Lecture 14: Overview of Anomaly-based Intrusion Detection Systems

Professor Ehab Al-Shaer
CyberDNA Center
Software and Information Systems,
School of Computing and Informatics
University of North Carolina, Charlotte, NC
Misuse Vs. Anomaly Detection

Activity believed To be **ABNOMAL**

- Estimated normal
- Actual normal

Activity believed To be **NOMAL**

- False negative
- False positive
Classification of IDS Systems

- **Misuse detection** is based on extensive knowledge of patterns associated with known attacks provided by human experts
  - Existing approaches: pattern (signature) matching, expert systems, state transition analysis, data mining
  - Major limitations:
    - Unable to detect novel & unanticipated attacks
    - Signature database has to be revised for each new type of discovered attack and this takes time

- **Anomaly detection** is based on profiles that represent normal behavior of users, hosts, or networks, and detecting attacks as significant deviations from this profile
  - Major benefit - potentially able to recognize unforeseen attacks.
  - Major limitation -
    - possible high false alarm rate, since detected deviations do not necessarily represent actual attacks
  - Major approaches: statistical methods, expert systems, clustering and outlier detection, supported vector machine, neural networks, information theory
Elements of Intrusion Detection

- Primary assumptions:
  - System activities are observable
  - Normal and intrusive activities have distinct evidence
- Components of intrusion detection systems:
  - From an algorithmic perspective:
    - Features - capture intrusion evidences
    - Models - piece evidences together
  - From a system architecture perspective:
    - Audit data processor, knowledge base, decision engine, alarm generation and responses
Anomaly Detection

Relatively high false positive rate - anomalies can just be new normal activities.
Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)

- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)
Intrusion Detection Expert System (IDES) – Anomaly-based detection

- IDES uses two behavioral measures (historical profile is compared with the user profile)
  - ordinary or continuous: numeric count
    - CPU time used
    - Number of audit records produced
  - Categorical or discrete: identity and frequency of occurrence
    - Binary
      - Whether the file was accessed
      - Whether the server has been updated
    - Linear -- majority of measures
      - # of times each command was used
      - # of failed login within a time period
      - # files modified
Anomaly Detection

- IDES or Dorothy Denning (ISR) Mode (1986!)
  - Operational Model: event counters and threshold
  - Mean and standard Deviation (d) Model: if the behavior away from the mean by more than $d \rightarrow$ anomalous
  - Multivariate Model: performs correlation among two or more metrics (e.g., cpu load and number of sessions)
  - Markov Process Model:
    - events/states with transitions indicate the frequency/probability
    - Anomalous activity is found if the its probability as determined by the previous state and value in the state transition matrix is too low.

- Quantitative Analysis Approach
  - Very widely used in both misuse and anomaly detection
  - Threshold detection: periodically or over sliding window (# of E per sec)
  - Heuristic Threshold detection: threshold determined as a function
  - Target-based integrity check: e.g., file integrity check
bec the prob is based on the prev state and the next state (transition)
Ehab Al-Shaer, 4/30/2007

ex of threshold more than 3 faild login attempt
Ehab Al-Shaer, 4/30/2007

ex of heurisitc: any abonormal number of faild login
Ehab Al-Shaer, 4/30/2007

instead of using a fixed failed logins, take the mean and STD for failed login is calculated and the any number if failed logins is compared with this. also Gaussian (or Chi-suare) can be used to constrcut the prob distribution fucltion.
Ehab Al-Shaer, 5/14/2007
### Anomaly Detection (cont)

**Statistical Measures:**
- **IDES:** statistical profile (vector) of normal user actions (e.g., file access, sessions, cpu used..etc) is compared with current actions. Vectors decayed exponentially.
- **Haystack – US Air Force:** Using threshold-based comparison identify the exceeded measures and then their associated weights are summed. This is used to rank the session behavior suspicious. Then deviation from this score (user’s session activity) based on all previous session is determined.
- Each element in the vector consider an aspect of the behavior and updated periodically.
- **Drawback (high false alarms):**
  - Applied in patch mode which prevents automated response
  - Precludes the capability to take into account the sequential relation between events

**Nonparametric Statistical Measures:**
- Preprocessing is performed to represent users activity in a vector of features
- Clustering is performed on these data vectors to divide them into two groups: normal and anomalous groups. (e.g., of clustering k-nearest neighbor algorithm)
ESS14  no specific assumption can be made regarding the distribution of the data being analyzed
Ehab Al-Shaer, 5/14/2007

