Parameterized Anomaly Detection System with Automatic Configuration

Bruno B. Zarpelão and Leonardo S. Mendes
School of Elect. & Comp. Engineering
University of Campinas (UNICAMP)
Campinas, Brazil
{bzarpe, lmendes}@decom.fee.unicamp.br

Mario L. Proença Jr.
Comp. Science Department
State Univ. of Londrina (UEL)
Londrina, Brazil
proenca@uel.br

Joel J. P. C. Rodrigues
Instituto de Telecomunicações
DI, University of Beira Interior
Covilhã, Portugal
joeljr@ieee.org

Abstract— **This work proposes a parameterized anomaly detection system, based on the method known as profile based. The analysis of network elements is performed in two levels: (*i*) analysis of Simple Network Management Protocol (SNMP) objects data using a hysteresis-based algorithm to detect behavior deviations; (*ii*) analysis of alerts generated in the first level using a dependency graph, which represents the relationships between the SNMP objects. The proposed system is also able to configure its own parameters automatically, aiming to meet the network administrator needs. Tests were performed in a real network environment and great results were obtained.**

Keywords: Alarms, Network management, SNMP, MIB-II

I. Introduction

Network anomalies are defined as situations where network traffic levels show a sudden deviation from their normal behavior. They usually have a great impact on quality of services provided for end users. Besides, anomalies can cause the degradation of overall network performance, leading to the operations' disruption in the worst cases [1-3]. Among the various events that can cause anomalies, we can mention flash crowds, malfunctioning, network elements failures, vendor implementation bugs, misconfigurations, transfer of very large files, outages and malicious attacks such as DoS (Denial of Service), DDoS (Distributed Denial of Service) and worms [4-7].

In the literature, there are distinct approaches to define if a traffic behavior deviation is an anomaly or not. Thottan and Ji [3] consider as anomalies only the behavior deviations that end in operations' disruption. Lakhina *et al.* [1], Roughan *et al.* [5] and Tapiador *et al.* [8] showed some events that were not reported on *syslogs* and did not cause the operations' disruption, but they reflected in the quality of service provided for end users and the anomalies should have been detected.

This work proposes a parameterized anomaly detection system, based on the method known as profile-based, aiming to detect volume anomalies. This method establishes a profile for the normal behavior of the network by studying the data collected previously. The detection is accomplished by searching for significant behavior changes that are not coherent with the profile.

Our anomaly detection system applies heuristics in order to analyze the network elements in two different levels. In the first one, the system compares data collected from SNMP objects to their profiles of normal behavior, detecting anomalous activities when the SNMP object data deviate from the profile of normal behavior. A hysteresis-based algorithm is used for this purpose. In the second level, the alerts that were generated for each SNMP object are analyzed using a dependency graph, which represents the relationships between the SNMP objects. When the anomaly occurrence is confirmed in the second level, the system notifies the network administrator, presenting a map of the anomaly propagation in the network element.

Moreover, the proposed anomaly detection system is parameterized, i.e., it can be configured in order to provide results that meet the administrator requirements. Another important contribution of this work is an algorithm which is responsible for configuring the parameters of anomaly detection system automatically, aiming to fulfill the network administrator requirements. The network administrator requirements include the goals for detection and false positive rates and the characteristics of the deviations considered as anomalies. This algorithm receives as inputs the anomalies that should have been detected in previous weeks and the goals for detection and false positive rates. Various combinations of parameters' values are tested in the historical data. Then, the parameters values that have produced the results that are the closest to the administrator goals are returned as the algorithm output. Finally, the configuration that was selected by the algorithm is applied in detection of anomalies during the current period of monitoring.

The remainder of this paper is organized as follows. Section 2 presents some related work. The traffic characterization model applied in our work is showed in Section 3. In Section 4, we present the anomaly detection system, while Section 5 brings the details on configuration algorithm. Section 6 shows the results of the anomaly detection system evaluation, using a

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

real network element and, finally, the conclusions and future work are presented in the Section 7.

