Experimental Results of Cross-Site Exchange of Web Content Anomaly Detector Alerts

Nathaniel Boggs[1], Sharath Hiremagalore[2], Angelos Stavrou[2], Salvatore J. Stolfo[1]
[1]Department of Computer Science, Columbia University
{boggs,sal}@cs.columbia.edu
[2]Department of Computer Science George Mason University
{shiremag,astavrou}@gmu.edu

Abstract

We present our initial experimental findings from the collaborative deployment of network Anomaly Detection (AD) sensors. Our system examines the ingress http traffic and correlates AD alerts from two administratively disjoint domains: Columbia University and George Mason University. We show that, by exchanging packet content alerts between the two sites, we can achieve zero-day attack detection capabilities with a relatively small number of false positives. Furthermore, we empirically demonstrate that the vast majority of common abnormal data represent attack vectors rather than false positives. We posit that cross-site collaboration enables the automated detection of common abnormal data which are likely to ferret out zero-day attacks with high accuracy and minimal human intervention.

I. Introduction

Automated zero-day and polymorphic attacks pose a critical widespread threat to web servers. Network-based Anomaly Detection (AD) is regarded as a potential defense to this very difficult to identify threat. However, AD sensors often suffer excessive false positive rates that require an unacceptable amount of human effort to properly resolve. When an alert is generated, the operator does not have a well defined "attack signature" to analyze – she must drill down to packet content in order to understand the nature and validity of the alert. Therefore, to distinguish a true attack from a legitimate but anomalous packet, operators have to manually sift through the alerts and offending packet content. On the other hand, any attempt to reduce the rate of false positives by modifying the sensitivity of the AD sensor may reduce the true positive detection rate.

In this paper, we present results using a working alert exchange architecture with an extensive section on model comparisons over an extended period of time. Currently, to limit the false positives, AD sensor outputs are typically correlated with other evidence to distinguish true attacks from false alarms. For instance, see shadow servers in [1]. *We propose a large scale network of AD sensors distributed across disjoint domains that exchange and correlate web server content alerts to identify widespread zero-day attacks in real time.* This network would analyze ingress traffic to some sample of collaborating web servers. Sensors could be deployed at each domain or at a common peering point. The zero-day attacks found could then be communicated broadly.

The results from an initial deployment of the system across two administrative domains on the Internet, support the feasibility and accuracy of such a system. Additionally, we present a method of comparing normal models between sites to potentially identify sites with distinct normal traffic flows. We conjecture that each administrative domain may detect zero-day attack vectors as abnormal content since, by definition, zero-day attacks are data delivered to a service that have not been seen before and are not contained in a signature database. Each site will also will classify some legitimate traffic as abnormal thus generating false positives. However, it is likely that this traffic will only be seen at a single site: errors of this nature are not likely to be similar at different domains because normal traffic flows will be different. The same (or nearly the

same) abnormal packet content seen at two or more sites is most likely a widespread attack vector rather than a false positive. Hence, correlating abnormal data across two or more sites in real-time may detect and accurately identify zero-day attacks. In presented results, we manually confirm the attack vectors in our correlated alerts. The number of zero-day attacks specifically depends on which signature engine the attacks are compared to. Furthermore, we claim that real-time filtering of zero-day attacks against web servers is feasible with essentially no human intervention by automatically filtering common abnormal content.

The strategy to correlate common abnormal content will detect zero-day attacks that do not use sophisticated polymorphic engines. In the case of polymorphic attacks, where each infection produces an entirely new version of the attack for each propagation attempt [2], it is unlikely this cross-domain correlation strategy will work. One should not expect to see any common attack vectors. In those cases, correlating AD alerts with host-based instrumented shadow servers is a likely better strategy to detect zero-days and reduce false positives. However, as it now stands today, most web-based attacks we have detected deliver their payload as relatively short PHP arguments, and do not contain polymorphic attack engines nor have we seen code attempting to download polymorphic variants. Hence, we posit that the cross-domain correlation strategy is effective against the large class of zero-day attacks targeting web-based applications and services.