ESS15  K-nearest alg groups each vector with k of its nearest neighbors. k is a function of the number of vectors in the sample set, not a fixed value,
Ehab Al-Shaer, 5/14/2007
Limitations of Statistical Approaches

- Most of the tests are for a single attribute
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution
Anomaly Detection Using Clustering

- General Steps
  - Build a profile of the “normal” behavior
    - Profile can be patterns or summary statistics for the overall population
  - Use the “normal” profile to detect anomalies
    - Anomalies are observations whose characteristics differ significantly from the normal profile

- Types of anomaly detection schemes
  - Graphical & Statistical-based
  - Distance-based
  - Model-based
Graphical Approaches

- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)
- Limitations
  - Time consuming
  - Subjective
Rule-based Approach (learning-based)
- Rules represent and store usage patterns
- E.g., time-based inductive machine (TIM)
- Looks at pattern in sequences of events (not single event)
- It effectively implements Markov transition probability model
- Check to see if a chain of events corresponds to what is expected based on observation of historical event sequences
- E.g., from history $e_1\rightarrow e_2\rightarrow e_3$: 0.9, then $e_1\rightarrow e_2\rightarrow e_4$ is anomalous
- Rules are created or deleted based on this inductive learning
- The accuracy dependent on the quality training set
- Other Approaches: Neural Network, Immune-based, Bio-based, Genetic approaches
Threshold-based Anomaly detection

- The relative frequency of the occurrence of activity $x$ is measured and the probability $P(x)$ is then calculated.
- Commonly used anomaly detector is the following inverse function:
  - $A(x) = -\log_2(P(x))$
- It is high for unexpected events and low for expected ones
- Examples
  - Combine dest IP and port. In the training phase we found out:
    - $x_1: P(10.0.0.1, 25) = 0.9$ and $x_2: P(10.0.0.1, 80) = 0.1$
    - Thus, $A(x_1)=0.15$ and $A(x_2)=3.32$
then $A(x)$ is expanded from 0 to infinity

see [http://en.wikipedia.org/wiki/Logarithm#The_logarithm_as_a_function](http://en.wikipedia.org/wiki/Logarithm#The_logarithm_as_a_function)

Ehab Al-Shaer, 4/30/2007

relative frequency here is calculate based the total activities. Eg. for pkts sent to port 25 for a specific machine:

$P = \text{number of pkt send to 25} / \text{total send to all ports}$

also

$p = \frac{\text{number of pkts to specific machine}}{\text{total number of pkts}}$

Ehab Al-Shaer, 2/26/2008
Why taking the log

\[
\log(x)
\]

\[1, 10\]
then $A(x)$ is expanded from 0 to infinity

see http://en.wikipedia.org/wiki/Logarithm#The_logarithm_as_a_function
Ehab Al-Shaer, 4/30/2007
Anomaly detection with two threshold

- L: lower threshold. U: upper threshold
- If $A(x) < L \Rightarrow$ normal activity; continue observing
- $U > A(x) > L \Rightarrow$ Suspicious activity; further investigation is required including temporary isolation and rehabilitation
- If $A(x) > U \Rightarrow$ anomalous activity; counter measure must be immediate (switch port deactivating)
SPADE: Statistical-based Anomaly Detection System

- **History**
  - SPADE was a DARPA funded project
  - DARPA ceased funding in 2003
  - Currently hosted by Computer Security
  - Online and BleedingSnort

- **What is it?**
  - It is a pre-processor plug-in for the Open Source IDS Snort
  - SPADE detects “anomalous” packets on the network
  - It is used in addition to a signature based IDS system
  - SNORT plugin -SPADE automatically detects stealthy port scans
**SPADE: How does it work?**

- SPADE reviews all packets that are received by Snort.
- It maintains a probability table for packets:
  - SPADE examines TCP-SYN packets and maintains the count of packets observed on (dest IP, dest Port) tuples.
  - SPADE checks the probability of every new packet on the (dest IP, dest Port) tuple.
  - The lower the probability, the higher the anomaly score.
- Tables are historically weighted:
  - More recent packets are more significant.
  - Older packets are phased out.
- It produces an “anomaly score” for each packet.
- Anomaly Scoring:
  - Raw: using $A(x)$ described before.
  - Relative: normalizing $A(x)$ by dividing it over the maximum one.
- SPADE uses Snort alarm and can also produce notifications of alert threshold adjustments.
Raw Anomaly Score (RawAS)

