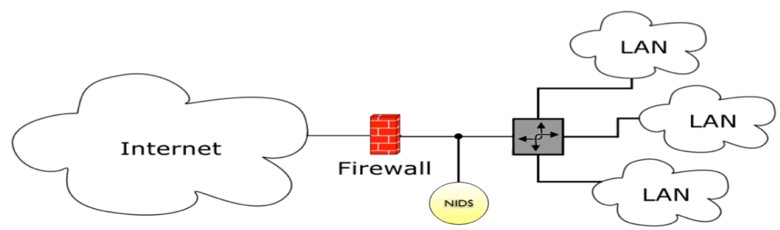
CS259D: Data Mining for CyberSecurity

Outline

- Introduction
- Challenges with using ML
- Guidelines for using ML
- Conclusions

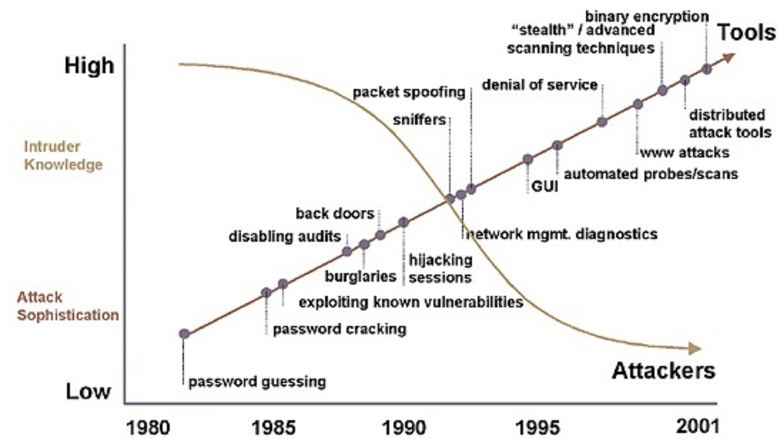
Types of network intrusion detection

- Misuse detection
  - Exact descriptions of known bad behavior
- Anomaly detection
  - Deviations from profiles of normal behavior
  - First proposed in **1987** by Dorothy Denning (Stanford Research Institute)



Why ML for security: Attack landscape

- Attacks sophistication
  - **403M** new variants of malware created in 2011
  - **100K** unique malware samples daily in 2012 Q1
- Required attacker knowledge decreasing
- Highly motivated attackers



Why ML for security: Reactive defense failing

Median time between breach and awareness	**300-400+** days
Duration of zero-day attacks	up to **30** months, median **8** months
% of attacks discovered by a third party	**61%**
% of businesses that share breach info	**2-3%**

ML success in other domains

- Product recommendations
  - Amazon, Netflix
- Optical character recognition
  - Google
- Natural language translation
  - Google, Microsoft
- Spam detection
  - Google, Yahoo, Microsoft, Facebook, Twitter

Fact

- Almost all NIDS systems used in operational environments are misuse-based
    - Despite lots of research on anomaly detection
    - Despite appeal of anomaly detection to find new attacks
    - Despite success of ML in other domains

Challenges

- Outlier detection
- High cost of errors
- Lack of appropriate training data
- Interpretation of results
- Variability in network traffic
- Adaptive adversaries
- Evaluation difficulties

Challenge: Outlier detection

	Classification	Outlier detection
Training samples	Many from both classes	Almost all from one class
Required quality	Enough to distinguish two classes	Perfect model of normal

- Premise: Anomaly detection can find **novel** attacks

- Fact: ML is better at finding **similar** patterns than at finding outliers
  - ✓Example: Recommend similar products; similarity: products purchased together

- Conclusion: ML is better for finding **variants** of known attacks

Challenge: Outlier detection

- Underlying assumptions
  - Malicious activity is anomalous
  - Anomalies correspond to malicious activity
- Do these assumptions hold?
  - Former employee requests authorization code
    - Account revocation bug? Insider threat?
    - Username typo
  - User authentication fails 10K times
    - Brute force attack?
    - User changed password, forgot to update script

Challenge: High cost of errors

	Cost of False Negatives	Cost of False Positives
Product recommendation	**Low**: potential missed sales	**Low**: continue shopping
Spam detection	**Low**: spam finding way to inbox	**High**: missed important email
Intrusion detection	**High**: Arbitrary damage	**High**: wasted precious analyst time

