ANTIDOTE: Understanding and Defending against Poisoning of Anomaly Detectors\textsuperscript{1}

Vikram Ravindra

TU München

vikram.ravindra@tum.de

23rd April 2013

\textsuperscript{1}Based on an article with the same name, by Rubinstein et al.
Overview

1. Motivation
2. Scope of this talk
3. PCA based Anomaly Detection
   - Background
   - Method due to Lakhina et al
4. Poisoning Strategies
5. ANTIDOTE
6. Results
Motivation

- Statistical Machine Learning methods are employed to improve network security (email spam filtering, intrusion detection, enterprise network fault diagnosis etc.)
- For intrusion detection, these methods model the “normal” behaviour of network traffic.
- Deviations from the “normal” behaviour is classified as an “intrusion”
Motivation contd...

- Such methods are susceptible to adversaries
- Adversaries can poison the training phase, thereby affecting the decision taken during the test phase
- Adversaries can affect systems which employ SVM, neural network, PCA etc.
Scope of this talk

- a PCA-based subspace method for detecting anomalies in backbone networks
- The effects of poisoning of such a method
- Poisoning strategies/schemes
- ANTIDOTE
PCA based Anomaly Detection

Background

- Network traffic anomaly detectors mine network traffic matrices which specify the traffic volume between all pairs of Points-of-Presence (PoP) in a backbone network.
- It contains the collected traffic volume time series for each origin-destination (OD) flow.
- Traffic anomalies occur due to Denial of service (DoS), device failure, misconfigurations, etc.
- $N$ links
- $F$ OD flows
A is the routing matrix with dimensions $N \times F$.

$$A_{ij} = \begin{cases} 
1 & \text{flow } j \text{ contains link } i \\
0 & \text{otherwise}
\end{cases}$$

$X$ is the traffic matrix with dimensions $T \times F$ which describes the time series of all OD flows.

$Y$ of dimensions $T \times N$

$$Y = XA^T \quad (1)$$
An example

Consider 3 flows:
A → B → C
E → D → C
E → D → A → B
A =

\[
\begin{pmatrix}
1 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 1
\end{pmatrix}
\]
X =

\[
\begin{pmatrix}
4 & 4 & 7 \\
3 & 5 & 1 \\
0 & 7 & 3
\end{pmatrix}
\]

Y =

\[
\begin{pmatrix}
11 & 4 & 4 & 7 & 0 & 0 & 7 \\
4 & 3 & 5 & 1 & 0 & 0 & 1 \\
3 & 0 & 7 & 3 & 0 & 0 & 3
\end{pmatrix}
\]

Therefore, Y represents the amount of traffic on a given link at a given point in time.
PCA based Anomaly Detection
Method due to Lakhina et al

Observations

- High levels of traffic aggregation on ISP backbone links cause OD flow volume anomalies to often go unnoticed because they are buried within normal traffic patterns.

- Although the measured data has high dimensionality, \( N \), normal traffic patterns lie in a subspace of low dimension \( K \ll N \)
Therefore, one could adopt a dimensionality reduction technique like PCA. For an Abilene network, most variance is captured with $K = 4$. 
Principal Component Analysis

- PCA chooses $K$ orthogonal components to form a $K$ dimensional subspace.
- PCA is used to separate traffic measurements into normal and anomalous subspaces.
- The $K$ dimensional subspace spanned by the first $K$ components (corresponding to the normal/innocuous traffic) is $V_{1:K} = [v_1, v_2, ..., v_k]$
- The $(N - K)$ dimensional subspace spanned by the remaining components corresponds to abnormal traffic.
PCA based Anomaly Detection
Method due to Lakhina et al

Projections onto principal components showing normal and anomalous traffic variation.

- Examine the projection on each principal axis in order; as soon as a projection is found that exceeds the threshold, that principal axis and all subsequent axes are assigned to the anomalous subspace.
- All previous principal axes are assigned to innocuous/normal subspace.
PCA based Anomaly Detection
Method due to Lakhina et al

\[ y(t) = y_n(t) + y_a(t) \]  

(2)

where,

- \( y_n(t) \) is the normal traffic
- \( y_a(t) \) is the anomalous traffic

\[ c(y(t)) = \begin{cases} 
  \text{anomalous} & \|y_a(t)\|^2 > Q_\beta \\
  \text{normal} & \|y_a(t)\|^2 \leq Q_\beta 
\end{cases} \]
Poisoning Strategies

Adversary’s goals
- Launch a DoS on some victim
- Attack traffic must cross the network without detection

Overview of the strategy
- Add additional traffic called *chaff* over the given OD flow.
- Attack

The amount and the time period for which chaff is introduced depends on the adversary’s knowledge of the network.
- Uninformed Chaff Selection
- Locally Informed Chaff Selection
- Globally Informed Chaff Selection
In each scheme, the adversary decides on the quantity of $c_t$ chaff to add to the target flow time series at a time $t$.

