Clustering as an add-on for firewalls

C. Caruso & D. Malerba
Dipartimento di Informatica, University of Bari, Italy

Abstract

The necessary spread of the access points to network services makes them vulnerable to many potential and different types of attackers: script kiddies, hackers, and misfeasors. Although the network services produce a great quantity of data logged by hosts, it is impossible for a security officer, and generally for a network administrator, to monitor daily generated traffic in order to control attacks. Currently a LAN is defended with a mixture of solutions adopted at different levels. Commercial firewalls typically use descriptive statistics to give the security officer information about the quantitative characteristics of the TCP/IP traffic as a whole. In this work, we generate information on the "profile" of connections by means of clustering techniques. This approach makes the security officer able to detect connections that are far away from the mass. We use different clustering techniques in order to study their response for this type of problem. Results on real traffic data are reported and commented.
Keywords: live network traffic analysis, anomaly detection, intrusion detection, clustering, data preprocessing.

1 Introduction

For a network administrator it is important to have a complete description of the connections behaviour so to understand the development of his/her own network. This aspect is becoming more and more relevant and in fact commercial firewalls include modules which, though the computation of simple descriptive statistics, try to inform the security officer on the qualitative nature of network traffic. Firewalls have no means to give information about the mass of connections. The built-in modules permit to analyse every aspect of packet streams; some firewalls also possess an SQL dialect to query its own logs but SQL queries give answers about something the user already know or "suspect".

Data Mining V, A. Zanasi, N. F. F. Ebecken & C. A. Brebbia (Editors)
© 2004 WIT Press, www.witpress.com, ISBN 1-85312-729-9

Personal experience as netadmins teaches us that some types of attack strategies against the network are discovered by chance. Often, netadmins have to read the firewall logs to notice anomalous connections neither IDS nor firewall itself would be able to notice. Although a firewall offers a privileged viewpoint with respect to other points in the network because of its concentration of all the pass-through traffic, the network security can only be guaranteed by agents distributed in different points that collaborate each other; a firewall is just one of this point.

These considerations justify some interest towards the application of data mining techniques to firewall logs in order to develop better tools for network security. Surprisingly, few examples of firewall logs analysis are reported in the literature and are mainly based on some statistical and heuristic measures [1]. The exploration of network traffic behaviour passing through a firewall actually requires more sophisticated analyses that would facilitate the recognition of anomalies (legitimate or not) in the mass of connections and would provide the network officer with the context knowledge necessary to target a connection as suspicious. In fact, it is impossible to correctly classify anomalies without having a precise description of the network to protect, that is, a model.

Clustering is a global model of discovery and summarization used as a basis for more precise models [2] and it widely employed in different sciences. In this work we study the applicability and the performance of clustering techniques to a descriptive task of real data to build a "daily" model. In particular, we investigate the potentialities of some clustering methods in capturing regularities in daily network traffic as well as the possible usage of descriptions returned by the combined used of clustering and decision tree learning techniques to recognize and to show traffic changes. The intuition is to adopt a human-oriented approach to unknown environments: initially summarize with respect to common aspects and then learn the peculiarities.

In network security literature, clustering is used by Portnoy et al. [3] as a predictive tool over KDDCup '99 dataset [4] to discover anomalies. The strategy of anomaly detection builds models of normal behaviour and detects anomalies as deviations from it. The model is generated over normal data [5], that is, data with no form of attack or anomalies. An important novelty of our study is that training data used are extracted from real historical logs of our firewall and no prior information about quantitative or qualitative aspects of illegitimate traffic, attacks or anomalies is available. As far as we know, the only research project based on totally real data is Minds [6] a project of network security of Minnesota University.

The paper is organized as follows. Information present in the firewall logs, extracted data and preprocessing procedures are described in Section 2. In Section 3, methods, experiments and results are presented.

2 Data collection and preprocessing

Data are extracted from the log files of our departmental firewall. Generally, the behaviour of a firewall depends on its security policy, which is specified by a set of rules that determine if a network packet is accepted or rejected. More specifically, our firewall accepts everything with few exceptions; it has few and "light" rules which above all apply to incoming requests. Network security is very important but a university department uses Internet in a more intensive way than a business company because it needs to collaborate fast and dynamically with other universities spread all over the world. This makes difficult to adopt a narrow security policy. However, we think that this is a chance more than an obstacle for a researcher in network security.

The original dataset is given by the packets logs of our firewall for a period of two weeks, from Monday to Sunday: 9th-15th, June 2003 and 17th-23rd May 2004. The two periods are very far and they represent different stages in the history of our network. We have 14 files (800MB all together) in *cvs* format. A transaction is a single logged packet, each of which is described by the following attributes: counter (integer), date, time, protocol (tcp/udp/icmp), direction (inbound/outbound), action (accept, drop, rejected, refused), source IP, destination IP, service port, source port, length, rule, diagnostic message, sys_message and others. Files have been cleaned from our service traffic generated by internal servers or internal broadcast. No missing or incorrect values have been observed in extracted data.