- \( P(\text{dip}=10.10.10.10, \text{dport}=80) = 10\% \)
- \( \text{RawAS} = -\log_2(P(X)) \)
- So for Packet Type 1 RawAS1 = 3.32
- \( P(\text{dip}=10.10.10.10, \text{dport}=79) = 0.1\% \)
- So for Packet Type 2 RawAS2 = 9.97
- \( \text{RelAS1} = \frac{3.32}{9.97} = 35\% \)
- \( \text{RelAS2} = \frac{9.97}{9.97} = 100\% \)
Relative Anomaly Score (RelAS)

- Simplifies use
- Divide the RawAS by the maximum RawAS
- Value between 0 and 1
- Using the RelAS
  - At any given time the sensor has a threshold set
  - Above get reported
  - Below get ignored
  - Problem - False Positive / False Negative
  - Can be tuned on the fly
SPADE Detectors

- Detectors are the internal SPADE components that look for an anomaly over a certain set of packets
- You can run any number simultaneously
- There are five types of detector
  - **closed-dport**
    - Traditional SPADE detector type
    - Looks for packets going to closed or infrequently used ports
    - Good for spotting port scans - legitimate traffic tends to head towards open ports
**SPADE Detectors**

**dead-dest**
- Looks for packets going to IP Addresses that are not in use
- Detects port scanning and network probes as legitimate traffic tends to go towards live hosts!

**odd-dport**
- Looks for unusual destination port usage
- Traffic from one host to another tends to stay within a small range of ports
- Attempts to connect to alternative ports could be indicative of a compromise of the source machine

**odd-port-dest**
- This looks for outgoing connections **from** a machine being made to unusual destination ports
- This hopefully would detect a compromise of an internal machine

**odd-typecode**
- Detects unusual ICMP type and code values
Adding My own SPADE Detectors

odd-Roc (our own)

• detecting DDOS based on Rate of change (RoC) of source IP address to a specific machine
• \( x = \frac{\text{#of New Src-IP}}{\text{#of distinct Src-IP address}} \) in \( W \) min
• Calculate the frequency of \( x \) (\( F_x \)) over \( n \) windows (\( n \times W \))
• Then \( P(x) = \frac{F_x}{n} \) (e.g., \( P(x=20)=0.5 \), \( P(x=300)=.01 \))
• Calculate the \( A(x) \) for each number
• set threshold (\( s \)) (low and max) based on RawA(\( x \)) or RelA(\( x \))
SPADE Detectors Examples

- Straight from the example spade.conf

preprocessor spade-detect: type=closed-dport tcpflags=synonly wait=3
preprocessor spade-detect: type=dead-dest tcpflags=weird wait=2
preprocessor spade-detect: type=odd-dport proto=tcp wait=2
preprocessor spade-detect: type=odd-typecode
preprocessor spade-detect: type=odd-port-dest proto=tcp Xdports=80

- Auto Thresholding
  - You can set thresholds manually.
  - There are four automatic facilities to help you get your threshold right.
  - Three of these adjust the threshold on the fly using weighted averaging
  - One adjuster per detector.
wait: the number of seconds a report that is held on the waiting queue will wait before timing out.

Ehab Al-Shaer, 4/30/2007
SPADE Limitations

- Drawback: SPADE raises false alarms on legitimate traffic for which \{dest ip, dest port\} combinations are infrequent
- Can’t say if a packet is Good or Bad
- Only can tell you how “Odd” it is
- Can’t look at packet contents (yet …)
- Won’t group events
Attack Graphs
Attack Graph Example

Figure 1: Example network model.

<table>
<thead>
<tr>
<th>Host</th>
<th>Vulnerability</th>
<th>CVE#</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTP Server 196.216.0.10</td>
<td>Ftp .rhost attack</td>
<td>1999-0547</td>
</tr>
<tr>
<td></td>
<td>Ftp Buffer overflow</td>
<td>2001-0755</td>
</tr>
<tr>
<td></td>
<td>Ssh Buffer overflow</td>
<td>2006-2421</td>
</tr>
<tr>
<td>SMTP Server 196.216.0.1</td>
<td>Ftp .rhost attack</td>
<td>1999-0547</td>
</tr>
<tr>
<td>Terminal 196.216.0.3</td>
<td>LICQ remote-2-user</td>
<td>2001-0439</td>
</tr>
<tr>
<td></td>
<td>“at” heap corruption</td>
<td>2002-0004</td>
</tr>
<tr>
<td>Data Server 196.216.0.2</td>
<td>LICQ remote-2-user</td>
<td>2001-0439</td>
</tr>
<tr>
<td></td>
<td>suid Buffer overflow</td>
<td>2001-1180</td>
</tr>
</tbody>
</table>