Each strategy has an attack parameter $\theta$, which controls the intensity of the attack.

**Uninformed Chaff Selection**

At every time $t$, a decision is made as to whether or not chaff is to be added to the network based on the Bernoulli Random variable $c_t = \theta$

“Random” scheme, as it is independent of the network traffic
Locally Informed Chaff Selection

- the attacker knows the volume of traffic in the ingress link he controls, $y_s(t)$
- add chaff only when the existing traffic is already reasonably large
- $c_t = (\max(0, y_s - \alpha))^\theta$
Globally Informed Chaff Selection

- The adversary has complete knowledge of \( Y, A \) and future measurements \( \tilde{y}_t \)

- The adversary can introduce the chaff along any flow

\[
\begin{align*}
\max_{C \in \mathbb{R}^{TxF}} & \quad \| (\tilde{Y} + C)A_f \|_2 \\
\text{subject to} & \quad \| C \|_1 \leq \theta, \quad \forall t, \quad nC_{tn} \geq 0.
\end{align*}
\]

\[ C^{(0)} \propto \tilde{Y}A_fA_f^T \]

- Iteratively take a step towards the objective’s gradient and then project onto a feasible set.

- From the equation, it is clear that it makes sense to add chaff along the links \( A_f \) which is associated to a flow \( f \)
Poisoning Strategies

Boiling Frog Attacks

- Use any of the preceding schemes to ascertain the value of $c_t$
- Initialise $\theta$ to a small value and then increase it slowly over time
- PCA is retrained every week. Any anomalies detected is excluded from the training data
- The training data contains malicious events that were not detected by the previous detector
- Repeat this until the week of attack
ANTIDOTE = Laplace Threshold + PCA grid
PCA v/s PCA-Grid

- PCA and PCA grid aim to maximise a dispersion measure
- PCA uses standard deviation as the dispersion measure

\[ S^{SD} = \left( \frac{1}{n-1} \sum_{i=1}^{n} (r_i - \bar{r}) \right)^{1/2} \]

- To estimate the centre of the data, mean is used
- PCA grid uses median absolute deviation (MAD) as the dispersion measure

\[ S^{MAD}(r_1, ..., r_n) = \omega \cdot \text{median} |r_i - \text{median}r_j| \]

for scalars \( r_1, ..., r_n \)

- To estimate centre of the data, spatial mean is used.
Figure: PCA v/s PCA Grid
PCA grid

- 2D case: To find the best candidate between 2 unit vectors $a_1$ and $a_2$, the search space is given by a unit circle

$$a_\phi = \cos(\phi)a_1 + \sin(\phi)a_2$$

where, $\phi \epsilon [-\pi/2, \pi/2]$  

- The grid splits the domain $\phi$ into $Q + 1$ candidates $\phi_k = \frac{\pi}{2} \left( \frac{2k}{Q} - 1 \right)$ for $k = 0, ..., Q$.  

- The vector $a_{\phi_k}$ that maximises $S(Ya_{\phi_k})$ is the approximate maximiser $\hat{a}$  

- For a more general $N$ dimensional case, the search iteratively finds the best candidate $\hat{a}$ by performing a grid search between $\hat{a}$ and each unit vector $e_i$.  

- With each iteration, the range of angles progressively narrows around $\hat{a}$
Laplace Threshold

- The Q-statistic is the estimate for threshold in PCA.
- However, the residuals of PCA and PCA grid are empirically not normally distributed which makes Q-statistic a poor choice.
- The threshold for ANTIDOTE is Laplacian.
Performance Measures

- Evasion Success (FNR)
- DoS Detection Rate (TPR)
Results

Evasion Success of PCA v/s Antidote

Evasion Success of PCA v/s Antidote
ROC Curves for PCA v/s Antidote
Results

AUC PCA v/s Antidote

Single Poisoning Period: Flows’ AUCs at 10% Chaff

AUC PCA v/s Antidote
Results

Boiling Frog poisoning in Antidote

Boiling Frog Poisoning: Evading PCA

Boiling Frog Poisoning: Evading ANTIDOTE

Evasion success (average test FNR)

Growth rates
- 1.01
- 1.02
- 1.05
- 1.15

Attack duration (weeks)
References

Diagnosing Network-wide Traffic Anomalies
*proc. SIGCOMM.*

ANTIDOTE: Understanding and Defending against Poisoning of Anomaly Detectors
*proc. IMC.*
Thank You