Application of Neural Networks in Network Control and Information Security

Ángel Grediaga[1], Francisco Ibarra[1], Federico García[2], Bernardo Ledesma[1], and Francisco Brotóns[1]

[1] Alicante University, P.O. 99 Alicante, E-03080, Spain
angel.grediaga@ua.es, ibarra@dtic.ua.es
http://www.ua.es/tia/
[2] Miguel Hernández University, Avda. de la Universidad, s/n. Elche E-03202, Spain
fedeg@umh.es

Abstract. The increment of intrusions and bad uses in computer systems and internal networks of a great number of companies has caused an increase in concern for computer security. For some time one comes applying measures based on fire walls and in systems of intrusion detection (IDS). In this document we present an alternative to the problem of the IDS based on rules, using two different neural networks, a Multi-Layer perceptron, and a self organizative map. A series of experiments are carried out and the results are shown to be better than others found in the literature.

Keywords: intrusion detection system, neural networks, self organizative map, perceptron.

1 Introduction

Security in computers has been studied as a discipline since 1970. It refers to the measures and controls that protect the information systems against the negation of service and the authorisation absence (accidental or deliberately) to reveal, to modify, or to destroy the information and data systems [1][2]. However, the scientific community dedicated to the computer system security, has noted that technical prevention is not enough to protect the systems, such and like it was proven in the year 2000 with the socalled Distributed attacks Denial of Service (DDoS) [3]. These incidents demonstrated that technical prevention is inadequate, although the true reason is that the developed systems of information are not certain, since they can have cracks in the implementation, as well as shortcomings in the design. Intrusion detection was proposed as a complement of technical prevention.

2 Experimentation Proposal

The proposal to palliate the main limitation of the traditional IDS and their inability to recognize attacks lightly modified regarding the patterns with those

J. Wang et al. (Eds.): ISNN 2006, LNCS 3973, pp. 208–213, 2006.

that carry out the comparisons, is the use of neural networks. By means of this proposal it is sought to prove that by means of using a neural network, it is possible to carry out the distinction among packages that represent normal flow in a computer network, and packages that represent attacks. Concretely we will carry out the experimentation with a ML perceptron, and with a self organizative map (SOM) [6] as for the type of IDS that we will implement using neural networks, it will be an IDS host.

2.1 Selection of the Data

The first thing that it is necessary to think about when training a neural network is what data will contain the samples that it uses as entrances, so much to train as for checking. This belongs since the most important decision that can betaken, will depend in great measure the distinction capacity that can acquire the neural network works once the training is concluded. For the election of these data, they have taken as reference the data that a traditional IDS uses, concretely 'Snort ', the one which, apart from using numerous data characteristic of the packages TCP, UDP, or ICMP, it also considers the content of the very package. This is since something that outlines a serious problem, in principle that this was the most important fact to distinguish among a dangerous inoffensive and other package, but it is very complicated to introduce the content of the data of the package in a neural network. Finally we have opted to take out four characteristics of the same one in statistical form of probability of the four more frequent characters that appear in it. The data used is the Table 1. In total there are 29 entrances, which have been introduced to the network for each package, but keeping in mind that when the package belongs to a protocol, the data of the other protocols will be represented as nonexistent (for example introducing the fact 0, or any other value that suits). When introducing the previous data to the neural network, they have been normalized between 0 and 1 choosing a maximum value for each item.

2.2 Obtaining of the Data

To obtain the dangerous packages two machines are used, one from which the attacks rushed using the scanner "Nessus" and another from which those packages were captured using the traditional IDS "Snort". In total 433 dangerous packages were obtained. To obtain the inoffensive packages, a real computer network is used, that is to say, where the habitual packages flow in any company or corporation where it seeks to settle an IDS. Finally the study has been carried out using a small departmental net. The fact that the IP address origin and the destination are the same one in all the packages doesn't rebound in the obtained results, since that information isn't kept in mind. Between Windows and Linux 5731 inoffensive packages were obtained. Next the two neural networks, the ML perceptron and the self organizative map, are explained. It is necessary to indicate that because the inoffensive training packages are much bigger in number that the dangerous training packages, when training both neural networks, it

