Network Attack Detection Based on Peer-to-Peer Clustering of SNMP Data*

Walter Cerroni, Gabriele Monti, Gianluca Moro, and Marco Ramilli

DEIS – University of Bologna, v. Venezia 52, 47521 Cesena (FC), Italy
{walter.cerroni,gabriele.monti4,gianluca.moro,marco.ramilli}@unibo.it

Abstract. Network intrusion detection is a key security issue that can be tackled by means of different approaches. This paper describes a novel methodology for network attack detection based on the use of data mining techniques to process traffic information collected by a monitoring station from a set of hosts using the Simple Network Management Protocol (SNMP). The proposed approach, adopting unsupervised clustering techniques, allows to effectively distinguish normal traffic behavior from malicious network activity and to determine with very good accuracy what kind of attack is being perpetrated. Several monitoring stations are then interconnected according to any peer-to-peer network in order to share the knowledge base acquired with the proposed methodology, thus increasing the detection capabilities. An experimental test-bed has been implemented, which reproduces the case of a real web server under several attack techniques. Results of the experiments show the effectiveness of the proposed solution, with no detection failures of true attacks and very low false-positive rates (i.e. false alarms).

Keywords: Network security, distributed intrusion detection, SNMP, data mining, data clustering, peer-to-peer.

1 Introduction

Network security is one of today's most important issues that must be dealt with by system engineers in their everyday work as well as by the research community. In particular, the problem of detecting malicious network traffic and promptly trigger alerts and/or suitable countermeasures has been widely studied in the last decade and is still of high interest. To this purpose, Network-based Intrusion Detection Systems (NIDSs) [7] have been developed with the ability to analyze network traffic, detect possible attacks and notify the network administrators. The NIDS operations are executed according to two possible approaches, respectively signature-based and anomaly-based [15].

The first approach relies on the idea that, by comparing well-known malicious network behaviors with the current network activity by means of traffic signatures, it is possible to detect the presence of harmful traffic with a good

* Work partially funded by the european project DORII: Deployment of Remote Instrumentation Infrastructure Grant agreement no. 213110.

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 417–430, 2009.

level of confidence and reliability. Unfortunately, signature-based schemes suffer from the so-called synonymous attack, where the attacker is able to bypass the signature check by using a different stream pattern with the same harmful meaning.

Anomaly-based NIDSs, on the other hand, are capable of detecting a threat by looking at the specific behavior of the network traffic: what is known is considered as "normal" activity, whereas any behavior that differs from normal traffic is considered as anomaly. The most difficult challenge of these systems is to figure out what is actually normal activity and what is not. In particular, this approach becomes tricky and very difficult to apply to networks characterized by heterogeneous user behaviors and highly variable traffic patterns.

A significant research effort has been spent in the last few years with the objective of increasing NIDS efficiency. For instance, by applying fuzzy logic to intrusion detection [8] [3] or by adopting an approach based on artificial neural network [30] [16]. Other solutions include the use of an agent-oriented paradigm to build a multi-agent system able to detect threats [31] or the development of an embedded NIDS inside a Network Interface Card (NIC) [11]. Finally other studies focused their attention on software engineering aspects of intrusion detection [26].

A common assumption made in most of the published work on NIDS is the analysis of network traffic through raw packet capture techniques. However, this is a very critical aspect, since packet-by-packet analysis may become a system bottleneck in case of very large traffic volumes. In fact, some packet sampling techniques have been recently investigated [19] [1] that are seeking a trade-off between detection accuracy and performance. In some cases, using raw packets it is not even possible to distinguish normal traffic from Denial of Service (DoS) attacks [22].

A viable alternative to raw traffic capture performed by NIDSs is the use of the monitoring facilities provided by the Simple Network Management Protocol (SNMP), the de-facto standard adopted in Network Management Systems (NMS) [12]. A first proposal for a methodology that integrates NMS and NIDS has been introduced with reference to proactive detection of Distributed Denial of Service (DDoS) attacks [4]. Other studies include anomaly detection using signal processing techniques on SNMP data [25] and SNMP-based traffic flooding attack detection [29].

The contribution of this paper is to follow a new approach based on data mining techniques, in particular applying data clustering to information collected through SNMP. In this context, one of the data clustering peculiarities is the capacity to perform successful detection without a training phase, which, instead, is required by all the supervised techniques, such as the popular ones based on decision trees. The training phase is a costly and time-consuming activity because a significant amount of data must be correctly classified in advance by human experts.

