Event and Performance Logs in System Management and Evaluation

Chapter · January 2011

CITATION
1

READS
180

3 authors, including:

Janusz Sosnowski
Warsaw University of Technology
182 PUBLICATIONS **680** CITATIONS
SEE PROFILE

Piotr Gawkowski
Warsaw University of Technology
84 PUBLICATIONS **284** CITATIONS
SEE PROFILE

EVENT AND PERFORMANCE LOGS IN SYSTEM MANAGEMENT AND EVALUATION

Janusz Sosnowski, Piotr Gawkowski, Krzysztof Cabaj

Institute of Computer Science, Warsaw University of Technology

Abstract. The paper outlines the space of event and performance logs which can be collected in computer systems to support managing and evaluating system operation. We concentrate on methodology of finding anomalies and assessing system dependability or resilience. General considerations are illustrated with more details on storage problems and network attacks.

Keywords. System logs, Performance, Reliability, Dependability evaluation, System administration.

1. INTRODUCTION

Contemporary IT systems are becoming more and more complex in hardware and software. Hence various problems may occur due to reliability issues, design faults, configuration inconsistencies, component interactions, external attacks, etc. Detection and diagnosis of these problems is not trivial. This process can be supported by collecting and analysing various event and performance logs.

Most publications devoted to system logs (e.g. [2,9,11] and references) are targeted at system reliability issues (detection and prediction of failures). Moreover, they are limited to some specific systems. We extend this approach for more general problems related to anomalies and system resilience (capability of adapting to changing conditions). The format of registered logs is not uniform and their informative contents is quite often ambiguous [2,9]. We have got some experience in these aspects monitoring many computers used in our Institute (e.g. [11]). This experience allowed us to develop more systematic approach to dealing with two aspects: finding operational profiles of the systems (in different time perspectives: instantaneous behaviour and trends) and identifying anomalies by combining and correlating various logs (covering different resources, their state changes, performance issues, etc.). In particular we consider the problem of selecting the most sensitive monitoring measures related to these aspects. The presented considerations are related to the developed data repository system.

Section 2 presents the scope and goals of system monitoring. Section 3 outlines some specific issues of storage monitoring – crucial point in many systems. Section 4 comments detection of external attacks, section 5 concludes this work.

2. THE SPACE AND GOALS OF SYSTEM MONITORING

Computer systems are instrumented to provide various logs on their operation. These logs comprise huge amounts of data describing the status of system components, operational changes related to initiation or termination of services, configuration modifications, execution errors, security threats, etc. Events are stored in various logs, for example: security, system, application logs, etc. The list of possible events in Windows systems exceeds 10000 [11]. In Unix and Linux systems over 10 sources of events and more priority levels are distinguished.

The formats of registered events have some loosely defined general scope. In particular we can distinguish various data fields comprising specific information in textual or numerical form with some specific brackets, etc. Some fields can be considered as parameters (compare section 4). Usually, at the beginning, we have the time stamp (the time of event registering), name of the event source (e.g. disk, application program, process PID, etc.), text describing the related event problem, severity of the problem. However, events of different classes can be stored in different log files (e.g. security events specifying authorization problems, user login and logout events). The included texts can be very general and of low information value or more specific. Their meaning can be better interpreted after gathering some practical experience within a longer observation time period.

Having checked the capacity of collected logs in some sample computers within the Institute (and some external ones) we can state that the number of registered events is quite high even in systems with low activity. In most cases the system operates correctly, so identifying critical or warning situations is to some extent problematic. Only some events are critical, on the other hand some event sequences or contexts can also be considered as interesting for the further analysis. Targeting at such analysis we should start with identifying different classes or types of events. If we take into account complete event specification than each event is distinct at least within the time stamp field (however, due to limited registration time granularity, the same time stamp is also possible). Having rejected the time stamp field we still notify a large number of different event reports, so some form of more sophisticated categorisation is needed. In particular it may be reasonable to eliminate in this categorisation argument fields (assuming various value e.g. PID). Moreover, such categorisation may be user oriented so it seems reasonable to introduce some flexible mechanism for this purpose e.g. regular expressions, to abstract the events [11]. In the case of Windows it is simpler due to explicit event

ID. Moreover, some time and space filtering can be used to eliminate some redundancy in events.

Another issue is getting long term experience by monitoring event logs for longer periods and on different hardware and software platforms. This monitoring should be correlated with systematic users and administrators observations, their reports on operation anomalies, occurred system crashes, power blackouts, network disconnections, system overloading or other problems. All these situations should be described and registered in some special repository. This can be confronted with the collected logs at the time of problem appearance or in a postponed log analysis.

In parallel with event logging, various data on performance can be collected in appropriate counters (e.g. provided by Windows, Linux) and according to some sampling policy [4]. These counters are correlated with performance objects such as processor, physical memory, cache, physical or logical disks, network interfaces, server of service programs (e.g. web services), I/O devices, etc. For each object many counters (variables) are defined characterising its operational state, usage, activities, abnormal behaviour, performance properties, etc. Special counters related to developed applications can also be added. These counters provide data useful for evaluating system dependability, predicting threats to undertake appropriate corrective actions, etc. The number of possible performance variables is quite big (many hundreds) and monitoring all of them is too expensive due to the additional load to the system processors and memory [11,13]. Hence, an important issue is to select those variables which can provide the most useful information. This depends upon the goals of monitoring, the sensitivity of variables to the monitored properties of the system, the system usage profile, etc.