II. Related Work

Anomaly detection in computer networks has been studied by many researchers. Surveys about anomaly detection were presented in [2],[8]. Both works also proposed taxonomies in order to classify anomaly detection systems. Tapiador *et al.* [8] classified a list of systems from research projects. On the other hand, Lim and Jones [2] focused on classifying commercial products related to behavioral analysis.

Thottan and Ji [3] proposed a system that collects data from SNMP objects and organizes them as time series. An auto regressive process is used to model those time series. Then, the deviations are detected by a hypothesis test based in the method GLR (Generalized Likelihood Ratio). The behavior deviations that were detected in each SNMP object are correlated later according to the objects characteristics.

Kline *et al.* [9] have developed a new detection algorithm named S3. The first component of the algorithm applies wavelets to detect abrupt changes in the levels of ingress and egress packet counts. The second one searches for correlations in the structures of ingress and egress packets, based on the premise of traffic symmetry in normal scenarios. The last component uses a Bayes network in order to combine the first two components, generating alarms.

Other solutions were proposed to detect network-wide traffic anomalies [1, 11]. In [1], authors analyzed the traffic flow data using the technique known as PCA (Principal Component Analysis). It separates the measurements into two disjoint subspaces: the normal subspace and the anomalous subspace, allowing the anomaly detection. Li *et al.* [10] characterized the normal network-wide traffic using a Spatial Hidden Markov Model (SHMM), combined with topology information. The CUSUM algorithm (Cumulative Sum) was applied to detect the anomalies.

This work proposes the application of simple parameterized algorithms and heuristics in order to detect anomalies in network devices, building a lightweight solution. Besides, the system can configure its own parameters, meeting network administrator's requirements and decreasing the need for human intervention in management.

III. Traffic Characterization: DSNS and BLGBA

The traffic characterization used by our model for anomaly detection is focused on Digital Signature of Network Segment (DSNS) generated through the application of Baseline for Automatic Backbone Management (BLGBA) model on real historical network data. The BLGBA model and the DSNS were both proposed in [11]. This characterization should reflect the normal behavior expected for the traffic along the day.

The BLGBA model was developed based on statistical analyses. It is used to perform analyses for each second of the day, each day of the week, respecting the exact moment of the collection, second by second for twenty-four hours, preserving the characteristics of the traffic based on the time variations along the day.

The BLGBA algorithm is based on a variation in the calculation of statistical mode, which takes the frequencies of the underlying classes as well as the frequency of the modal class into consideration. The calculation takes the distribution of the elements in frequencies, based on the difference between the greatest G_{aj} and the smallest S_{aj} element of the sample, using only 5 classes. This difference, divided by five, forms the amplitude h between the classes, $h = (G_{\mathrm{aj}} - S_{\mathrm{aj}})/5$. Then, the limits of each L_{Ck} class are obtained. They are calculated by $L_{\mathrm{Ck}} = S_{\mathrm{aj}} + \mathrm{h}*\mathrm{k}$, where Ck represents the k class (k = 1...5).

The generated DSNS is constituted of elements named Bl_{i}. The Bl_{i} will be defined as the greatest element inserted in class with accumulated frequency equal or greater than 80%. The purpose is to obtain the element that would be above most samples, respecting the limit of 80%.

Figure 1 shows a chart of the daily movement of State University of Londrina (UEL) Web server, and its respective DSNS. The traffic levels expected from the estimative found in the DSNS are represented as a line. The real traffic is represented as vertical bars. The lighter bars show that the real traffic is below the DSNS and the darker ones mean that it overcame the DSNS. It is possible to observe a great adjustment between the real traffic and the DSNS.

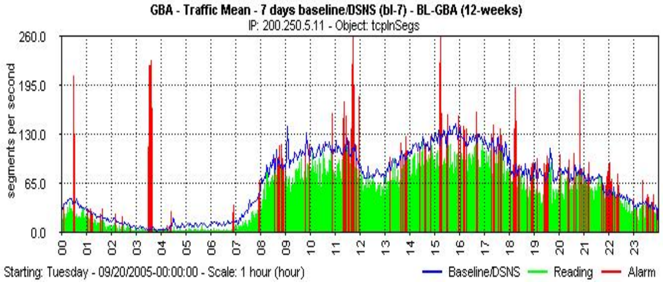

Figure 1. Real traffic and DSNS of UEL's Web Server.