The experiments performed on a test-bed using real traffic traces show that the proposed methodology is capable of detecting many different network attacks,

such as DoS, DDoS and several flavors of TCP port scanning, with a very high accuracy, with no detection failures of true attacks and very low false alarm rates. In addition, a thorough analysis driven by the adopted data mining approach allows to understand which pieces of information collected through SNMP are really essential for attack detection.

The paper is organized as follows. Section 2 provides a brief overview of data clustering techniques, with particular reference to the k-means approach used in this paper. Then section 3 describes the architecture and operations of the proposed framework, followed by section 4 which presents the experimental test-bed and the obtained results. Finally, section 5 concludes the work.

2 Background on Data Clustering

Data mining is the extraction of implicit, previously unknown, and potentially useful information from data [10]. Data mining techniques are usually divided in unsupervised and supervised according to the learning (i.e. the information extraction) method adopted. The unsupervised mining, differently from the supervised one, do not require training phases saving the classification cost performed by human experts to define a valid training set.

The goal of data clustering, which is the unsupervised technique included in our framework, is to extract new potential useful knowledge from a generally large data set by grouping together similar data items and by separating dissimilar ones according to some defined dissimilarity measure among the data items themselves. The literature on data clustering offers a large number of algorithms, generally grouped in hierarchic (e.g. BIRCH [32]), density-based approaches (e.g. DENCLUE [13]), linkage-based, statistics-based methods and partitive solutions (e.g. k-means [18]).

The hierarchic methods can be further divided in agglomerative (i.e. bottom-up) or divisive (i.e. top-down), according to how the algorithms begin the formation of groups, namely with each element as a separate cluster which is gradually merged into successively larger clusters, or alternatively dividing the whole set into successively smaller clusters.

In density-based approaches the idea is that similarity is expected to be high in densely populated regions of the given data set. Consequently, searching for clusters may be reduced to searching for dense regions of the data space separated by regions of relatively lower density. Popular methods in this class have been investigated in the context of non-parametric density estimation [24] and data mining [9] [13] [28].

Partitive approaches, in particular k-means that has been used in the proposed framework test-bed, aim to partition observations into k clusters specifying randomly in advance k centroids (cluster centers). Each observation is associated to its closest centroid according to a distance metric and then each centroid updates its position according to its associated observations; the process iterates until the k centroids no longer change their positions. Once the iteration stops, each point is assigned to its nearest cluster center, so the overall effect is to minimize the

total squared distance from all points to their cluster centers. In general this is a local minimum and the final result depends on the initial position of k centroids, however there are valid heuristics to select their positions to achieve suboptimal solutions [2]. In general it is almost infeasible to find globally optimal clusters with any kind of clustering algorithms.

A description of each group of solutions above mentioned, which is beyond the scope of this paper, is available in [27].

The data clustering problem has been investigated also in the distributed setting where data cannot be concentrated on a single machine, for instance because of privacy reasons or due to network bandwidth limitations, or because of the huge amount of distributed data. Several algorithms have been developed for distributed data clustering [14] [17] [23]. A common scheme underlying all approaches is to first locally extract suitable aggregates, then send the aggregates to a central site where they are processed and combined into a global approximate model. The kind of aggregates and combination algorithm depend on the data types and distributed environment under consideration, e.g. homogeneous or heterogeneous data, numeric or categorical data.

A k-means algorithm for clustering data distributed over a large, dynamic network is presented in [6], suited for overlay peer-to-peer systems [21] [20]. The algorithm requires only local communication and synchronization at each iteration, namely each node cooperates only with its neighboring nodes. Authors achieved high accuracy levels with less than 3% on average of misclassified with respect to the centralized version of k-means.

3 Description of the Proposed Framework

The reference scenario considered in the attack detection approach proposed in this paper is sketched in Fig. 1. The basic idea is to have several monitoring stations to share their knowledge of the traffic behaviors and their attack detection capabilities according to any peer-to-peer (P2P) collaborative paradigm; namely according to any unstructured or structured P2P overlay network. Each monitor is based on a standard SNMP management station configured to collect traffic data from a number of SNMP agents running on hosts, servers, workstations, laptops, etc. This is a very common situation, as most organizations are using SNMP to manage their networks.

Data collected from SNMP agents are represented as objects according to a standard language (ASN.1) and organized in a tree-structured database called Management Information Base (MIB). Each MIB object provides information about the corresponding feature being managed, e.g. the number of packets received on a network interface, the amount of disk space available on a server, the availability of a given service and so on. In particular, for the purpose of the methodology presented here, the MIB objects related to IP and TCP are considered.