Table 1: Initial vulnerability per host in example network.
Figure 3: Attack tree of example network model.
Figure 2: Example attack tree.
Distributed IDS using Bayesian Methods

- Goal: track the attackers sequence of actions around the network and provide early detection
- Approach: using Bayesian multiple hypothesis tracking (BMHT) to classify intrusion detection system events into attack sequences reorganizing the collecting data from different sensors to detect intrusion
- Hypothesis generation
  - Each hypothesis consist of a set of tracks that map events measured (by sensors) to targets i.e, a series of events that describe the motion or the activities of an individual target
  - Each event appears exactly once (in each track)
Distributed IDS using Bayesian Methods

- **Hypothesis Likelihood Evaluation**
- **For multiple events (independent observations)**

\[
P(x \mid y) \propto L(y \mid x)P(x)
\]

\[
P(x \mid y_1, y_2) = L_2(y_2 \mid x)L_1(y_1 \mid x)P(x)
= L_2(y_2 \mid x)\frac{\alpha}{\beta}P(x \mid y_1)
\]

- And so on. This could be done recursively
- y is the set of sensor readings defining a track, and x the set of target states we believe to have caused these readings \(\Rightarrow\) i.e., \(p(y \mid x)\) is the misuse detection like prob of finding byte sequence XYZ for (given) virus W
- Example of x: rate of change in Dest IP, y: DOS, zombie collection .. goals
- When event is observed, it will be inserted in track that gives the highest hypothesis value. The valuation of the hypothesis is then calculated
- Basically this is representing the motion characteristics of the attackers
Distributed IDS using Bayesian Methods

<table>
<thead>
<tr>
<th></th>
<th>Denial of Service</th>
<th>Zombie Collection</th>
<th>Directed Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technique RoC</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Source IP RoC</strong></td>
<td>High</td>
<td>Low/None</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Dest. IP RoC</strong></td>
<td>Low/None</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Dest. Port RoC</strong></td>
<td>Unknown</td>
<td>Low/None</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Time Rate of Events</strong></td>
<td>Quick</td>
<td>Unknown</td>
<td>Unknown</td>
</tr>
<tr>
<td><strong>Type of Events</strong></td>
<td>DoS</td>
<td>Scan</td>
<td>Reconnaissance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote-Access</td>
<td>Scan</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Remote-Access</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Privilege Level</td>
</tr>
</tbody>
</table>

Table 2: Modeling Types of Attacker Behavior
ESS18

ROC-DoS (first row) = means DoS might occur and RoC is low but must be high when direct attack occur

Ehab Al-Shaer, 3/4/2008
IDS Performance Evaluation
Measuring IDS performances

- In order to compare different IDSs, a measure of their performances is needed.
- Of all the measurable characteristics mentioned before, the *true positive rate* and the *false positive rate* are the most important for comparing IDSs.
- The true positive rate and the false positive rate are included in various sublimation metrics for comparing IDSs.
Measuring IDS performances

<table>
<thead>
<tr>
<th></th>
<th>Intrusion</th>
<th>( \neg \text{Intrusion} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alarm</strong></td>
<td>True Positive</td>
<td>False Positive</td>
</tr>
<tr>
<td>( \neg \text{Alarm} )</td>
<td>False Negative</td>
<td>True Negative</td>
</tr>
</tbody>
</table>
Evaluating IDS

- Probability of false alarms
  - Suppose that we have $N$ IDS decisions, of which:
    - In $TP$ cases: intrusion – alarm.
    - In $TN$ cases: no intrusion – no alarm.
    - In $FP$ cases: no intrusion – alarm.
    - In $FN$ cases: intrusion – no alarm.
  - Total intrusions: $TP+FN$
  - Total no-intrusions: $FP+TN$
  - $N=TP+FN+FP+TN$
  - Base-rate – the probability of an attack:

$$P(I) = \frac{TP + FN}{N}$$
Testing IDS

- Probability of false alarms (cont.)
  - Events: Alarm $A$, Intrusion $I$
  - The following rates are defined:
    - True positive rate $TPR$
      \[
      TPR = \frac{TP}{TP + FN} = P(A|I)
      \]
    - True negative rate $TNR$
      \[
      TNR = \frac{TN}{FP + TN} = P(\neg A|\neg I)
      \]
Testing IDS

