Annales UMCS Informatica AI X, 1 (2010) 51-59
DOI: 10.2478/v10065-010-0035-7

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Analysis of neural networks usage for detection of a new attack in IDS

Przemysław Kukiełka[1*], Zbigniew Kotulski[2†]

[1] *Institute of Telecommunications, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland.*
[2] *Institute of Fundamental Technological Research, Polish Academy of Sciences, Swietokrzyska 21, 00-049 Warsaw, Poland.*

Abstract – Generally, Intrusion Detection Systems (IDS) work using two methods of identification of attacks: by signatures, that are specific defined elements of the network traffic possible to identify and by anomalies being some deviation form of the network behaviour assumed as normal. Recently, some attempts have been made to implement artificial intelligence method for detection of attacks. Many such implementations use for testing and learning process the data set provided by KDD (Knowledge Discovery and Data Mining Competition) project in 1999. Unfortunately, KDD99 data set was created more than eight years ago and during this time many new attacks have been discovered. In this paper we present our research on updating KDD99 data with traces of attacks of new types. After updating, the data set was used for training and testing MLP (Multi Layer Perceptron) neural network architecture IDS.

1 Introduction

Because of their generalization feature, neural networks are able to work with imprecise and incomplete data. It means that they can recognize also patterns not presented during a learning stage. That is why neural networks could be a good solution for detection of a well known attack, which has been modified by an aggressor

*przemyslaw.kukielka@telekomunikacja.pl
†zkotulsk@ippt.gov.pl

in order to pass through the firewall system. In that case, traditional Intrusion Detection Systems (IDS), based on signatures of attacks or expert rules, may not be able to detect the new version of this attack.
Moreover, IDS based on the artificial intelligence system does not require building a complicated set of signatures for each attack type or profiles of user's normal behaviour because they are created during the learning stage automatically.

We use the KDD 99 data set as the input vectors for training and validation of the tested neural network. In order to update this data set we used Metasploit framework which allows to simulate a new attack type. In this paper we focus on the attacks which cause buffer overflow in the servers providing the network services like: ftp, http, telnet, tftp and smtp, and in the case of success allow an aggressor to execute payload code on the victim's host.

The result of simulation is the information about the attack detection accuracy, represented as a number of false alarms and not detected attacks in comparison to the number of validation vectors for each type of used neural network.

2 Neural network: a way of work

An artificial neural network is a system simulating work of the neurons in the human brain. Fig. 1 presents the diagram of neuron's operation.

The neuron consists of some inputs emulating dendrites of the biological neuron, a summation module, an activation function and one output emulating an axon of the biological neuron. The importance of a particular input can be intensified by the weights that simulate biological neuron's synapses. Then, the input signals are multiplied by the values of weights and next the results are added in the summation block. The sum is sent to the activation block where it is processed by the activation function. Thus, we obtain neuron's answer to the input signals "x".

2.1 MLP (Multi Layer Perceptron)

One neuron cannot solve a complex problem that is why the neural network composed of many neurons is used. One of the most often used architecture is the Multi Layer Perceptron. In such a network, all neurons' outputs of the previous layer are connected with the neurons' inputs of the next layer. The MLP architecture consists of one or more hidden layers. A signal is transmitted in the one direction from the input to the output and therefore this architecture is called feed-forward. The MLP networks are learned using the Backward Propagation algorithm (BP). For the simulation procedure in our research, the Matlab toolbox was used.

3 KDD99 Input data

KDD99 data set was created based on the DARPA (Defence Advanced Research Project Agency) intrusion detection evaluation program. MIT Lincoln Lab that participates in this program has set up simulation of typical LAN network in order to acquire raw TCP dump data [1]. They simulated LAN operated as a normal environment, which was infected by various types of attacks. The raw data set was processed into connection records.