Besides the typical network management tasks that may or may not be implemented, each monitoring station uses the queried MIB objects to extract its

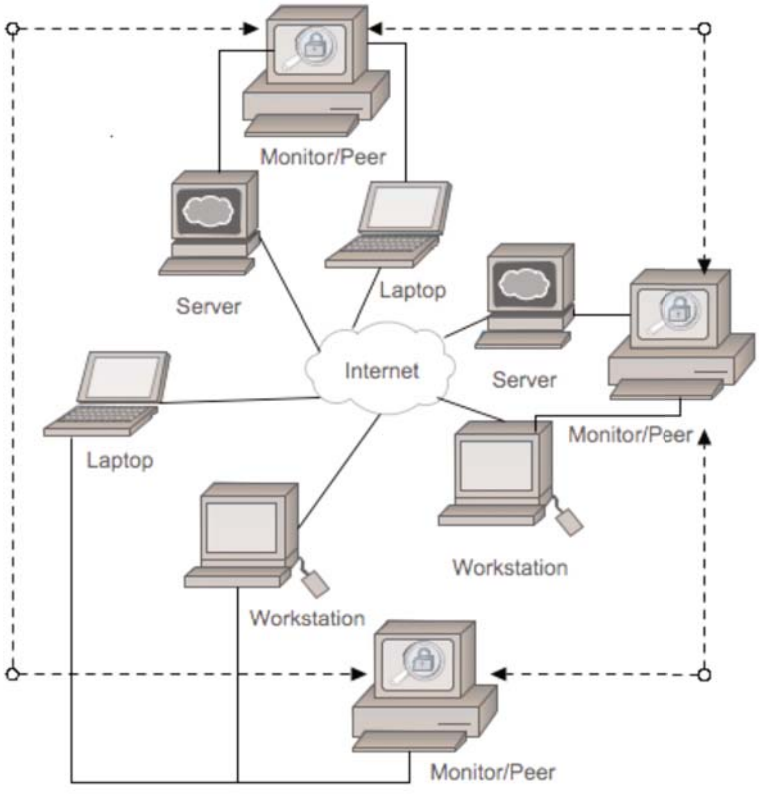

Fig. 1. Reference scenario of the SNMP-based attack detection framework

knowledge of the network behavior, according to which it is able to distinguish normal traffic and several kinds of attack. This process is performed by applying a clustering algorithm to observations whose schema (i.e. the relevant variables) and the corresponding instances are derived from collected SNMP data. The knowledge (i.e. the data clustering model) is represented as a set of centroids (i.e. cluster centers), therefore the memory required is less than a couple of Kilobytes and this guarantees high detection efficiency once the clustering model is applied to new incoming SNMP data. In other words, the framework can work in real-time manner.

Each monitor then periodically collects the content of the network-related MIB objects and process them by updating the clustering model in background. The model updates improve the detection accuracy. In fact an increasing effectiveness has been observed in the test-bed described in the next section when the SNMP data set becomes larger.

Figure 2 shows the methodology adopted by the implemented framework working as a monitor. The software running on the monitor machine reads the SNMP TCP stack information from the monitored machines and generates the