- Probability of false alarms (cont.)
  - False positive rate $FPR$
    \[
    FPR = \frac{FP}{FP + TN} = P(A|\neg I)
    \]
  - False negative rate $FNR$
    \[
    FNR = \frac{FN}{TP + FN} = P(\neg A|I)
    \]
Measuring IDS performances

- It is important to determine the probability of intrusion, if an alert has been generated.
- This gives rise to a Bayesian probabilistic measure for characterising IDS performances.
- We need the total probability of alert in order to determine the probability of intrusion given the alert.
Measuring IDS performances

$I, \neg I$ mutually exclusive $\Rightarrow A = (I \cap A) \cup (I \cap \neg A)$

$$P(A) = P(I)P(A|I) + P(I)P(A|\neg I)$$

Total probability of alert

Prob of True Alarms $\Rightarrow$ Prob of False Alarms
Measuring IDS performances

\[ P(A|I) = TPR = \frac{TP}{TP + FN} \]

\[ P(\neg I) = 1 - P(I) \]

\[ P(I) = \frac{TP + FN}{N} \]

\[ P(A|\neg I) = FPR = \frac{FP}{FP + TN} \]
Measuring IDS performances

- A performance measure: Bayesian detection rate:

\[
P(I \mid A) = \frac{P(I)P(A \mid I)}{P(I)P(A \mid I) + P(\neg I)P(A \mid \neg I)}
\]

- The greater the detection rate, the better the IDS, but...
- Conclusion: If the base-rate is low, the false alarm rate must be extremely low.
IDS Evaluation – Bayesian Detection Rate

• Base-rate fallacy
  • Even if false alarm rate $P(A \mid \neg I)$ is very low, Bayesian detection rate $P(I \mid A)$ is still low if base-rate $P(I)$ is low
  • E.g. if $P(A \mid I) = 1$, $P(A \mid \neg I) = 10^{-5}$, $P(I) = 2 \times 10^{-5}$, $P(I \mid A) = 66$

• Implications to IDS
  • Design algorithms to reduce false alarm rate
  • Deploy IDS to appropriate point/layer with sufficiently high base rate
Measuring IDS performances

• **Base-rate fallacy**
  
  • Even if false alarm rate $P(A|\neg I)$ is very low, Bayesian detection rate $P(I|A)$ is still low if base-rate $P(I)$ is low
  
  • Example 1: if $P(A|I) = 1$, $P(A|\neg I) = 10^{-5}$, $P(I) = 2\times10^{-5}$, $P(I|A) = 66\%$
  
  • Example 2: if $P(A|I) = 1$, $P(A|\neg I) = 10^{-5}$, $P(I) = 10^{-1}$, $P(I|A) = 99.99\%$
  
  • Example 3: if $P(A|I) = 1$, $P(A|\neg I) = 10^{-9}$, $P(I) = 2\times10^{-5}$, $P(I|A) = 99.99\%$
Base Rate Fallacy

- Bayes theorem:

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

- More generally:

\[ P(A|B) = \frac{P(A) \cdot P(B|A)}{\sum_{i=1}^{n} P(A_i) \cdot P(B|A_i)} \]
The base-rate fallacy is best described through example.² Suppose that your doctor performs a test that is 99% accurate, i.e. when the test was administered to a test population all of whom had the disease, 99% of the tests indicated disease, and likewise, when the test population was known to be 100% free of the disease, 99% of the test results were negative. Upon visiting your doctor to learn the results he tells you he has good news and bad news. The bad news is that indeed you tested positive for the disease. The good news however, is that out of the entire population the rate of incidence is only 1/10000, i.e. only 1 in 10000 people have this ailment. What, given this information, is the probability of you having the disease? The reader is encouraged to make a quick “guesstimate” of the answer at this point.
Base Rate Fallacy

\[ P(S|P) = \frac{P(S) \cdot P(P|S)}{P(S) \cdot P(P|S) + P(\neg S) \cdot P(P|\neg S)} \]

\[ P(S|P) = \frac{1/10000 \cdot 0.99}{1/10000 \cdot 0.99 + (1 - 1/10000) \cdot 0.01} \approx 1\% \]