IV. Anomaly Detection

In profile-based anomaly detection systems, anomaly detection is performed by comparing the profile of normal traffic to the real data, in order to identify sudden changes in the traffic levels. The anomaly detection system must be effective and present a low rate of false positives, besides generating a reduced amount of notifications that do not overload the network administrators.

Figure 2 presents the reference model of the anomaly detection system. The GBA tool (Automatic Backbone Management) [11] is responsible for collection and storage of samples, and the execution of the BLGBA model in order to generate the DSNS. The Alarm system reports the deviation detected through the comparison between the DSNS and the real movement pictured by the SNMP objects. The Correlation system gathers these alarms and analyzes them using the Correlation graph. Its function is to verify the occurrence of an anomaly and to offer a map of its behavior to the network administrator.

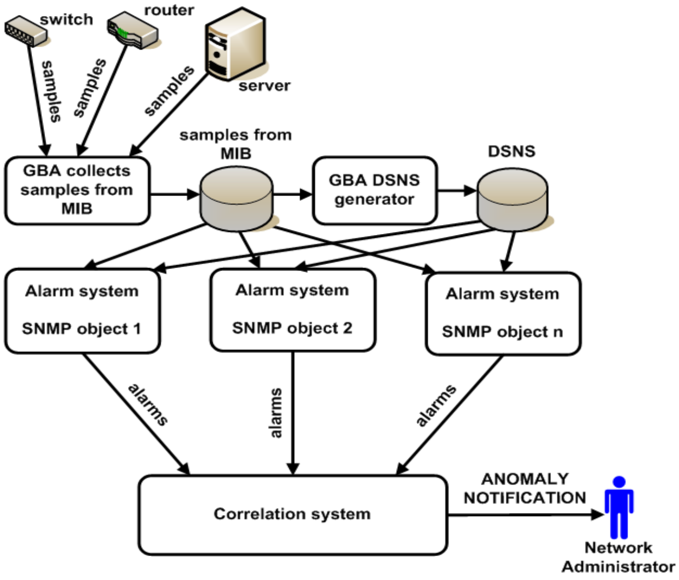

Figure 2. Anomaly detection system components.

A. Alarm System

The Alarm system indicates the occurrence of a behavior deviation in a specific SNMP object, generating an alarm when an event 3 happens. There are the following three possible events:

- Event 1: the real sample analyzed deviates from the limit established by the *DSNS'*. It represents the beginning of the hysteresis interval *t*. *DSNS'* is presented in (1).
- Event 2: during the hysteresis interval, the real sample analyzed overcomes the one concerning the previous occurrence of event 2. If it is the first occurrence of event 2, the current sample is compared to the one related to the event 1.
- Event 3: the number of occurrences of event 2 in the hysteresis interval *t* overcomes the value of a parameter named $\boldsymbol{\delta}$.

The occurrence of these three events is required to characterize a significant behavior deviation aiming to avoid the generation of false alarms. The parameters *t* and $\boldsymbol{\delta}$ can be configured in order to make the anomaly detection system results to meet the management policies established by the network administrator.

In event 1, it is possible to observe that the real sample is compared to *DSNS'* and not to *DSNS*. The *DSNS'* is the *DSNS* increased according to the value *factor*, as it is presented in (1). The *factor* is the third parameter of the system. It can be configured in order to change the number of hysteresis interval occurrences, changing also the sensitivity of the anomaly detection system.

$$DSNS' = DSNS + (DSNS \times factor) \qquad (1)$$

B. Correlation System

The correlation system analyzes all the first level alarms that were generated during the same five-minute time frame, based on the dependency graph presented in Figure 3. This graph is built from the relationships and properties of SNMP objects.

