ULISSE, a Network Intrusion Detection System

Stefano Zanero
Dipartimento di Elettronica e Informazione
Politecnico di Milano
stefano.zanero@polimi.it

ABSTRACT

In this paper we present a tool for network anomaly detection and network intelligence which we named ULISSE. It uses a two tier architecture with unsupervised learning algorithms to perform network intrusion and anomaly detection. ULISSE uses a combination of clustering of packet payloads and correlation of anomalies in the packet stream. We show the experiments we conducted on such architecture, we give performance results, and we compare our achievements with other comparable existing systems.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorized access (e.g., hacking, phreaking)*

General Terms

Experimentation

Keywords

Anomaly Detection; Intrusion Detection

1. INTRODUCTION

Our dependence on complex, networked computer system grows constantly. Ensuring the overall security of such systems is a daunting task, as they are prone to vulnerabilities and exploits. One of the key processes in the management of network security is therefore the detection of security incidents, followed by identification and appropriate reaction. *Intrusion Detection Systems* [3] have been the subject of much research. The majority of network IDS systems deployed today are misuse-based, meaning that they use a knowledge base to recognize directly the signatures of intrusion attempts. This makes them powerless against the so called "zero-day" (i.e. new) attacks, and against a wide array of evasion techniques [19]. In other words, such systems are prone to false negatives, even if they generate few, if any, false positives.

A better solution would be to use anomaly detectors, creating a model of normal behavior, and flagging any deviation as suspicious. Not requiring "a priori" knowledge of the attacks, they are theoretically able to detect any type of misbehavior in a statistical way; however, they tend to be more difficult to design than misuse detectors, and are particularly prone to false positives: thus, they were rarely used in practice.

But on critical networks, with a growing number of threats, relying on systems that ensure few false positives at the cost of many false negatives is not an acceptable option anymore. This is why shifting to anomaly detection is a move towards network intelligence, where we use the word "intelligence" to denote information that is current and potentially relevant, even if lacking in detail or accuracy.

This is the point of view with which we developed ULISSE (Unsupervised Learning IDS with 2Stage Engine) [27], a network-based anomaly detection IDS with a novel two-tier architecture which analyzes network packets overcoming the dimensionality problems which arise in the application of unsupervised learning techniques to network-based anomaly detection. In particular, we use the two-tier architecture to avoid discarding the network packet contents: the first tier is an unsupervised clustering algorithm which reduces the network packets payload to a tractable size. The second tier is a traditional anomaly detection algorithm, whose efficiency is improved by the availability of data on the packet payload content.

The remainder of this work is organized as follows. In Section 2 we describe our two-tier approach, and compare it with earlier works. In Section 3 we describe the algorithms we used for clustering TCP/IP payloads. In Section 4 we describe the algorithm used for outlier detection in multivariate time series. In Section 5 we describe our experimental evaluation. In Section 6 we draw our conclusions and outline recent developments of our work.

2. A TWO-TIER ARCHITECTURE FOR NETWORK INTRUSION DETECTION

To perform network intelligence at the packet level, we need to analyze the flow of packets. While the packet header data can be easily mapped onto a multivariate time series, the varying size of the payload data and its heterogeneous nature defy a compact representation as a single feature. Most existing research on the use of unsupervised learning

CSIIRW '08 May 12-14, Oak Ridge, Tennessee, USA

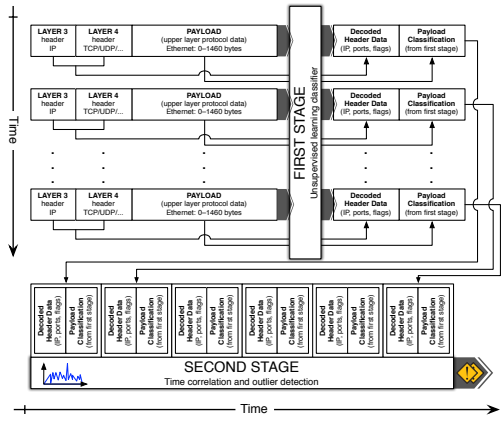

Figure 1: Scheme of the overall architecture of the network-based IDS

algorithms for network intrusion detection avoid this problem altogether by discarding the payload and retaining only the information in the packet header [14, 5, 12, 17, 15].

