Analyzing TCP Traffic Patterns Using Self Organizing Maps

Stefano Zanero

D.E.I.-Politecnico di Milano,
via Ponzio 34/5 - 20133 Milano Italy
zanero@elet.polimi.it

Abstract. The continuous evolution of the attacks against computer networks has given renewed strength to research on anomaly based Intrusion Detection Systems, capable of automatically detecting anomalous deviations in the behavior of a computer system. While data mining and learning techniques have been successfully applied in host-based intrusion detection, network-based applications are more difficult, for a variety of reasons, the first being the curse of dimensionality. We have proposed a novel architecture which implements a network-based anomaly detection system using unsupervised learning algorithms. In this paper we describe how the pattern recognition features of a Self Organizing Map algorithm can be used for Intrusion Detection purposes on the payload of TCP network packets.

1 Introduction

The continuous evolution of the attacks against computer networks and systems has given renewed strength to research on "anomaly based" Intrusion Detection Systems (IDS). The "misuse based" approach, which tries to define what constitutes an attack in order to detect them, has been widely successful in the past, but is increasingly showing its limits.

The growing number of attacks requires a continuous update of the knowledge base of misuse based IDSs. In addition, there is also an unknown number of discovered but undisclosed vulnerabilities (the so called "zero-days") that are not available for analysis and inclusion in the knowledge base. Most attacks are also polymorph, and skilled attackers exploit this polymorphism to evade detection.

The obvious solution would be to use an anomaly detection approach, modeling what is *normal* instead than what is *anomalous*. This is surprisingly similar to the earliest conceptions of what an IDS should do [1]. However, while a number of host based anomaly detection systems have been proposed and implemented, both in literature and in practice, network anomaly detection is still an open field for research.

In a previous work [2] we proposed a novel architecture for applying unsupervised learning techniques to a network based IDS. Unsupervised learning techniques are natural candidates for this type of task, but while they have been

successfully applied in host based intrusion detection [3], their application to network based systems is still troublesome, mainly due to the problems of input selection, data dimensionality and throughput. In particular, analyzing the payload of TCP/IP packets is a challenging task, which most earlier researches avoided to deal with.

In this paper we will we analyze how our architecture uses the pattern recognition capabilities of a Self Organizing Map (SOM) algorithm [4] in order to solve this problem. By using a SOM, we are able to retain part of the information related to the payload content, characterizing in an unsupervised manner the recurring patterns of packet payloads and "compressing" them into a single byte of information. On most network segments, the traffic belongs to a relatively small number of services and protocols, regularly used, and a good learning algorithm can map it onto a relatively small number of clusters. Our experimental results show that a SOM can successfully learn and recognize these patterns. We also analyze how this unsupervised characterization can help in detecting anomalous traffic payloads. A comparison with various alternative approaches to the problem is also presented.

The remainder of the paper is organized as follows: in section 2 we will describe the proposed architecture for our IDS; in section 3 we will describe how a SOM algorithm can be used for detecting anomalous patterns in payloads; in section 4 we will report on preliminary detection results of the overall architecture; finally, in section 5 we will draw our conclusions and outline some future work.

2 The Proposed Architecture

In a previous work ([2]) we proposed a novel architecture for building a network based anomaly detection IDS, using only unsupervised learning algorithms. These algorithms exhibit properties such as the ability of detecting outliers and of building a model of "normality" without the need of a priori knowledge input; this makes them good candidates for anomaly detection tasks.

We thus reformulated the problem of detecting network anomalies as an instance of the multivariate time sequence outlier detection problem, where the stream of observations is composed of "packets". However, mapping TCP/IP packets as a multivariate time sequence is not straightforward. Each packet has a variable size, and outlier detection algorithms are designed to work on multivariate data with a fixed number of dimensions or "features". The network and transport layer headers can be normalized into a fixed number of features (it is important to note, however, that in the case of connection-oriented protocols the transport layer headers may need inter-correlation in order to be fully deciphered). On the contrary, the data carried by the packet (the payload) cannot be easily translated into a fixed set of features, since each different application layer protocol would require its own set of variables, increasing complexity and decreasing generality. This would also require a full reconstruction of traffic ses-

