Online Training and Sanitization of AD Systems

Extended Abstract

Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo
Department of Computer Science, Columbia University
{gcretu, locasto, sal}@cs.columbia.edu
Department of Computer Science, George Mason University
astavrou@gmu.edu

ABSTRACT

In this paper, we introduce novel techniques that enhance the training phase of Anomaly Detection (AD) sensors. Our aim is to both improve the detection performance and protect against attacks that target the training dataset. Our approach is two pronged: we employ a novel sanitization method for large training datasets that removes attacks and traffic artifacts by measuring their frequency and position inside the dataset. Furthermore, we extend the training phase in the spatial dimension to include model information from other collaborative systems. We demonstrate that by doing so we can protect all the participating systems against targeted training attacks.

Another aspect of our system is its ability to adapt and update the normality model when there is a shift in the nature of inspected traffic that reflects actual changes in the back-end servers. Such "on-line" training appears to be the "Achilles' heel" of AD sensors because they fail to adapt when there is a legitimate deviation in the traffic behavior, thereby flooding the operator with false positives. To counter that, we discuss the integration of what we call a *shadow sensor* with the AD system. This sensor complements our techniques by acting as an oracle to analyze and classify the resulting "suspect data" identified by the AD sensor. We show that our techniques can be applied to a wide range of unmodified AD sensors without incurring significant additional computational cost beyond the initial training phase.

1. INTRODUCTION

Recent research indicates that signature-based network intrusion detection systems (NIDS) are quickly becoming ineffective in identifying malicious traffic [4, 1, 5], especially that generated by polymorphic attack engines [5]. Relying on anomaly detection (AD) sensors to detect 0-day attacks has become a necessity rather than an option. Effective anomaly detection, however, requires highly accurate modeling of normal traffic — a process that remains an open problem [7] and the subject of this paper. Traditional approaches to modeling network traffic in this context typically measure only a few features based on a limited training data set and ignore the hard problem of vetting updates to the environment to distinguish between valid, sanctioned divergence from this initial measurement and invalid or attacker-driven changes. Ideally, an AD should achieve 100% detection accuracy, *i.e.,* true attacks are all identified, with 0% false positives. Reaching this ideal is very hard due to the following problems:

- The generated model can under-fit the actual normal traffic. Under-fitting for an AD system means that the AD system will erroneously flag traffic as "normal" leading to an overly generalized model. Attackers who have sufficient room to disguise their exploit as normal can bypass a poorly defined normality model, thus increasing the "false negatives" of the AD sensor.
- The model of normal traffic can over-fit the training data: non-attack traffic that is not observed during training can be regarded as anomalous. Over-fitting may generate an excessive amount of false alerts or "false positives."
- Unsupervised anomaly sensors often lack a measure of ground truth to compare to and verify against. The presence of an attack in the training data "poisons" the normal model, thus rendering the AD system incapable of detecting future or closely related instances of this attack. In short, the AD system may produce false negatives. This risk becomes a limiting factor of the size of the training set [6].
- Even in the presence of ground truth, creating a single model of normal traffic which includes all non-attack traffic can result in under-fitting and over generalization.

These problems appear to stem from a common source: the quality of the normality model that an AD system employs to detect abnormal traffic. This single and monolithic normality model is the product of a training phase that traditionally uses all traffic from a non-sanitized training data set. **The problem that we address in this paper is the difficulty of creating and maintaining robust normality models that can be used for efficient anomaly detection.** Applications range from single detectors operating on a local dataset to large systems distributed over space and/or time.

2. APPROACH

The first component that we have developed extends the AD training phase to successfully **sanitize training data** [2], while achieving both a high rate of detection and a low rate of false positives. Instead of using a normal model generated by a single AD sensor trained on a single large set of data, we use multiple AD instances trained on small data slices. Therefore, we produce multiple normal models, which we call *micro-models*, by training AD instances on small, disjoint subsets of the original dataset. Each of these micro-models represents a very localized view of the training data. Armed with the micro-models, we are now in a position to assess the quality of our training data and automatically detect and remove any attacks or abnormalities that should not be considered part of the normal model. Our approach shares elements with the ensemble method [3] because we also construct a set of classifiers and then classify the new data points using a (weighted) vote to decide. However, we modify the training phase by generating models from slices of the training data and also we lack the existence of a ground truth. The threat model for this approach consists of persistent and/or targeted attacks, or other anomalies that persist throughout the majority of the training set.

The next logical step is to develop methods for **sanitization of AD models**. Undertaking this type of sanitization becomes useful when additional information becomes available after the training phase has concluded. We apply model sanitization methods in a novel distributed strategy which leverages the location diversity of collaborating sites to exchange information that can be used to improve each site's model. Even if the identities of the collaborating sites become known, attacking all the sites with targeted or blending attacks is a challenging task: the attacker would have to generate mimicry attacks against all collaborators and blend the attack traffic using the individual sites' normal data models. Model sanitization techniques can also be applied in MANET environments: in such situations, trusted devices usually lack the resources necessary to build models from scratch (*i.e.*, from the training data). Sharing already-built models reduces the burden.

