Using Prediction to Improve Network Intrusion Detection Performance

Sunny James Fugate
University of New Mexico
Department of Computer Science

Abstract

Signature-based and anomaly/behavioral detection offer complementary approaches with respect to precision and recall. My current research effort focuses on signature-based detection due to the need for significant expansion of Network Intrusion Detection System (NIDS) coverage while maintaining precision and improving performance. Modern NIDS offer precise detection of known threats but suffer poor recall and poor coverage of new threats and polymorphic variants.

My research focuses on three complementary techniques to achieving better coverage, performance, and scalability for NIDS: partitioning of large-scale decision procedures into semantic equivalence classes; prediction of equivalence class likelihoods based on known priors; and decision procedure assignment to in-situ information streams to improve performance. These refinements allow us to "bootstrap" and improve detector performance by (counter-intuitively) expanding IDS coverage. The predictor is then used to perform an an intelligent prioritization of future IDS rule applications which results in better performance per signature.

Introduction

My approach is inspired by biological cognitive systems which perceive objects within the world via mechanisms of predictive bias (Gregory 1994; Summerfield et al. 2006; Nowak and Hermsdörfer 2006; Norris and Kinoshita 2008) (e.g. masked priming, spreading activation, selective attention, sensory feedback, context effects, etc.). Without such bias, accurate and timely perception of a large number of objects is not a likely phenomenon. Such perceptual bootstrapping mechanisms enable the perceptual apparatus of almost any known organism to dwarf the capabilities of even our most sophisticated computing systems. It stands to reason that such organisms must reason abductively about almost everything that is perceived, leaping to conclusions first and only checking as time permits or necessity mandates. I believe that this biologically inspired approach has general applicability to any detection task which requires a large decision procedure that can be partitioned and which has sufficient structure in the temporal relationships of incoming data for accurate predictions of future events.

In its most general form the problem of optimizing NIDS performance is a resource allocation optimization. In the literature this has been done globally across the entire IDS configuration either as component of IDS tuning(Yu, Tsai, and Weigert 2008) or as a late optimization based on measured performance degradation(Lee et al. 2002). Central to my approach is the notion of "wasted information" in terms of mutual information between partitions of large decision procedures. Wasted information (information gain) is used to model the fitness of a predictively refined decision procedure. In this paper I will discuss the details of the approach, its ancillary benefits, and describe my progress. I will also discuss a prototype system and some initial results.

Problem

Whether or not existing Network Intrusion Detection System (NIDS) approaches adequately address current threats is a matter of debate. A review of existing literature and familiarity with the commercial capabilities suggests that existing network-based IDS approaches generally lack adequate coverage, have at best linear complexity scaling, and suffer from poor performance (inadequate to cover all possible exploits and polymorphic variants).

More precisely, we can define these three aspects of an IDS or similar decision procedure as follows:

- A *scalable* system grows in computing cost at a rate sublinear in respect to the growth of its input size.
- Alert *coverage* A, is the union of the sets of vulnerabilities $\mathcal{E}_v$, exploits $\mathcal{E}_x$, victim characteristics $\mathcal{C}_v$, and attacker characteristics $\mathcal{C}_x$ which are accurately identified (in respect to true positives and true negatives) by an IDS (i.e. $A = \mathcal{E}_v \cup \mathcal{E}_x \cup \mathcal{C}_v \cup \mathcal{C}_x$).
- *Performance* is defined using the conventional measures of precision, recall, accuracy, and specificity.

Ignoring differences in the cost of different types of signatures, current NIDS techniques require $O(n \cdot p)$ comparisons of p packets with n signatures. More state-full IDS (over TCP sessions for example) have an equivalent complexity. This is true even when clever algorithms are used to construct optimal decision trees over a set of feature signatures (Li and Ye 2001; Kruegel and Toth 2003). Decision tree search degenerates to exhaustive enumeration of the leaves of a much smaller binary tree. $O(p \log n)$ scaling is possible, but requires all branch feature values to participate in feature discrimination at each branch of the tree. If only single depth-first traversals are performed important alerts may be missed.

