To appear in the Proceedings of the 2009 ACM SIGMETRICS, June 2009

# Stealthy Poisoning Attacks on PCA-based Anomaly Detectors

Benjamin I. P. Rubinstein[1] Blaine Nelson[1] Ling Huang[2] Anthony D. Joseph[1,2]
Shing-hon Lau[1] Satish Rao[1] Nina Taft[2] J. D. Tygar[1]

[1]Computer Science Division, University of California, Berkeley
[2]Intel Research, Berkeley

## ABSTRACT

We consider systems that use PCA-based detectors obtained from a comprehensive view of the network's traffic to identify anomalies in backbone networks. To assess these detectors' susceptibility to adversaries wishing to evade detection, we present and evaluate short-term and long-term data poisoning schemes that trade-off between poisoning duration and the volume of traffic injected for poisoning. Stealthy *Boiling Frog* attacks significantly reduce chaff volume, while only moderately increasing poisoning duration. ROC curves provide a comprehensive analysis of PCA-based detection on contaminated data, and show that even small attacks can undermine this otherwise successful anomaly detector.

## Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; C.4 [**Performance of Systems**]: Modeling Techniques; I.2.6 [**Artificial Intelligence**]: Learning

## General Terms

Measurement, Performance, Security

## Keywords

Network Traffic Analysis, Principal Components Analysis, Adversarial Learning

## 1. INTRODUCTION

We explore vulnerabilities associated with using techniques based on Statistical Machine Learning (SML): specifically how adversaries can subvert the learning process [1]. Since SML is an increasingly popular tool for analyzing and improving network design and performance, it is important to understand the security of SML within the context of Internet measurement. In particular, SML plays an important role in *dynamic network anomography* [7]: the problem of inferring network-level Origin-Destination (OD) flow anomalies from aggregate network measurements. Network anomography techniques aggregate network measurements and employ various SML techniques [7] to diagnose network traffic anomalies. One popular technique [2] is based on Principal Components Analysis (PCA).

Consider a network with $N$ links and $F$ OD flows, which represent traffic over $T$ time intervals. Given $\mathbf{X}$, the $T \times F$ traffic matrix (TM) containing the time-series of all OD flows, a detector can simply flag flow $f$ as anomalous at time $t$ if $\mathbf{X}_{t,f}$ is large. However, the observed network link traffic represents the superposition of OD flows: the $T \times N$ link TM $\mathbf{Y}$ containing the traffic time-series of all links equals $\mathbf{X}\mathbf{A}^\top$, where $\mathbf{A}$ is the $N \times F$ routing matrix. Performing OD-flow anomaly detection given only $\mathbf{Y}$ is more difficult. PCA is one proposed method that works directly with $\mathbf{Y}$. PCA identifies the principal components of the link TM; i.e., the eigenvectors of its covariance matrix. The first $K \ll N$ (typically $3-4$) components model normal traffic: the *normal subspace* spanned by these vectors captures most of the variance between link TM rows. Second, we choose a threshold $Q_\beta > 0$. Link traffic that is more than $Q_\beta$ from the normal subspace is flagged as *anomalous*.

We previously showed that an adversary can generate OD traffic flow patterns that mislead this network anomography technique and lead it to miss anomalous traffic flows [5, 6]. We demonstrated data poisoning schemes that increase the variance along the links of a target flow during the training phase of the algorithm by strategically injecting additional high variance traffic along the flow; we refer to this adversarial traffic as *chaff*. The adversary subsequently launches a large-scale Denial of Service (DoS) attack that evades detection along the same flow. The attacker's goal is to increase the false negative rate (FNR; the percentage of anomalies that are undetected) of PCA to evade detection. The chaff is of low average volume, so poisoning is hard to detect.

Ringberg et al. showed that routing outages can pollute the normal subspace [4]; a kind of non-adversarial disturbance to the subspace. We expand on this work by quantifying PCA's sensitivity to *adversarial* contamination. Adversarial contamination can be much more subtle than incidental outages since attackers can adapt their attacks based on current network traffic levels.

## 2. RESULTS

As in our previous work [6], we train the PCA detector on an initial link TM and the learned principal components (PCs) are subsequently used for anomaly detection. Each week the detector relearns the PCs; i.e., the PCs used in any week $m$ are those learned in week $m-1$. During data poisoning, the attacker poisons flow $f$ by adding chaff of volume $c_t$ to $\mathbf{X}_{t,f}$, flow $f$ at time $t$. Once satisfied with the amount of poisoning, the attacker launches a DoS attack along flow $f$ corresponding to a large flow volume at one time. Following the validation methods of [2], we evaluate the efficacy of data poisoning by testing the poisoned detector on link data of known normality/abnormality. We validate PCA and our poisoning methods on data from the Internet2's Abilene network of 12 PoPs and 15 inter-PoP connections (comprising