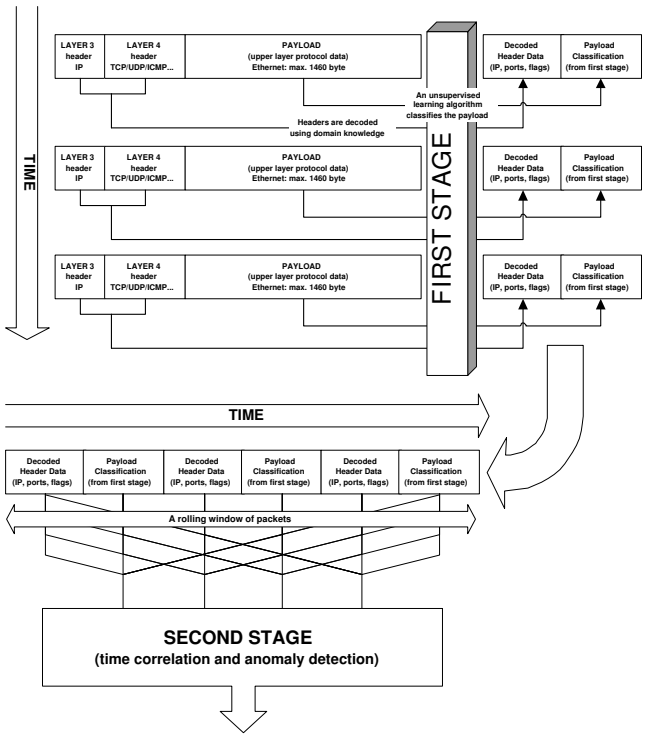

Fig. 1. Architecture of the IDS

sions, which would expose the IDS to reconstruction problems, possibly leading to attack windows [5].

Most existing researches on the use of unsupervised learning algorithms for network intrusion detection purposes avoid this problem by discarding the payload and retaining only the information in the packet header (e.g. [6–8], the only partial exception being [9], which uses a rule-based algorithm). Ignoring the payload of packets, in our opinion, leads to an unacceptable information loss: most attacks, in fact, are detectable only by analyzing the payload of a packet, not the headers alone. Nevertheless, these algorithms show interesting, albeit obviously limited, intrusion detection properties.

Our approach to the problem consists of a two-tier architecture (shown in Figure 1), which allows us to retain at least part of the information related to the payload content. In the first tier of the system, an unsupervised clustering algorithm operates a basic form of pattern recognition on the payload of the packets, observing one packet payload at a time and "compressing" it into a single byte of information. This classification can be added to the information decoded from the packet header (or to a subset of this information), and passed on to the second tier. On most networks, the traffic belongs to a relatively small

number of services and protocols, regularly used, and a good learning algorithm can map it onto a relatively small number of clusters.

The second tier algorithm instead tries to detect anomalies, both in each single packet and in a sequence of packets. It is worth noting that most of the solutions proposed by previous researchers in order to analyze the sequence of data extracted by the packet headers can be used as a second tier algorithm, complemented by our first tier of unsupervised pattern recognition and clustering.

3 Detecting Patterns in Packet Payloads

In the first tier of our architecture the algorithm receives in input the payload of a TCP or UDP over IP packet, which is a sequence of bytes of variable size (on an Ethernet segment, limited to 1460 bytes). The algorithm must be able to classify such information in a "sensible" way. By sensible we mean that it should preserve three important properties:

1. Preserve as much information as possible about the "similarity" between packets
2. Separate, as much as possible, packets from different protocols in different groups
3. Most importantly, since our final goal is to detect intrusions, separate packets with anomalous or malformed payload from normal packets

"The grouping of similar objects from a given set of inputs" [10] is a typical clustering problem; but it can also be seen as an instance of a pattern recognition problem, where we are trying to characterize the recurring patterns in packet payloads in order to detect anomalies.

We have shown [2] that a Self Organizing Map algorithm [4] is indeed able to sensibly cluster payload data, discovering interesting information in an unsupervised manner, and that it performs much better than a K-means algorithm, or a Principal Direction Divisive Partitioning Algorithm. A previous research showed that neural algorithms can recognize protocols automatically [11], while another paper later independently confirmed our result that the payload of the packets indeed shows some interesting statistical properties [12].

There are multiple reasons for choosing a SOM for this purpose. The algorithm is robust with regard to the choice of the number of classes to divide the data into, and it is also resistant to the presence of outliers in the training data, which is a desirable property. In addition, we have compared various algorithms and shown that the SOM had the best performance trade-off between speed and classification quality.

As it is known, however, the computational complexity of unsupervised learning algorithms scales up steeply with the number of considered features, and the detection capabilities decrease correspondingly (one of the effects of the so called "curse of dimensionality"). There are alternative algorithms for clustering which are much faster in the learning phase than SOM, for example, the well known

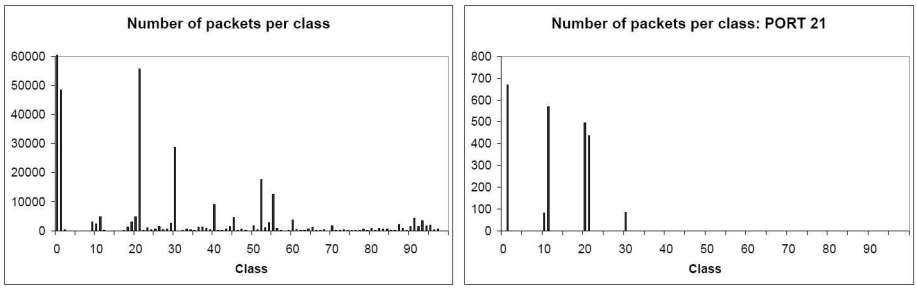

Fig. 2. Comparison between all the traffic and the subset with destination set to port 21/TCP

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%

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

K-means algorithm is one of the fastest. But during recognition even K-means is not more efficient than a SOM, so we cannot solve this problem by simply choosing a different algorithm.