A graph *G* is a data structure defined by $G = (V, E)$, where *V* represents the set of vertices of the graph and *E* the set of edges that link vertices respecting a specific relation between them. For directed graphs, edges express a unidirectional relationship between two vertices and are represented by ordered pairs (x, y). In the dependency graph, an ordered pair (x, y) defines that an anomaly can propagate from the SNMP object represented by vertex *x* to the SNMP object represented by vertex *y*.

The correlation algorithm that gathers all the alarms generated and verifies the occurrence of an anomaly through the dependency graph was built based on the depth search algorithm [12]. The difference is that in the depth search algorithm the graph is processed going from a vertex to its adjacent, while in the algorithm used at the correlation system the graph is processed going from a vertex to its correlated. Two vertices are considered as correlated when they are adjacent and there are alarms generated for both SNMP objects in the same five-minute time frame.

V. Parameters Configuration

In this work, it is proposed a parameterized anomaly detection system with the following parameters: hysteresis interval length, $\boldsymbol{\delta}$ and DSNS' factor. They can be configured in many different ways, changing the sensitivity of the system and, consequently, situations that are considered as anomalous, detection rates and false positive rates. Since the system is well configured, results obtained from the operation of the anomaly detection system may be closer to the network administrator needs. However, it is not feasible to delegate to the network administrator the task of testing different combinations of values for the parameters, in order to find the best solution. Thus, it is proposed an algorithm that is responsible for configuring the parameters according to the administrator requirements.

The algorithm receives the following data input: anomalies occurred in last weeks, the goal for the detection rate, and the goal for the false positive rate. The values for the parameters are selected by the configuration algorithm aiming to ensure that the detection and false positive rates will be equal to or better than the goals.

In order to decide which configuration is the best to monitor a network element in the week w_n, the configuration algorithm tests a lot of different combinations of parameter values in the period from w_1 to w_{n-1}, which is named training period. The algorithm compares the anomaly notifications that were generated in the training period for each combination of parameter values with the list of anomalies that were inserted by the network administrator, calculating the detection and false positive rates. The configuration that presents the closer results with respect to the goals is selected to the week w_n.

Next sub-sections present how sets of possible values for the parameters are constructed.