- Even though the test is 99% certain, your chance of having the disease is 1/100, because the population of healthy people is much larger than sick people.
Base Rate Fallacy in Intrusion Detection

- I: intrusive behavior,
  ¬I: non-intrusive behavior
- A: alarm
  ¬A: no alarm

Detection rate (true positive rate): $P(A|I)$
False alarm rate: $P(A|\neg I)$
Goal is to maximize both
  - Bayesian detection rate, $P(I|A)$
  - $P(\neg I|\neg A)$
Detection Rate vs False Alarm Rate

\[ P(I|A) = \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)} \]

- Suppose:
  \[ P(I) = 1 - \frac{1 \cdot 10^6}{2 \cdot 10} = 2 \cdot 10^{-5}; \]
  \[ P(\neg I) = 1 - P(I) = 0.99998 \]

- Then: False alarm rate becomes more dominant if \( P(I) \) is very low

\[ P(I|A) = \frac{2 \cdot 10^{-5} \cdot P(A|I)}{2 \cdot 10^{-5} \cdot P(A|I) + 0.99998 \cdot P(A|\neg I)} \]
Detection Rate vs False Alarm Rate

We need a very low false alarm rate to achieve a reasonable Bayesian detection rate.
Measuring IDS performances

- Another performance measure: **ROC**
  - Receiver **Operating** Characteristic
  - Used widely in systems for detection of signals in noise (radars, etc.)
  - $TPR$ vs. $FPR$ curve
  - An ideal system has $TPR=1$ and $FPR=0$. 
Example ROC Curve

% False Alarm Rate

True Alarm Rate % (Detection Rate)

IDS1

IDS2

Worst case

- Ideal system should have 100% detection rate with 0% false alarm.
- If a parameter of an IDS is varied, the ROC curve is obtained, instead of a single point.
IDS2 has a max accuracy 75% but IDS1 can reach higher accuracy

but IDS1 false alarm increases sharply after 75%. the rate of FP increase for IDS1 is less than the rate of %detect increase which makes tuning IDS12 to tuning IDS1 to these values is undesirable

for IDS2, it is bad to choose the highest detect% bec it might give you unlimited number if false positive (fp)

Ehab, 5/31/2006
How to create ROC Curve for an IDS

Assume \#alarms = a, \#false negative = f, \#attacks occurred = o, \#detected attacks = d, t1, t2, \ldots tn are range of thresholds that can be used with an IDS.

- The false negative (F)% = \( f/a \times 100 \)
- Detection (D)% = \( d/o \times 100 \)
- For all threshold \( ti: 1 \rightarrow n \), calculate F and D and call them \( F_{ti} \) and \( D_{ti} \) for each IDS \( \rightarrow \) IDS RoC curve
- \( \text{RoC} [t_i - t_{i+x}] = dD/df = D_{t_{i+x}} - D_{t_i} / F_{t_{i+x}} - F_{t_i} \)
- You can find the avg RoC over different window of time in order to compare different IDSs
- To find out the optimal threshold, ti, you want to look for the maximum ti after which RoC is slow (e.g., < 1)
IDS2 has a max accuracy 75% but IDS1 can reach higher accuracy. But IDS1 false alarm increases sharply after 75%. The rate of FP increase for IDS1 is less than the rate of %detect increase which makes tuning IDS12 to tuning IDS1 to these values is undesirable.

Ehab Al-Shaer, 5/30/2006

- **Entropy Definition 1**: For a dataset $X$ where each data item belongs to a class $x \in C_x$, the entropy of $X$ relative to this $|C_x|$-wise classification is defined as (where $P(x)$ is the probability of $x$ in $X$):

$$H(X) = \sum_{x \in C_x} P(x) \log \frac{1}{P(x)}$$

- The entropy is smaller when the class distribution is skewer.
- Each *unique* data record represents a class, so the smaller the entropy the better for A-IDS.
Entropy – intuition

- Measure of impurity or regularity

\[ H(X) = 0 \quad \text{H(X) > 0 but low} \quad \text{H(X) \sim= 1} \]
Information-theoretic Measures for Anomaly Detection

- The fewer the number of different activities/records (higher redundancies), means low entropy $\Rightarrow$ good for IDS
- If the entropy is large, data is partitioned into more regular subsets evenly $\Rightarrow$ high entropy (bad for IDS)
- Any deviation from achieved entropy indicates potential intrusion
- Anomaly detector constructed on data with smaller entropy will be simpler and more accurate
Example: Detecting Anomaly based on IP source