To validate our claims, we study the outcome of two weeks of real network data capture and an automated exchange of AD alerts between Columbia University and George Mason University over the Internet. Our empirical results confirm our theory: the more distinct each normal model may be, the more likely common AD alerts will identify and filter true zero-day attacks. Indeed, by comparing the normal models from different domains we establish that each site has a distinct model of normal content. Moreover, throughout the two week study, we found 11787 common alerts. Furthermore, we analyze the time between each site first detecting each attack and verify that real time exchange is feasible. With this baseline of common content-based web server Anomaly Detector alerts comprised almost entirely of attacks, the few false positives can be quickly identified and shared to reduce the human workload. These experimental results support our conjecture that cross-domain content-based AD correlation deployed at a large-scale could effectively detect and mitigate zero-day web attacks.

II. Related Work

Previous work on distributed intrusion detection has focused mainly on the exchange of data within a single organization. Much of the early work, *e.g.,* [3], [4] focused on limited distribution within an enclave. In [5], the authors discuss methods for cooperatively correlating alerts from different types of intrusion detection systems. Krugel *et al.* [6], [7] concluded that only a relatively small number of messages (seldom more than two) need to be exchanged to determine an attack is in progress, making decentralized intrusion detection feasible and appropriate. DShield [8] is the most active volunteer-based DIDS project on the Internet that we are aware of, focusing on "top 10"-style reports and blacklists; however, it uses a centralized model, is reliant on reports from volunteers, and generally scrubs data. In [9], [10] the authors describe more general mechanisms for node "cooperation" during attacks.

DOMINO [11] is probably the closest decentralized framework in scope to ours. The paper measured, using DShield alert logs, the notion of information gain, however, DOMINO does not incorporate alerts generated by Anomaly Detectors. In addition, Farroukh *et al.* proposed a distributed and collaborative intrusion detection system called DaCID [12] based on the Dempster Shafer theory of evidence of fusing data. Additionally, in [13] the authors used a decentralized analyzer. Tian *et al.* introduced an alert correlation model based on hierarchical architecture [14].

The AD system we employ is based on STAND by Cretu *et al.*, [1] a derivative of an earlier system call Worminator [15]. A collaborative technique where the sites exchange abnormal models to im-

prove detection was presented in [1]. In [16], the authors completely automated the process of determining the optimal AD sensors parameters for a single sensor. As user interactions with systems change overtime, the current model becomes stale and may incorrectly classify new traffic patterns [17]. However, in all the above systems, the authors did not explore the benefits and caveats of exchanging anomaly detector content alerts. This paper provides the novel contribution that cross-domain AD alert exchange can identify zero-day attack vectors.

III. Experiment Architecture

We call our complete system AutoSense, which consists of an expanded Worminator [15] alert exchange and storage system integrated with deployed local AD sensors to exchange alerts and abnormal models in real time. Using a client-server architecture, each administrative domain has a Worminator client install that receives the raw alerts and models from a sensor. Each client then encodes alerts by inserting the n-grams of the content into bloom filters [18] as needed before sending the alerts and models to the server over an encrypted channel. On the Worminator server alerts and models are stored in a database. Separate threads perform correlation on the stored alerts continuously matching all non-private content from the host domain to all bloom filter representations of private alerts.

We use the combination of STAND [16], an automated training and sanitizing process with the Anagram sensor [19] as our network based Anomaly Detection sensors. We have sensors deployed at Columbia University and George Mason University inspecting inbound network traffic. To allow a proper training period and still have time for the data exchange to reveal common alerts, we collected traffic over the period of 2 weeks. The sensors' automated training process requires around 50-60 hours. For testing, we ran our sensors on TCP port 80 traffic inbound to two web servers: *www.cs.columbia.edu* and *www.gmu.edu*. For the collection, aggregation, and correlation of the HTTP data, we used VMWare virtual machines running Ubuntu Server 9 64bit. Each virtual machine was equipped with 16GB ram and 2-4 CPUs. The AD sensors are designed to sample a subset of packet traffic by parsing and normalizing TCP port 80 GET request URIs. In order to reduce variability, we strip the URI down to just the string of arguments and then remove numbers and decode hex characters. The resulting normalized argument strings that are less than 17 characters are dropped as attack vectors are much longer. This subset of packet content allows our machines to still see a wide range of potential attacks as numerous applications have a web service front-end. For each alert we log the IP address, timestamp, and content string. After each site ran the sensor on their two weeks of data from the same time period, we correlated the resulting alert content strings.