While $O(p \cdot n)$ is linear scaling and not immediately alarming, the number of threats appears to be increasing exponentially (Figure 1). The number of required signatures is growing as a function of the number of instances, versions, and vulnerabilities of deployed software applications, hardware instantiations, and protocols. As a result, without strong sub-linear scaling these systems are not sufficiently scalable. As an example, the Snort IDS ruleset is distributed with approximately 4,500 rules enabled with around 20,000 rules available. The number of potentially detectable threats is already large by comparison, ranging from 100s of thousands to millions of unique event types. The number of rules is kept small to achieve the best cost per performance ratio for a deployed IDS. Numerous statistical anomaly detection methods have also been created to achieve better scalability, but generally result in unacceptably high false-positive rates.

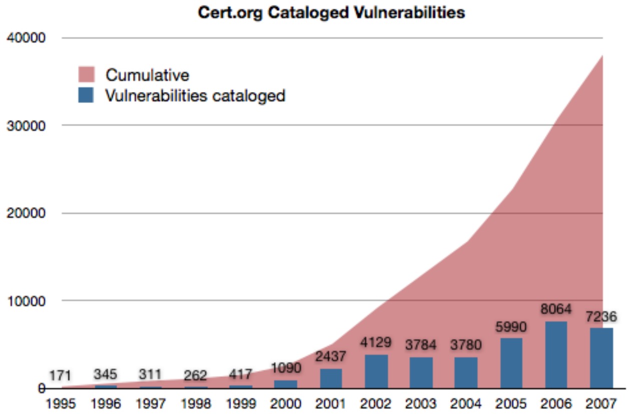

Figure 1: Cataloged vulnerabilities per calendar year (Carnegie Mellon CERT 2011)

IDS researchers, developers, and users need better performance, better coverage, and long-term scalability. It is my perspective that there is no need to trade performance for coverage. Instead, we can intelligently leverage increased IDS coverage to directly improve performance.

Research

It is the thesis of my research that in the context of large-scale detection tasks such as those of NIDS, predictive refinements to a decision procedure can enable an effective prioritization of signature applications, saving processing cycles which would otherwise be wasted. Current approaches are signature-limited. A conventional detection engine requires computing resources proportional to the total number of signatures. A cleverly designed predictive approach should decrease average computing cost per signature when new signatures provide "good" predictors over a set of semantic equivalence classes. More research and more formal analysis is needed to show that this is the case. It is clear, however, that such signatures will need to use the same set of features as the original ruleset. The intent is that the quality of each predictor should improve with increased coverage. A simple example would be a set of signatures which identify host operating system and provide the straight-forward prediction that only rulesets which are relevant to a previously detected operating system are relevant. Such an approach is bounded by the specificity and accuracy of the predictions. This proposed bootstrapping approach should require a diminishing average cost per additional signature and result in the desired sub-linear scaling.

Interestingly, there may also be several ancillary benefits to utilizing a predictive bootstrapping approach to detection. The approach should result in fault-tolerance to several forms of attack to the NIDS itself (e.g. flooding, denial-of-service, scanning, misdirection). This would result from the predictor "short-circuiting" the decision procedure to a specific equivalence class for attacks that utilize easily predicted repetitions and sequences. Additionally, the use of multiple independent predictors has the potential for increased robustness against NIDS attacks due to ready parallelizability of the resulting decision procedure. Each predictor operates over subunits of the global decision procedure. This decoupling should enable advantageous cache working set behavior, although these effects may be small. Finally, the prediction threshold can be dynamically varied to perform intelligent dynamic load-shedding. In conventional approaches these incidental beneficial properties are commonly dealt with using specialized (and often costly) preprocessing.

This research effort thus far has focused on signature-based detection, although the proposed approach may be equally suited for online optimization of other decision procedures, including statistical anomaly detection. The approach should apply equally well to those IDS (such as BroIDS) which detect events over a longer period and more state (e.g. TCP sessions instead of IP packets).