- Finding the entropy (skewness) of IP sources distribution in connections to a server, $S$.
- Suppose that $a$ is the value of a specific IP source that connected to $S$, then $P(a)$ is as follows:
  - $P(a) = \frac{\text{Frequency}(a)}{\text{(Total # of connections)}}$
- Thus, if $H(IPS)$ is low then very few number of IP source values connect to this server which is good for A-IDS.

$$H(IPS) = \sum_{a \in \text{Value}(a)} P(a) \log \frac{1}{P(a)}$$
Statistical Optimization for Traffic-aware Adaptive Firewall Configurations

A Case Study
Internet Traffic Analysis - Observations

- Several packet traces from U of Auckland (NLANR) and DePaul U (sizes 3M to 10 M packets over different times)

About 20% of the flows (that carry about 60% of the total traffic) last 5 seconds or more,

About 15% of the flows (of 10 packets or more) carry about 70% of the total traffic.

Figure 1: CCDF distribution of flow size.

Figure 1: CCDF distribution of flow duration.
Locality of Matching Property

- **Skewness** of a field value is an indication of the high frequency of few values of a particular field compared with the frequency of others values in the traffic. Can be expressed as follows:

\[ S_f = 1 + \sum_{i=1}^{n} \frac{p_i \log p_i}{\log n} \]
The majority of traffic matches a small subset of field values in the policy

This “skewness” is likely to stay for sufficient time
ESS19

- there is a significant skewness in the source port and dest IP (>60% and 45%)

- this skewness is increasing over sampling

- the skewness is persistent over time (particularly for the src-port and dest ip)

Ehab Al-Shaer, 9/14/2005
Problems and Motivation: Statistical Network Security Policy Optimization

- Packet filtering is very critical for security devices
  - Enterprise Firewalls have 5K-25K rules has as reported by Cisco
  - IDS are expected to have in order of 10K (150K is not a surprise)
- Most of security devices still match rules sequentially; in this case the filtering cost:
  \[
  \text{Cost} = \sum_{i \in \text{matched}}^{n} \text{Rate}_i \times \text{Depth}_{R_i} + \sum_{j \notin \text{matched}}^{n} \text{Rate}_i \times \text{Policy Size}
  \]
- It is very possible for an attacker to target default-deny rule ➔ causing the max matching overhead
- Exploiting “locality of matching” in Internet traffic
Statistical Policies Optimization
Problems

- How can policy reconfigured dynamically to math packets based on real-time traffic properties such that average matching of accepted traffic is minimized? ➔ Statistical Traffic-aware Optimization
- How can security devices learn the traffic trend at real-time?
- How can this implemented and deployed?

[JSCA 06, JHSN 06, INFOCOM 05, INFOCOM 06, ASIACCS 06]
Optimizing the Accept Path: Statistical Tree Matching
Figure 1: Search tree for the destination port statistics in table: (a) binary tree, and (b) statistical tree.
Basic Idea
- Building a statistical search tree using the values of each field such that high frequent values comes first in the tree
- We use Alphabetic trees (vs. Hofmann tress) because it retains binary tree simplicity in building and searching
- Although the Alphabetic tree inserts values in the leave based on the weight (frequency), the inherent order is preserved like binary search è easy to search based on value
- The most frequent field values will exert less packet matching è more “skewness”, more performance gain
- Overall average filtering time of all flows is reduced
- In the worst case (uniform distribution of all the values), the Alphabetic tree can not be worst than Binary tree --unlikely

Ehab Al-Shaer, 8/23/2006
Search Aggregate in Alphabetic Tree Filtering

- Aggregate matching tree structure for: (a) Cascaded matching, (b) Parallel matching.
- Complexity: $O(n \log n)$ for Construction, $O(n)$ for space in cascade.
- In parallel, the intersected rules selected.
- Parallel tree might be faster but does not give less matches than cascade.
Internet traffic traces from NLANR and DePaul University backbone
Anonymized Policies
Random policy generation from traces
Evaluation of Early Rejection Technique

41% (optimal gain is 50% assuming RR has no overhead)

Early rejection (a) performance gain, (b) the number of RR for three polices with varying percentage of default rule traffic