A traditional approach to the problem would use dimensionality reduction techniques such as dimension scaling [13] or Principal Component Analysis [14]. But our experiments demonstrated that they are quite ineffective in this particular situation, since by their nature they tend to "compress" outliers onto normal data, and this is exactly the opposite of what we want to achieve.

Since no alternative solution was viable, we developed various approximate techniques to speed up the SOM algorithm [15]. The throughput of an implementation of the original Kohonen algorithm on common hardware running Linux is on average of 3400 packets per second, which is enough to handle a 10 Mb/s Ethernet network, but insufficient for a 100 MB/s network. Our improved algorithm uses a combination of heuristics in order to reduce the average number of operations to find the best matching neuron, at the expense of precision of matching. The modified algorithm runs at a speed of 10.500 packets/second, which is high enough to handle a normal 100 Mb/s link, with a minimal precision loss which does not impact pattern recognition capabilities. If necessary, performance could be further improved by reducing the number of bytes of the payload considered by the algorithm: it can be shown that this has just minimal impact on the recognition capabilities.

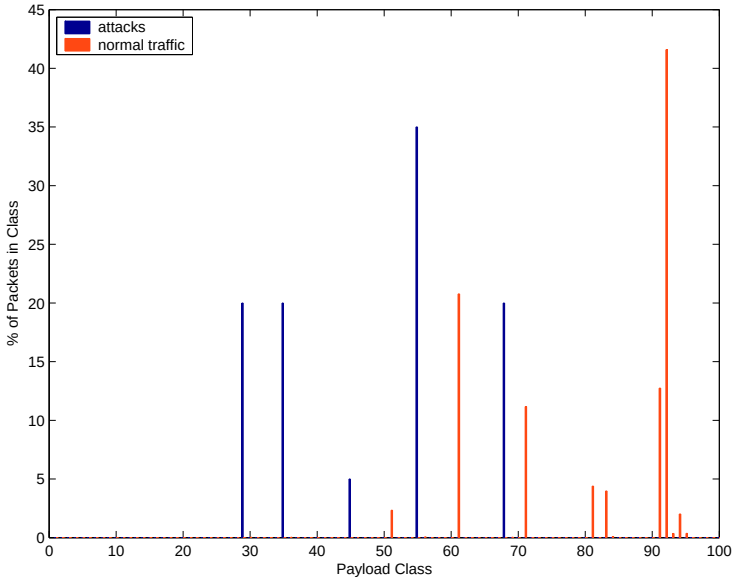

Fig. 3. A comparison between the classification of attack payloads and normal traffic payloads on port 80/TCP

4 Evaluation of Pattern Recognition Capabilities

In order to evaluate the recognition capabilities of the algorithm, we must see if it can usefully characterize traffic payloads for different protocols, and detect anomalous attack payloads from normal payloads. The data used for training and testing the prototype are subsets of the "1999 DARPA IDS Evaluation dataset". In [16] the shortcomings of the DARPA traffic sample sets are analyzed, and we share many of the author's observations. Thus, we positively validated our results using also smaller dumps collected on our own internal network.

In Figure 2 we present a demonstration of the recognition capabilities of a 10×10 SOM (with hexagonal topology). The network was trained for 10.000 epochs on TCP packet payloads. The histograms represent the number of packets in each of the 100 clusters. For graphical reasons, the scale of y-axis is not the same.

On the left handside, we can see the classification of a whole day of TCP traffic drawn from the dataset. On the right handside, we can see how the network classifies the subset of packets with destination port set to 21/TCP (FTP service command channel). It can be observed how all the packets fall in a narrow group of classes, demonstrating a strong, unsupervised characterization of the protocol.

We also analyzed how the SOM classifies packets from the attacks contained in the DARPA datasets. In Figure 3 we can see that attack packets consistently fall into different classes than normal packets (as an example, we used the packets destined to port 80/TCP).