Another important aspect of anomaly detection models is that they have to reflect the dynamic changes that are exhibited in the system's behavior. We apply what we call a *progressive model update*; we employ this procedure when changes are caused by observable external factors that cannot otherwise be controlled (for example, a previously obscure Web page becomes popular). We propose to "frequently" update the sanitized model to reflect the dynamic changes in the network or users (including attackers) behavior. Updating the sanitized model implies updating the micro-models and applying the voting mechanism on the same training dataset as the one used for building the micro-models. We define this type of sanitization as introspective. We propose to age the older micro-models when new ones are built, but even so the use of micro-models might introduce a delay in the updating process. This delay occurs when there is a strong relationship between the mutation speed and the time granularity of the micro-models.

Finally, an important aspect of our work regards the efficiency of the resulting AD systems. Many papers comment on anomaly detectors having too high a false positive rate, thus making them less than ideal sensors. We see such comments as the **"*false* false positive problem."**. In this paper, we describe the use of a heavily instrumented host-based "shadow sensor": a system that behaves as an oracle for the AD sensor to accurately distinguish between alerts that represent false positive and those that represent real attacks. Such systems perform substantially slower than the uninstrumented application. As a result, we only wish to infrequently employ the services of the oracle so that most traffic enjoys a fast path of service; only traffic that is alerted on is sent to the oracle for confirmation of an attack. We focus on producing a sensor that identifies few "suspect data" items that are subjected to further but time-expensive tests. In this way, real attacks do not cause damage to the system under protection, and false positives do not flood an operational center with too many alarms. Instead, the shadow sensor processes both true attacks and incorrectly classified packets to validate whether a packet signifies a true attack. These packets are still processed by the intended shadowed application and only cause an increased delay for network traffic incorrectly deemed an attack.

Our experiments employ two content-based anomaly detectors Anagram [9] and Payl [8]. These AD sensors have very different learning algorithms. The experimental results indicate that the alerts generated when using the "sanitized" AD model represent a small fraction of the total traffic. The model detects approximately 5 times more attack packets than the original unsanitized AD model. In addition, the AD system can detect more threats both online and after an actual attack, since the AD training data are attack-free. In case the local sanitization is evaded, we extend our methodology to support sharing models of abnormal traffic among collaborating sites. A site can cross-sanitize its local training data based on the remote models. Our results show that, if the collaborating sites were targeted by the same attack and they were able to capture it in their abnormal models, the detection rate can be improved up to 100%.

3. REFERENCES

[1] Crandall, J. R., Su, Z., Wu, S. F., and Chong, F. T. On Deriving Unknown Vulnerabilities from Zero-Day Polymorphic and Metamorphic Worm Exploits. In *ACM Conference on Computer and Communications Security* (Alexandria, VA, 2005).

[2] Cretu, G. F., Stavrou, A., Stolfo, S. J., and Keromytis, A. D. Data Sanitization: Improving the Forensic Utility of Anomaly Detection Systems. In *Workshop on Hot Topics in System Dependability (HotDep)* (2007).

[3] Dietterich, T. G. Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science 1857* (2000), 1–15.

[4] Newsome, J., Karp, B., and Song, D. Polygraph: Automatically Generating Signatures for Polymorphic Worms. In *IEEE Security and Privacy* (Oakland, CA, 2005).

[5] Song, Y., Locasto, M. E., Stavrou, A., Keromytis, A. D., and Stolfo, S. J. On the infeasibility of Modeling Polymorphic Shellcode for Signature Detection. In *Columbia University Computer Science Department Technical Report, CUCS 007-07* (2007).

[6] Tan, K. M., and Maxion, R. A. Why 6? Defining the Operational Limits of stide, an Anomaly-Based Intrusion Detector. In *Proceedings of the IEEE Symposium on Security and Privacy* (May 2002), pp. 188–201.

[7] Taylor, C., and Gates, C. Challenging the Anomaly Detection Paradigm: A Provocative Discussion. In *Proceedings of the* 15^{th} *New Security Paradigms Workshop (NSPW)* (September 2006), pp. 21–29.

[8] Wang, K., Cretu, G., and Stolfo, S. J. Anomalous Payload-based Worm Detection and Signature Generation. In *Proceedings of the Symposium on Recent Advances in Intrusion Detection (RAID)* (September 2005).

[9] Wang, K., Parekh, J. J., and Stolfo, S. J. Anagram: A Content Anomaly Detector Resistant to Mimicry Attack. In *Proceedings of the Symposium on Recent Advances in Intrusion Detection (RAID)* (September 2006).