It is important to note that the proposed approach only applies to decision procedures which are sufficiently complex (and costly). Predictors which operate over trivial decisions (e.g. branch prediction) cannot be improved using this method. All possible equivalence classes are already represented. It is also necessary that a fraction of detected events are conditionally dependent on prior events. It is not necessary for events to be causally related, although causal relationships provide a more sound justification for decision prioritization.

The predictor can be constructed using a machine learning approach (e.g. learning a stochastic matrix) or constructed based on expert domain knowledge (e.g. attack graphs). The benefit the former is the automation of predictor construction and adaptation to new network environments. However, the latter would provide better explanations for the current state of the predictor. Determination of equivalence classes may be performed in a number of ways: k-means, hierarchical agglomerative clustering, or even imposing or extracting taxonomic relations from alert annotations.

Various temporal logic semantics can be used to learn the stochastic matrix, in particular Next and Finally. The most straightforward is the Next semantics calculated on a per-connection basis over IDS alerts. Figure 2 describes the

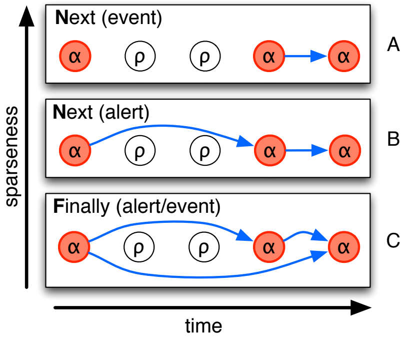

Figure 2: Temporal logic semantics over IDS events (ρ) and alerts (α)

differences between the temporal logic semantics being assessed in the context of IDS events and output alerts. The primary issue with the Next semantics is the sparseness of the resulting stochastic matrix. This semantics also assumes that being predicted as the next event is more important than predicting an event which will *eventually* happen. The benefit of the Next semantics is that the algorithm used to generate the predictor needs to retain much less state (i.e. at most a single reference to an event ID or equivalence class per connection tuple, $O(1)$, versus $O(p)$ per connection tuple). For reasonably sized datasets and offline learning this difference does not matter. The Next semantics over events (Figure 2 A) is expected to only capture predictions of the simplest types of noisy probes, scans, and denial-of-service. The Next semantics captures the instances where each packet (or event) in a sequence results in an IDS alert.

Progress

I am currently in the process of producing my initial research. I am looking for guidance from the community and searching for potential collaborators.

An initial architecture has been defined and a prototype is being constructed (Figure 3). The prototype as currently conceived performs prediction external to the IDS. This is intended to allows swapping the forward inferencing component (the IDS) and testing refinement of fundamentally different IDS engines (e.g. BroIDS, Snort, and one or more anomaly detection systems). The current design utilizes the experimental PF_RING kernel module for dynamic packet filtering and forwarding (Deri 2007). The PF_RING module filters and forwards packets to the appropriate IDS instances, of which there is one per equivalence class. I am in the process of evaluating the approach when applied to both the Snort and Bro intrusion detection systems utilizing both a private corporate data-set as well as the 1998 DARPA IDS Evaluation data-set (McHugh 2000; 2000).

In Figure 3 the decision procedure T maps packet features $\{\rho \in P\}$ to alerts $\{\alpha \in A\}$. The attack predictor G maps dependencies between attacker actions β and associated probabilities $p(\beta)$. The function $\Phi(\alpha)$ maps alerts to attacker actions and $\Gamma(\beta)$ maps predicted actions to equivalence classes of alerts $\xi = \{\alpha \in A | \alpha \sim \xi\}$.

