ANOMALY DETECTION 3.0 FOR SNORT®

Maciej Szmit, Sławomir Adamus,
Sebastian Bugała, Anna Szmit,
Computer Engineering Department,
Technical University of Lodz, Poland

Snort® is an open source intrusion detection system based on signature detection. In the paper we present information about the third version of Snort AD – preprocessor designed to log and analyze network traffic information developed by us.

AD preprocessor

Snort is the most popular open source intrusion detection system based on signature detection (see e.g. [2], [14], [18], [19]). The modular construction of Snort allows one to extend its capabilities by creating own pre- or postprocessors and/or plugins. The best-known Snort tools are for instance ACID (see [15]), BASE (see [16]) or SAFE (see [17]).

In our works we develop Snort preprocessor designed to enhance Snort possibilities to monitor, analyze and detect network traffic anomalies using NBAD (Network Behavioral Anomaly Detection) approach (see e.g. [6], [7], [20]). The first version of Anomaly Detection preprocessor (see [21]) for Snort version 2.4x was published in a Master's Thesis [11] in 2006. Next the project has been developed (see e.g. [10], [12], [9], [8]) till the current version 3.0 designed for Snort 2.9.x which periodically, with a given interval, logs information about 29 parameters of the network traffic as a number of TCP/UDP packets sent/received from outside/inside the current IPv4 subnet, www download/upload speed, a number of UDP 53 (DNS) datagrams etc. Values of these parameters are the logs into a file in CSV (Comma Separated Values) format, with header line containing description of each parameter (see Figure 4).

DD-MM-YY, HH:MM:SS, Day of the Week, Time interval [s], TCP summary [number of packet], TCP outgoing [number of packet], TCP incoming [number of packet], TCP from this subnet [number of packet], UDP summary [number of packet], UDP outgoing [number of packet], UDP incoming [number of packet], UDP from this subnet [number of packet], ICMP summary [number of packet], ICMP outgoing [number of packet], ICMP incoming [number of packet], ICMP from this subnet [number of packet], TCP with SYN/ACK [number of packets], WWW outgoing - TCP outgoing to port 80 [number of packet], WWW incoming - TCP incoming from port 80 [number of packet], DNS outgoing - UDP outgoing to port 53 [number of packet], DNS incoming - UDP incoming from port 53 [number of packet], ARP-request [number of packet], ARP-reply [number of packet], Not TCP/IP stacks packet [number of packet], Total [number of packet], TCP upload speed [kBps], TCP download speed [kBps], WWW upload speed [kBps], WWW download speed [kBps], UDP upload speed [kBps], UDP download speed [kBps], DNS upload speed [kBps], DNS download speed [kBps]

```
04-10-
11,16:52:16,Tue,30,0,0,0,0,16,0,0,16,60,0,0,60,0,0,0,0,0,29,0,22,127,0.00,0.00,0.00,0.00,0.00
,0.00,0.00,0.00

04-10-
11,16:52:46,Tue,30,0,0,0,0,20,0,0,20,60,0,0,60,0,0,0,0,0,40,0,29,149,0.00,0.00,0.00,0.00,0.00
,0.00,0.00,0.00
```

Figure 4. **AD log file. Source: own research.**

The next function of the preprocessor is generating alerts. Preprocessor reads a predicted pattern of the network traffic (of all parameters) from the 'profile' file and generates alert when the current value exceeds 'minimum' to 'maximum' range for the current moment (the moment is given by day of the week, hour, minute and second corresponding to the intervals from the log file) from the profile file (see Figure 5).

