# THE REAL WORLD APPLICATION OF ML TO CYBER SECURITY

TIM CROTHERS

# BACKGROUND

- >30 years in Information Technology and >20 years in Infosec
- Authored/Co-Authored 16 books to date
- Engineer and Maker
- Unabashed Math & Computer Science Geek
- Spent several years “on the ground at some of the largest breaches”

# OPPORTUNITY

- Breaches continue to grow in number and severity year after year
- Severe shortage in Cyber Security Subject Matter Expertise
- Venture capital funds and research opportunities are readily available

# 92%

# COMMON FAILURE #1

- Pure anomaly detection
  - Real world networks are messy
  - Real world systems are inconsistently configured
  - Real world vendor applications are usually abnormal
  - Real world hosts are all unique within a few minutes of the end user taking possession

# COMMON FAILURE #2

- Trying to be all security things to all security people
  - Determining optimal parameters and features for a tightly scoped use case is pretty easy
  - As the width of the use case increases the difficulty increases exponentially

# COMMON FAILURE #3

- Failing to leverage deep cyber security subject matter expertise
  - It’s hard to solve a problem you don’t understand well
  - Interesting != security problem
  - Security problem != something that will improve security
  - Success in a lab is much easier than success in a real world environment

# COMMON FAILURE #4

- Failing to leverage deep ML subject matter expertise
  - Proper parameter and feature selection is critical
  - Proper algorithm selection is really important
  - Proper testing and refinement is critical

# SUCCESS IS POSSIBLE!

Clearcut – https://github.com/DavidJBianco/Clearcut

- Finds interesting security entries in HTTP Proxy Logs

Malicious Macro Bot – https://github.com/egaus/MaliciousMacroBot

- Is a document macro malicious?

Assimilate – https://github.com/Soinull/assimilate

- Finds interesting security HTTP/HTTPS headers

# KEYS TO SUCCESS

- Tightly scoped problem statement or use case
- Decide on approach
- Appropriate data
- Determine proper parameters and features
- Test & tune

# TIGHTLY SCOPED PROBLEM

- Find the malicious activity in my DNS that my signature based detection isn't finding
- Find malicious PowerShell activity in Windows event logs that isn't being detected otherwise
- Find unknown malicious traffic posing as legitimate applications

# DECIDE ON APPROACH

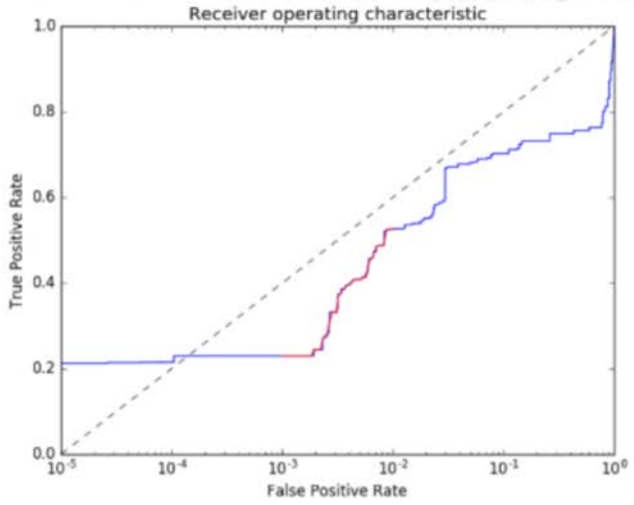
- Supervised
  - Generally best for solving specific problems
  - Needs ’labeled’ data
- Unsupervised
  - Essentially anomaly detection
  - Needs large piles of real world data
  - Inherent assumption attacks are rare

# APPROPRIATE DATA

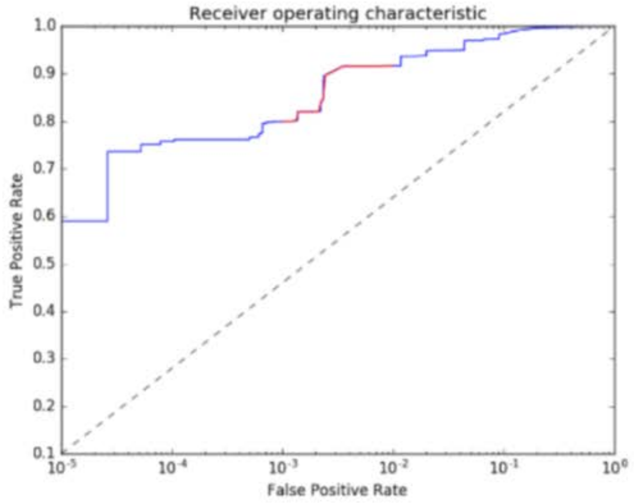
- Hardest part of doing in the real world
- Data appropriate to the problem you selected
- Known good & known bad
  - Bad Samples: https://www.malware-traffic-analysis.net/
- Use 80% of each so you can use the other 20% for testing & tuning

# DETERMINE PROPER FEATURES

- Blend of ML and Cyber Security Expertise really critical
  - Start with Cyber Security
  - Validate using standard Data Science techniques




(a) The initial parameters.




(b) The modified parameters.

**Figure 6:** The ROC Curve Produced by the Model under Different Settings of Parameters.

Excerpt from “Practical Cyborgism” by David J. Bianco and Chris McCubbin :
https://speakerdeck.com/davidjbianco/practical-cyborgism-getting-started-with-machine-learning-for-incident-detection

# ASSIMILATE BUILD STEP-BY-STEP

- Gathered the real world network data (one week > 10TB)
- Used Bro to convert the packet captures into metadata (HTTP)
- Compiled over a years worth of packet captures from malware and converted with Bro similarly
- Cleaned the Malicious Bro metadata of the non-malware activity
- Used the malicious data to clean the real world network data
- Tested for algorithm, parameters and features
- Coded trainer & model application, tested, iterated
- 🍺

# PACKET CAPTURES (PCAP) PROCESSING

```
# Example script to iterate over pcap files to get corresponding http.log and httpheader.log files
for file in ../*.pcap
do
   name=${file##*/}
   echo $name
   base=${name%.pcap}
   echo $base
   cp ../"$file" .
   bro -r "$file" custom/BrowserFingerprinting/http-headers.bro
   mv http.log ../"$base"_http.log
   mv httpheaders.log ../"$base"_httpheaders.log
   rm -f *.log *.pcap
done
```

# TEST & TUNE

- Standard ML best practices apply
- If the accuracy is too low:
  - Is your sample data solid?
  - Is your parameter & feature selection strong?
  - Try swapping different algorithms

# RECOMMENDED RESOURCES

Real world bad traffic – https://www.malware-traffic-analysis.net/

Basics - https://speakerdeck.com/davidjbianco/introduction-to-data-analysis-with-security-onion-and-other-open-source-tools

Mid-level - https://speakerdeck.com/davidjbianco/practical-cyborgism-getting-started-with-machine-learning-for-incident-detection

# THANK YOU!

badsecurity@gmail.com

@soinull

linkedin.com/in/tim-crothers-5458738/

https://github.com/soinull/ML_for_Cyber