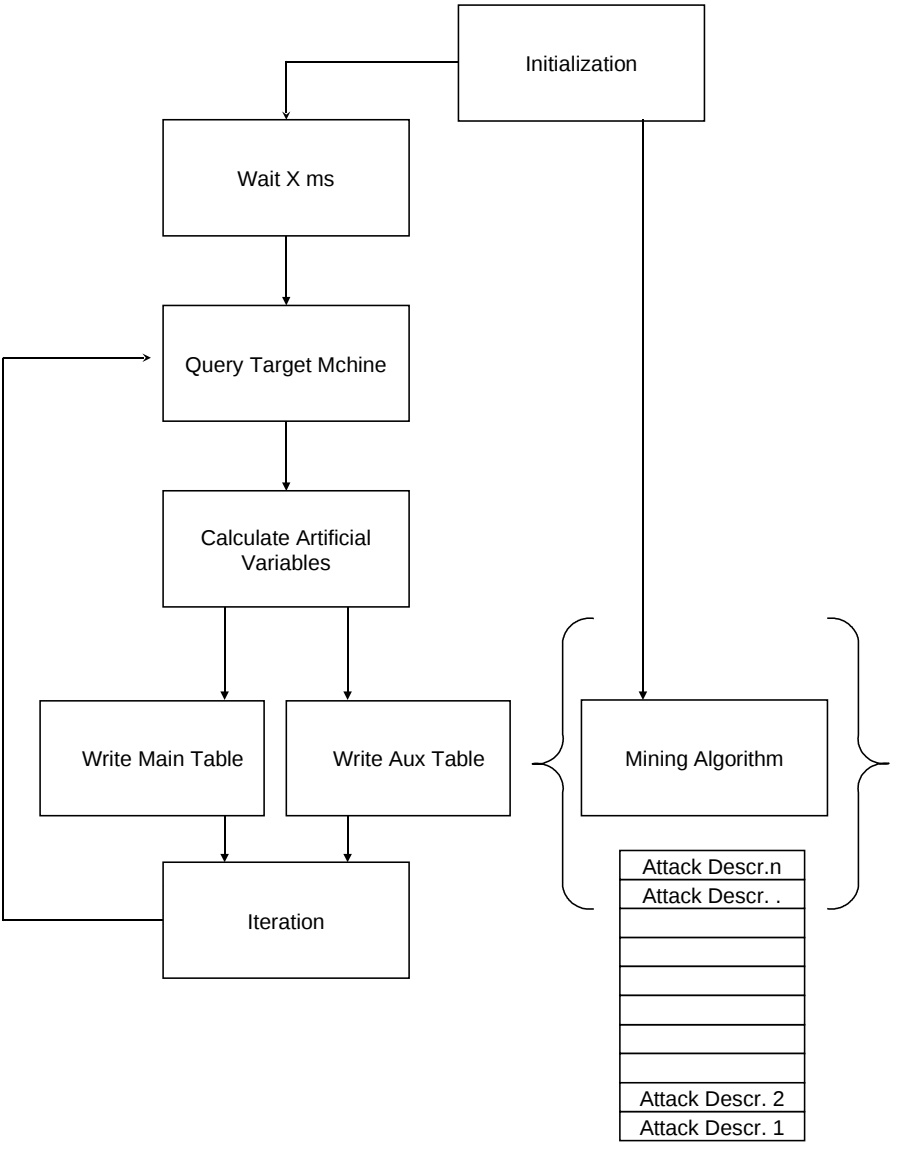

Fig. 2. Logical software flow

observations producing two tables used by the data clustering algorithm. One table, called the *main table*, includes general TCP stack information as well as some computed variables, both useful to discern attacks from normal traffic. The other table, called the *auxiliary table*, summarizes some correlations from the main table related to connected hosts and connection details, useful to differentiate attacks.

Table 1 reports the list of relevant variables of each observation that have proved to guarantee high and stable levels of detection accuracy; we have defined them by evaluating combinations of hundreds of SNMP parameters by using several mining techniques for feature extractions. This scheme of variables determines a multi-dimensional space, where each variable represents a dimension and each observation is a point whose coordinates correspond to the variable values.

Table 1. Relevant Clustering Variables Derived from SNMP Data

Features Derived from SNMP Data
Number of processes in TCP listen state
Number of open TCP connections (any possible TCP state)
Number of TCP connections in time-wait state
Number of TCP connections in established state
Number of TCP connections in SYN-received state
Number of TCP connections in FIN-wait state
Number of different remote IP addresses with an open TCP connection
Remote IP address with the highest number of TCP connections
Remote IP address with the second highest number of TCP connections
Remote IP address with the third number of TCP connections
Local TCP port with the highest number of connections
Number of connections to the preceding TCP port
Local TCP port with the second highest number of connections
Number of TCP RST segments sent out

More specifically, the clustering algorithm we used in the test-bed of this framework is the k-means, introduced in the previous section. The number of clusters specified in advance must be two or more in order to learn a model able to at least discern normal traffic from attacks. In general, the number of clusters should correspond to the number of different attacks to be detected plus one. However, it is important to clarify that the clustering model does not indicate which cluster corresponds to which attack. This meaning association occurs by interpreting the knowledge discovered. Anyway, the same kind of attacks perpetrated against different machines using the same features, like those we have introduced above, become points which fall in the same zone of the multi-dimensional space, leading naturally to similar clusters everywhere in the P2P network.

With our framework this convergence of clusters in the P2P network is further enhanced thanks to the collaboration among peers. In fact, each peer, i.e. each monitoring station, may share with one or more neighbours its observations, which do not represent any network transmission content, or it may simply share its knowledge, namely its local cluster centroids with its cardinality (i.e. the number of associated observations). In the latter case, the traffic among peers is almost negligible since it corresponds to less than a couple of Kilobytes.

Moreover, the frequency of this information exchange is as low as the number of times the local knowledge is updated, therefore even the sharing of observations is a practicable method. The observations coming from one or more neighbours are simply added to the local ones in order to contribute to improve the next update of the local clustering model. The same happens with the transmission of cluster centroids. In the first case the clustering algorithm behaves as usual updating its centroids using the new observations together with its local ones, while in the second case its local centroids are updated according to the weight (i.e. the cardinality) of the received centroids as well.