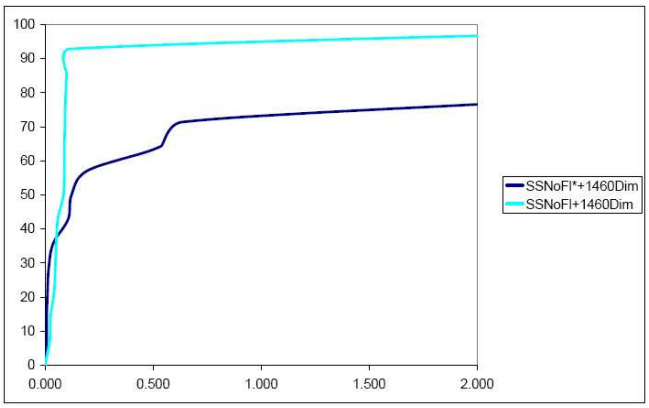

Fig. 4. ROC curves comparing the behavior of Smart Sifter with (lighter) and without (darker) our architecture

For the second stage we used a modified version of the unsupervised outlier detection algorithm SmartSifter [17]. We ran the prototype over various days of the 1999 DARPA dataset. The average results are reported in Table 1. The first column contains the sensitivity threshold of the algorithm, and is a statistical predictor of the percentage of data that will be flagged as outliers by the algorithm. As we can see, 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. In comparison, in [12] the best overall result leads to the detection of 58.7% of the attacks, with a false positive rate that is between 0.1% and 1%. Our prototype shows thus a better detection rate, with a number of false positives which is between one and two order of magnitudes lower than a comparable system. In Figure 4 we further show how our 2-tier architecture benefits the detection rate by comparing the ROC curves of the system with and without the payload classification stage. The results are clearly superior when the first stage of unsupervised clustering is enabled, proving the usefulness of our approach.

5 Conclusions and future work

We have described how we used pattern recognition algorithms in order to build an anomaly based network intrusion detection system. We have described the overall architecture of the system, and shown how the first stage of clustering performs an efficient, unsupervised pattern recognition on packet payloads. The results on the detection rate and false positive rate of the complete system demonstrate that it outperforms a similar, state-of-the-art system by almost an order of magnitude in term of false positive reduction; comparison of ROC curves demonstrates that our approach is indeed the key . Our future work will

focus on further false positives reduction, and on the empirical evaluation of the IDS under practical workloads.

Acknowledgments

This work was partially supported by the Italian FIRB Project "Performance evaluation for complex systems". We need to thank prof. Sergio M. Savaresi for his support and helpful suggestions, and also our student Matteo F. Zazzetta for his invaluable support in software development and lab testing.

References

1. Anderson, J.P.: Computer security threat monitoring and surveillance. Technical report, J. P. Anderson Co., Ft. Washington, Pennsylvania (1980)
2. Zanero, S., Savaresi, S.: Unsupervised learning techniques for an intrusion detection system. In: Proc. of the 14th Symp. on Applied Computing, ACM SAC 2004. (2004)
3. Kruegel, C., Mutz, D., Valeur, F., Vigna, G.: On the detection of anomalous system call arguments. In: Proc. of ESORICS 2003. (2003)
4. Kohonen, T.: Self-Organizing Maps. 3 edn. Springer-Verlag, Berling (2001)
5. Ptacek, T.H., Newsham, T.N.: Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical Report T2R-0Y6, Secure Networks, Calgary, Canada (1998)
6. Mahoney, M., Chan, P.: Detecting novel attacks by identifying anomalous network packet headers. Technical Report CS-2001-2, Florida Institute of Technology (2001)
7. Yeung, D.Y., Chow, C.: Parzen-window network intrusion detectors. In: Proc. of the 16th Int'l Conf. on Pattern Recognition. Volume 4. (2002) 385–388
8. Labib, K., Vemuri, R.: NSOM: A real-time network-based intrusion detection system using self-organizing maps. Technical report, Dept. of Applied Science, University of California, Davis (2002)
9. Mahoney, M.V., Chan, P.K.: Learning rules for anomaly detection of hostile network traffic. In: Proc. of the 3rd IEEE Int'l Conf. on Data Mining. (2003) 601
10. Hartigan, J.A.: Clustering Algorithms. Wiley (1975)
11. Tan, K., Collie, B.: Detection and classification of TCP/IP network services. In: Proc. of the Computer Security Applications Conf. (1997) 99–107
12. Wang, K., Stolfo, S.J.: Anomalous payload-based network intrusion detection. In: RAID Symposium. (2004)
13. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Monographs on Statistics and Applied Probability. Chapman & Hall (1995)
14. Jolliffe, I.T.: Principal Component Analysis. Springer Verlag (1986)
15. Zanero, S.: Improving self organizing map performance for network intrusion detection. In: SDM 2005 Workshop on "Clustering High Dimensional Data and its Applications", submitted for publication. (2004)
16. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory. ACM Trans. on Information and System Security **3** (2000) 262–294
17. Yamanishi, K., ichi Takeuchi, J., Williams, G.J., Milne, P.: On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. In: Proc. of the 6th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. (2000) 320–324