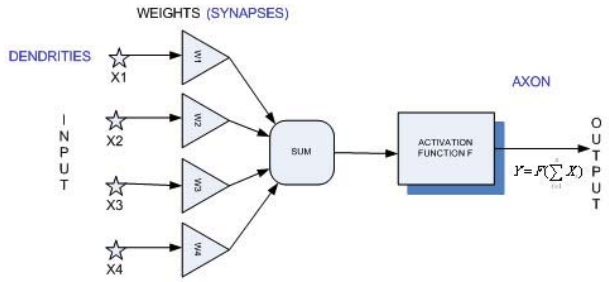

Fig. 1. A scheme of an artificial neuron.

For each connection, 41 various features were extracted. Each connection was labelled as normal or under a specific type of attack. Four main categories of attacks were simulated:

- **DoS** (Denial of Service): an attacker tries to prevent legitimate users from using a service e.g. TCP SYN Flood, Smurf .
- **Probe:** an attacker tries to find information about the target host. For example: scanning victims in order to get knowledge about available services, using Operating System etc.
- **U2R** (User to Root): an attacker has local account on victim's host and tries to gain the root privileges.
- **R2L** (Remote to Local): an attacker does not have local account on the victim's host and tries to obtain it.

The KDD data sets are divided into three subsets: 10%KDD, corrected KDD, whole KDD. Basic characteristics of KDD data sets are shown in Table 1. It includes a number of connections assigned to the particular class (DoS, Probe etc.).

Table 1. KDD99 data subsets.

Dataset	**DoS**	**Probe**	**U2r**	**U2l**	**Normal**
10%KDD	391458	4107	52	1126	97277
Corrected KDD	229853	4166	70	16347	60593
Whole KDD	3883370	41102	52	1126	972780

3.1 10%KDD

The 10%KDD data set is used for the training process of the IDS. It includes the connections simulating the following 22 types of the attacks: back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop. The attacks are not represented by the same number of connections. The most of the simulated attacks are of DoS group because of the nature of this type of attacks that use many IP packets in order to block network services.

3.2 Corrected KDD

The Corrected KDD data set is used for the testing process of IDS. It includes additional 14 types of new attacks not presented in 10%KDD and whole KDD. Owing to them, it is possible to check if the tested IDS is able to detect a new attack not presented in the training phase.

In our research only 10% of KDD and corrected KDD data set were used.

4 Preprocessing of new attcks to input vectors

4.1 Simulating a new attack with Metasploit

For simulation of a new attack we used the tool provided by the Metasploit project. The Metasploit Framework is a development platform for creating security tools and exploits. It could be used by network security professionals to perform penetration tests and development of IDS signatures. The framework was created in 2003 using the Perl scripting language and later was rewritten in the Ruby programming language. In our simulation we used the Metasploit framework in version 3.2. It consists about 300 types of exploits that can be used against the network services working on OS like Windows, Linux, Unix, MAC. In the case when the exploit succeeds, the payload is executed on the victim and the user could be provided with a shell to interact with the payload. More information about the Metasploit project could be found in [9].

For our research we decided to choose the following exploits:

- Learning data set: linksys_apply_cgi, tftp32longfilename, goodtech_telnet, filecopa_list_overflow, wsftp_server_503_mkd, warftpd_165_pass, warftpd_165_user, mailcarrier_smtp.
- Testing data set: tftpdwin.long_filename, goodtech_telnet, cesarftp_mkd, slimftpd_list_concat, easyfilesharing_pass, freeftpd_user, netterm_netftpd_user, mailcarrier_smtp, atftp.long_filename, linksys_apply_cgi .

The common feature for all chosen attacks is that they cause buffer overflow by sending a very long string to the target as password, login or arguments of commands used by the network services like ftp, telnet, smtp, telnet etc.

4.2 Attacks preprocessing

During simulation of an attack we collected traces in the tcpdump format with the network sniffer (ethereal). Besides the traces consisting of attack we also simulated and collected normal traffic sent to the same service as used by attack. Owing to them later we could check level of false alarms generated by the tested IDS when we provided as an input data normal traffic with many features similar to those used in the attack traffic. In KDD 99 there are also normal traces but sometimes there is a lack of traffic similar to the attack that we decided to use for updating. For example, there is only one connection for tftp traffic in the "corrected" data set, also normal traffic with the flag RSTO is not represented for the service like smtp or ftp.