IV. Model Comparison

We theorize that correlating AD content alerts between sites with distinct normal traffic flows will reduce the false positive rate since legitimate requests will be less likely to be similar. For a large scale system, we will want to minimize the alert comparisons between similar sites to prevent more false positives. Therefore, a method to quickly compare the similarity of normal traffic between sites will be vitally important. In this section, we use each site's normal model as an approximation of its normal traffic flow for cross-site comparisons.

In order to explain our model comparison results, here is a brief explanation of the model generation process. For a complete description please see previous work on STAND [1]. STAND has a sanitizing process where small micromodels are continuously created on small sets data once little new traffic is seen, generally around 3 hours worth depending on data volatility. Then after 25 of these micromodels are created, they use a voting process to test all the data the micromodels used. All the data is then voted on by micromodels. Data passing the vote is added to a bloom filter to create a, now sanitized, normal Anagram model. Once the first sanitized model is created then each time a new micromodel is made the process is repeated. The new micromodel replaces the oldest one so that the latest 25 micromodels are always used to create a new sanitized Anagram model.

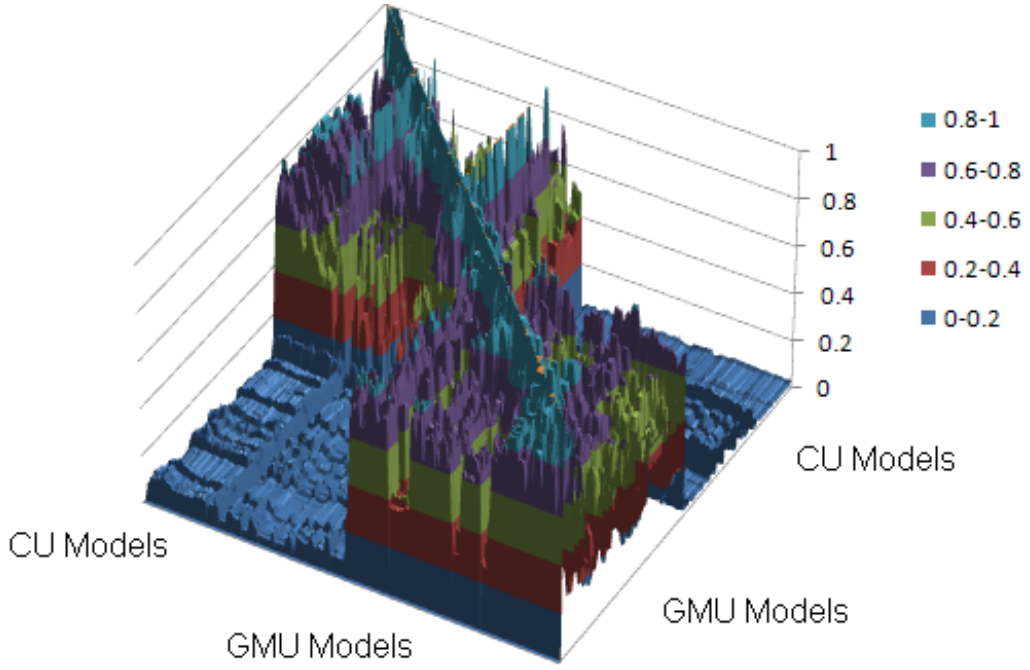

Fig. 1. Normal Models Compared across Site and Time

Each Anagram model is a bloom filter with a 2^{28} long bit array initialized to 0s. Data is added to the model by hashing n-grams from the data into the bit array setting some bits to 1. While not an exact measurement, we believe that since the same hash functions are used for all models that directly comparing the bits that are set on in each bloom filter gives a general idea of how similar two models are. In all our comparisons we find the number of 1 bits that both bloom filters share and divide by the total number of 1 bits in the bloom filter. This comparison operation C is represented below:
Let B1, B2 be bloom filters from Anagram models with bits $\{1...i...2^{28}\}$.
C(B1,B2) = (Number of bits i such that $B1[i] = 1$ and $B2[i] = 1$) / (Number of bits i such that $B1[i] = 1$)

Each AD sensor computes models of normal data for consecutive epochs at each site, producing a set of time-ordered models. Fig. 1 displays each individual normal model compared against each other normal model, both those from the same and collaborating site. The top quadrant shows Columbia University models compared to each other, while the bottom quadrant shows George Mason University models compared to each other. The models are placed in time order from top to bottom. Notice that the content flows at each site slowly change over time. The comparison of the time-ordered models computed at the same site shared 49% of set bits. Furthermore, the left side of the graphic in Fig. 1 shows George Mason University models compared to Columbia University models and the right side shows Columbia University models compared to George Mason University models. Notice that the content flows at each site are truly distinct. The important lesson is that the CU and GMU anomaly detectors have learned entirely different "normal" models at each site.