Ignoring the payload of packets, however, inevitably leads to information loss: most attacks, in fact, are detectable only by analyzing the payload of a packet, not the headers alone. Some earlier works tried to deal with this problem: [18] uses a rule-based algorithm to evaluate the payloads but, on the contrary, ignores totally the meaning of the header fields; ALAD [16] detects "keywords" in the protocols in a rather limited manner; PAYL [21] uses statistical techniques on the payloads, ignoring the headers.

We proposed a two-tier architecture (shown in Figure 1) for building a network-based anomaly detector using only unsupervised learning algorithms [28]. In the first tier of the system, we apply a clustering algorithm on the payload of the packets, classifying them on a single value. This classification can then be added to a subset of the information decoded from the packet header, and passed on to the second tier.

The second tier is an anomaly detection algorithm for multivariate time series, both intra-packets and inter-packets.

3. A PAYLOAD CLUSTERING ALGORITHM

The first tier algorithm needs to cluster sensibly *pattern vectors* of variable size (the payloads of the packets). "Sensibly" means that the transformation should preserve as much information as possible about the "similarity" between packets; it should separate, as much as possible, packets from different protocols in different groups; most importantly, it should also separate, as much as possible, anomalous or malformed payloads from normal payloads. This is a typical problem of clustering, i.e., trying to group objects in classes, so that intra-class similarity is maximized and inter-class similarity is minimized [7]. It can also be seen as an instance of a pattern recognition problem applied to packet payloads [25].

We need to choose an algorithm which is robust to an arbitrary choice of the number of classes, and that is robust w.r.t. the presence of outliers in training data. An outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism [8]. We tested various algorithms and selected a Self Organizing Maps (SOM) [11]. We showed that the SOM is able to discover interesting information in an unsupervised manner. Additionally, it is robust with regard to the choice of the number of clusters, and it is also resistant to the presence of outliers in the training data.

In order to use it on such high-dimensional data with good performance we developed various approximate techniques to speed up the algorithm at runtime, by introducing minimal errors in the classification [26].

Another problem which arises when using similarity-based algorithms in high-dimensional space, is the choice of a good similarity criterion. The two most used distance criteria in SOM literature are the inner product and the euclidean metric. Since the inner product is closely related to the so-called cosine distance, it is particularly useful in those cases where attributes have values whose characteristic is to be either zero or nonzero. We have a range of discrete values instead, so we resorted to the euclidean distance. While this choice has no specific theoretical support, our experiments have shown that it works well. Furthermore, in [4, 9] it is shown that in high dimensional spaces the concept of proximity and distance may not be meaningful, even qualitatively. However, as most of the hypotheses of such theoretical works do not hold for our variables, we have experimentally observed that in our setup this effect does not happen. We explored the application of the alternative distance metrics proposed in [9, 2], but in our particular application they seem to lump all the data in a few cluster, diminishing the overall recognition capabilities of the algorithm instead of enhancing it. We are currently studying the applicability of wavelet-based distance metrics such as the ones proposed in [1].

4. MULTIVARIATE TIME SERIES OUTLIER DETECTION

The second tier algorithm must detect anomalies and outliers over a multivariate time series with at most 30 features, finding both intra and inter-packet correlations. A survey of outlier detection techniques can be found in [10], but they are mostly limited to continuous variables and to strictly ordered series. Packets are neither totally numeric nor strictly ordered. Additionally, since in a real world situation it would be difficult to collect a large base of attack-free traffic in order to train the algorithm, we need it to be resistant to the presence of outliers in the training dataset.

Excluding supervised algorithms, we are left with a handful of candidates. We found that using a modified version of the discounting learning algorithm SmartSifter [23, 24, 22] gives good results. SmartSifter is designed for online usage, and it uses a "forgetting factor" in order to adapt the model to non-stationary data sources. The output is a value expressing the statistical distance of the new observation, which means "how much" the new observation would modify the model currently learned. SmartSifter has also the great advantage to be able to use both categorical and metric variables.