4 Test-Bed Setup and Results

To prove the feasibility and accuracy of the proposed network attack detection metodology, an experimental test-bed has been set up emulating a typical situation where some standard web servers might be under attack. Fig. 3 shows the particular scenario where a machine controls what is happening on the monitored server. This scenario has been reproduced ten times to collect collaborative data of ten distinct servers for a real consistent experiment. The web server is connected to a De-Militarized Zone (DMZ), whereas legitimate clients as well as attacking hosts from the external network are able to contact the server through a router. The monitoring station is connected to the server through a separate private network, which is also a typical network management situation where the monitoring and management traffic is kept safe and isolated from the public Internet, e.g. on a dedicated VLAN.

Another machine has been used to generate synthetic traffic replicating requests directed to a web server according to real traces collected on a public

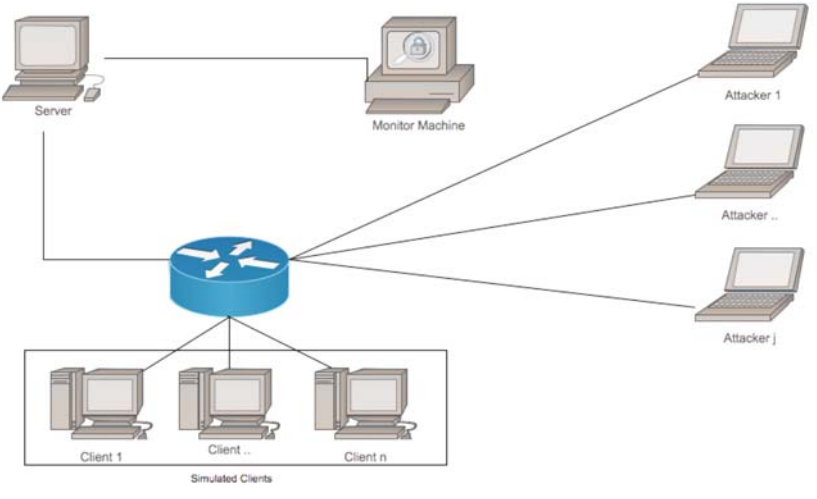

Fig. 3. Test-bed setup: One cell to which corresponds one peer

backbone link [5]. With this approach it was possible to emulate the timings of a realistic and significant amount of HTTP traffic under a controlled environment. A bunch of additional machines has then been set up to perform several different kinds of network attacks:

1. Denial of Service
2. Distributed Denial of Service
3. TCP Port Scanning using different techniques: FIN, SYN, ACK, WINDOW, NULL, XMAS
4. SSH Denial of Service
5. SSH Brute Force

The experiment has been executed in five different sessions, plus a session of normal traffic only. For each session, 1000 samples of the network-related MIBs have been collected and stored in the main and auxiliary tables on the local file system. The tables have been further processed to include, besides the natural SNMP MIBs, some more specific information according to Table 1. All these variables are useful to figure out which host might attack the monitored system. Once the monitor has collected enough data, it is ready to communicate its results to other monitor peers. The communication can be performed by sending all collected data or the learned models only, as previously explained.

The following section describes the results of all possible scenarios emulated through different kinds of simulations, such as: a non collaborative host, two collaborative hosts, three collaborative hosts and so forth until nine collaborative hosts for each peer.

4.1 Results

To validate the results, in order to measure the efficacy of our framework, the observations generated from SNMP data, have been labeled according to the belonging network attack session, including the one of normal traffic, as mentioned in the previous section.

We highlight that the observation labeling has been totally ignored by the data clustering algorithm during the model learning, just because the approach is not supervised. The labels have been used only in test phases to compute the efficacy of the clustering model in the following two cases:

1. for discerning attacks from normal traffic, without distinguishing the kind of attack;
2. for detecting even the kind of attack together with the normal traffic.

Formally the accuracy is defined as follows:

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP is the number of true positive observations, namely the number of attacks correctly detected as attacks, TN is the number of observations correctly

detected as normal traffic, FP regards the false positives, that is the amount of normal traffic erroneously detected as attacks and finally the false negative, namely attacks wrongly interpreted as normal traffic.