After that "bro" tool [10] was used to process the data into the connection record in the same way as it was done for the KDD 99 project.

4.3 Proposal of adding a new feature

An attack could be detected properly using existing features. It was confirmed also in the first simulations. Many representations of an attack (not all) against the services like ftp, telnet were detected correctly. Unfortunately, the attacks aimed at tftp and smtp services were not detected at all. When we compare the connection records related to each attack type with normal traffic we notice that the attacks related to ftp or telnet connection may be detected based on asymmetry between the destination and source bytes sent features. For normal traffic in the services like telnet or ftp (only for 21 TCP port transmission) usually more data are sent from the destination to the source during one connection. In the case of attack types that we plan to add more data in the connection record is sent from the source to the destination. This observation is not correct for the services like tftp and smtp where also during normal connection more data could be sent from the source to the destination. Moreover, we noticed that asymmetry between the source and the destination bytes is not enough to detect all attacks related to ftp, http or telnet connection. For the reasons mentioned above we decided to add a new feature to the KDD99 data set. This feature represents the longest string sent during connection as password, login or arguments of commands like list, mcd, snmp hello.

5 Training process

5.1 Process of Selection of the Input Vector

The 10% KDD data set includes a large number of connections, which affects time duration of training, high requirements for efficient implementation of neural network and hardware to it work. Therefore the research presented in this publication for a training purpose three randomly selected small data sets were used. Table 2 shows how many connections are assigned to the particular training data set. Because the data used for a training process should represent statistically all connections presented in

the whole 10%KDD data set we decided to modify the process of random selection of input vector for the training process in the following way:

- There was chosen the same number of connections representing each type of attack. In the case when the number of connections for a particular type of attack was lower than the assumed one all connections for this group were selected.
- For normal traffic there was chosen the representation of all services presented in 10%KDD data set. Also in the case when a number of normal connections for a particular service was lower than the assumed one all connections for this group were selected.

Table 2. Data subsets for the training process.

Data Set name for training process	**Number of normal connection**	**Number of connection labelled as attack**	**Training set description**
nauIbiza2009f1	1176	879	Only existed connection from 10%KDD
nauIbiza2009f1+new	1184	887	Vectors related to new attack added
Nau_Ibiza2009f1+new-ext	1184	887	New feature was added

Because some features of connection (e.g., protocol, flags) existed as a character string, they were transformed to numerical representation.

Main Assumption for the Training Process of MLP:

- Learning method: Quasi Newton BFGS and Levenberg-Marquardt
- Number of Epochs: 1000.
- MSE (Mean Square error): 0.01.
- Learning rate: 0,9.
- Activation function: log-signoid.
- Number of neurons in the hidden layer: 8.
- Number of neurons in the output layer: 1.
- Update of weights – batch mode (after presentation of the entire training data set).

6 Results of the test

The neural network architectures were tested using the whole data set from the "Corrected KDD" updated with a new connection that represents simulated new attacks and normal traffic. The data sets used for the testing stage were shown in Table 3. For the analysis of neural network reply, it was assumed that the value from 0 to 0.5 refers to "normal" and from 0.5 to 1 "attack".

Table 3. Data subset for the testing stage.

Data Set name for testing process	**Number of normal connections**	**Number of connections labelled as an attack**	**Test set desc**
Test+new	60604	250445	New attack and normal added
Test_new-ext	60604	250445	New feature was added

Our research was divided into three stages. In the first we built MLP neural network IDS and checked its ability to classify of connections already presented in the KDD99 data set and new added connection. The training stage is performed before the KDD99 connection is accomplished. In the second we added to the 10%KDD data set new attack simulated with the usage of Metasploit framework and similar to added attacks normal traffic. In comparison to the first stage here new attacks and new normal traffic were used in both training and testing processes. In the third stage we extended with a new feature training and test data sets from the second stage. The first one could be used to compare the results with second and third in order to observe influence of adding new input vector and features on classification of existing representation of attacks and normal traffic.