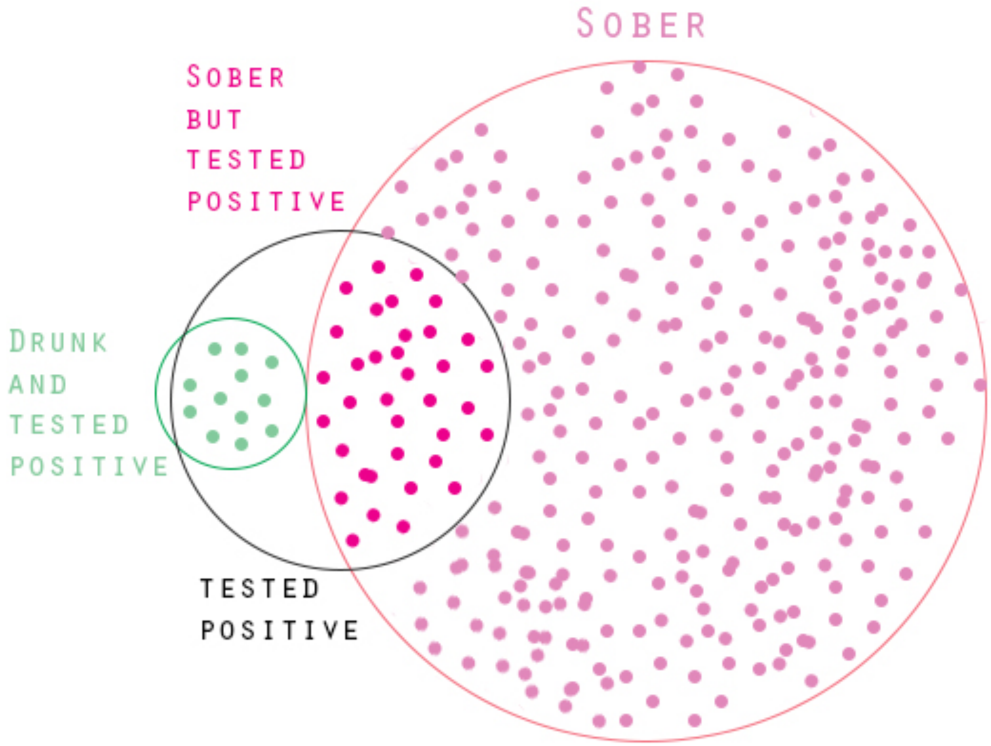
Post-processing:

- ✓Spelling/grammar checkers to clean up results
- ✓Proofreading: Much easier than verifying a network intrusion

Thought experiment

- Assume:
  - Breathalyzer gets the answer right **90%** of the time
  - It detects a driver as drunk
- Question:
  - What is the probability the driver is actually drunk?

Base rate fallacy



Challenge: Lack of appropriate training data

- Attack free data hard to obtain
- Labeled data expensive to obtain

	Training
Product recommendation	Supervised
Spam detection	Supervised
Intrusion detection	Unsupervised

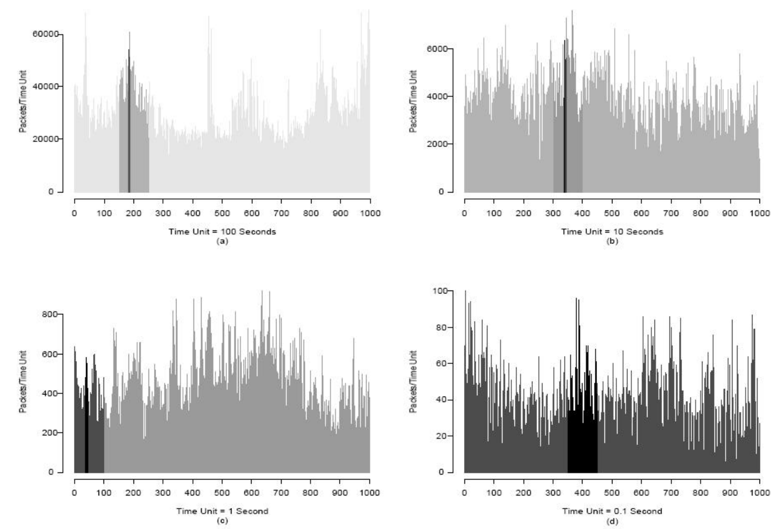
Challenge: Interpretation of results

	Goal
Product recommendation	**Classify**
Spam detection	**Classify**
Intrusion detection	**Classify and Interpret**