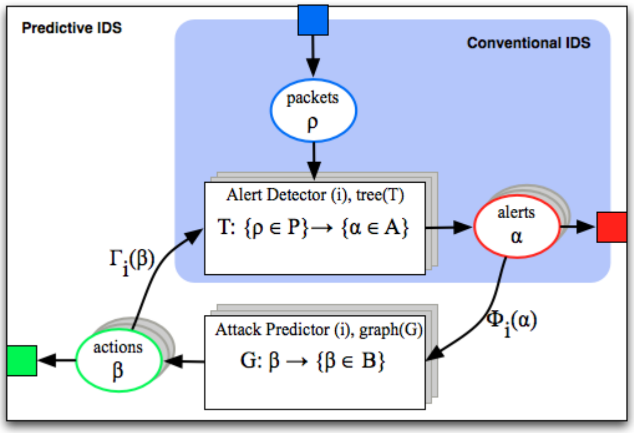

Figure 3: Predictive IDS architecture

The proposed dual-layer architecture implements a forward-inferencing decision procedure which is ”matched” with a Naive Bayes predictor. This entails learning a stochastic matrix of alert predictors from training data and then clustering alerts predictors (rows of the stochastic matrix) over the distributions of alert predictions.

Packets are directed to a particular IDS instance based on prior activity for the same “connection”, where a connection is defined as a tuple over IP address pairs, ports, protocol, and potentially other Layer 3 or Layer 4 features. The specific tuple which is most useful for prediction is under investigation but will most likely depend upon the type of alert and the network layer in which the attack is performed.

Table 1: High Occurrence Alert Sequences

sid → sid	Occurrences	Description
469 → 469	13997	ICMP PING NMAP → ICMP PING NMAP
1620 → 1620	478128	Non-Standard IP protocol → Non-Standard IP protocol
1620 → 13949	123893	Non-Standard IP protocol → Spoof of domain
1620 → 15934	48766	Non-Standard IP protocol → DNS for 172.16/12
1620 → 15935	13821	Non-Standard IP protocol → DNS for 192.168/16
13310 → 2925	17638	Apache DOS attempt → Web bug 1x1 gif attempt
13310 → 13310	138106	Apache DOS attempt → Apache DOS attempt
13948 → 1620	24901	DNS cache poisoning → Non-Standard IP protocol
13949 → 1620	109765	Spoof of domain → Non-Standard IP protocol
13949 → 13948	24888	Spoof of domain → DNS cache poisoning
15934 → 1620	54953	DNS for 172.16/12 → Non-Standard IP protocol
15935 → 1620	14923	DNS for 192.168/16 → Non-Standard IP protocol
	1,049,782	

Figure 4 shows a stochastic matrix generated from 360,591 alerts from Snort’s stateful preprocessing engines prior to clustering. The first column of this matrix represents events for which no subsequent event was seen. The first row represents events for which no prior event was seen. The diagonal represents events which predict sequences of identical events (e.g. scans, malformed packets, ICMP activity, statistical threshold violations, etc.). The built-in event generators for Snort represent many of these classes of events. For the purposes of my research, I will be ignoring these stateful detectors (and stream processors) and focusing on the non-stateful detection which represents the bulk of the signature set and computing cost.

Figure 5 represents a stochastic matrix produced over

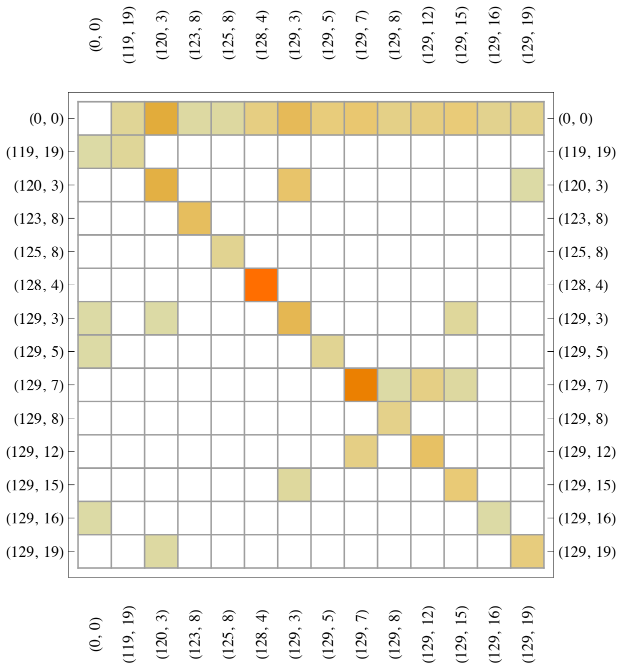

Figure 4: Example stochastic matrix generated from the DARPA 1998 training dataset and Snort's default alert generators (genid,sid) and the Next temporal semantics. The y-axis are event priors and the x-axis the consequents. The first row and column are events with no priors and events with no consequents respectively.