We modified SmartSifter using an approximate calculation of the *Hellinger distance* (as pointed out in [24]). Also, in order to automatically tune the anomaly threshold we introduced a short training phase during which the distribution of the anomaly scores is approximated, and an estimated quantile of the distribution is also computed. In this way we can directly set the IDS sensitivity as the per-

Table 1: Detection rates and false positive rates for our prototype

Threshold	Detection Rate	False Positive Rate
0.03%	66.7%	0.031 %
0.05%	72.2%	0.055 %
0.08%	77.8%	0.086%
0.09%	88.9%	0.095%

centage of packets we want to consider as outliers. As opposed to this totally unsupervised outlier determination, in [22] the authors of SmartSifter proposed a mixed supervised/unsupervised approach.

Feature selection is an important step for any learning application [6]. We wish to stress this point, since the algorithms proposed in the literature have been applied to more or less arbitrary selections of features of the packets. As no reliable method exists to perform feature selection on categorical variables, we tested different combinations of the variables, and found that a set containing source port, destination port, TCP flags, source and destination address and the payload classification worked well.

5. EXPERIMENTAL EVALUATION

In order to evaluate our architecture in a repeatable manner, we ran the prototype over various days of traffic drawn from the 4th week of the 1999 DARPA dataset [13]. We also added various attacks against the Apache web server and against the Samba service generated through the Metasploit framework (`www.metasploit.org`). The average results are reported in Table 1. The first column contains the sensitivity threshold set for the algorithm, which is, as we noted, a good statistical predictor of the percentage of data that will be flagged as outliers by the algorithm. Therefore it is also a good predictor of the false positive rate, if the attack rate is not too high. The prototype is able to reach a 66.7% detection rate with as few as 0.03% false positives.

For comparison, the authors of SmartSifter in [24] tested their algorithm against the KDD Cup 1999 [20] dataset, which is extracted from the same dataset we used. They claim a 18% detection rate, with a 0.9% false positive rate (6421 connections). Our algorithm can instead reach a 92% detection rate with a 0.17% false positive rate (2035 packets), thus demonstrating a highly superior performance.

PAYL [21] is the only other prototype of IDS in literature which uses part of the payload of packets. The best overall results reported for PAYL show a detection rate of 58.7%, with a false positive rate that is between 0.1% and 1%. Our architecture can reach the same detection rate with a false positive rate below 0.03%, thus an order of magnitude better than PAYL, or on the other hand reach a 88.9% detection rate with no more than a 1% rate of false positives.

In Figure 2 we further show how our 2-tier architecture benefits the detection rate by comparing the ROC curves of the SmartSifter system with and without the payload classification tier by including and excluding the feature. The results are clearly superior when the first tier of unsupervised clustering is enabled, proving the usefulness of our approach.

6. CONCLUSIONS

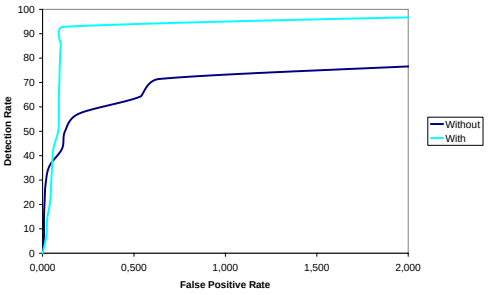

Figure 2: ROC curves comparing the behavior of SmartSifter with (lighter) and without (darker) our architecture

In this short paper we described ULISSE, an innovative model of anomaly based network intrusion detection system, completely based on unsupervised learning techniques. Using a first tier of clustering (based on Self Organizing Maps and on specific heuristics for speedup) it can perform an efficient, unsupervised pattern recognition on packet payloads. A modified version of the SmartSifter outlier detection algorithm completes the prototype.

We have given results on the detection rate and false positive rate, showing that the system outperforms a similar, state-of-the-art system by almost an order of magnitude in term of false positive reduction. Future works on this system will strive to further improve its speed, as well as to reduce the false positive rate as much as possible. A theme we are beginning to research on now, and which is the natural evolution of this work, is how to integrate ULISSE with S^2A^2DE (Syscall Sequence and Arguments Anomaly Detection Engine), a host-based systems we designed, in order to use the results of both to automatically filter out false positives and to improve correlation and alert quality.