- Network operator needs **actionable** reports
  - What does the anomaly ***mean***?
  - Abnormal activity vs. Attack
  - Incorporation of site-specific security policies
  - Relation between features of anomaly detection & semantics of environment

Challenge: Variability in network traffic

- Variability across all layers of the network
  - Even most basic characteristics: bandwidth, duration of connections, application mix
- Large bursts of activity



Challenge: Variability in network traffic

- What is a stable notion of *normality*?
- Anomalies ≠ Attacks
- One solution: Reduced granularity
  - Example: Time-of-Day, Day-of-Week
  - Pro: More stable
  - Con: Reduced visibility

Challenge: Adaptive adversaries

- Adversaries adapt
  - ML assumptions do not necessarily hold
    - I.I.D, stationary distributions, linear separability, etc.
- ML algorithm itself can be an attack target
  - Mistraining, evasion

Challenge: Evaluation

- Difficulties with data
  - Data's sensitive nature
  - Lack of appropriate public data
    - Automated translation: European Union documents
  - Simulation
    - Capturing characteristics of real data
    - Capturing *novel* attack detection
  - Anonymization
    - Fear of de-anonymization
    - Removing features of interest to anomaly detection

Challenge: Evaluation

- Interpreting the results
  - “HTTP traffic of host did not match profile”
  - Contrast with spam detection: Little room for interpretation
- Adversarial environment
  - Contrast with product recommendation: Little incentive to mislead the recommendation system

Root cause

- **Using tools** borrowed from ML in **inappropriate** ways
- Goal: Effective adoption of ML for large-scale operational environments
  - Not a Black box approach
  - Crisp definition of context
  - Understanding semantics of detection

Guidelines

- Understand the threat model
- Keep the scope narrow
- Reduce the costs
- Use secure ML
- Evaluation
- Gain insights to the problem space

Guideline: Understand the threat model

- What kind of target environment?
  - Academic vs enterprise; small vs large/backbone
- Cost of missed attacks
  - Security demands, other deployed detectors
- Attackers' skills and resources
  - Targeted vs background radiation
- Risk posed by evasion

Guideline: Keep the scope narrow

- What are the specific attacks to detect?
- Choose the right tool for the task
  - ML not a silver bullet
  - Common pitfall: Start with intention to use ML or even worse a particular ML tool
  - No Free Lunch Theorem
- Identify the appropriate features

Example

- **Features**: Byte frequencies in packet payloads
- **Algorithm**: Detect packets with anomalous frequency patterns
- **Assumption**: Attack payloads have different payload byte frequencies
- **Question**: Where does this assumption come from?

Example

- **Threat model**: Web-based attacks using input parameters to web applications
- **Why anomaly detection**: Attacks share conceptual similarities, yet different enough in their specifics for signatures
- **Data**:
  - Successful GET requests to CGI apps, from web server Access Logs
- **Features**:
  - Length of attribute value, Character distribution of attribute value
- **Why is this feature relevant**
  - Length: Buffer overflow needs to send shellcode and padding
  - Character distribution: Directory traversal uses too many "." & "/"