As correct detection we mean that IDS correctly recognizes attack and also "normal" traffic with many features similar to the detected attack type. We noticed, for instance, that neural network classified an attack based only on the flag feature. It is incorrect behaviour because normal connection could be also finished with the flags RST0 or RSTR not only SF as it exists in 10%KDD data set. Therefore we decided to add new normal traffic input vectors. The results of tests for all phases are presented in Table 4. All new attacks were detected only in the third stage when we added a new feature.

Table 4. Results of three tests phases.

Test set/training set	**False alarm Number/Percent of false normal connections**	**Not detected/ Detection rate**	**Not detected new attack/false new normal**
Test+new/nauRST	1746 2,9%	22697 90,9%	atak_nettermftp_user, atak_slimftp_list, smtp_multicarier, goodtech_telnet tftp_prosysyst /telnet,ftp,http

Table 4. Continued.

Test+new/ nauIbiza2009f1 +new	1506 2,5%	21752 91,3%	atak_nettermftp_user, atak_slimftp_list, smtp_multicarier,, goodtech_telnet tftp_prosysyst /telnet,ftp,http
Test+new-ext/ NAU_IBIZA2009F1 +NEW-EXT	3089 5,1%	29815 88%	All new attacks detected /telnet, ftp

7 Conclusions

Usage of neural networks for intrusion detection with the input data from the DARPA project was presented in many papers. Unfortunately, since 1999 when KDD 99 was published many new attacks have been launched. Therefore for more credible tests of IDS it is important to update the KDD 99 data set. The goal of this research was to find the way of adding new connection records represented of new attacks and check how it influences on classification of already presented network traffic. The main conclusions are:

- Selection of the input data is a very important issue. Representation of all types of attacks and normal activity should be included in the learning data set. Therefore besides new attack connections, representation of normal traffic not presented before in KDD99 was added.
- In order to properly evaluate ability of detection of new attack also to testing set "normal" traffic similar to attack that we would like to detect should be added. When both attack and normal connections were properly classified we can say that a new attack will be properly detected.
- A new attack could be detected based on existing KDD 99 features but for the others a new feature should be created
- Adding a new feature makes some influence on the classification process. The reason can be that for longer input a vector is more difficult for training and testing process of neural network. Probably there should be used more complex architecture of neural network but it is the issue for future research.
- In our experiment all attacks were properly detected only when we added a new feature. Concerning a false alarm rate for that case only two new added "normal" connections are not properly classified. For the first and second stages of research a false alarm rate for a new added connection was significantly worse.

References

[1] Lee W., Stolfo S. J., A framework for constructing features and models for intrusion detection systems, ACM Transactions on Information and System Security (TISSEC) 3(4) (2000): 227–261.

[2] Rutkowski L., Metody i Techniki Sztucznej Inteligencji (in Polish) (PWN, Warszawa, 2005).

[3] Lee W., Stolfo S.J., Data mining approaches for intrusion detection, Proc. of the Seventh USENIX Security Symposium (SECURITY '98) (San Antonio, 1998).

[4] Lippmann R., Haines J. W., Fried D. J. et al., The 1999 darpa off-line intrusion detection evaluation, Computer Networks: The International Journal of Computer and Telecommunications Networking 34 (2000): 579–595.

[5] Paxson V., Bro: A system for detecting network intruder in real time, Proceedings of the 7th USENiX Security Symposium (San Antonio, 1998).

[6] Elkan Ch., Results of the KDD'99 classifier-learning contest (1999), http://www.cse.ucsd.edu/#elkan/clresults.html.

[7] Osowski S., Sieci Neuronowe do Przetwarzania Informacji (in Polish) (Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000), ISBN: 83-7207-187-X.

[8] Kukielka P., Kotulski Z., Analysis of different architectures of neural networks for application in intrusion detection systems, International Multiconference on Computer Science and Information Technology (Wisła, 2008): 20–22.

[9] The Metasploit Project, www.metasploit.com.

[10] Bro-Intrusion Detection System, www.bro-ids.org.