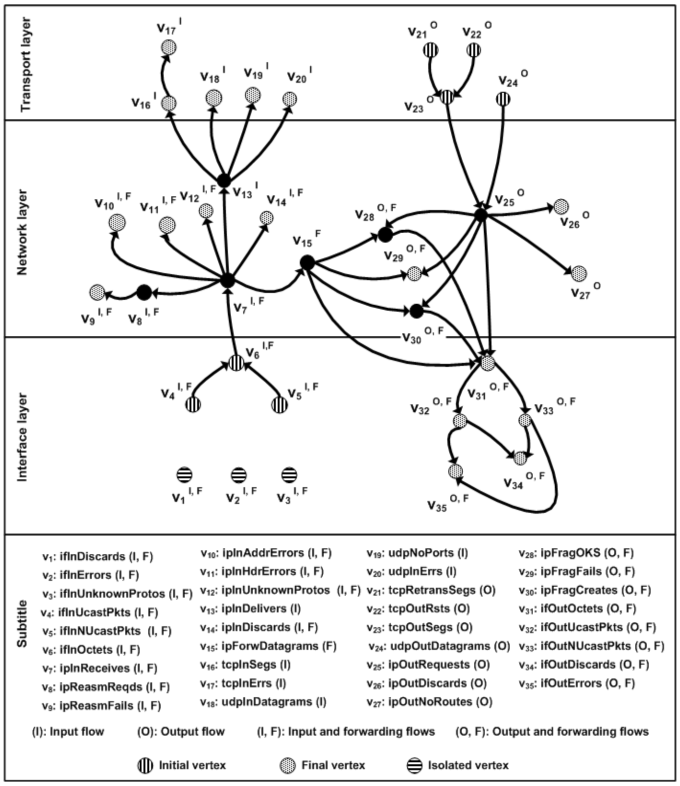
Transport layer
Network layer
Interface layer
Subtitle
v_{17} I
v_{16} I
v_{18} I
v_{19} I
v_{20} I
v_{21} O
v_{22} O
v_{23} O
v_{24} O
v_{10} I, F
v_{11} I, F
v_{12} I, F
v_{13} I
v_{14} I, F
v_{15} F
v_{28} O, F
v_{29} O, F
v_{25} O
v_{26} O
v_{27} O
v_9 I, F
v_8 I, F
v_7 I, F
v_{30} O, F
v_6 I,F
v_4 I, F
v_5 I, F
v_{32} O, F
v_{31} O, F
v_{33} O, F
v_{34} O, F
v_{35} O, F
v_1 I, F
v_2 I, F
v_3 I, F
v_1: ifInDiscards (I, F)
v_2: ifInErrors (I, F)
v_3: ifInUnknownProtos (I, F)
v_4: ifInUcastPkts (I, F)
v_5: ifInNUcastPkts (I, F)
v_6: ifInOctets (I, F)
v_7: ipInReceives (I, F)
v_8: ipReasmReqds (I, F)
v_9: ipReasmFails (I, F)
v_{10}: ipInAddrErrors (I, F)
v_{11}: ipInHdrErrors (I, F)
v_{12}: ipInUnknownProtos (I, F)
v_{13}: ipInDelivers (I)
v_{14}: ipInDiscards (I, F)
v_{15}: ipForwDatagrams (F)
v_{16}: tcpInSegs (I)
v_{17}: tcpInErrs (I)
v_{18}: udpInDatagrams (I)
v_{19}: udpNoPorts (I)
v_{20}: udpInErrs (I)
v_{21}: tcpRetransSegs (O)
v_{22}: tcpOutRsts (O)
v_{23}: tcpOutSegs (O)
v_{24}: udpOutDatagrams (O)
v_{25}: ipOutRequests (O)
v_{26}: ipOutDiscards (O)
v_{27}: ipOutNoRoutes (O)
v_{28}: ipFragOKS (O, F)
v_{29}: ipFragFails (O, F)
v_{30}: ipFragCreates (O, F)
v_{31}: ifOutOctets (O, F)
v_{32}: ifOutUcastPkts (O, F)
v_{33}: ifOutNUcastPkts (O, F)
v_{34}: ifOutDiscards (O, F)
v_{35}: ifOutErrors (O, F)
(I): Input flow
(O): Output flow
(I, F): Input and forwarding flows
(O, F): Output and forwarding flows
Initial vertex
Final vertex
Isolated vertex

Figure 3. Dependency graph.

A. Hysteresis interval

The set *H = {60}* defines a single value for the hysteresis interval. It ensures that the first level alerts will be generated in a maximum time of 60 seconds.

B. Delta (δ)

The set of possible values for ***δ*** is defined in the arithmetic progression presented in (2).

$$D = \{d_1, \quad d_2, \quad \ldots, \quad d_n\} \tag{2}$$

In order to define the first and last elements of *D*, we analyzed a sample of four weeks of data from State University of Londrina network.

The amount of hysteresis intervals where the event 2 occurs only one time is significant. Thus, the first element of the set *D* is defined as $d_1 = 1$.

The amount of hysteresis interval where the event 2 occurs more than 30 times represents less than 1% of the analyzed sample, considering a hysteresis interval of 60 seconds and a polling interval of 1 second. Thus, the last element d_n of *D* is defined in (3), where c_m represents the smallest polling interval in all the objects e h_i represents the hysteresis interval.

$$d_n = \left\lceil \frac{30}{c_m} \right\rceil * \frac{h_i}{60}, \tag{3}$$

The common difference of arithmetic progression*,* represented as *r*, is calculated according to (4), where c_m represents the smallest polling interval and h_i represents the hysteresis interval:

$$r = \left\lceil \frac{5}{c_m} \right\rceil * \frac{h_i}{60} \tag{4}$$

The ***δ***, which is applied in each SNMP object, depends on the polling interval of the object. So, there are two different ways of applying the ***δ***:

