Warning!

This is an simplified, electronic version of our chapter published in “Information Systems Architecture and Technology. Information Systems and Computer Communication Networks”. You can use it in your articles/thesis/books etc. but you have to include bibliographical note of the citation (full description below). Please send questions or suggestions to me mailto: ***maciej.szmit@gmail.com*****.**
Please visit ***http://www.anomalydetection.info***

Maciej Szmit, Radosław Wężyk, Maciej Skowroński, Anna Szmit: "Traffic Anomaly Detection with Snort", [in:] Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, Wydawnictwo Politechniki Wrocławskiej, Wrocław 2007 ISBN 978-83-7493-348-3

Maciej SZMIT*
Radosław WĘŻYK**
Maciej SKOWROŃSKI**
Anna SZMIT***

TRAFFIC ANOMALY DETECTION WITH SNORT

Snort is open source intrusion detection system based on signature detection. In the paper we present information about the second version of anomalydetection – preprocessor designed to log and analyze network traffic information. We also collect network traffic information from a few local area networks and made a few simply traffic statistical analysis which could be usefull to anomalies detection.

1. INTRODUCTION

Snort is open source network intrusion detection and prevention system (IDS and IPS) utilizing a rule driven language, which possibility of use signature, protocol and anomaly based inspection methods. Snort authors said that it is the most widely deployed intrusion detection and prevention technology worldwide and has become the de facto standard for the industry [1]. The most popular form of using Snort is to build an Network based Intrusion Detection System (NIDS) based on standard signatures database or build an IPS by adding active response program, like Guardian [2], which works in conjunction with Snort to automatically update firewall (usually iptables [3]) rules based on alerts generated by Snort, but there are a lot of other possibilities: using Snort in inline mode [4], using Snort in bridge mode computer (which can be good idea from security point of view because of prevention from attacks against IDS machine) or using Snort like a sniffer etc. (see: [1], [5], [6]).

Because of Snort popularity there are a lot of accessories – beginning from front ends – visualization and analysis tools (like ACID - Analysis Console for Intrusion Databases [7] or BASE – Basic Analysis and Security Engine [8]), through experimental preproc-

* Computer Engineering Department, Technical University of Lodz, Maciej@szmit.info
** Computer Engineering Department, Technical University of Lodz
** Computer Engineering Department, Technical University of Lodz
*** Faculty of Organization and Management, Technical University of Lodz

essors (like Snort+AI [10] or SPADE – Statistical Packet Anomaly Detection Engine [11]), finishing on dedicated hacker tools like Snot – a Snort alert generator and general NIDS decoy utility [12].

2. ANOMALYDETECTION PREPROCESSOR

Our project – Anomalydetection [13] is a simply preprocessor designed to log and analyze information about network traffic and detect possibly network traffic anomalies. You can find more information about idea and program data structures in our articles [14] and [15], in thesis [6] and in project page [13]. Basically: Anomalydetection logs 25 parameters of network traffic:

1) number of TCP packets,
2) number of outgoing TCP packets,
3) number of incoming TCP packets,
4) number of TCP packets in its own subnet (LAN),
5) number of UDP datagrams,
6) number of outgoing UDP datagrams,
7) number of incoming UDP datagrams,
8) number of UDP datagrams in its own subnet (LAN),
9) number of ICMP messages,
10) number of outgoing ICMP messages,
11) number of incoming ICMP messages,
12) number of ICMP messages in its own subnet (LAN),
13) number of TCP packets with set SYN and ACK flags,
14) number of outgoing TCP packet send on port 80 (WWW),
15) number of incoming TCP packet from port 80 (WWW),
16) number of outgoing UDP datagrams send on port 53 (DNS),
17) number of incoming UDP datagrams from port 53 (DNS),
18) outgoing IP bitrate [kBps],
19) incoming IP bitrate [kBps],
20) outgoing TCP port 80 bitrate [kBps],
21) incoming TCP port 80 bitrate [kBps],
22) outgoing UDP bitrate [kBps],
23) incoming UDP bitrate [kBps],
24) outgoing UDP port 53 bitrate [kBps],
25) incoming UDP port 53 bitrate [kBps].