the entire DARPA 1998 training dataset. This dataset represents approximate 3GB of capture packet data (13,620,149 packets) which results in 1,096,099 packet-level Snort alerts using a recent release of the Snort VRT ruleset (ignoring stream and specialized preprocessor alerts)(SourceFire 2011). Of particular interest are the events which show a high correlation with future alerts for the same connection tuple. Table 1 describes all sequences which occur more than 10,000 times.

These events account for 95% of the alerts. At least for the DARPA dataset, we have a small set of superb predictors which account for almost the entire set of alerts and predict temporal correlations with relatively high degree of confidence. The extent to which the predictor events cover the packet events also gives the upper bound on performance speedups when each distinct signature is an equivalence class. For the DARPA dataset, with perfect predictors this would result in at best $\frac{1049782}{13620149} = 7.7\%$ of the events being predicted and detected with a $O(1)$ computing cost. This is consistent with an identical set of tests run against a 2GB sample of a real-world corporate dataset.

It is interesting to note that there are a large number of symmetries in the stochastic matrix. These symmetries account for 50% of the correlations in the DARPA testing dataset and 62% of the correlations in the corporate sample dataset. For the sample dataset the significantly higher symmetry is most likely due to the short time-frame (6 minutes) over which the sample extends. These symmetries may also represent an artifact of the SnortIDS processor or rulesets. Further analysis is needed to determine the underlying meaning of the symmetries.

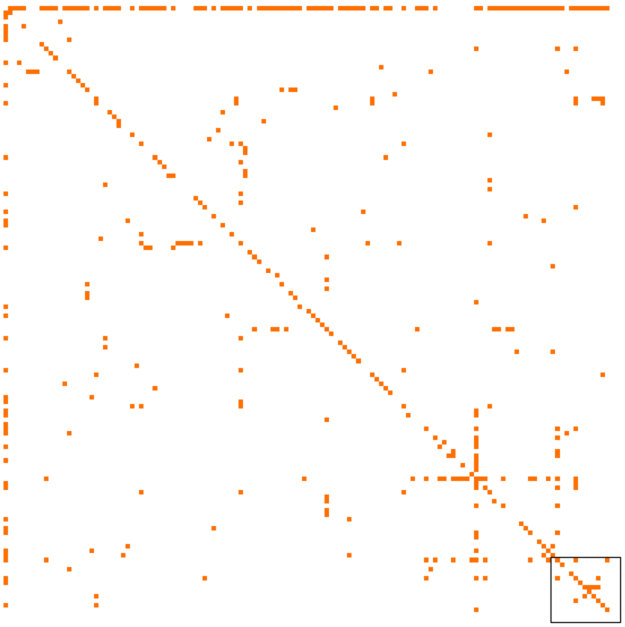

Figure 5: Stochastic matrix generated from the DARPA 1998 training dataset and the entire set of available rules (genid,sid) and the Next temporal semantics. For clarity, this plot is shown using using a fixed value for all events matching the temporal constraint. The bottom right box is the same as that in Figure 4

The sample dataset also exhibits similar super-exponential event frequency distribution (Figure 6). This implies that even for the case where the ruleset is kept constant that significant gains can be achieved by utilizing trivial predictors over small sets of noisy alerts.

The relatively small proportion of alerts to packets also elucidates one of the primary performance issues with these types of detection systems and a thesis of this work. Over 90% of the information gained in using the decision procedure against incoming packet data is discarded. Since no alert fires, any features extracted (either real or potential) cannot be used for future optimizations. Each packet passes through the decision procedure. If new rules were added to provide better predictors over the set of packets *not* associated with an alert, then the performance gains which might be achievable span the entire dataset.