1. All the SNMP objects have the same polling interval: the common difference is calculated with this polling interval. When the anomaly detection system is executed, the ***δ*** values are obtained directly from the arithmetic progression and they are applied to all the SNMP objects.
2. Polling intervals are different. The smallest polling interval is used to calculate the common difference, and the arithmetic progression is built. Then, for each SNMP object, the ***δ*** is calculated according to (5), where d_j is the value obtained directly from the arithmetic progression, *d'$_j$* is the value adapted for the given SNMP object and c_M is the polling interval of the given SNMP object.

$$d'_j = \left\lceil d_j \Big/ \frac{c_M}{c_m} \right\rceil \tag{5}$$

C. DSNS' factor

The possible values for DSNS' factor are defined in the arithmetic progression presented in (6). The first element of *F* is *$f_1 = 0$*. The last element f_n will be defined when the DSNS' factor is high enough to result in a detection rate of 0. The common difference is 0.1.

$$F = \{f_1, \quad f_2, \quad \ldots \quad f_n,\} \tag{6}$$

D. Combination for buiding the parameters set

After *H*, *D* and *F* have been defined, the set of different combinations for parameter values can be constructed, according to (7). Each combination of parameter values is defined as *p*. As it is showed in (8), each element *p* is defined as a 3-uple, where h_i, d_j and f_k represent the values obtained from *H*, *D* and *F* respectively.

$$P = \{p_1, \quad p_2, \quad \ldots \quad p_n,\} \tag{7}$$

$$p_l = \{h_i, \quad d_j, \quad f_k,\} \tag{8}$$

The anomaly detection system analyzes the training period using all the elements of *P*. The detection and the false positive rates are calculated for each element *p*. The detection rate is represented in the function *f(p, t),* where *t* is the period to which the rate was calculated. The false positive rate is represented in the function $g(p, t)$.

Finally, after all the combinations of parameter values have been tested, the system classifies all the elements of P to construct P', using the following metric:

- p_i is better than p_j if the condition presented in (9) is true. The function h is defined in (10), where o_d is the goal for detection rate and o_f is the goal for false positive rate.

$$\mathrm{h}(\mathrm{f}(p_i,t),o_d,\mathrm{g}(p_i,t),o_f) < \mathrm{h}(\mathrm{f}(p_j,t),o_d,\mathrm{g}(p_j,t),o_f) \qquad (9)$$

$$\mathrm{h}(\mathrm{f}(p,t),o_d,\mathrm{g}(p,t),o_f) = \frac{(|\mathrm{f}(p,t)-o_d|)+(|\mathrm{g}(p,t)-o_f|)}{2} \qquad (10)$$

The element p_1 from P' represents the best combination of values for the parameters, considering the goals established by the administrator. These values will be applied in the monitoring of the next week after the training period.

E. Use case

This sub-section presents a use case of the anomaly detection system, focusing on the configuration algorithm. It was performed using data collected from the proxy server of State University of Londrina, in a period of four weeks, from May 13, 2007 to June 9, 2007. The first week was only used as training period. The polling interval of the objects is 10 seconds.

At first, the anomalies that occurred from May 13 to May 19 were registered in the system. The objective was to find the best configuration to monitor the week from May 20 to May 26. The goal for detection rate was 80% and for false positive rate was 20%. The configuration algorithm tested different combinations of values for the parameters in the training week from May 13 to May 19, saving the obtained detection and false positive rates. According to the algorithm, during the training period, the configuration with hysteresis interval = 60 seconds, $\boldsymbol{\delta}$ = 1 and DSNS' factor = 0.8 presented the rates that were the closest to the goals: 80% and 11%. Therefore, the algorithm selected this configuration to be applied in the week from May 20 to May 26. Its application resulted in a detection rate = 81% and in a false positive rate = 7%. Both were better than goals.

Then, anomalies for the period from May 20 to May 26 were inserted in the system. The goal for false positive rate was decreased to 10%. The goal for detection rate remained the same. The algorithm selected the following configuration taking into account the training period of two weeks from May 13 to May 26: hysteresis interval = 60 seconds; $\boldsymbol{\delta}$ = 1; DSNS' factor = 0.8. The following results were obtained for the week from May 27 to June 02: detection rate = 83%; false positive rate = 2%.