After a period of working in collecting data mode, anomalydetection builds “network profile” which contains information about mean and variation of each parameter. In anomaly detection mode the current version of our preprocessor generates alert when

one of the parameter exceed value meaning plus (or minus) X standard deviation, where X is parameter setting by administrator.

The second version of anomalydetection, which will be released soon (a beta release is available on project pages) will also detect ARP-spoofing attack. This attack (and a few similar like DNS-spoofing) can be easy detected by comparation numbers of request with number of answers. When you receive more than one ARP-reply with different content you can expect that you are under attack.

In our research we collected data from two campus networks (Net A and Net C on figures below). The first was rather small (about 20 computers), the second was larger (more than 400 computers). We tried to make simply statistical analysis of these time series to recognize their characteristics and – in the future – implement additional alert generating mechanism in our preprocessor.

3. NETWORK TRAFFIC – A STATISTICAL OVERVIEW

Because the both of networks were similar, we expected day and eventually week seasonality (see [6], [14]). The first parameter which we investigate was number of TCP packets which usually should be about 80% of network traffic (see. figure 1). On figure 2 and figure 3 you can see average overall TCP traffic in each of the networks in each day of the week. As we could expect, each of the two networks has different characters and the Network C with smallest variation and more regular traffic seems to be better for statistical analysis.

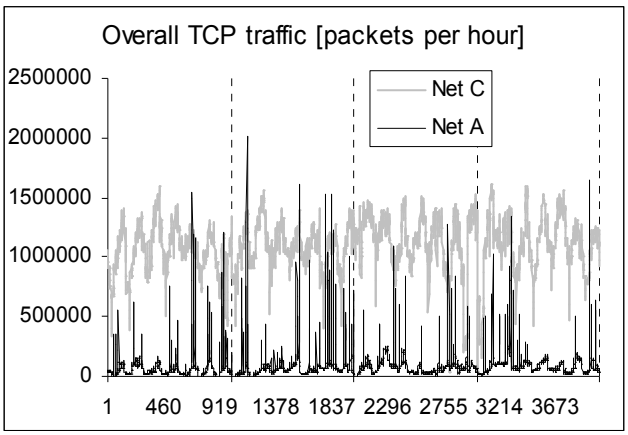

Fig. 1. Overall TCP traffic in the Net A and the Net C

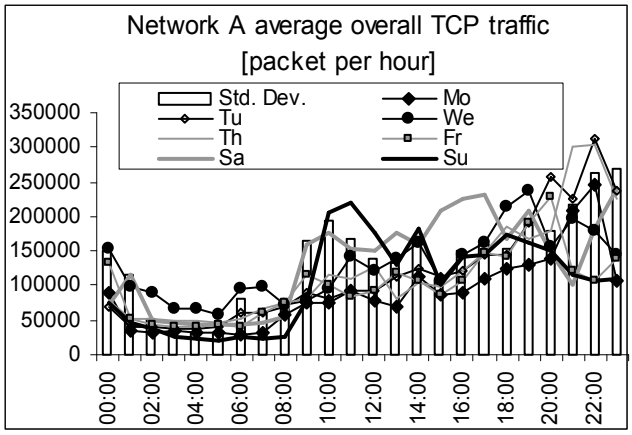

Fig.2. The Net A average overall TCP traffic

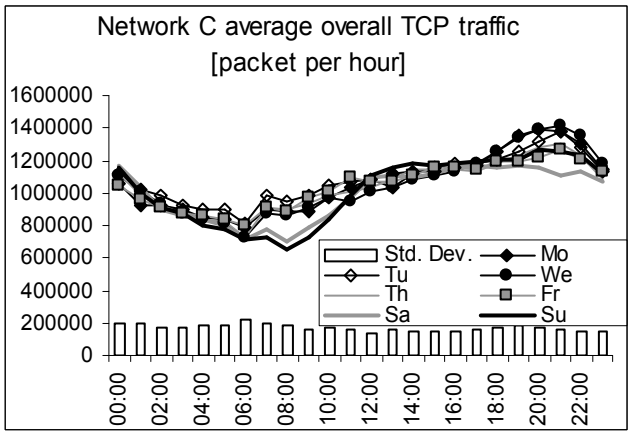

Fig.3. The Net C average overall TCP traffic

Coefficient of determination R^2 (proportion of variation of each kind of traffic explained by hour is shown on table 1).

