destination Port	Destination port
Protocol	IP protocol
Initial Flags	Forward first-packet TCP flags
Union Flags	Forward nth-packet TCP flags union
Reverse Initial Flags	Reverse first-packet TCP flags
Reverse Union Flags	Reverse nth-packet TCP flags union
End reason	Indicate whether the flow was ended normally (i.e., by TCP RST or FIN), expired by idle timeout, or expired by active timeout.
Destination IP – Destination Port	Combination of Destination IP and Destination Port
Destination IP – Initial Flags	Combination of Destination IP and Initial Flags
Source IP - Destination IP	Combination of Source IP and Destination IP
Source IP - Initial Flags	Combination of Source IP and Initial Flags

Data Conditioning
Feature Engineering (1)



- Each flow contains 21 features such as IP addresses, ports, …
- Many of these are either unchanging or unlikely to help us look for anomalous behavior
- We used domain knowledge, trial-and-error and luck to pick features

Image Source: https://elitedatascience.com/feature-engineering

Data Conditioning
Feature Engineering (2)

Using Entropy

- Entropy is a measure of uncertainty associated with a random variable
- Used extensively in information theory
- Typically measured using Shannon Entropy:

$$H = -\sum p_i log p_i$$

- For our pipeline:
  - We compute the Shannon entropy associated with each of our features: H_{src_ip}, H_{dst_ip}, H_{src_port}, …

Intuition is that the overall entropy of the system should stay reasonably constant without external ”mechanisms”

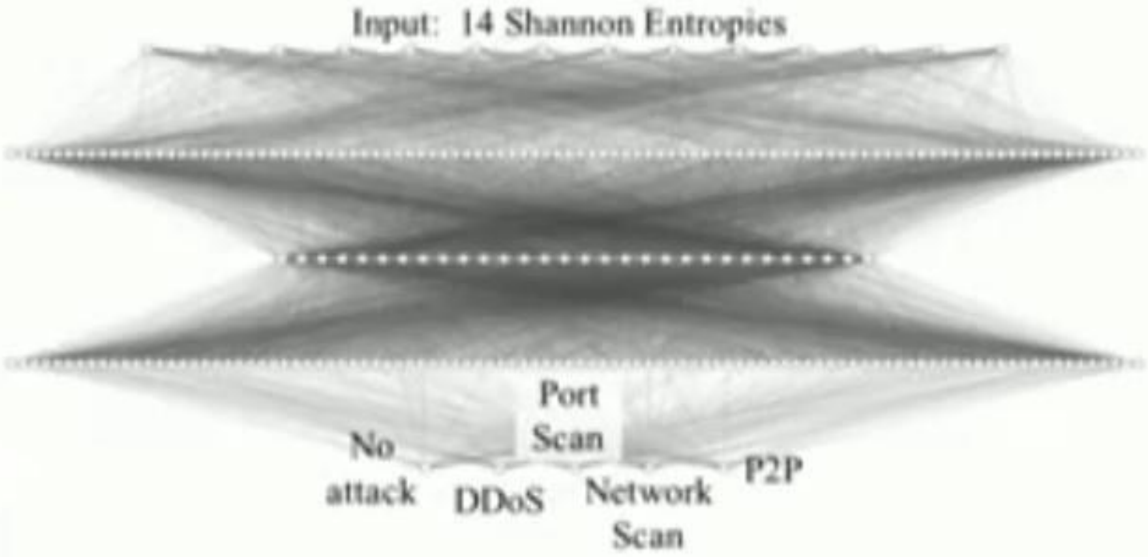
Hawkins, Simon, et al. "Outlier detection using replicator neural networks." *International Conference on Data Warehousing and Knowledge Discovery*. Springer, Berlin, Heidelberg, 2002.

Cordero, Carlos García, et al. "Analyzing flow-based anomaly intrusion detection using replicator neural networks." 2016 14th Annual *Conference on Privacy, Security and Trust (PST)*. IEEE, 2016.