Guideline: Reduce the costs

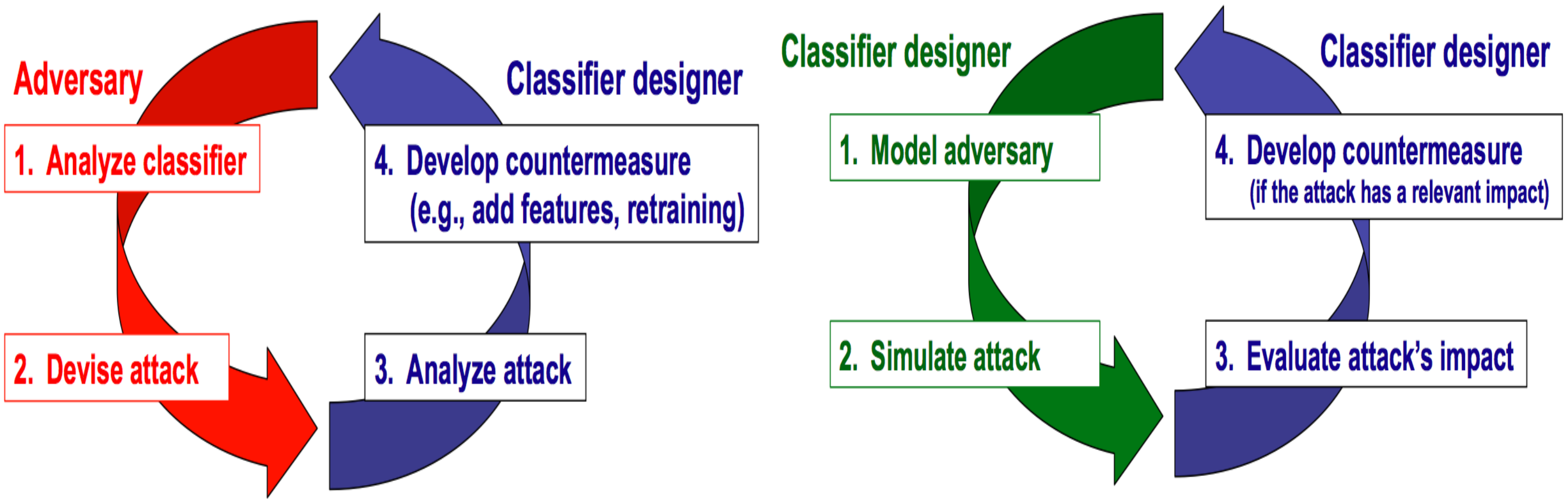
- Reduce the system's scope
- Classification over outlier detection
- Aggregate features over suitable intervals
- Post-process the alerts
- Provide meta-information to analyst to speed up inspection

Guideline: Use secure ML

"If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle."

– Sun Tzu, The Art of War

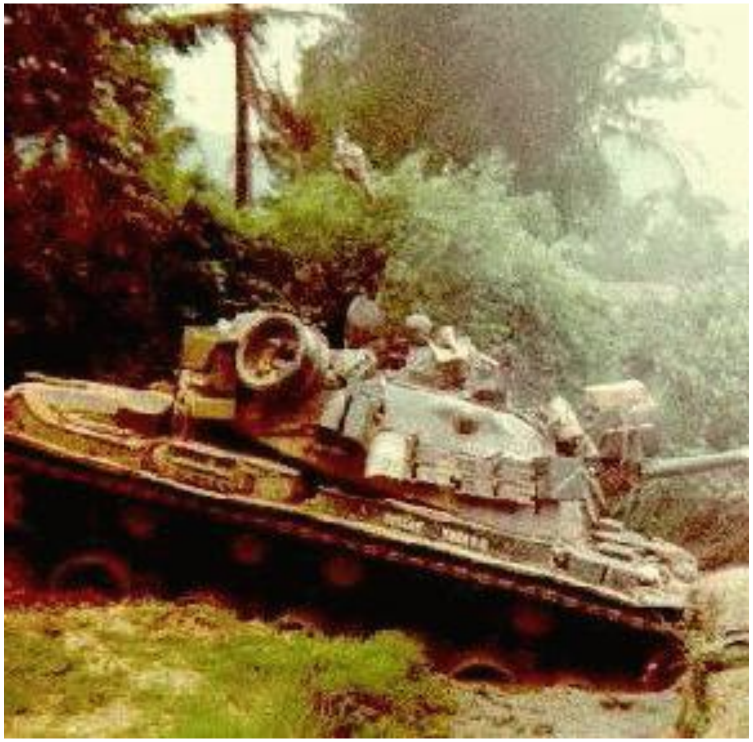
Guideline: Use secure ML



Guideline: Evaluation

- Develop insight into anomaly detection system's capabilities
  - What can/can't it detect? Why?

Guideline: Evaluation



Guideline: Gain insights to the problem space

- ML as means to identify important features
- Use those features to build non-ML detectors
- ML as a means to an end

Reference

- “Outside the closed world: On using machine learning for network intrusion detection”, Sommer-Paxson, 2010
- “Challenging the Anomaly Detection Paradigm: A Provocative Discussion”, Gates-Taylor, 2007
- “The Base-Rate Fallacy and Its Implications for the Difficulty of Intrusion Detection”, Axelsson, 1999