TCP overall	TCP outgoing	TCP incoming	ICMP incoming	TCP with SYN/ACK	UDP port 53 incoming
47%	61%	62%	22%	65%	36%

Tab. 1. Coefficient of determination values

On figure 4 you can see average values of TCP and other protocols traffic in the Network C in 24-hours period.

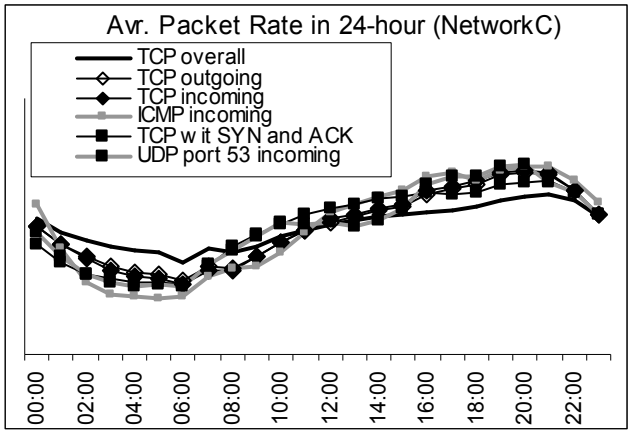

Fig.4. Average traffic in the Net C (selected protocols)

The most interesting protocol for detecting anomalies is of course ICMP, because we can expect high ICMP traffic when any problem or error occurs. In current version of anomalydetection the alerts are generated based on average and standard deviation values, but it seemed to be interesting to analyze histogram of incoming ICMP distribution. In figure 5 we presented histograms of average number of TCP packets and ICMP messages for the Network C at 6 am. and in figure 6 – the same histogram for 9 pm.

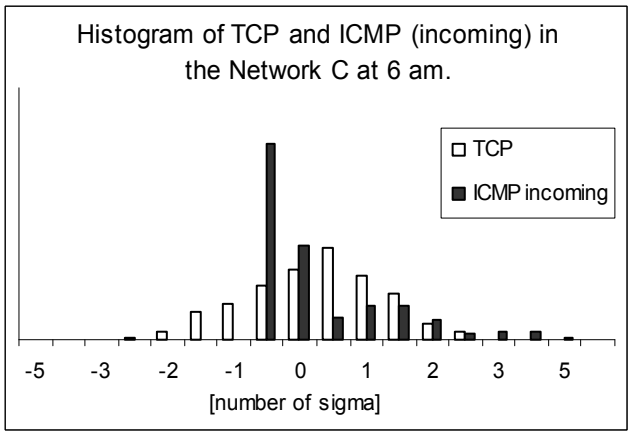

Fig.5. Histogram of TCP and ICMP (incoming) in the Net C at 6 am.

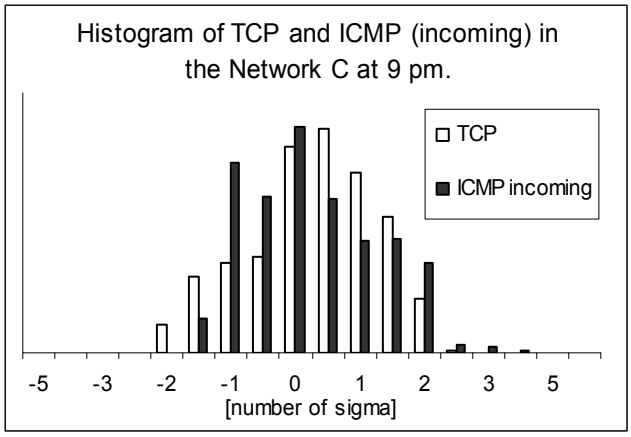

Fig.6. Histogram of TCP and ICMP (incoming) in the Net C at 9 pm.

As you can see ICMP traffic has asymmetric distribution (or even not unimodal distribution) so it should be investigated if average and standard deviation can be used. We are going to investigate this problem in details in future, after getting more time series from a few anomalydetection which we have installed in a few networks.

