Traffic Data Preparation for a Hybrid Network IDS

Álvaro Herrero, and Emilio Corchado

Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
{ahcosio, escorchado}@ubu.es

Abstract. An increasing effort has being devoted to researching on the field of Intrusion Detection Systems (IDS's). A wide variety of artificial intelligence techniques and paradigms have been applied to this challenging task in order to identify anomalous situations taking place within a computer network. Among these techniques is the neural network approach whose models (or most of them) have some difficulties in processing traffic data "on the fly". The present work addresses this weakness, emphasizing the importance of an appropriate segmentation of raw traffic data for a successful network intrusion detection relying on unsupervised neural models. In this paper, the presented neural model is embedded in a hybrid artificial intelligence IDS which integrates the case based reasoning and multiagent paradigms.

Keywords: Computer Network Security, Network Intrusion Detection, Artificial Neural Networks, Unsupervised Learning, Projection Methods, Artificial Intelligence, Hybrid Artificial Intelligence Systems.

1 Introduction

Intrusion Detection Systems (IDS's) [1] are becoming more and more popular as network security tools, although firewalls are still the most widely used tool of this kind. A network IDS (NIDS) monitor the activity of a network with the purpose of identifying intrusive events and can take actions to abort these risky events.

Up to now, a wide range of Artificial Intelligence (AI) techniques have been used to build IDS's. On the one hand, there have been some previous attempts to take advantage of agents and Multiagent Systems (MAS) [2] in the field of Intrusion Detection (ID) [3], [4], [5]. It is worth mentioning the mobile-agents approach [6], [7]. On the other hand, some different machine learning models – including Data Mining techniques and Artificial Neural Networks (ANN) – have been successfully applied for ID [8], [9], [10], [11], [12]. As can be seen, there have been a huge number of attempts to apply ANN to the detection of intrusions.

Additionally, some other AI techniques have been combined (genetic algorithms and fuzzy logic [13], genetic algorithms and K-Nearest Neighbor (K-NN) [14] or K-NN and ANN [15] among others) in order to face ID from a hybrid point of view. In some cases they provide intelligence to MAS. This paper employs the Mobile Visualization Connectionist Agent-Based IDS (MOVICAB-IDS) [2], a dynamic

multiagent architecture for network intrusion detection that includes deliberative agents capable of learning and evolving with the environment. Some previous work has been devoted to the managing and processing of network traffic data [16], [17], [18], [19].

The remaining four sections of this paper are structured as follows: traffic data management is outlined in section 2, section 3 contains an overview of MOVICAB-IDS; some experimental results are described in section 4; finally, section 5 shows a comparative study and puts forward a number of conclusions.

2 Traffic Data Preparation

As in the case of traffic management [16], we can divide ID into three well defined tasks:

1. Data collecting: prior to getting the data from the network, some issues must be addressed; selecting the sources from where the information is going to be extracted, specifying the parameters (time length, filters, etc.) of the data gathering, defining the final format of the collected data, etc.
2. Data processing: once the data is gathered, it is processed for the identification of intrusions and attacks.
3. Intrusion abortion: some IDS's (lately referred as Intrusion Prevention Systems) incorporate any mechanism for stopping intrusions at the very moment they are identified.

This paper focuses on the Data collecting task, taking into account the connectionist techniques applied in the processing task. For the data collecting task, a 4 stage framework is proposed. These four stages are described in the following sections.

Stage 1.- Data Capture

As this framework is proposed for a NIDS, a continual data flow must be managed. This data flow contains information about all the packets travelling along the network to be monitored. For this information to be captured, one of the network interfaces of a host within this network must be set up in promiscuous mode. Thus, this interface is able to "see" all the packets travelling along the network, whatever its destination host is.

Stage 2.- Data Selection

NIDS's have to deal with the practical problem of high volumes of quite diverse data [17]. To deal with the problem of high diversity, we propose the splitting of the traffic into different groups, taking into account the protocol (either UDP, TCP, ICMP…) over IP as TCP and UDP packets are quite different. Once the captured data is classified by the protocol over IP, it can be processed in quite different ways. There have been several approaches dealing with traffic data summarized by TCP

connections, such as the well-known KDD dataset [20]. On the contrary, we propose a packet-based approach, where each instance in the final datasets corresponds to a single packet. This packet-based approach has proven to be successful in the identification of some anomalies [2], [12].