<table>
<thead>
<tr>
<th>Policy</th>
<th>Number of Rules</th>
<th>Accepted Traffic (α)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy 1</td>
<td>1000</td>
<td>50%</td>
</tr>
<tr>
<td>Policy 2</td>
<td>1000</td>
<td>75%</td>
</tr>
<tr>
<td>Policy 3</td>
<td>200</td>
<td>50%</td>
</tr>
</tbody>
</table>
Optimization Effectiveness for Individual Fields

The reduction of packet matching relative to binary search for each filtering field on the firewall (a) inbound interface, (b) outbound interface. *(NOTE: reduction relative to Binary search)*
Optimization Effectiveness for Individual Fields – over 24 hours

Relative matching reduction for each field for different times of day
Average optimal and measured relative matching reduction with varying update interval for (a) Cascaded search, (b) Parallel search
Performance-trigger Update Intervals (during rush hour)

Relative matching reduction the source port during one hour interval
Continuing on
Information-theoretic Measures for Anomaly Detection of Sequence of Events
(EXTRA SLIDES)
Information-theoretic Measures for Anomaly Detection of Sequence of Events

- **Conditional Entropy Definition 2**: the conditional entropy of $X$ given $Y$ is the entropy of the probability distribution $P(x \mid y)$ that is

$$H(X \mid Y) = \sum_{x \in C_x, y \in C_y} P(x, y) \log \frac{1}{P(x \mid y)}$$

- Where $P(x, y)$ is the joint probability of $x$ and $y$ and $P(x \mid y)$ is the conditional prob of $x$ given $y$
- Measures the regularity in sequence of events; i.e. How regular these events comes sequential
- Conditional entropy $H(X \mid Y)$ tells how much uncertainty remains in sequence of events $X$ after we have seen subsequence $Y$ ($Y \in X$)
Relative Conditional Entropy

- Measures the distance of regularities between two audit datasets
- **Definition 3:** the relative entropy between two probability distributions \( p(x) \) and \( q(x) \) that are defined over the same \( x \) that belongs to \( C_x \) is

\[
realEntropy(p \mid q) = \sum_{x \in C_x} p(x) \log \frac{p(x)}{q(x)}
\]
Relative Conditional Entropy

**Definition 4:** the relative conditional entropy between two probability distributions $p(x \,|\, y)$ and $q(x \,|\, y)$ that are defined over the same $x$ that belongs to $Cx$ and $y$ belongs to $Cy$ is

$$\text{relConEnt}(p \,|\, q) = \sum_{x \in Cx, \, y \in Cy} p(x, y) \log \frac{p(x \,|\, y)}{q(x \,|\, y)}$$
Information Gain and Classification

- Definition 5: the information gain of attributes (i.e, features) $A$ on dataset $X$ is

$$Gain(X, A) = H(X) - \sum_{v \in \text{Values}(A)} \frac{|X_v|}{|X|} H(X_v)$$

Where $\text{Values}(A)$ is the set of possible values of $A$ and $X_v$ is the subset of $X$ where $A$ has value $v$

- The good feature is the one that its entropy values is low (causes less reduction in the total entropy) $\Rightarrow$ it has information gain
Information Gain and Classification

- In anomaly-based IDS (A-IDS), the higher information gain of the feature the better.
- When the regularity of the sequential dependencies is used directly in A-IDS, there is a direct connection between conditional entropy and information gain.
- Example, if 1..N-1 sequence of states represent feature A is observed, then the following shows how good A for predicting the state N.
- When we model a sequence, the smaller the entropy the higher the information gain and hence the better performance of the model.

\[ Gain(X, A) = H(X) - H(X | A) \]
Information Gain and Classification

- In analyzing network traffic or logs for anomaly detection classification, we should look for feature(s) that express strong regularity (low conditional entropy).
- Conditional entropy can be used in the feature construction process to suggest what features can be added so that the feature set contains information on both current and previous events.
- Example: \(<t, f_1, f_2, \ldots f_n, c>\) where \(t\) is the time stamp, \(f_i\) is a feature duration of connections, \# of bytes, \ldots etc and \(c\) is the label of this class.
  - Suppose we want to model how each service normally behaves.
  - If there is a strong regularity (low conditional entropy) on the sequence of services (or combination of service and other features), we can add these features that explain the strong regularity.
References

- www.snort.com
- SPADE white paper
- N. Einwechter, An Introduction to Distributed IDSs, 2001.
References

- www.snort.com
- SPADE white paper
- N. Einwechter, An Introduction to Distributed IDSs, 2001.