Table 1. Data used

Head IP of the package:	**Head TCP of the package:**
Port origin	Flag 1
Port destination	Flag 2
Protocol	Flag OR
TOS	Flag TO
Size of the head IP	Flag P
Total size of the package	Flag R
Reserved Bit	Flag S
Don't Fragment bit	Flag F
Fragments bit live	Size of the window
Number of options IP	Size head TCP
	Number of options TCP
Head UDP of the package:	**Content Package:**
Size of the head UDP + data (Len)	Percentage of the most frequent datos1
	Percentage of the most frequent datos2
	Percentage of the most frequent datos3
	Percentage of the most frequent datos4
Head ICMP of the package:	
Message type (Type)	
Code of the message (Code)	
Protocol of the original package	

has left introducing alternating a dangerous inoffen-sive and other package, and when these last they have finished, then they went by the network again until all the inoffensive packages finish happening.

3 Multi-layer Perceptron

For our case of ML perceptron, guided to distinguish among inoffensive packages and packages that represent attacks to a host in a computer net, the configuration that we have adopted is the following one: The activation function for each neuron should be continuous and derivable, so the network can store information on the exit of each neuron in a more precise way, only not differing between 0 and 1, that is to say that the exit can take values in the continuous one between 0 and 1. For this end, a sigmoidea function has been used. The number of neurons in the first layer is similar to the number of data that is extracted of each package, in this case 29, and in the exit layer, since we will only distinguish between two possible values, we only use a neuron whose exit will indicate that the analyzed package is inoffensive when it is smaller or the same as 0.5, and will indicate that the analyzed package is dangerous when the exit is bigger than 0.5, since the exit is enclosed between 0 and 1 for the activation function, and the training exits have been introduced as 1 for dangerous packages, and 0 for inoffensive packages. Only a hidden layer has been used. After carrying out diverse tests varying the number of neurons of this layer, we obtained the best results with a

number of 30. The learning rate that has been used is variable. It begins with a value of 0.5, and when the error begins to oscillate (the oscillations are detected when the error ascends 4 times after 8 histories) it diminishes in 0.2 units. If the error continues oscillating, it must descend the value again of a, until it arrives at 0.1. If it continues oscillating it is lowered up to 0.06, and it no longer is lowered more. After proving several values for the moment, the one that had given The half error of each history in the learning, something that will be good to check if the error is lowered in each history, is calculated adding the error of all the samples of the history and dividing it among the number of samples, where the error of each sample is indicated in the equation (1)

$$E(hist) = \frac{1}{N} \cdot \sum_{p-1}^{N} E(p). \tag{1}$$

4 Self Organizative Map

The second proposal of neural networks work that has been proven to distinguish inoffensive packages from dangerous packages is based on unsupervised learning, concretely it is a self organizative map [6]. For our case of recognition of inoffensive packages and dangerous packages of a computer net, where the extracted information of each package is composed of 29 data, a rectangular SOM has been used (toroidal) size 40 x 40, since it has been proven with sizes of 5 x 5, 10 x 10 and 20 x 20, and the error of each history oscillated too much, indicating that clusters was superimposed, and therefore it lacked space so that these groupings are formed without blocks among them. It has also been proven to use sizes of 30 x 30 and 50 x 50, but 50 x stops 50 the results they were the same ones that using a size of 40 x 40, and 30 x stops 30, the results were a little worse. It is also necessary to consider that as much as adult is the size of the map, more will take a long time as much in the training as in the recognition, and this last is something to keep in mind if one wants to implant the system in a computer net, and that it works in real time, the minimum map size has been chosen for the reason that offered better results. The distance function used to measure the difference between a training sample and each neuron of the SOM has been the distance euclidea. As for the coefficients of upgrade of weights (Cp) for the winning neuron, and their neighbours, the following ones have been used, after having proven other values and to see that they didn't work better: Cp for the winning neuron: 0.9 Cp for the neighbouring neurons of level 1: 0.1, Cp for the neighbouring neurons of level 2: 0.005, Cp for the neighboring neurons of level 3: 0.0005]. The error of each sample is measured by the distance to the winning neuron, equation(1).