DD-MM-YY, HH:MM:SS, Day of the Week, Time interval [s], TCP summary MAX [number of packet], TCP summary MIN [number of packet], TCP outgoing MAX [number of packet], TCP outgoing MIN [number of packet], TCP incoming MAX [number of packet], TCP incoming MIN [number of packet], TCP from this subnet MAX [number of packet], TCP from this subnet MIN [number of packet], UDP summary MAX [number of packet], UDP summary MIN [number of packet], UDP outgoing MAX [number of packet], UDP outgoing MIN [number of packet], UDP incoming MAX[number of packet], UDP incoming MIN[number of packet], UDP from this subnet MAX [number of packet], UDP from this subnet MIN [number of packet], ICMP summary MAX [number of packet], ICMP summary MIN [number of packet], ICMP outgoing MAX [number of packet], ICMP outgoing MIN [number of packet], ICMP incoming MAX [number of packet], ICMP incoming MIN [number of packet], ICMP from this subnet MAX [number of packet], ICMP from this subnet MIN [number of packet], TCP with SYN/ACK MAX [number of packets], TCP with SYN/ACK MIN [number of packets], WWW outgoing - TCP outgoing to port 80 MAX [number of packet], WWW outgoing - TCP outgoing to port 80 MIN [number of packet], WWW incoming - TCP incoming from port 80 MAX [number of packet],], WWW incoming - TCP incoming from port 80 MIN [number of packet], DNS outgoing - UDP outgoing to port 53 MAX [number of packet], DNS outgoing - UDP outgoing to port 53 MIN [number of packet], DNS incoming - UDP incoming from port 53 MAX [number of packet], DNS incoming - UDP incoming from port 53 MIN [number of packet], ARP-request MAX [number of packet], ARP-request MIN [number of packet], ARP-reply MAX [number of packet], ARP-reply MIN [number of packet], Not TCP/IP stacks packet MAX [number of packet], Not TCP/IP stacks packet MIN [number of packet], Total MAX [number of packet], Total MIN [number of packet], TCP upload speed MAX [kBps], TCP upload speed MIN [kBps], TCP download speed MAX [kBps], TCP download speed MIN [kBps], WWW upload speed MAX [kBps], WWW upload speed MIN [kBps], WWW download speed [kBps] MAX, WWW download speed MIN [kBps], UDP upload speed MAX [kBps], UDP upload speed MIN [kBps], UDP download speed MAX [kBps], UDP download speed MIN [kBps], DNS upload speed MAX [kBps], DNS upload speed MIN [kBps], DNS download speed MAX [kBps], DNS download speed MIN [kBps]

```
04-10-11,16:52:16,Tue,
30,0,10,4,1,0,0,0,0,0,0,1,0,1,0,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10.00,5.0
0,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
```

Figure 5. **AD profile file. Source: own research.**

The profile can be generated "manually" or by a Profile Generator using appropriate model based on historic values from the log file. The architecture affords easy implementation of different statistical models of the traffic and usage of different tools (i.e. statistical packets) for building profiles. For easy implementation of adaptive models (which have to generate profile 'incrementally' after getting current values of the traffic parameters), the values from the profile file are loaded by AD to the log file after each saving of the current traffic values.

Profile Generator

In the current version of AD the profile generator was designed based on R environment (see:[13]).

The profile generator produces four files:

- File containing predicted pattern (expected future values of parameters) of the network traffic based on the statistical model for a given future time period.
- Profile file containing minimum and maximum values of the parameters (limitations for alert generation).
- File containing calculated values of the model parameters.
- File containing traffic pattern (theoretical values of the parameters) for the past time (the same time as in the log file). This values are used to evaluate the quality of the model.

(See Figure 6), gray solid arrows means saving to and the black dotted ones – reading from the file.

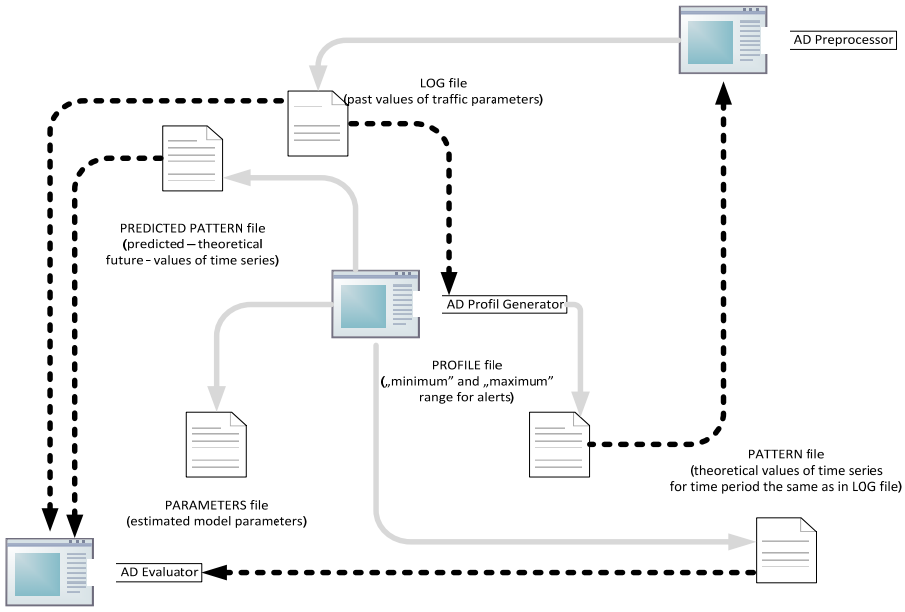

Figure 6. **AD Data flow diagram. Source: own research.Evaluator**

In the current version the Profile generator can build profiles based on four methods: Moving average, Naïve method, Autoregressive time series model and Holt-Winters model (see e.g. [4], [3], [1], [5], [8]).