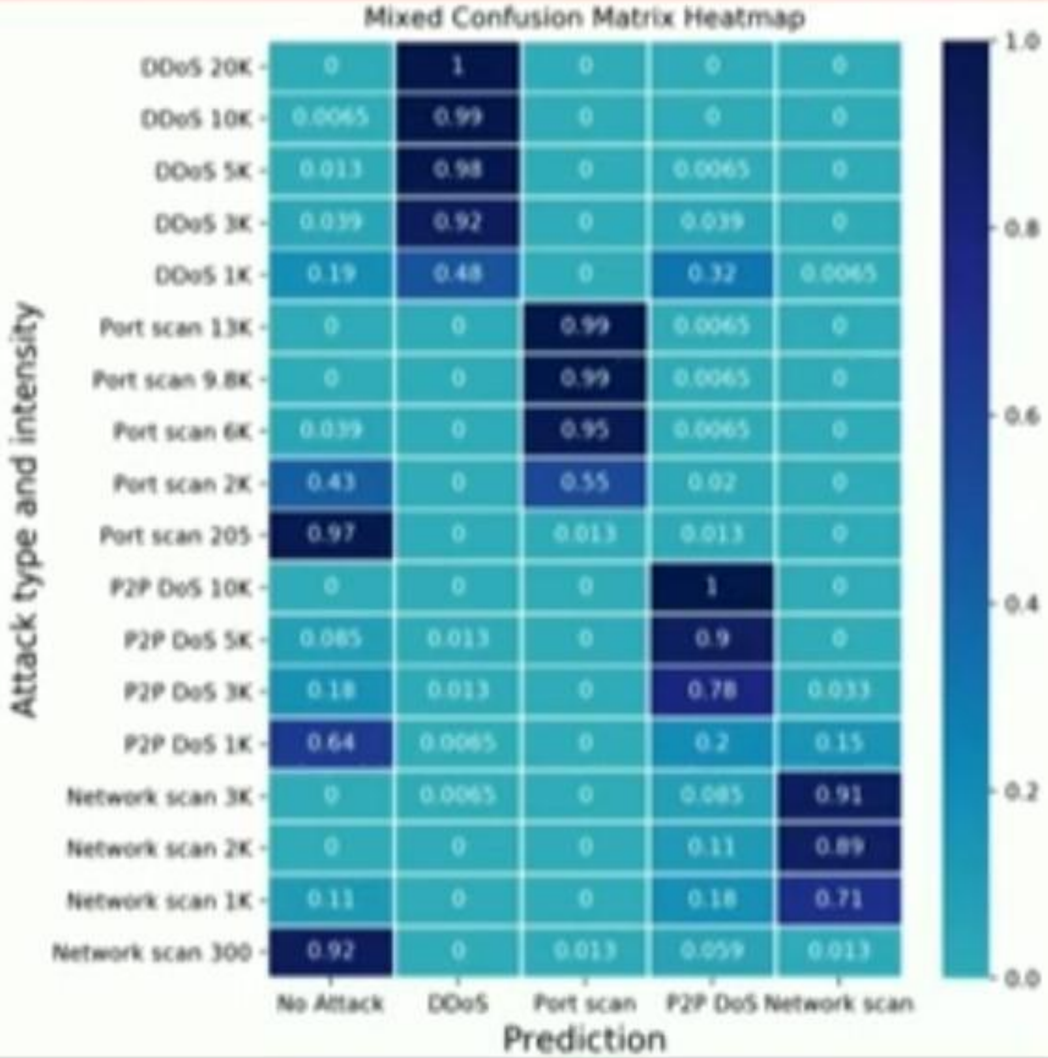
Kotani, Gaku, and Yuji Sekiya. "Unsupervised Scanning Behavior Detection Based on Distribution of Network Traffic Features Using Robust Autoencoders." *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018.

Machine Learning Model
-Training-

- Our model is a fully connected feed-forward network that takes in the 14 entropies as input features and attemps to classify them as one of 5 different classes
  - Input layer, 3 hidden layers (100, 30, and 100 nodes), and an output layer
  - ReLU activation
- Five output classes: No-Attack, DDoS-Attack, Port-Scan-Attack, Point2Point-DoS-Attack, and Network-Scan-Attack



Evaluation



Mixed Confusion Matrix Heatmap

	No Attack	DDoS	Port scan	P2P DoS	Network scan
DDoS 20K	0	1	0	0	0
DDoS 10K	0.0065	0.99	0	0	0
DDoS 5K	0.013	0.98	0	0.0065	0
DDoS 3K	0.039	0.92	0	0.039	0
DDoS 1K	0.19	0.48	0	0.32	0.0065
Port scan 13K	0	0	0.99	0.0065	0
Port scan 9.8K	0	0	0.99	0.0065	0
Port scan 6K	0.039	0	0.95	0.0065	0
Port scan 2K	0.43	0	0.55	0.02	0
Port scan 205	0.97	0	0.013	0.013	0
P2P DoS 10K	0	0	0	1	0
P2P DoS 5K	0.085	0.013	0	0.9	0
P2P DoS 3K	0.18	0.013	0	0.78	0.033
P2P DoS 1K	0.64	0.0065	0	0.2	0.15
Network scan 3K	0	0.0065	0	0.085	0.91
Network scan 2K	0	0	0	0.11	0.89
Network scan 1K	0.11	0	0	0.18	0.71
Network scan 300	0.92	0	0.013	0.059	0.013

On going research!

Summary

- Good results on detecting and classifying network attacks from internet backbone traffic

- Key notes:
  - Data conditioning consisted of a number of steps:
    - Cleaning up collected data
    - Generating "labeled" data using a synthetic attack generator
    - Feature engineering to determine which features and form of the features were likely to get the best results