Future Work

While my initial analysis is promising I have not yet demonstrated the principal thesis of the work: that these types of systems can be optimized using prediction such that adding rules improves overall performance. I hope to have initial performance results using the rudimentary prototype prior to the student conference. This summer I plan to extend the approach to see or improve predictive performance gains in several ways:

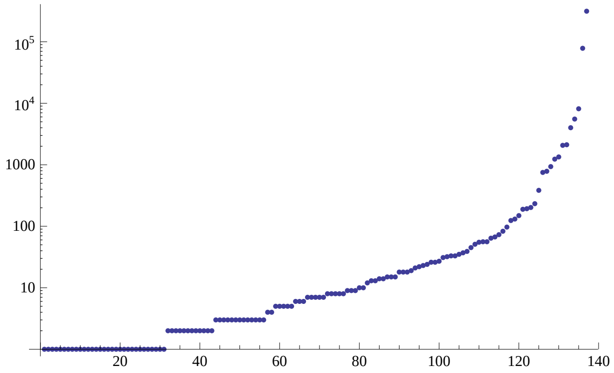

Figure 6: Marginal frequency histogram of event priors of a 2GB corporate dataset.

- A better theoretical presentation of the concept and results
- Increasing the overall coverage of the IDS
- Including signatures which identify exposure of host characteristics
- Exploring clustering methods for determining useful equivalence classes
- Exploring more robust predictors and ensemble methods
- Exploring alternative approaches as elucidated by interactions with the broader research community
- Demonstration of sub-linear scaling at the same performance points

References

CMU CERT 2011. Cert statistics (historical). [data tables]. Retrieved from http://www.cert.org/stats on Feb 5, 2011.

Deri, L. 2007. High-speed dynamic packet filtering. *Journal of Network and Systems Management* 15(3):401–415.

Dreger, H.; Feldmann, A.; Paxson, V.; and Sommer, R. 2008. Predicting the resource consumption of network intrusion detection systems. In *In Proceedings of Recent Advances in Intrusion Detection*, 135–154.

Gregory, R. 1994. Seeing as thinking: An active theory of perception. In Gibson, E., ed., *An Odyssey in Learning and Perception*. MIT Press.

Hartendorp, M. O.; Van der Stigchel, S.; Burnett, H.; Jellema, T.; and Postm, A. 2008. The influence of priming on the interpretation of an ambiguous figure. *Perception* 120.

Kruegel, C., and Toth, T. 2003. Using decision trees to improve signature-based intrusion detection. *Lecture Notes in Computer Science* 2820:173–191.

Lee, W.; Fan, W.; Miller, M.; Stolfo, S.; and Zadok, E. 2002. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security* 10:5–22.

Li, X., and Ye, N. 2001. Decision tree classifiers for computer intrusion detection. *Parallel and Distributed Computing Practices* 4(2):179–190.

McHugh, J. 2000. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security . . .* 3(4):262–294.

McHugh, J. 2000. The 1998 lincoln laboratory ids evaluation. volume 1907, 145–161.

Norris, D., and Kinoshita, S. 2008. Perception as evidence accumulation and Bayesian inference: Insights from masked priming. *Journal of Experimental Psychology: General* 137(3):434–455.

Nowak, D. A., and Hermsdörfer, J. 2006. Predictive and reactive control of grasping forces: on the role of the basal ganglia and sensory feedback. *Experimental Brain Research* 173(4):650–660.

SourceFire 2011. SourceFire VRT Ruleset Version 2.9.0.3. [configuration files]. Retrieved from http://www.snort.org/downloads/846 on March 14, 2011.

Summerfield, C.; Egner, T.; Greene, M.; Koechlin, E.; Mangels, J.; and Hirsch, J. 2006. Predictive Codes for Forthcoming Perception in the Frontal Cortex. *Science* 314(5803):1311–1314.

Yu, Z.; Tsai, J.; and Weigert, T. 2008. An adaptive automatically tuning intrusion detection system. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 3(3).