Acknowledgments

We wish to thank our students Matteo F. Zazzetta and Federico Maggi for their precious support in software development and lab testing. This work has been partially supported by the European Commissions through project IST-216026-WOMBAT funded by the 7th framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission. Earlier works on this subject by the same author have been partially supported by the Italian FIRB Project “Performance evaluation for complex systems”, and by a PhD scholarship of the Italian Ministry for the University and Research (MIUR), under the guidance of Prof. Giuseppe Serazzi, whose support we gratefully acknowledge.

7. REFERENCES

[1] C. C. Aggarwal. On effective classification of strings with wavelets. In *KDD ’02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 163–172, New York, NY, USA, 2002. ACM Press.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. *Lecture Notes in Computer Science*, 1973, 2001.

[3] R. G. Bace. *Intrusion detection*. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 2000.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? *Lecture Notes in Computer Science*, 1540:217–235, 1999.

[5] C. Chow. Parzen-Window network intrusion detectors. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 4*, pages 385–388, Washington, DC, USA, aug 2002. IEEE Computer Society.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[7] J. Han and M. Kamber. *Data Mining: concepts and techniques*. Morgan-Kauffman, 2000.

[8] D. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.

[9] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *The VLDB Journal*, pages 506–515, 2000.

[10] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.

[11] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 3 edition, 2001.

[12] K. Labib and R. Vemuri. NSOM: A real-time network-based intrusion detection system using self-organizing maps. Technical report, Dept. of Applied Science, University of California, Davis, 2002.

[13] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. In *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection*, pages 162–182, London, UK, 2000. Springer-Verlag.

[14] M. Mahoney and P. Chan. Detecting novel attacks by identifying anomalous network packet headers. Technical Report CS-2001-2, Florida Institute of Technology, 2001.

[15] M. V. Mahoney. Network traffic anomaly detection based on packet bytes. In *Proceedings of the 19th Annual ACM Symposium on Applied Computing*, 2003.

[16] M. V. Mahoney and P. K. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 376–385, New York, NY, USA, 2002. ACM Press.

[17] M. V. Mahoney and P. K. Chan. A machine learning approach to detecting attacks by identifying anomalies in network traffic. Technical Report CS-2002-08, Florida Institute of Technology, 2002.

[18] M. V. Mahoney and P. K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Proc. of the 3rd IEEE Int'l Conf. on Data Mining*, page 601, 2003.

[19] T. H. Ptacek and T. N. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical Report T2R-0Y6, Secure Networks, Calgary, Canada, 1998.

[20] H. S. and B. S. D. KDD Cup '99 Dataset. `http://kdd.ics.uci.edu/`, 1999.

[21] K. Wang and S. J. Stolfo. Anomalous payload-based network intrusion detection. In *RAID Symposium*, September 2004.

[22] K. Yamanishi and J. ichi Takeuchi. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 389–394, New York, NY, USA, 2001. ACM Press.

[23] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Min. Knowl. Discov.*, 8(3):275–300, 2004.

[24] K. Yamanishi, J.-I. Takeuchi, G. J. Williams, and P. Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Knowledge Discovery and Data Mining*, 8(3):275–300, 2004.

[25] S. Zanero. Analyzing tcp traffic patterns using self organizing maps. In F. Roli and S. Vitulano, editors, *13th International Conference on Image Analysis and Processing - ICIAP 2005*, volume 3617 of *Lecture Notes in Computer Science*, pages 83–90, Cagliari, Italy, September 2005. Springer.

[26] S. Zanero. Improving self organizing map performance for network intrusion detection. In *SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications"*, 2005.

[27] S. Zanero. *Unsupervised Learning Algorithms for Intrusion Detection*. PhD thesis, Politecnico di Milano T.U., Milano, Italy, May 2006.

[28] S. Zanero and S. M. Savaresi. Unsupervised learning techniques for an intrusion detection system. In *Proc. of the 2004 ACM Symposium on Applied Computing*, pages 412–419. ACM Press, 2004.