Stage 3.- Segmentation

The two first stages do not deal with the problem of continuity in network traffic data. As it is said before, most neural models can not process traffic data "on the fly". Therefore, in order to overcome this shortcoming, we propose a way of creating limited datasets from this continuous data flow by segmenting it. Two kinds of segments are proposed:

- **Simple segments**. Each simple segment contains all the packets whose timestamps are between the initial and final time limits of the segment. There must be a time overlap between each pair of consecutive simple segments because anomalous situations could conceivably take place between simple segment S_x and S_{x+1} (where S_{x+1} is the next segment following S_x). In this case, it would be necessary to consider some packets twice in order to visualize the end of the anomalous situation and the evolution between simple segments.
- **Accumulated segments**. Each one of these segments contains several consecutive simple ones (removing the time overlap). The main considerations are, firstly, to present a long-term picture of the evolution of network traffic and, secondly, to allow the identification of attacks lasting longer than the length of a simple segment.

There are some key issues concerning segmentation that are quite important, such as the length (time duration) of the simple segments, the overlap time and the number of simple segments making up the accumulated segments among others.

Stage 4.- Data Preprocessing

Finally, the different datasets (simple and accumulated segments) must be preprocessed before presenting them to the neural model. In this case of network traffic data, it is not needed the application of some techniques such as denoisification, outlier detection, missing data managing and some others. Depending on the neural model to be applied, only data normalization is needed.

3 MOVICAB-IDS

For the experimental verification (see section 4) of the proposed data segmentation framework, MOVICAB-IDS, is proposed. It may be roughly defined as a hybrid NIDS formed of different software agents [21] that work in unison in order to detect anomalous situations by taking full advantage of an unsupervised connectionist model [22], [23]. The traffic data preparation is performed by MOVICAB-IDS in the following steps:

– 1st step.- Traffic Data Capture: packets travelling over the different network segments are captured.
– 2nd step.- Data Selection: the captured data is selected. A set of packets and features contained in the headers of the captured data is extracted from the raw network traffic.
– 3rd step.- Segmentation: the data stream is divided into simple and accumulated segments.

Once the data is ready, the detection of intrusions goes through:
– 4th step.- Data Analysis: after preprocessing the data, the Cooperative Maximum Likelihood Hebbian Learning (CMLHL) model [22], [23] is applied to analyse it (see section 3.1 for further details).
– 5th step.- Visualization: the projections of traffic data are presented to the network administrator for the supervision and monitoring (see section 5 for samples).

MOVICAB-IDS makes use of the hybrid approach to perform all these steps, combining the following paradigms:
- Multiagent systems (MAS) [2]: this NIDS employs deliberative agents capable of learning and evolving with the environment.
- Case-Based Reasoning (CBR) [24]: some of the agents contained in the previously described MAS are known as CBR-BDI agents [25] because they integrate the BDI (Believes, Desires and Intentions) [26] model and the CBR paradigm.
- Artificial Neural Networks: the connectionist approach fits the intrusion-detection problem mainly because it allows a system to learn empirically the input-output relationship between traffic data and its subsequent interpretation. The previously described CBR-BDI agents incorporate the CMLHL neural model [22], [23].

The combination of these paradigms let us take advantage of some of the properties of ANN (generalization that allows the identification of previously unseen attacks), CBR (learning from past experiences) and agents (reactivity, proactivity and sociability), making the ID task possible.

3.1 The Neural Model

The Data Analysis step previously mentioned is based on the use of the unsupervised neural model called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [22], [23]. It is based on Maximum Likelihood Hebbian Learning (MLHL) [22], [27]. Consider an N-dimensional input vector, $\mathbf{x}$, and an M-dimensional output vector, $\mathbf{y}$, with W_{ij} being the weight linking input j to output i and let η be the learning rate. MLHL can be expressed as:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \forall i \,. \qquad (\mathbf{1})$$

The activation (e_j) is fed back through the same weights and subtracted from the input:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \,. \tag{2}$$

Weight change:

$$\Delta W_{ij} = \eta . y_i . sign\left(e_j\right) | e_j |^{p-1} . \tag{3}$$

Where: η is the learning rate and p is a parameter related to the energy function.