With an average of 5% rate of common bits set between models from separate sites, we show that both sites are quite diverse. This confirms the intuition that distinct sites have radically different models of normal data and supports the ability of Anomaly Detectors to recognize attack traffic from distinct domains. With distinct models, traffic

detected as abnormal at separate sites is much more likely to be an attack. Attackers attempting to put together a mimicry attack against multiple sites likely face an impossible task. They would have to target it specifically to one site since the sites have diverse normal content flows.

V. Alert Correlation

The initial correlation process began with 41,232 alerts observed at Columbia University and 20,678 at George Mason University. We compare each local alert to each bloom filter encoded alert from the remote site until we find a match. To account for simple polymorphism that could exist in the alerts, we consider a match to be 80% of n-grams from the local alert being present in the bloom filter and the alert lengths to be within 80% of each other. We found 11787 common alerts, 7989 at Columbia University and 3798 at George Mason University.

We confirm our online results from the bloom filter comparisons with an offline study using the Levenshtein string distance [20]. We normalized this distance by the length of the longer string to find equivalent alerts. The Levenshtein algorithm is a simple and effective way to correlate content alerts offline. For more information on suitable algorithms for content correlation see [21]. After testing, we set the threshold for matching at 0.2. This means that we allow for up to 2 changes in every 10 characters. In addition, the unique sets of alert content strings from each site were clustered using the normalized Levenshtein distance to obtain the common alerts. This correlation results in 96 common content alert string clusters representing 12353 total alerts, 8570 at Columbia University and 3783 at George Mason University. Using these two different methods and seeing similar results confirms that encoding the alerts in bloom filters still allows for accurate correlation.

Using manual inspection, we see that all but 4 of the attack clusters are indeed true attacks in both correlation methods. An Internet Explorer automated browser request related to an office toolbar made up 2554 alerts. This alert is caused by an IE browser extension and will likely be seen as an anomalous request by all web servers. This distinct but highly repetitive alert should be identified as a false positive once upon its first occurrence, and subsequently ignored by simply filtering it as "noise." Since it is so common, after filtering this specific alert the remaining false positive rate drops precipitously. Three additional clusters of false positives produces a net false positive rate of 0.69% out of common alerts and $1.2{*}10^{-5}\%$ (68 of 549 millions packets) out of total incoming packets to the web servers. For this dataset, covering a 2 week period, a human operator would have had to manually inspect 96 clusters to identify the 3 false positives. This is strong evidence that cross-domain alert exchange is a valuable security measure with minimal amounts of human interaction needed. Out of the 3 false positives, two are iframe tags and the other is related to a twitter feed. This makes intuitive sense as one of the only ways legitimate traffic would appear at both sites would be a generic request with never-before-seen parameters. Nevertheless, manual inspection of a false positive cluster from one operator could save all other sites from having to identify the alert. We believe that other false positives will most likely also be fairly generic mistakes. Therefore, the limited number of these generic argument strings will be identified as false positives, and then any future results will have an even lower low false positive rate.