Moreover the following rates represent the incidence of false alarms and of undetected alarms (i.e. detection failures of true attacks):

$$\frac{FP}{TN + FN} \tag{2}$$

$$\frac{FN}{TP + FP} \tag{3}$$

$$\frac{FP}{TP + FP} \tag{4}$$

Usually the variables of expression 1 are represented in a squared matrix, called *confusion matrix*, in which the numerators are along a diagonal, moreover FP and FN are in the same row with TP and TN respectively. In the two kinds of test phases above mentioned, we have computed a series of both 2x2 and 6x6 confusion matrixes respectively.

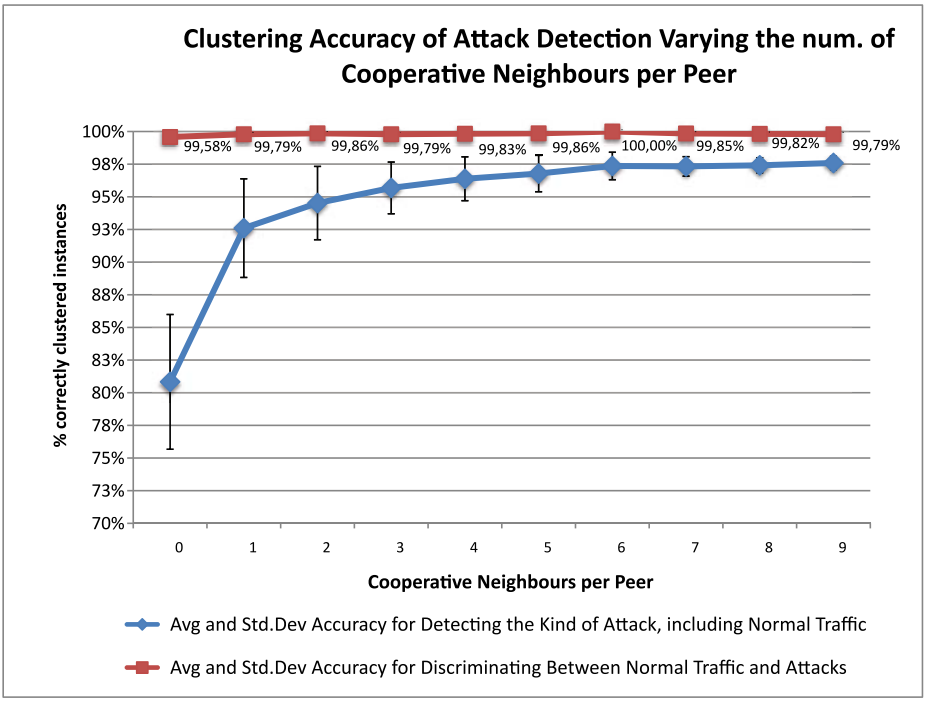

Fig. 4. Detection Accuracy Based on SNMP Data Clustering Varying the num. of Cooperative Neighbours per Peer

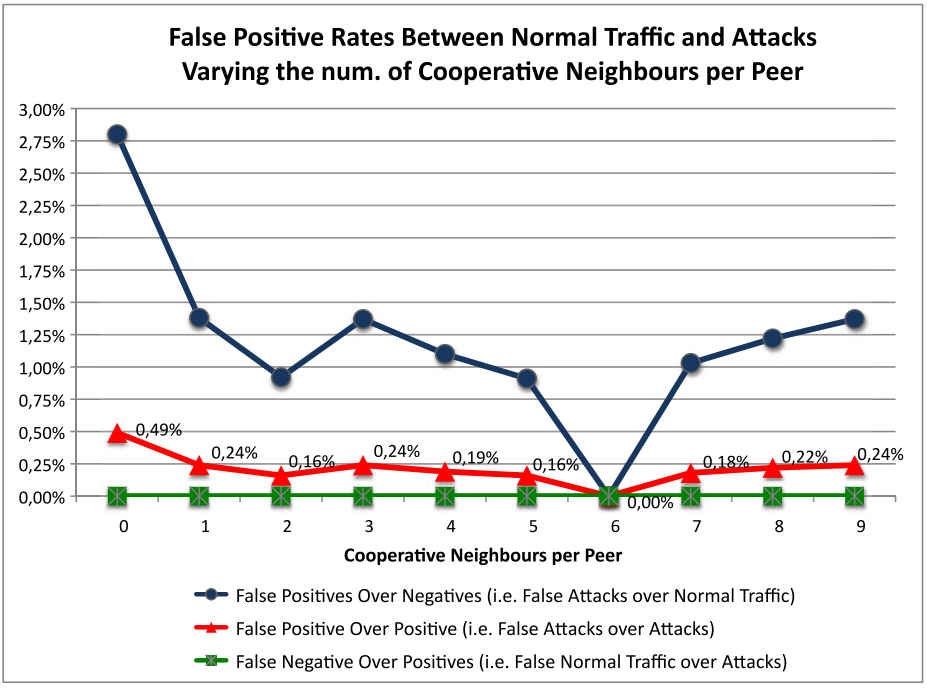

Fig. 5. False Positive Rates Between Normal Traffic and Attacks To Measure False Alarms Varying the num. of Cooperative Neighbours per Peer

Experiments and corresponding measures have been performed by varying, for each peer, the number of collaborative neighbours from zero (i.e. no collaboration) to nine, moreover the same experiment has been repeated ten times with different random seed in order to compute average values and standard deviations.