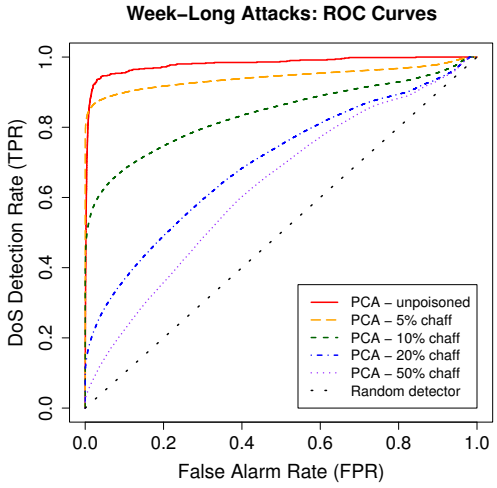


**Figure 1: ROC curves for PCA under *Week-Long* attacks that increase the volume of traffic by the indicated percent. The curves are obtained by averaging the curves from poisoning each flow separately.**

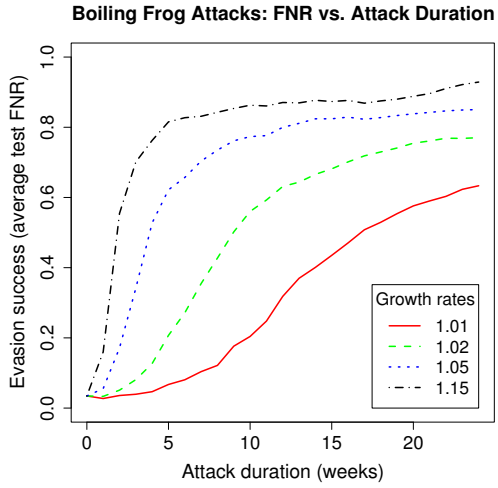


**Figure 2: *Boiling Frog* attacks using the *Add-More-If-Bigger* chaff method for four geometric growth rates $R$: the volume of traffic from week $m-1$ to week $m$ increases by a factor $R$ due to poisoning.**

144 OD-flows and 54 links) [7]. Here we present results for the *Add-More-If-Bigger* chaff selection scheme, which uses $c_t = (\max\{0, \mathbf{Y}_{t,S} - \alpha\})^{\theta}$ where $S$ is the ingress link of the target flow $f$, $\alpha$ is the attacker's estimate of the flow's mean, and $\theta$ controls the mean and variance of the chaff. We consider a semi-informed attacker capable of obtaining local information about the ingress link $S$ from network monitoring resources such as MRTG [3].

Our contributions in this work are twofold. First, we conduct a broader study of the impact of data poisoning on the performance of the PCA detector. Like the FNR, the detector's false positive rate (FPR; the percentage of normal traffic flagged as anomalous) is also affected by data contamination. Indeed our Receiver Operating Characteristic (ROC) curves in Figure 1 show that by increasing the traffic volume as little as 10%, poisoning undermines the detector's ability to successfully detect anomalies (high true positive rate (TPR)) without an intolerably high FPR. Further, when the increase in traffic volume due to chaff exceeds 20%, the PCA detector approaches the performance of a random detector.

Second, we show a new stealthy form of data poisoning that trades off the duration of poisoning with the volume of chaff. SML techniques are vulnerable because they often need to be retrained to capture evolving trends in changing data. In previous usage scenarios [2], the PCA detector is retrained regularly (e.g., weekly), allowing attackers to poison PCA slowly over long periods of time. By perturbing the principal components gradually, the attacker decreases the chance that the poisoning activity itself is detected. We design such an attack strategy, called a *Boiling Frog* attack. In the previous *Week-Long* attacks, the attacker must increase the volume on every link of the target flow by an average of 18% to increase the DoS's chance of successful evasion from detection from 4% to 50%. Under the *Boiling Frog* attack the same result can be achieved with a modest 5% volume increase from week-to-week over a 3 week period. Figure 2 shows the FNRs resulting from *Boiling Frog* attacks of increasing durations for each of four geometric growth rates in the amount of poison traffic used per week.

Future work will include investigation of *globally informed* poisoning methods, which will provide a standard with which to compare our locally informed and uninformed poisoning methods. We are also investigating several variants of robust PCA from the field of Robust Statistics as potential defenses against variance injection attacks.

## 3. REFERENCES


[1] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. "Can machine learning be secure?". In *Proc. ASIACCS '06*, 2006.

[2] A. Lakhina, M. Crovella, and C. Diot. "Diagnosing network-wide traffic anomalies". In *Proc. SIGCOMM '04*, pages 219–230, 2004.

[3] T. Oetiker. *The Multi Router Traffic Grapher*. `http://oss.oetiker.ch/mrtg/`, 2008.

[4] H. Ringberg, A. Soule, J. Rexford, and C. Diot. "Sensitivity of PCA for traffic anomaly detection". *Proc. SIGMETRICS '07*, 35(1):109–120, 2007.

[5] B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S. Lau, N. Taft, and D. Tygar. "Compromising PCA-based anomaly detectors for network-wide traffic". Technical Report No. UCB/EECS-2008-73, EECS Department, University of California, Berkeley, 2008.

[6] B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S. Lau, N. Taft, and J. D. Tygar. "Evading anomaly detection through variance injection attacks on PCA" (extended abstract). In *Recent Advances in Intrusion Detection*, volume 5230/2008 of *Lecture Notes in Computer Science*, pages 394–395, 2008.

[7] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. "Network anomography". In *Proc. IMC '05*, pages 1–14, NY, NY, USA, 2005.