Intrusion Detection, Firewalls, and Intrusion Prevention

Professor Patrick McDaniel
ECE 590 – 03 (Guest lecture)
Intrusion Detection Systems

• Authorized eavesdropper that listens in on network traffic or host behavior (e.g., packets, system calls)
• Makes determination whether behavior contains/indicates malware
• Typically usually compares payload/activity to malware signatures
• If malware is detected, IDS somehow raises an alert
• Intrusion detection is a **classification** problem

This becomes a **intrusion prevention system** if it actively attempts to block activity ...
Example Network Setup
Detection via Signatures

- Signature checking (pattern matching)
  - does packet match some signature
  - suspicious headers
  - suspicious payload (e.g., shellcode)
  - great at matching known signatures ...

- Problem: not so great for zero-day attacks
Detection via Machine Learning

- Use ML techniques to identify malware
- Underlying assumption: malware will look different from non-malware
- Supervised learning
  - IDS requires learning phase in which operator provides pre-classified training data to learn patterns
  - Sometimes called anomaly detection (systems)
  - \{good, 80, “GET”, “/”, “Firefox”\}
  - \{bad, 80, “POST”, “/php-shell.php?cmd=’rm -rf /’”, “Evil Browser”\}
  - ML technique builds model for classifying never-before-seen packets

- **Problem**: is new malware going to look like training (data) malware?
Q: What is an intrusion?

- What constitutes an intrusion/anomaly is really just a matter of definition
  - A system can exhibit all sorts of behavior
- Quality determined by the consistency with a given definition
  - Context sensitive

Q: Which of these events would you consider an attack on the grading system?

- A student Bob changes the final grade of Gina in this class?
- A TA Alice changes the final grade for Gina in this class?
- A professor Patrick changes the final grade for Gina in this class?
Detection
Theory
Confusion Matrix

• A Confusion matrix is a table describing the performance of some detection algorithm
  • True positives (TP): number of correct classifications of malware
  • True negatives (TN): number of correct classifications of non-malware
  • False positives (FP): number of incorrect classifications of non-malware as malware
  • False negatives (FN): number of incorrect classifications of malware as non-malware
Metrics
(from perspective of detector)

• False positive rate:
\[ FPR = \frac{FP}{FP + TN} = \frac{\# \text{benign marked as malicious}}{\text{total benign}} \]

• True negative rate:
\[ TNR = 1 - FPR = \frac{TN}{FP + TN} = \frac{\# \text{benign unmarked}}{\text{total benign}} \]

• False negative rate:
\[ FNR = \frac{FN}{FN + TP} = \frac{\# \text{malicious not marked}}{\text{total malicious}} \]

• True positive rate:
\[ TPR = 1 - FNR = \frac{TP}{FN + TP} = \frac{\# \text{malicious correctly marked}}{\text{total malicious}} \]
**Recall** (also known as sensitivity)
- fraction of correct instances among all instances that actually are positive (malware)
- \( \frac{TP}{TP + FN} \)

**Precision**
- fraction of correct instances (malware) that algorithm believes are positive (malware)
- \( \frac{TP}{TP + FP} \)

Recall: percent of malware you catch  
Precision: percent correctly marked as malware
Bayes Rule

- Pr(x) function, probability of event x
  - Pr(sunny) = .8 (80% of sunny day)

- Conditional probability
  - Pr(x|y), probability of x given y
  - Pr(cavity|toothache) = .6
    - 60% chance of cavity given you have a toothache

- Bayes’ Rule (of conditional probability)

\[
Pr(B|A) = \frac{Pr(A|B) \cdot Pr(B)}{Pr(A)}
\]

- Assume: Pr(cavity) = .5, Pr(toothache) = .1
- What is Pr(toothache|cavity)?
Base Rate Fallacy

- Occurs when assessing $P(X|Y)$ without considering probability of $X$ and the total probability of $Y$
  - Example:
    - Base rate of malware is 1 packet in a 10,000
    - Intrusion detection system is 99% accurate (given known samples)
    - 1% false positive rate (benign marked as malicious 1% of the time)
    - 1% false negative rate (malicious marked as benign 1% of the time)
  - Packet X is marked by the NIDS as malware. What is the probability that packet X actually is malware?
  - Let’s call this the “true alarm rate,” because it is the rate at which the raised alarm is actually true.
Base Rate Fallacy

- How do we find the true alarm rate? [i.e., \( Pr(\text{IsMalware} | \text{MarkedAsMalware}) \)]
  \[
  Pr(\text{IsMalware} | \text{MarkedAsMalware}) = \frac{Pr(\text{MarkedAsMalware} | \text{IsMalware}) \cdot Pr(\text{IsMalware})}{Pr(\text{MarkedAsMalware})}
  \]

- We know:
  - 1% false positive rate (benign marked as malicious 1% of the time); TNR = 99%
  - 1% false negative rate (malicious marked as benign 1% of the time); TPR = 99%
  - Base rate of malware is 1 packet in 10,000

- What is?
  - \( Pr(\text{MarkedAsMalware} | \text{IsMalware}) \) = ? TPR = 0.99
  - \( Pr(\text{IsMalware}) \) = ? Base rate = 0.0001
  - \( Pr(\text{MarkedAsMalware}) \) = ?

\[
Pr(\text{Marked}) = Pr(\text{Marked} | \text{Malware}) \cdot Pr(\text{Malware}) + Pr(\text{Marked} | !\text{Malware}) \cdot Pr(!\text{Malware})
\]

\[
Pr(\text{Marked}) = (0.99 \cdot 0.0001) + (0.01 \cdot 0.9999) = 0.01
\]
Base rate fallacy ...