Finally, anomalies for the period from May 27 to June 3 were inserted in the system. The goals remained the same. The algorithm analyzed the training period of three weeks from May 13 to June 2 and decided that the best configuration was the following: hysteresis interval = 60 seconds; $\boldsymbol{\delta}$ = 1; DSNS' factor = 0.8. The following results were obtained for the week from June 3 to June 9: detection rate = 77%; false positive rate = 4%.

Figure 4 presents the comparison between results and goals for false positive rates in weeks 2, 3, and 4 of the use case. The first week of the use case was used only for training. Results are satisfactory, since all false positive rates are lower than goals.

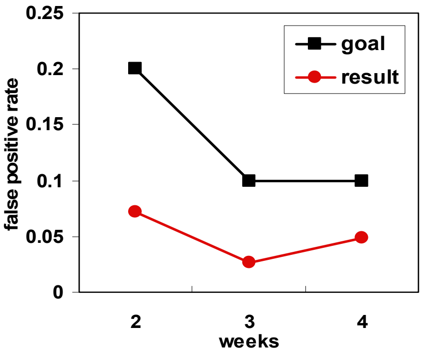

Figure 4. False positive rates in use case.

Figure 5 presents goals and results for detection rates in weeks 2, 3 and 4. Results are satisfactory even if the last week, where the detection rate was lower than the goal, is taken into account. The difference between the goal and the detection rate in the last week was about 2 percentage points.

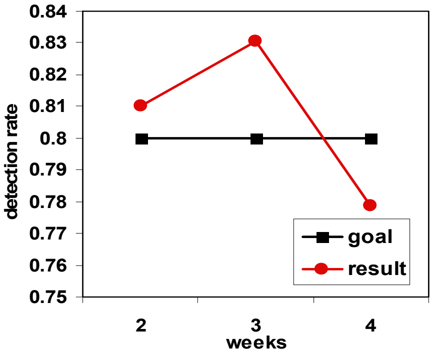

Figure 5. Detection rates in use case.

VI. Results and Evaluation

Tests were performed at the proxy server of the network of the State University of Londrina (UEL). It interconnects 4000 computers to the Internet. Besides, it controls the access to unauthorized web pages. The following SNMP objects were monitored during the tests: *ipInReceives, ipInDelivers, tcpInSegs, udpInDatagrams, tcpOutSegs, udpOutDatagrams* and *ipOutRequests*. The polling interval for all the objects is 10 seconds.

The main evaluation objective is to verify if the obtained rates are close to or better than initial goals. Tests were performed for 9 different pairs of goals, combining three different values for detection rate goals with three different values for false positive rate goals. The values used were 70%, 80% and 90% for the detection rate goal, and 10%, 20% and 30% for the false positive rate goal. Tests were performed in a period of 4 weeks from May 13, 2007 to June 09, 2007. As the first week was only used as a training period, we collected results from the other three weeks. The results are the following:

- False positive rate: 25 out of 27 false positive rates were better than the goal. The two situations where the false positive rates were worse than the goals were:
    - Detection rate goal = 90% and false positive rate goal = 10% in the week from May 20 to May 26, the false positive rate were 11%, showing only 1 percentage point of difference to the goal;
    - Detection rate goal = 90% and false positive rate goal = 20% in the week from May 20 to May 26, the false positive rate were 22%, showing only 2 percentage points of difference to the goal;
- Detection rate: 20 out of 27 detection rates were better than the goals. In the 7 cases where the detection rates were worse than the goal, 6 cases occurred in the last week. It shows that the last week has different characteristics in relation to the three previous weeks that were used in the training. However, the results were satisfactory even in those situations. The worst result was a detection rate of 83% when the goal was 90%.

It can be observed in the results that the configuration algorithm selects, in the most of times, values for the parameters that produce rates, which are close to the goals inputted by the network administrator.

The false positive and detection rates are related and present a trade-off. If the parameters' values are stricter, the anomaly detection system becomes less sensitive, the false positive rates get better and detection rates get worse. On the other hand, if the parameters' values are more lenient, the anomaly detection system becomes more sensitive, the false positive rates get worse and detection rates get better. The proposed configuration algorithm handled successfully these characteristics.