3. CONCLUSIONS

Statistical analysis (like self-similarity analysis [20], time series analysis [21] etc.) and Artificial Intelligence based methods (like Neural Networks [10], Genetic Algorithms [16], Immunity based algorithms [17], Data Mining methods [18], Simulated Annealing algorithms [19] etc.) of network anomalies is perceived as very interesting and promising methods for Intruder Detection and Prevention Systems. It possibly can detect new or unknown methods of attack, like the zero day exploits. But one should remember that anomaly based detection (neither nor signature based detection) is not The Silver Bullet. For example: a lot of attacks can be not recognized by network traffic anomaly detection systems because of small amount of data which they use (even only one packet) and a lot of false positives and false negatives could be generated by them because of using bandwidth management methods which can falsify traffic profile. On the other hand, traffic anomalies detection can be useful for even non technical attacks (for example: spy, who transfer a big amount of data from secure network to external server). So we think that this method can be used as subsidiary method in “classic”, signature-based IDS.

REFERENCES

[1] Snort homepage http://www.snort.org (01.03.2007)
[2] Guardian homepage http://www.chaotic.org/guardian (01.03.2007)
[3] Iptables and Netfilter project pages http://www.netfilter.org (01.03.2007)
[4] Snort inline project page http://snort-inline.sourceforge.net (01.03.2007)
[5] Szmit M., Gusta M., Tomaszewski M.: *101 zabezpieczeń przed atakami w sieci komputerowej*, Helion, Gliwice 2005
[6] Skowroński M., Wężyk R.: *Systemy detekcji intruzów i aktywnej odpowiedzi*, praca magisterska napisana w Katedrze Informatyki Stosowanej Politechniki Łódzkiej pod kierunkiem Macieja Szmita, maszynopis, Łódź 2006
[7] Analysis Console for Intrusion Databases project page http://acidlab.sourceforge.net (01.03.2007)
[8] Basic Analysis and Security Engine homepage http://base.secureideas.net (01.03.2007)
[9] Snort Setup for Statistics howto http://www.faqs.org/docs/Linux-HOWTO/Snort-Statistics-HOWTO.html (01.03.2007)
[10] Snort+AI project page http://afrodita.unicauca.edu.co/%7Eaarboleda/snort_ai.htm (01.03.2007)
[11] SPADE CVS repository http://www.bleedingsnort.com/cgi bin/viewcvs.cgi/?cvsroot=SPADE (01.03.2007)
[12] Copy of old (2001) Snot page http://web.archive.org/web/20010424080846/http://www.geocities.com/sniph00/ (01.03.2007)
[13] Anomalydetection project page http://www.anomalydetection.info (01.03.2007)
[14] Skowroński M., Wężyk R., Szmit M., *Detekcja anomalii ruchu sieciowego w programie Snort*, Hakin9 nr 3/2007
[15] Maciej Skowroński, Radosław Wężyk, Maciej Szmit, "Preprocesory detekcji anomalii dla programu Snort" w: "Sieci komputerowe Tom 2. Aplikacje i zastosowania", WKŁ 2007, pp. 333-338
[16] Li W.: *Using Genetic Algorithm for Network Intrusion Detection*, United States Department of Energy Cyber Security Group 2004 Training Conference, Kansas City, 2004
[17] Dasgupta D.: *Immunity Based Intrusion Detection System: A General Framework*, 22nd National Information Systems Security Conference (NISSC), 1999
[18] Cichocki R.: *Algorytmy indukcji reguł decyzyjnych w Systemach Wykrywania Intruzów*, XII konferencja Sieci Komputerowe , Zakopane 2005
[19] Kruk T. J., Wrzesień J.: *Korelacja w wykrywaniu anomalii*, Materiały konferencji CERT Secure 2003, Warszawa 2003
[20] Kolbusz J., Lewicki A., Majdański A., Karmelita S.: *Badanie samopodobieństwa ruchu w sieciach LAN – metody i narzędzia*", Informatyka Stosowana ISSN 83-914678-6-4, VII Lubelskie Akademickie Forum Informatyczne, Kazimierz Dolny 2003, s. 97-103.
[21] Hao Y., Chuang L. Berton S. Bo L. Geyong M., *Network traffic prediction based on a new time series model: Research Articles*, International Journal of Communication Systems, Volume 18, Issue 8, 2005