Figure 4 shows the two series of accuracy corresponding to the two test phases above mentioned. In both series the greatest accuracy increase occurs from zero to one collaborative neighbour. The accuracy in the experiments regarding the discerning between normal traffic and attacks is in the worst case 99.58%, while in the best case is 100%. In the experiments for the detection also of the kind of attack, the worst accuracy is 80.8% without any collaboration, while the best one is 97.6% with nine collaborative neighbours. Another interesting result is that the standard deviations of both series decrease by increasing the number of collaborative neighbours.

Figure 5 illustrates the rates about false alarms and undetected attacks according to the expressions 2, 3 and 4 varying the number of cooperative neighbours per peer. The first important results is that, according to expression 4, the rate of false negative over positives, namely the undetected attacks, is always zero. Moreover the false allarms, corresponding to normal traffic erroneously

detected as attacks, decreases drastically from 2.80%, which is the worst result, to 1.38% with only one cooperative neighbour; this rate on average is 1.21% and its best value is 0%. The incidence of false attacks over attacks is always less than 0.5%.

Finally, in the test-bed, we have observed that the accuracy of clustering models is very well preserved over new incoming observations. In fact, the loss of accuracy, in case of missing model updates, is on average only 0.39%, when the amount of new observations, generated from new network traffic, is greater than an order of magnitude of the cardinality of the data set from which the clustering model has been generated.

5 Conclusion

This paper described a novel methodology for network attack detection based on data mining of traffic information collected via SNMP by multiple monitoring stations, which are organized in a peer-to-peer network with the purpose of sharing the gained knowledge. In particular, the use of unsupervised clustering techniques on network-specific MIB objects allows to effectively detect malicious network behaviors, such as the ones due to DoS, DDoS and port scanning attacks, while still distinguishing between normal and harmful traffic profiles with very high accuracy.

Experimental results, obtained by emulating the real traffic of ten web servers under several kinds of attack, demonstrated the effectiveness of the proposed solution, reaching high accuracy levels with no detection failures and a false-positive rate as low as 1.21% on average. The accuracy levels of discerning normal and harmful traffic is on average greater than 99.58%. Moreover the detection accuracy can be increased by increasing the number of collaborative neighbours per peer, particularly the accuracy of identifying also the kind of attack.

Finally, the experiments highlighted that the loss of detection accuracy of not updated clustering models, over new incoming observations, is on average only 0.39%, after that the amount of the new SNMP traffic is an order of magnitude greater than the one used to learn the corresponding model.

Such promising results will be the basis to extend the current work to more complex network scenarios, where experiments will be conducted on SNMP traffic collected from a larger set of heterogeneous hosts and servers as well as from interconnecting equipment such as routers and switches.

References

1. Androulidakis, G., Chatzigiannakis, V., Papavassiliou, S.: Network anomaly detection and classification via opportunistic sampling. IEEE Network 23(1), 6–12 (2009)
2. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: Proceedings of the 15th International Conference on Machine Learning (ICML 1998), pp. 91–99. Morgan kaufmann, San Francisco (1998)