Lateral connections [28], [23] have been derived from the Rectified Gaussian Distribution [29] and applied to the MLHL model. The resultant net (CMLHL) can find the independent factors of a data set but do so in a way that captures some type of global ordering in the data set. So, the final CMLHL model is as follows:

Feed forward step: Equation (1)

Lateral activation passing: $y_i(t+1) = \left[y_i(t) + \tau\left(b - Ay\right)\right]^+ .$ (4)

Feed back step: Equation (2)

Weight change: Equation (3)

Where: τ is the "strength" of the lateral connections, *b* is the bias parameter and *A* is a symmetric matrix used to modify the response to the data. Its effect is based on the relation between the distances among the output neurons.

4 Experimental Study

Among all the implemented network protocols, there are several of them that can be considered potentially dangerous, such as the Simple Network Management Protocol (SNMP) [30], ICMP (Internet Control Message Protocol), TFTP (Trivial File Transfer Protocol) and so on. SNMP was identified as one of the top five most vulnerable services by CISCO [31], specially the two first versions of this protocol that are the most widely used at present time. An attack based on this protocol may severely compromise the security of the whole network [32]. Thus, this research line is focused on the identification of anomalous situations concerning SNMP.

Most of the security tools focus their attention on attacks coming from the internet but attacks are just as likely to come from inside the network as from the outside, however. MOVICAB-IDS helps network administrators to identify one of the most dangerous set of inside attacks: those related to the SNMP.

SNMP was oriented to manage nodes in the Internet community [30]. That is, it is used to control routers, bridges, and other network elements, reading and writing a wide variety of information about the devices, such as operating system, version,

routing tables, default TTL (Time To Live) and so on. The Management Information Base (MIB) can be defined in broad terms as the database used by SNMP to store information about the elements that it controls.

Three main anomalous situations related to the SNMP are analyzed in this section, namely: network/port scans, SNMP community search and MIB information transfers. These SNMP-related anomalous situations are described in this section.

Network Scan

A network scan may be seen as series of messages sent to the same port number of different host to gain information on protocols/services activity status. These messages can be sent by an external agent attempting to access a host to find out more about the network services running. A network scan provides information on where to probe for weaknesses, for which reason scanning generally precedes any further intrusive activity [33].

SNMP Community Search

The community string can be seen as the SNMP password for versions 1 and 2. This anomalous situation is characterized by the intruder sending SNMP queries to determine the SNMP community string. Once the community string has been obtained, all the sensitive information stored in the MIB is available for the intruder.