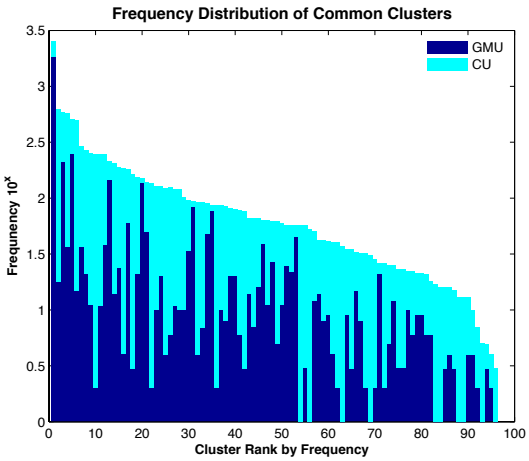

Fig. 3. Frequency of Common Alerts - Normalized Levenshtein distance

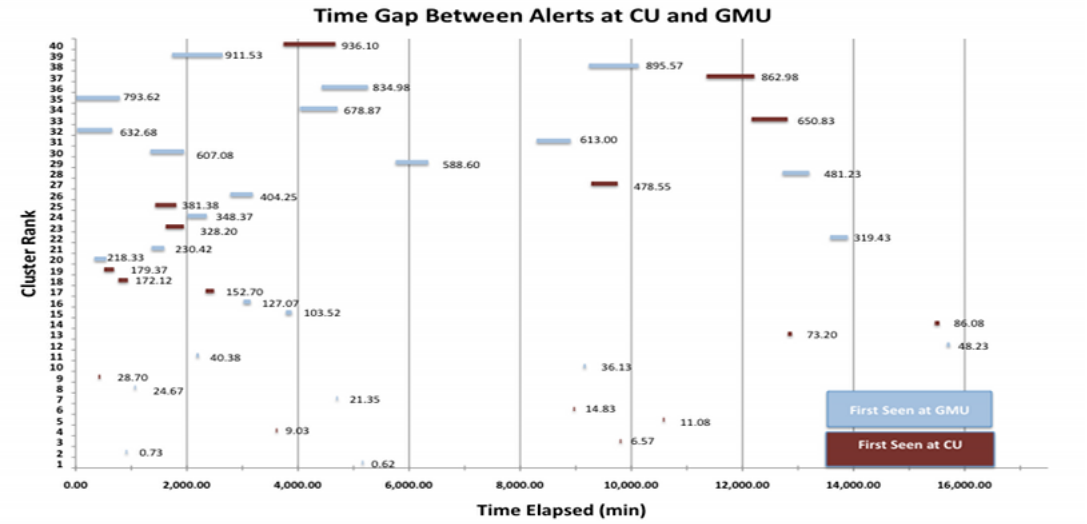

Fig. 2. Time gap between Common Alerts at CU and GMU for 40 shortest time gap clusters

Now that we have the alerts common to both sites clustered, we analyze the timing of alerts. An interesting measurement is the time lag between each cluster's first detection at each site. Fig. 2 shows the time lag which varied widely from the shortest delay being 37 seconds to attacks being seen at the second site after 8 days. Without information from a third site, we cannot directly estimate the utility of broadcasting "filters" to many other collaborating sites. Neither can we measure the impact of filters on reducing the infection rate of a large-scale attack. Hence, with additional collaborating sites we may directly measure the response time needed to provide wide-area protection against an attack. Extrapolating the data from the two sites, we hypothesize that a real time exchange of a watch list with confirmed attack signatures would be able to filter even the more rapid attacks. Given that the large number of attacks fit into a small number of clusters, generating a signature for each cluster seems feasible. We also measured the duration of each alert cluster. Our results show that some attacks persist beyond the time that both sites have seen them. This suggests that a watch list still benefits the sites that first identify a new attack.

Furthermore, we compute the frequency distribution of the common alert clusters for Columbia University and George Mason University. In Fig. 3 we can see that the presented distribution of alerts follows an exponentially decreasing trend indicating that a small fraction of the alert clusters are responsible for the vast majority of the total alerts. This favors our collaborative approach because we can mark only a small number of dense clusters only at one of the sites to sift through the majority of the alerts. The rest of the small frequency clusters can be vetted out over time since their rate of appearance is also small, and thus manageable. Currently, our collaborative architecture is applied on feeds at two sites. This makes it difficult to estimate how the alert clustering and frequency distribution would change with the addition of other collaborating sites.

AutoSense can be employed as a means of extracting zero day attacks from web applications streams at a peering point or any set of distributed sensors across enterprises. The entire set of packet streams need not be analyzed, but rather an AD model may be computed from a sample of ingress packets destined to some selected web server. By comparing the models a group of "collaborating" servers may then be chosen from which a pool of correlated common anomalous web requests would be extracted by AutoSense. Those are likely zero day attacks as evidenced by the Columbia and George Mason University experiments. With the

addition of more peers, the process of exchanging and marking alerts clusters is going to require a more comprehensive approach for operator synchronization and prioritization. This warrants further investigation which we plan to complete in the future.

VI. Conclusions

We present and analyze empirical evidence supporting the benefits from deploying a distributed content-based Anomaly Detection system. Our findings demonstrate the potential for efficient large scale mitigation of the zero-day attacks and false positives by real-time filtering of common attacks. Indeed, a total of 11787 alerts were confirmed by both AD systems for a period of two weeks. Our correlation of real alerts between distinct sites demonstrated that, in most cases, we can boost the detection performance by identifying attack clusters and false positives in one of the sites ahead of time. With this number of alerts just between two sites, we posit that, if our system is expanded to a large scale, a significant portion of zero-day web attacks could be identified and mitigated. Our findings support our theory that collaborative cross-domain content-based AD correlation might be a potential solution to the web-based zero-day attacks.