Features are extracted with the purpose of creating groups of similar connections as to time length, number of packets, timestamp, requested service and world zone they come from. Extracted features should help to answer to these questions: Who are visitors of our network? Where do they come from? Which service do they ask for? Answers to these questions do help the network officer to identify regularities in the network traffic.

The target dataset is built by reconstructing the entire connections from single packets. In our analysis we have used only accepted outside connections and we have a file for every day involved in our study. Since the goal is to create the daily description of our connections, we have chosen to work with few but fundamental attributes, namely:

a) *Starthalfhour* (integer): the time field of a packet has the hhmmss format; we have several different connections beginning in the same instant; we have chosen to discretize this value and we have divided a day in 48 intervals, each of half an hour: 0.47. A connection belongs to the time interval it starts in.
b) *Protocol* (nominal): udp, tcp; few icmp values have been dropped when generating training data.
c) *DestinationIP* (integer): this field, which is composed of four bytes, can only have the value of our network addresses, so we have considered only the last byte (values 0.255).
d) *SourceIP* (nominal): this field has a very high cardinality because it represents the entire IPv4 space.
e) *Service port* (nominal): the requested service (http, ftp, smtp and many other ports).
f) *Numpackets* (integer): the number of connection packets.

g) *Length* (integer): the time length of the connection.
h) *NationCode* (nominal): the two digit code of the nation the source IP belongs to.

We observe that NationCode can be considered a generalization of SourceIP due to the partitioning of the IPv4 space. However, they are both kept because in this preliminary study no hint on the best level of detail of training data is available.

After this connection reconstruction step, the target dataset is made up by 14 files in *cvs* format with a total dimension of 30 MB.

3 Methods and experiments

3.1 Methods

Two clustering algorithms implemented in the Weka software (www.cs.waikato.ac.nz/ml/weka) have been considered in this study: K-means and EM. The former is very simple and reasonably efficient; it distributes instances between clusters deterministically by trying to minimize the distance (in our case the Euclidean one) between an instance and its cluster's centroid. The EM (expectation-maximization) algorithm distributes instances probabilistically; it tries to maximize likelihood looking for the most feasible parameters' values, in particular, the number of clusters.

The result of the clustering algorithms is simply a partitioning of the original dataset. To analyze the results of clustered connections, a decision tree has been generated by considering the membership cluster as the class of a connection. The decision tree learning system used in this work is C4.5 implementation of Weka package. The pruning factor has been set to 1.0 since our data analysis task has a descriptive rather than predictive nature, that is, overfitting is not an issue while it is important to have a complete tree that describes the clusters at best.

3.2 Experimental results

For every day we perform two experiments. In the first experiment we use a dataset, expressed in ARFF format, described by the features a-g; in the second one the last feature (h) is also considered. For each subset, both clustering algorithms (K-means and EM) are applied. For K-means, we have chosen the same number of clusters generated by the EM. In Tables 1 and 2 we report the number of clusters and instances for each experiment.

We observe that the number of training instances decreases towards the weekend in the week of May 2004, while it surprisingly increases during the weekend of June 2003. This difference is due to heavy traffic towards a specific server of our departmental network. The number of generated clusters is quite low, ranging from two to nine. The addition of the feature *NationCode* does affect the number of clusters, although its effect is unpredictable (no clear increase/decrease in the number of clusters). By examining the distribution of instances per cluster, we notice that while EM tends to create one big cluster and other significantly smaller, K-means tends to evenly distribute instances among clusters.

Table 1: May 17th-23th, 2004. Number of clusters and instances for each experiment.

Date	Feature set	Clusters	Training instances
Mon 17.05.04	a-g	3	56,693
	all	2	
Tue 18.05.04	a-g	3	53,147
	all	3	
Wed 19.05.04	a-g	2	55,026
	all	5	
Thu 20.05.04	a-g	6	61,328
	all	7	
Fri 21.05.04	a-g	4	23,471
	all	6	
Sat 22.05.04	a-g	5	3,937
	all	4	
Sun 23.05.04	a-g	3	1,583
	all	9	

Table 2: June 9th-15th, 2003. Number of clusters and instances for each experiment.

Date	Feature set	Clusters	Training instances
Mon 9.06.03	a-g	3	23,742
	all	4	
Tue 10.06.03	a-g	3	11,125
	all	2	
Wed 11.06.03	a-g	5	19,804
	all	7	
Thu 12.06.03	a-g	7	28,360
	all	3	
Fri 13.06.03	a-g	4	24,607
	all	6	
Sat 14.06.03	a-g	4	61,627
	all	4	
Sun 15.06.03	a-g	3	36,064
	all	4	

The number of internal nodes (size) and leaves of decision trees generated for each experiment present a strong variability which depends on the daily data. Tree depth is quite small although trees may have many leaves. Trees cannot be completely reported because of the page limitation of this paper. Figure 1 shows the complete trees generated for EM clusters for three days, May 17th -19th, 2004. For K-means only the high level tests in the decision trees are given. The entire results can be downloaded from the URL http://www.di.uniba.it/datamining/network.