5 Obtained Results

It is necessary to say that the obtained results have overcome the initial expectations, since the fact of being able to distinguish between inoffensive packages

and dangerous packages without totally looking at the content of data of the package was something that was thought it could only be gotten in a moderate percentage of successes, but the percentage of successes gotten in the test overcame 90%. The first thing that is necessary to decide in this stage is how to measure the results. Since This is a problem bigger than which was thought of in principle, we don't prepare more than 433 dangerous packages with those that to train and to test, and neither you can prove the result of the training of the neural network work installing it in a computer net, since the surest thing is it meets with packages completely different from the packages with those that it has trained in the small network of two machines used in the project. Concretely 80% of the packages has been chosen for training, and 20% for testing, as much for the inoffensive packages as for the obtained dangerous packages. The results will measure them in terms of percentage of successes obtained in the testing packages that will be: inoffensive, dangerous packages, and the entirety of them.

5.1 Results of the Multi-layer Perceptron

Next a summary of the error is shown after each history in the training of the multilayer perceptron that has given better results, since this depends in great measure of the random initialization of the weights. We will indicate the errors of the first 10 histories, and then we will go advancing more quickly, since it allowed to train until the error didn't lower more, and there were 2153 histories (min error= 0,002132). The results as for success Package types approximate number of packages of training Inoffensive packages. Dangerous packages the rate of success of all the test packages, approximately 1233, is of 0.984750. To check that the obtained results have not been fruit of chance when selecting the training packages and the test packages, they have been carried out 3 more trainings, each one with a division of training packages and of different test. The results, after 100 training test are the Table 2.

Table 2. 100 training test results

	Division 1	**Division 2**	**Division 3**
Success inoffensive test packages	0.863755	0.849877	0.947186
Success dangerous test packages	1.000000	0.988372	0.876712
Success in all the test packages	0.872340	0.859004	0.942997

5.2 Results of the Self Organizative Map

Next the same thing will be shown for the training of the self organizative map that has given better results. The configuration of the number of neurons of the map and other parameters are previously the suitable ones. It is necessary to keep in mind that the mensuration of the error is not calculated in the same way that in the ML perceptron, for what the comparison in that sense is not possible. Yes it will be possible to compare the rate of successes. We will indicate the errors of

the first 10 histories, and then we will go advancing more quickly, since it allowed to train until the error didn't lower more, and they were 85 histories (min error= 0.004966).. The rate of success of all the test packages, approximately 1233, it is of: 0.991235. To check that the obtained results have not been fruit of chance when selecting the training packages and the test packages, 3 more trainings have been carried out, each one with a division of training packages and of different test, The results, after 10 histories of training, are the Table 3.

Table 3. 100 training test results

	Division 1	**Division 2**	**Division 3**
Success inoffensive test packages	0.995442	0.996567	0.997567
Success dangerous test packages	0.924051	0.911111	0.891304
Success in all the test packages	0.990646	0.990438	0.990189

6 Conclusions

Managing the detection of attacks is something complicated, since continually new exploits arise and the at-tackers invent new ways to penetrate the systems. The use of the neural network can suppose a great advantage in the detection of these attacks, since they have the capacity to detect attacks that they have not memorized directly. As a final conclusion one can say that the intent of checking if a neural network work could learn how to distinguish inoffensive packages of dangerous packages has been an entire success, still without using the whole information of the content of each package.

References

1. Abrams, M.D., Jadodia, S., Podell, H.J. (eds.): Information Security: An Integrated Collection of Essays. IEEE Computer Society Press, Los Alamitos California (1995)
2. Pfleeger, C.P., Pfleeger S.L.: Security in Computing. 3rd edn. Prentice Hall Inc., New Jersey (2003)
3. CERT: Incident Note IN-99-07: Distributed Denial of Service Tools. CERT Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2001)
4. Kumar, S.: Classification and Detection of Computer Intrusion. PhD thesis, Purdue University, August (1995)
5. Denning, D.E.: An Intrusion-Detection Model. IEEE Trans. Software Engineering SE13 (2) (1987) 222-232.
6. Kohonen, T.: Self-Organization and Associative Memory Springer-Verlag, New York (1987)