References

[1] G. Cretu, A. Stavrou, M. Locasto, S. Stolfo, and A. Keromytis, "Casting out demons: Sanitizing training data for anomaly sensors," in *Security and Privacy, 2008. SP 2008. IEEE Symposium*, may 2008, pp. 81 –95.

[2] Y. Song, M. E. Locasto, A. Stavrou, A. D. Keromytis, and S. J. Stolfo, "On the infeasibility of modeling polymorphic shellcode," in *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2007, pp. 541–551.

[3] S. Staniford-Chen, S. Cheung, R. Crawford, and M. Dilger, "GrIDS - A Graph Based Intrusion Detection System for Large Networks," in *National Information Computer Security Conference*, Baltimore, MD, 1996.

[4] P. Porras and P. G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," in *National Information Systems Security Conference*, 1997.

[5] F. Cuppens and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework," in *IEEE Security and Privacy*, 2002.

[6] C. Kruegel and T. Toth, "Distributed Pattern for Intrusion Detection," in *Network and Distributed System Security (NDSS)*, 2002.

[7] C. Kruegel, T. Toth, and C. Kerer, "Decentralized Event Correlation for Intrusion Detection," in *International Conference on Information Security and Cryptology*, 2002.

[8] J. Ullrich, "DShield home page," 2005, http://www.dshield.org.

[9] K. G. Anagnostakis, M. B. Greenwald, S. Ioannidis, A. D. Keromytis, and D. Li, "A Cooperative Immunization System for an Untrusting Internet," in *IEEE International Conference on Networks*, 2003.

[10] K. G. Anagnostakis, M. B. Greenwald, S. Ioannidis, and A. D. Keromytis, "Robust Reactions to Potential Day-Zero Worms through Cooperation and Validation," in *Proceedings of the 9^{th} Information Security Conference (ISC)*, August/September 2006, pp. 427–442.

[11] V. Yegneswaran, P. Barford, and S. Jha, "Global Intrusion Detection in the DOMINO Overlay System," in *NDSS*, 2004.

[12] A. Farroukh, N. Mukadam, E. Bassil, and I. Elhajj, "Distributed and collaborative intrusion detection systems," in *Communications Workshop, 2008. LCW 2008. IEEE Lebanon*, may 2008, pp. 41 –45.

[13] S. Zaman and F. Karray, "Collaborative architecture for distributed intrusion detection system," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, july 2009, pp. 1 –7.

[14] D. Tian, H. Changzhen, Y. Qi, and W. Jianqiao, "Hierarchical distributed alert correlation model," in *IAS '09: Proceedings of the 2009 Fifth International Conference on Information Assurance and Security*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 765–768.

[15] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Towards Collaborative Security and P2P Intrusion Detection," in *IEEE Information Assurance Workshop*, West Point, NY, 2005.

[16] G. Cretu-Ciocarlie, A. Stavrou, M. Locasto, and S. Stolfo, "Adaptive Anomaly Detection via Self-Calibration and Dynamic Updating," in *Recent Advances in Intrusion Detection*. Springer, 2009, pp. 41–60.

[17] A. Stavrou, G. F. Cretu-Ciocarlie, M. E. Locasto, and S. J. Stolfo, "Keep your friends close: the necessity for updating an anomaly sensor with legitimate environment changes," in *AISec '09: Proceedings of the 2nd ACM workshop on Security and artificial intelligence*. New York, NY, USA: ACM, 2009, pp. 39–46.

[18] B. H. Bloom, "Space/time trade-offs in Hash Coding with Allowable Errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[19] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: A Content Anomaly Detector Resistant to Mimicry Attack," in *Symposium on Recent Advances in Intrusion Detection*, Hamburg, Germany, 2006.

[20] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals." *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966, doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[21] J. J. Parekh, K. Wang, and S. J. Stolfo, "Privacy-Preserving Payload-Based Correlation for Accurate Malicious Traffic Detection," in *Large-Scale Attack Detection, Workshop at SIGCOMM*, Pisa, Italy, 2006.