The third program in the project (named Evaluator) is designed to compare simply statistic $\frac{MAE}{M}$ for two files (e.g. log file and predicted value profile).

MAE means Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| = \frac{1}{n}\sum_{t=1}^{n}|e_t|$$

where $e_t = y_t - \hat{y}_t$ is a model residual in the moment t and M is arithmetic mean. This way the Evaluator can be used either for checking fit between the model and historical data or between the predicted and real values.

All of the programs can be downloaded for free from the AD project page http://www.anomalydetection.info

References

[1] J. D. Brutlag, 'Aberrant Behavior Detection in Time Series for Network Monitoring' *14th System Administration Conference Proceedings,* New Orleans 2000, Pp. 139-146, available at: http://www.usenix.org/events/lisa00/full_papers/brutlag/brutlag_html/

[2] A. Fadia, M. Zacharia, '*Network Intrusion Alert. An Ethical Hacking Guide to Intrusion Detection'*, Thomson Source Technology, Boston 2008

[3] P. Goodwin, 'The Holt-Winters Approach to Exponential Smoothing: 50 Years Old and Going Strong', *FORESIGHT Fall* 2010 pp. 30-34, available at: http://www.forecasters.org/pdfs/foresight/free/Issue19_goodwin.pdf

[4] R. Lawton, 'On the Stability of the Double Seasonal Holt-Winters Method', unpublished, available at: forecasters.org/submissions09/LawtonRichardISF2009.pdf

[5] E. Miller, 'Holt-Winters Forecasting Applied to Poisson Processes in Real-Time' (draft, version August 2010), unpublished, available at: http://www.scribd.com/doc/35521051/Miller-Automated-Error-Detection-in-Web-Production-Environment

[6] E. A. Patkowski 'Mechanizmy wykrywania anomalii jako element bezpieczeństwa' [*Anomaly Detection Mechanisms as Safety Component], Biuletyn Instytutu Automatyki i Robotyki* No. 26/2009, Wydawnictwo Wojskowej Akademii Technicznej, Warsaw 2009

[7] O. Siriporn, S. Benjawan, 'Anomaly Detection and Characterization to Classify Traffic Anomalies. Case Study: TOT Public Company Limited Network', *World Academy of Science, Engineering and Technology* 48/2008

[8] M. Szmit, A. Szmit, 'Use of Holt-Winters Method in the Analysis of Network Traffic. Case Study', *Springer Communications in Computer and Information Science* vol. 160, pp. 224-231

[9] M. Szmit, Využití nula-jedničkových modelů pro behaviorální analýzu síťového provozu,] Internet, competitiveness and organizational security, TBU Zlín 2011, pp. 266-299

[10] M. Szmit, R. Wężyk, M. Skowroński, A. Szmit, 'Traffic Anomaly Detection with Snort, Information Systems Architecture and Technology ISAT 2007, Information

Systems and Computer Communication Networks', Wydawnictwo Politechniki Wrocławskiej, Wrocław 2007
[11] Skowroński M., Wężyk R.: Systemy detekcji intruzów i aktywnej odpowiedzi, ['*Intruders Detection and Active Response Systems*'] Master's Thesis, Lodz 2005, available at: http://maciej.szmit.info/documents/wezyk_skowronski.zip, Łódź 2006
[12] Tynenski A.: Bezpieczeństwo sieci komputerowych. Autorska dystrybucja systemu Linux, Master's Thesis, Lodz 2008
[13] The R Project for Statistical Computing, http://www.r-project.org
[14] Rehman R. U.: Intruder Detection With Snort, Prentice HallPTR, New Jersey 2003
[15] ACID Analysis Console for Intrusion Databases – program homepage http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html
[16] BASE Basic Analysis and Security Engine program homepage http://base.secureideas.net
[17] SAFE Snort Analysis Front End – Virtual Machine with the software http://www.vmware.com/appliances/directory/835163
[18] Wang Y.: 'Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection', IGI Global 2009
[19] Axelsson S.: 'Research in Intrusion-Detection System: A survey' http://www.cs.unc.edu/~jeffay/courses/nidsS05/surveys/Axelson99-ids-survey.pdf
[20] Telecommunication Standardization Sector of ITU (ITU-T) Recommendation E.507 Telephone Network and ISDN Quality of Service, Network Management and Traffic Engineering. Models for Forecasting International Traffic. ITU-T 1999,1993
[21] Skowroński M., Wężyk R., Szmit M.: Detekcja anomalii ruchu sieciowego w programie Snort, „Hakin9" Nr 3/2007, s. 64-68