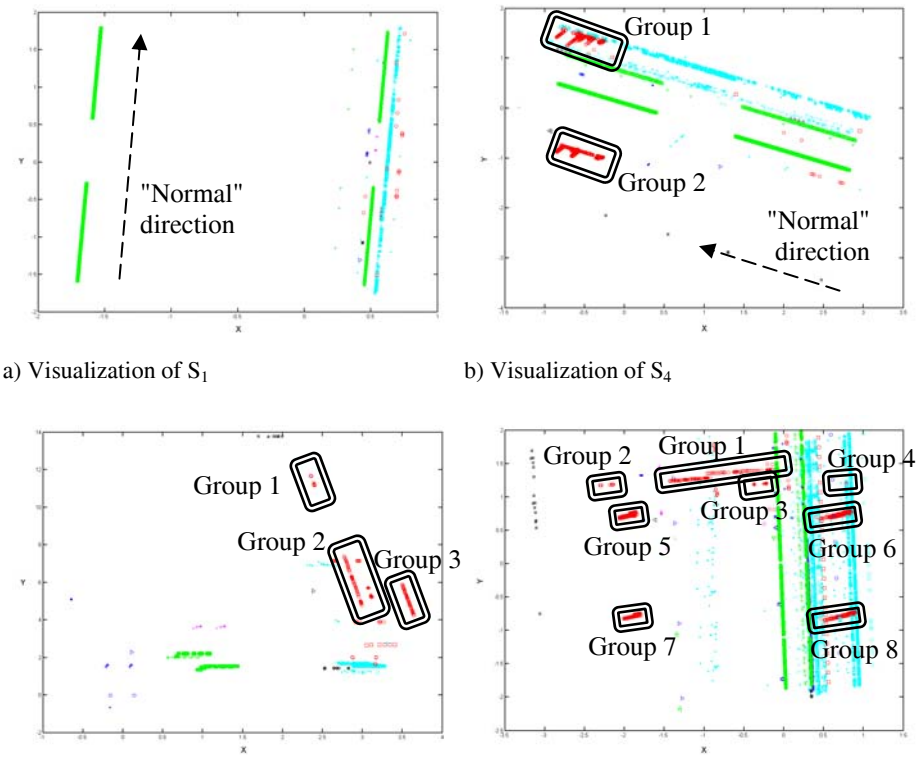
MIB Information Transfer

This situation is a transfer of some (or all the) information contained in the SNMP MIB. This kind of transfer is potentially quite a dangerous situation, but, on the other hand, the "normal" behaviour in the network includes queries to the MIB.

The segments analyzed in this section contain examples of all these anomalous situations. As SNMP is based on UDP, this section only deals with UDP traffic. Apart from the anomalous traffic previously described, information concerning normal traffic from a middle-size network is included as well. The MOVICAB-IDS visualization of the following four segments is depicted in Fig. 1:

- S_1: this simple segment does only contain normal data. The segment length is 600 seconds.
- S_4: this simple segment contains an MIB information transfer as well as normal data. The segment length is 600 seconds.
- A_3: this accumulated segment contains several network scans and SNMP community searches as well as normal data. The segment length is 1560 seconds.
- A_{13}: this accumulated segment contains several network scans, SNMP community searches and two MIB information transfers as well as normal data. The segment length is 6360 seconds.

All the packets contained in the three first segments (S_1, S_3, A_3) are also contained in A_{13}.



a) Visualization of S_1

b) Visualization of S_4

c) Visualization of A_3

d) Visualization of A_{13}

Fig. 1. Visualizations obtained by MOVICAB-IDS.

As can be seen in Fig. 1.a, normal traffic is depicted by MOVICAB-IDS as parallel straight lines. That is, all the packets (associated by protocol) evolve in the same direction (labelled in Fig. 1.a as "Normal" direction). In the case of an MIB information transfer, it can be identified in Fig. 1.b (labelled in Fig. 1.b as Groups 1 and 2) due to the high concentration of packets and the evolution of these packets in directions non-parallel to the one for normal traffic. It is easy to identify anomalous situations within simple segments.

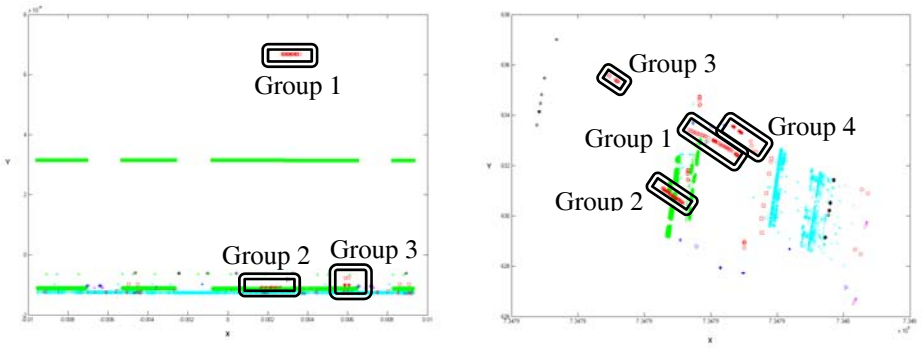
The other two anomalous situations, namely network scans and SNMP community searches are identified as anomalous in Fig. 1.c (Groups 1, 2 and 3). The non-parallel evolution of these groups enables us to label them as anomalous. It can be checked once again in Fig. 1.d where examples of all the three anomalous situations under study can be identified as anomalous due to their non-parallel evolution and high temporal concentration of packets: network scans (Group 1 in Fig. 1.d), SNMP community search (Group 2, 3 and 4 in Fig. 1.d) and the MIB information transfers

(Groups 5, 6, 7 and 8 in Fig. 1.d). In this case, as the analysed dataset is much bigger, the identification of some anomalous situations (specially the cases of the community search) is not as easy as in the other smaller segments.