K-means

```
protocol = udp: c0
protocol = tcp
   startHalfHour <= 25
      dst <= 107
         startHalfHour  <=  22:
c1
         startHalfHour > 22
            dst <= 9: c1
            dst > 9
               dst <= 10
                  src              =
217.95.11.95: c1
...omissis...
```

```
protocol = udp: c0
protocol = tcp
   startHalfHour <= 30
      dst <= 105
         startHalfHour <= 22: c1
         startHalfHour > 22
            dst <= 9: c1
            dst > 9
               dst <= 10
                  src              =
218.145.148.178: c1
...omissis...
```

```
protocol = udp: c0
protocol = tcp
   startHalfHour <= 25
      dst <= 87
         startHalfHour  <=  22:
c0
         startHalfHour > 22
            dst <= 9: c0
            dst > 9
               dst <= 10
                  src              =
217.95.11.95: c0
...omissis..
```

EM

```
numPackets <= 1
   dst <= 45
      dst <= 10: c1
      dst > 10: c0
   dst > 45: c1
numPackets > 1: c2
```

```
length<= 0
  dst <= 45
     dst <= 10
        numPackets <= 2: c1
        numPackets > 2: c0
     dst > 10: c2
  dst > 45
     numPackets <= 1: c1
     numPackets > 1
        numPackets <= 2: c1
        numPackets > 2: c0
length> 0: c0
```

```
protocol = udp
   dst <= 50
      numPackets <= 1: c1
      numPackets > 1: c0
   dst > 50: c0
protocol = tcp
   dst <= 45
      dst <= 10: c0
      dst > 10
         numPackets <= 1: c1
         numPackets > 1: c0
   dst > 45: c0
```

Figure 1: Decision trees learned for the days May 17th-19th, 2004. Only seven features (a-g) are used to describe training data. Nominal features are in bold; all the others are numeric.

Trees induced from clusters generated by EM are characterized by the dominant presence of numeric features, while trees associated to K-means clusters have several nominal features and are generally bigger.

It is noteworthy that the decision tree generated for EM clusters brings the nominal feature Protocol up to the root for the dataset of May, 19th. This can be explained by the fact that in this day the "udp" phenomenon in our network traffic is stronger than in the other ones. This phenomenon lasted for the first four days of the week, Monday to Thursday, and actually corresponded to a great activity of one of our computers as a peer-to-peer (P2P) system. For the last three days of the week the structure of the induced trees change: this actually corresponds to an important variation in the network traffic since the peer-to-peer system strongly reduced its activity.

We have performed the same experiments on the week June 9th-15th, 2003. The network configuration is completely different as to systems, services and firewall rules. The absence of udp packets in the traffic of June 2003 is evident in K-means trees, which start testing the StartHalfHour feature. As to EM trees, they still involve only numeric features and look similar to those of May 2004, meaning that the network traffic itself possesses regularities.

Finally, we observe that decision trees are very similar across the two different sets of features; the most notable difference is that K-means uses the NationCode feature in place of SourceIP.

For our experiments, we used a Pentium with 2GB Ram, 60 Gb Hd, CPU 3,06 GHz. The size of processed files varies from 900kb to 3Mb and the clustering phase needs almost 20 minutes for every file. The tree elaboration was always very fast.

4 Conclusions and future work

In this work, we reported a preliminary study on the application of data mining tools to real network traffic data with the aim of capturing daily regularities as well as changes occurring with time. Two different clustering methods have been tested. They build different models of the network traffic but both provides useful information on the nature of network traffic. In our analysis we have used general features extracted from firewall logs plus an additional feature (NationCode) obtained through the *whois*-service.

Future work will proceed along several directions. First, we want to define a formal measure of similarity for decision trees of rules sets derived from them, so that daily models of network connections can be properly compared. The similarity measure will take into account the background knowledge of the network officer, such as udp connections have length zero. Second, we intend to experiment additional descriptive data mining tasks, such as association rules and frequent episodes, which can capture additional regularities not disclosed by clustering algorithms. Finally, we intend to exploit the model of regular network traffic to detect anomalies.

References

[1] Porras, P.A., Valdes, A.. Live traffic Analysis of TCP/IP Gateways. In Proceedings of the ISOC Symposium on Network and Distributed Systems Security, 1998.

[2] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. Advances in knowledge discovery and data mining. AAAI Press/ The MIT Press, 1996.

[3] Portnoy, L., Eskin, E., Stolfo, S.J. Intrusion detection with unlabeled data using clustering. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001), Philadelphia, PA, 2001.

[4] http://kdd.ics.uci.edu/databases/kddcup99/Kddcup99.html, 1999.

[5] Lazarevic, A., Srivastava, J., Kumar, V. Data Mining for Intrusion Detection. Tutorial on the Pacific Asia-Conference on Knowledge Discovery in Databases, 2003.

[6] Ertoz, L., Eilerton, E., Lazarevic, A., Tan, P., Dokas, P., Kumar, V., Srivastava, J. Detection of Novel Network Attacks using data mining. Workshop on Data Mining for Computer Security (DMSEC '03), 2003.