3. Bridges, S.M., Vaughn, R.B.: Fuzzy data mining and genetic algorithms applied to intrusion detection. In: Proceedings of the National Information Systems Security Conference (NISSC), pp. 16–19 (2000)
4. Cabrera, J.B.D., Lewis, J.L., Qin, X., Lee, W., Mehra, R.K.: Proactive intrusion detection and distributed denial of service attacks—a case study in security management. Journal of Network System Management 10(2), 225–254 (2002)
5. CAIDA. The cooperative association for internet data analysis passive monitor (May 2009), `http://www.caida.org/data/monitors/passive-equinix-chicago.xml`
6. Datta, S., Giannella, C.R., Kargupta, H.: Approximate distributed k-means clustering over a peer-to-peer network. IEEE Transactions on Knowledge and Data Engineering 21(10), 1372–1388 (2009)
7. Denning, D.E.: An intrusion-detection model. IEEE Transactions on Software Engineering 13(2), 222–232 (1987)
8. Dickerson, J.E., Dickerson, J.A.: Fuzzy network profiling for intrusion detection. In: Proc. of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society, Atlanta, pp. 301–306 (2000)
9. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD 1996 Proceedings, pp. 226–231. AAAI Press, Menlo Park (1996)
10. Frawley, W.J., Piatetsky-shapiro, G., Matheus, C.J.: Knowledge discovery in databases: an overview. AAAI Press, Menlo Park (1992)
11. Ghoting, O.P., Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S.: Towards NIC-based intrusion detection. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 723–728. ACM Press, New York (2003)
12. Harrington, D., Presuhn, R., Wijnen, B.: An architecture for describing simple network management protocol (SNMP) management frameworks. IETF RFC 3411 (2002)
13. Hinneburg, A., Hinneburg, E., Keim, D.A.: An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD 1998), pp. 58–65. AAAI Press, Menlo Park (1998)
14. Johnson, E.L., Kargupta, H.: Collective, hierarchical clustering from distributed, heterogeneous data. In: Large-Scale Parallel KDD Systems, SIGKDD, pp. 221–244. Springer, Heidelberg (1999)
15. Kabiri, P., Ghorbani, A.A.: Research on intrusion detection and response: A survey. International Journal of Network Security 1, 84–102 (2005)
16. Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I.: On the capability of an SOM based intrusion detection system. In: Proceedings of the International Joint Conference on Neural Networks, July 2003, vol. 3, pp. 1808–1813 (2003)
17. Klusch, M., Lodi, S., Moro, G.: Distributed clustering based on sampling local density estimates. In: Proceedings of the Biennal International Joint Conference on Artificial Intelligence, pp. 485–490. Morgan Kaufmann, San Francisco (2003)
18. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
19. Mai, J., Sridharan, A., Chuah, C.-N., Zang, H., Ye, T.: Impact of packet sampling on portscan detection. IEEE Journal on Selected Areas in Communications 24(12), 2285–2298 (2006)

20. Monti, G., Moro, G.: Multidimensional range query and load balancing in wireless ad hoc and sensor networks. In: Wehrle, K., Kellerer, W., Singhal, S.K., Steinmetz, R. (eds.) Peer-to-Peer Computing, pp. 205–214. IEEE Computer Society, Los Alamitos (2008)
21. Moro, G., Ouksel, A.M.: G-grid: A class of scalable and self-organizing data structures for multi-dimensional querying and content routing in P2P networks. In: Moro, G., Sartori, C., Singh, M.P. (eds.) AP2PC 2003. LNCS (LNAI), vol. 2872, pp. 123–137. Springer, Heidelberg (2004)
22. Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA 2001), pp. 5–8 (2001)
23. Costa Da Silva, J., Klusch, M., Lodi, S., Moro, G.: Privacy-preserving agent-based distributed data clustering. Web Intelligence and Agent Systems 4(2), 221–238 (2006)
24. Silverman, B.W.: Density estimation for statistics and data analysis. Chapman and Hall, London (1986)
25. Thottan, M., Ji, C.: Anomaly detection in IP networks. IEEE Transactions on Signal Processing 51(8), 2191–2204 (2003)
26. Vigna, G., Valeur, F., Kemmerer, R.A.: Designing and implementing a family of intrusion detection systems. SIGSOFT Software Engineering Notes 28(5), 88–97 (2003)
27. Xu, R., Wunsch II, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
28. Xu, X., Ester, M., Kriegel, H.-P., Sander, J.: A distribution-based clustering algorithm for mining in large spatial databases. In: Proceedings of the Fourteenth International Conference on Data Engineering (ICDE 1998), Washington, DC, USA, pp. 324–331. IEEE Computer Society, Los Alamitos (1998)
29. Yu, J., Lee, H., Kim, M.-S., Park, D.: Traffic flooding attack detection with SNMP MIB using SVM. Computer Communications 31(17), 4212–4219 (2008)
30. Zanero, S., Savaresi, S.M.: Unsupervised learning techniques for an intrusion detection system. In: Proceedings of the 2004 ACM symposium on Applied Computing (2004)
31. Zhang, R., Qian, D., Bao, C., Wu, W., Guo, X.: Multi-agent based intrusion detection architecture. In: Proceedings of the 2001 International Conference on Computer Networks and Mobile Computing (ICCNMC 2001), Washington, DC, USA, p. 494. IEEE Computer Society, Los Alamitos (2001)
32. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, Montreal, Canada, pp. 103–114 (1996)