5 Comparative Study and Conclusions

For comparison purposes, two well-known projection neural models, namely Principal Component Analysis (PCA) and Curvilinear Component Analysis (CCA) were applied to A_3 segment (Fig. 2). It also validates the proposed framework for traffic data preparation. As can be seen in Fig 2.a, PCA failed to detect the anomalous situations (network scans and SNMP community search) contained in A_3. Both the network scans (Groups 1 and 2 in Fig. 2.a) and the SNMP community searches (Group 3 in Fig. 2.a) contained in A_3 are not identified as anomalous traffic as these packets evolve in parallel lines.

Fig. 2.b shows the projection of A_3 obtained by CCA using Euclidean distance. The anomalies are differentiated from normal traffic on the basis of parallel evolution as in the case of CMLHL. The network scans (Groups 1 and 2 in Fig. 2.b) together with the SNMP community searches (Groups 3 and 4 in Fig. 2.b) are depicted as lines non parallel to normal traffic. The main difference between the CMLHL projection and the one obtained by CCA is that CMLHL minimizes the overlapping between the different groups, getting clearer and more sparse projections.



a) PCA projection of A_3 b) CCA projection of A_3

Fig. 2. Comparison of A3 projections

We can conclude that a proper traffic data preparation enables a successful network intrusion detection relying on neural models. However, it does not ensure a successful identification of attacks as the applied neural model is decisive as well. On the other hand, there are some issues concerning segmentation whose importance is worth mentioning. That is the case of the length (time duration) of the simple segments, as the longer the segments are, the more likely an attack will be unnoticed.

Acknowledgments. This research has been partially supported by the project BU006A08 of the Junta de Castilla y León.

References

1. Anderson, J.P.: Computer Security Threat Monitoring and Surveillance. Technical Report, Washington, PA, James P. Anderson Co, (1980)
2. Herrero, Á., Corchado, E., Pellicer, M., Abraham, A.: Hybrid Multi Agent-Neural Network Intrusion Detection with Mobile Visualization. In: Innovations in Hybrid Intelligent Systems. Advances in Soft Computing, vol. 44, pp. 320-328. Springer, Heidelberg (2007)
3. Spafford, E.H., Zamboni, D.: Intrusion Detection Using Autonomous Agents. Computer Networks: The Int. Journal of Computer and Telecommunications Networking 34(4), 547-570 (2000)
4. Hegazy, I.M., Al-Arif, T., Fayed, Z.T., Faheem, H.M.: A Multi-agent Based System for Intrusion Detection. IEEE Potentials 22(4), 28-31 (2003)
5. Dasgupta, D., Gonzalez, F., Yallapu, K., Gomez, J., Yarramsettii, R.: CIDS: An agent-based intrusion detection system. Computers & Security 24(5), 387-398 (2005)
6. Wang, H.Q., Wang, Z.Q., Zhao, Q., Wang, G.F., Zheng, R.J., Liu, D.X.: Mobile Agents for Network Intrusion Resistance. In: APWeb 2006. LNCS, vol. 3842, pp. 965-970. Springer, Heidelberg (2006)
7. Deeter, K., Singh, K., Wilson, S., Filipozzi, L., Vuong, S.: APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection System. In: Mobility Aware Technologies and Applications. LNCS, vol. 3284, pp. 244-253. Springer, Heidelberg (2004)
8. Laskov, P., Dussel, P., Schafer, C., Rieck, K.: Learning Intrusion Detection: Supervised or Unsupervised? In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 50-57. Springer, Heidelberg (2005)
9. Liao, Y.H., Vemuri, V.R.: Use of K-Nearest Neighbor Classifier for Intrusion Detection. Computers & Security 21(5), 439-448 (2002)
10. Sarasamma, S.T., Zhu, Q.M.A., Huff, J.: Hierarchical Kohonenen Net for Anomaly Detection in Network Security. IEEE Transactions on Systems Man and Cybernetics, Part B 35(2), 302-312 (2005)
11. Zanero, S., Savaresi, S.: Unsupervised Learning Techniques for an Intrusion Detection System. In: Proc. of the ACM Symposium on Applied Computing. pp. 412-419 (2004)
12. Corchado, E., Herrero, A., Sáiz, J.M.: Detecting Compounded Anomalous SNMP Situations Using Cooperative Unsupervised Pattern Recognition. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3697, pp. 905-910. Springer, Heidelberg (2005)
13. Sindhu, S.S.S., Ramasubramanian, P., Kannan, A.: Intelligent Multi-agent Based Genetic Fuzzy Ensemble Network Intrusion Detection. In: Neural Information Processing. LNCS, pp. 983-988. Springer, Heidelberg (2004)
14. Middlemiss, M., Dick, G.: Feature Selection of Intrusion Detection Data Using a Hybrid Genetic Algorithm/KNN Approach. In: Design and application of hybrid intelligent systems. IOS Press. 519-527 (2003)
15. Kholfi, S., Habib, M., Aljahdali, S.: Best Hybrid Classifiers for Intrusion Detection. Journal of Computational Methods in Science and Engineering 6(2), 299 - 307 (2006)
16. Babu, S., Subramanian, L., Widom, J.: A Data Stream Management System for Network Traffic Management. In: Proc. of Workshop on Network-Related Data Management (NRDM 2001). pp. (2001)