• How do we find the true alarm rate? [i.e., $\Pr(\text{IsMalware} | \text{MarkedAsMalware})$]

$$
\Pr(\text{IsMalware} | \text{MarkedAsMalware}) = \frac{\Pr(\text{MarkedAsMalware} | \text{IsMalware}) \cdot \Pr(\text{IsMalware})}{\Pr(\text{MarkedAsMalware})}
$$

$$
= \frac{0.99 \cdot 0.0001}{0.01} = 0.0099
$$

• Therefore only about 1% of alarms are actually malware!
• What does this mean for network administrators?
### Where is Anomaly Detection Useful?

| System | Intrusion Density $P(M)$ | Detector Alarm $Pr(A)$ | Detector Accuracy $Pr(A|M)$ | True Alarm $P(M|A)$ |
|--------|---------------------------|------------------------|-----------------------------|-------------------|
| A      | 0.1                       |                        | 0.65                        |                   |
| B      | 0.001                     |                        | 0.99                        |                   |
| C      | 0.1                       |                        | 0.99                        |                   |
| D      | 0.00001                   |                        | 0.99999                     |                   |

\[
Pr(B|A) = \frac{Pr(A|B) \cdot Pr(B)}{Pr(A)}
\]
Where is Anomaly Detection Useful?

| System | Intrusion Density $P(M)$ | Detector Alarm $Pr(A)$ | Detector Accuracy $Pr(A|M)$ | True Alarm $P(M|A)$ |
|--------|--------------------------|------------------------|-----------------------------|---------------------|
| A      | 0.1                      | 0.38                   | 0.65                        | 0.171               |
| B      | 0.001                    | 0.01098                | 0.99                        | 0.090164            |
| C      | 0.1                      | 0.108                  | 0.99                        | 0.911667            |
| D      | 0.00001                  | 0.00002                | 0.99999                     | 0.5                 |

$$Pr(B|A) = \frac{Pr(A|B) \cdot Pr(B)}{Pr(A)}$$
Calibrating Detection
The ROC curve

• Receiver Operating Characteristic (ROC)*
  • Curve that shows that detection/false positive ratio (here, for a binary classifier system as its discrimination threshold is varied)

*AKA, Area Under the Curve (AUC)
You are told to design an intrusion detection algorithm that identifies vulnerabilities by solely looking at transaction length, i.e., the algorithm uses a packet length threshold $T$ that determines when a packet is marked as an attack. More formally, the algorithm is defined:

$$D(k,T) \rightarrow [0, 1]$$

where $k$ is the packet length of a suspect packet in bytes, $T$ is the length threshold, and (0,1) indicate that packet should or should not be marked as an attack, respectively if the transaction length > $T$. You are given the following data to use to design the algorithm.

- attack packet lengths: 1, 1, 2, 3, 5, 8
- non-attack packet lengths: 2, 2, 4, 6, 6, 7, 8, 9

Draw the ROC curve.
Solution

attack packet lengths: 1, 1, 2, 3, 5, 8

non-attack packet lengths: 2, 2, 4, 6, 6, 7, 8, 9

\[ TP\% = TPR = \frac{TP}{TP + FN} \]
\[ FP\% = FPR = \frac{FP}{FP + TN} \]

<table>
<thead>
<tr>
<th>T</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>TP%</td>
<td>0.00</td>
<td>33.33</td>
<td>50.00</td>
<td>66.67</td>
<td>66.67</td>
<td>83.33</td>
<td>83.33</td>
<td>83.33</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>FP</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>FP%</td>
<td>0.00</td>
<td>0.00</td>
<td>25.00</td>
<td>25.00</td>
<td>37.50</td>
<td>37.50</td>
<td>62.50</td>
<td>75.00</td>
<td>87.50</td>
<td>100.00</td>
</tr>
</tbody>
</table>
• The ROC curve shows the (natural) trade off between detection (detecting instances of malware) vs false positives
  • Systems are calibrated by picking a pareto point on the curve representing good accuracy vs. the cost of dealing with false positives
  • This is harder than you would think ...

Note: ROC curves are used to calibrate any detection systems, and is used in signal processing (e.g., cell phone reception), medicine, weather prediction, etc.
Practical IDS/IPS
Problems with IDSes/IPS

• VERY difficult to get both good recall and precision
• Malware comes in small packages
• Looking for one packet in a million (billion? trillion?)
• If insufficiently sensitive, IDS will miss this packet (low recall)
• If overly sensitive, too many alerts will be raised (low precision)
• Automated IPS can be induced to responding