VII. Conclusions

This paper proposed a parameterized anomaly detection system, based on the method known as profile-based. It is possible to change the behavior of the proposed anomaly detection system by configuring its parameters, which allows that the results meet the needs of the network administrator easily. Since it is not feasible for the network administrator to keep testing a lot of combinations of parameter values until the results are the expected, we also proposed an algorithm that is able to configure the system parameters automatically.

Satisfactory results were obtained in the experiments performed in an element from the State University of Londrina (Brazil) network. For the false positive rates, 25 out of 27 situations presented rates that were better than the goals required by the network administrator. For the detection rates, 20 out of 27 situations presented rates that were better than the goals. Even in cases where the detection rate was worse than the goal, results were close to the expected and they were satisfactory.

Future work includes the creation of a third level of analysis in the system, providing an anomaly localization system with a network-wide view. The anomaly notifications generated in the second level of analysis will be correlated, using information about the network topology in order to find the root-cause of anomalies.

Acknowledgments

Part of this work has been supported by The State of São Paulo Research Foundation (FAPESP), Brazil, and by *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Covilhã, Portugal.

References

[1] Lakhina, M. Crovella, C. Diot "Diagnosing Network-Wide Traffic Anomalies". ACM SIGCOMM Computer Communication Review, Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, v. 34, p. 219-230, 2004.

[2] S. Y. Lim and A. Jones "Network Anomaly Detection System: The State of Art of Network Behaviour Analysis" International Conference on Convergence and Hybrid Information Technology 2008, p. 459-465, 2008.

[3] M. Thottan, C. Ji "Anomaly Detection in IP Networks" IEEE Transactions in Signal Processing, v. 51, n. 8, p. 2191-2204, 2003.

[4] J. Li, C. Manikopoulos. "Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters." Proceedings of the 2003 IEEE Workshop on Information Assurance, United States Military Academy, p. 53-59, 2003.

[5] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, B. Freeman "IP Forwarding Anomalies and Improving their Detection Using Multiple Data Sources" Proceedings of the ACM SIGCOMM workshop on Network troubleshooting: research, theory and operations practice meet malfunctioning reality, p. 307-312, 2004.

[6] N. Saaman and A. Karmouch, "Network Anomaly Diagnosis via Statistical Analysis and Evidential Reasoning," IEEE Transactions on Network and Service Management, v. 5, no. 2, 2008.

[7] Y. Zhang, Z. Ge, A. Greenberg, M. Rhoughan. "Network Anomography". Proceedings of ACM SIGCOMM Internet Measurement Conference 2005 (IMC'05), p. 317-330, 2005.

[8] J. M. Tapiador, P. G. Teodoro and J. E. D. Verdejo. "Anomaly detection methods in wired networks: a survey and taxonomy". Computer Communications, 27, p. 1569-1584, 2004.

[9] J. Kline, S. Nam, P. Barford, D. Plonka and A. Ron "Traffic Anomaly Detection at Fine Time Scales with Bayes Net" The Third International Conference on Internet Monitoring and Protection, p. 37-46, 2008.

[10] M. Li, S. Yu and L. He "Detecting Network-wide Traffic Anomalies based on Spatial HMM" 2008 IFIP International Conference on Network and Parallel Computing, p. 198-203, 2008.

[11] M. L. Proença Jr., C. Coppelmans, M. Bottoli, L. S. Mendes. "The Hurst Parameter for Digital Signature of network Segment". 11th International Conference on Telecommunications (ICT 2004), 2004, Fortaleza. Springer-Verlag in the LNCS series. p. 772-781, 2004.

[12] J. L. Gersting "Mathematical Structures for Computer Science". 5 ed., W H Freeman, 2002.

[13] W. Stallings "SNMP, SNMPv2, SNMPv3, and RMON 1, 2 and 3". Addison-Wesley, 1998.

[14] K. McCloghrie, M. Rose "Management Information Base for Network Management of TCP/IP-based internet: MIB-II". RFC 1213, mar 1991.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.