17. Dreger, H., Feldmann, A., Paxson, V., Sommer, R.: Operational Experiences with High-volume Network Intrusion Detection. In: Proc. of the 11th ACM Conf. on Computer and Communications Security. ACM Press New York, NY, USA. 2-11 (2004)
18. Hall, M., Wiley, K.: Capacity Verification for High Speed Network Intrusion Detection Systems. In: Wespi, A., Vigna, G., Deri, L. (eds.) RAID 2002. LNCS, vol. 2516, pp. 239-251. Springer, Heidelberg (2002)
19. Lee, W., Stolfo, S.J., Mok, K.W.: Mining in a Data-Flow Environment: Experience in Network Intrusion Detection. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, San Diego, California, United States (1999)
20. Elkan, M.: Results of the KDD'99 Classifier Learning Contest. http://www-cse.ucsd.edu/users/elkan/clresults.html, (1999)
21. Wooldridge, M., Jennings, N., R.: Agent theories, architectures, and languages: A survey. Intelligent Agents (1995)
22. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203-225 (2004)
23. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. Int. Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447-1466 (2003)
24. Aamodt, A., Plaza, E.: Case-Based Reasoning - Foundational Issues, Methodological Variations, and System Approaches. AI Communications 7(1), 39-59 (1994)
25. Carrascosa, C., Bajo, J., Julián, V., Corchado, J.M., Botti, V.: Hybrid Multi-agent Architecture as a Real-Time Problem-Solving Model. Expert Systems with Applications: An International Journal 34(1), 2-17 (2008)
26. Bratman, M.E.: Intentions, Plans and Practical Reason. Harvard University Press, Cambridge, M.A. (1987)
27. Fyfe, C., Corchado, E.: Maximum Likelihood Hebbian Rules. In: Proc. of the 10th European Symposium on Artificial Neural Networks (ESANN 2002). pp. 143-148 (2002)
28. Corchado, E., Han, Y., Fyfe, C.: Structuring Global Responses of Local Filters Using Lateral Connections. Journal of Experimental & Theoretical Artificial Intelligence 15(4), 473-487 (2003)
29. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. Advances in Neural Information Processing Systems 10, 350-356 (1998)
30. Case, J., Fedor, M.S., Schoffstall, M.L., Davin, C.: Simple Network Management Protocol (SNMP). RFC-1157. (1990)
31. Cisco Secure Consulting. Vulnerability Statistics Report. (2000)
32. Myerson, J.M.: Identifying Enterprise Network Vulnerabilities. Int. Journal of Network Management 12(3), 135-144 (2002)
33. Stephen, L.: The Spinning Cube of Potential Doom. Commun. ACM 47(6), 25-26 (2004)