Monitoring various objects and selected variables we can identify operational profiles in relevance to different system workloads, operational times, etc. Moreover, they can provide some symptoms of anomalies, errors, etc. The anomaly can be identified by tracing its characteristic properties which differ it from the normal system workload. We can distinguish three classes of anomalies (this is a generalized and extended notion of [1]): i) *point anomalies* – individual measured features or attributes related to system components or the whole system are different from normal values or states; ii) *conditional anomalies* (in [1] called contextual) – the observed features and attributes can be considered as symptoms of anomalies under additional behavioral (not typical) or contextual (related to spatial ant time context) conditions. *Complex anomalies* (in [1] called collective) are specified by multidimensional features (involving combined state of some set of observed measures).

We can look for some statistical deviations (as compared with the normal operation) of the monitored variables e.g.: increase (M+) or decrease (M-) in the mean from the compared reference condition, unimodal left skewed (DUL) or right skewed (DUR), uniform, etc. This approach has been used in identifying 11 cyber-

attacks in [13]. More sophisticated approaches base on finding various correlations between variables, conditioned with events, etc.

Collecting appropriate data within observation window (ΔT_o) allows us to determine that within the time period [ΔT_p, $\Delta T_p + \Delta T_v$] (where ΔT_p - predicted time of problem occurrence starting from the end of the observation period, ΔT_v - time of prediction validity) a problem may appear. Sometimes this prediction can be supported with the probability of problem occurrence. Moreover, the time interval can also be defined in some probabilistic way (fuzzy interval). Some problems can be detected by specific mechanisms almost immediately (low ΔT_o e.g. parity error detection) in a deterministic way. In practice various situations are possible, in particular the problem really can happen in the previewed perspective (CR – correct prediction), beyond this perspective (sooner or later – IP imprecise positive prediction), will not occurs at all (FA – false alarm prediction), will occur despite no detected symptom in the observation time window (SP – skipped prediction or incorrect negative prediction), will not occur and will not be predicted to occur (CN – correct non prediction or true negative). The frequency of these situations defines the quality of the implemented prediction mechanisms. This qualification can be extended with problem attributes (fine granularity prediction). For example predicting faults we can specify their nature: permanent, intermittent, transient.

Searching for system anomalies we can be targeted at some specific problems and correlate them with appropriate events, environment conditions, performance variables, etc. Some of these problems can be obvious (e.g. system failures), others may need some practical experience (e.g. those related to performance issues). Another issue is defining unique and effective symptoms which assure high accuracy and precision of forecasting [14]. One step further is looking for unknown problems, which in fact can be hidden and not harmful at least for some time, however they grow systematically and result in dangerous situations. Looking for such potential threats needs observing many variables in the system, finding their profiles, trends, etc. For example some users may create problems resulting from not sufficient skill in using a new application (this can be identified by comparing operational profiles of many users), cyber-attacks may not cross the introduced firewalls, so we do not feel any harm, but it may be interesting to identify that the system is an object of hackers interest. Correlation of uneven resource or system load, communication traffic may also be attributed to various hidden unknown anomalies.

Analysing performance variables it is important to find mean, maximal, minimal values but also their trends in different time perspectives, distribution in time and amplitude of spike values, distribution of width (burst periods), correlations with other variables, events, etc. Dealing with resiliency we have to check system capabilities of using alternate options, handling additional demands without degradation or loss of functionality, arranging sufficient resources and services in critical

situations (emergencies) or environment changes. Deploying safety procedures, restoring system stability in time, etc., it is worth noting that generated warning signals by many systems (e.g. based on detecting some threshold crossing, specific event) are not accurate and often misleading, so it is reasonable to refine these mechanisms taking into account the experience of system administrators.

3. MONITORING STORAGE SYSTEM

Monitoring storage system covers several administrative aspects. First of all, the user can get the knowledge of current usage profile (e.g. storage capacity in use, read-only data volume, temporary data capacities and accessibility profile), usage trend (e.g. how much storage the company needs in the future? What our demands are – do we need higher performance for read or write, for sequential or for random IO operations, etc.), and finally, the reliability of the system. Among others, the storage reliability is a crucial issue. In [7] authors reported that over 70% of all components failures in large datacenters (more than 100000 servers) are related to the hard disk drive failures. To better understand the "space" of monitoring it is good to be aware of possible failures of the storage system, a heart of which is a set of magnetic hard disk drives organized as RAID (Redundant Array of Independent Disks) arrays for better availability and performance.

One of the important parameter of a hard disk (from the dependability perspective) is the Unrecoverable Error Rate (UER) which defines the rate of possible data loss during disk drive operation. This parameter is assumed to be 10^{14} for desktop-class HDD to 10^{16} bits for the enterprise-class disks. The user may expect the disk operation failure after processing the specified number of bits (i.e. 10^{14} bits is 12.5 terabytes of information). As the disk capacities are growing, using them dramatically decreases the RAID dependability. Assuming five 500GB disks with 10^{14} bits UER in the RAID-5 configuration, there is 20% probability of the second failure during the array rebuild process (array failure). Much better characteristic of enterprise-class HDDs is achieved by manufactures in several ways, both in mechanics and electronics [3, 12], e.g. fully certified magnetic platters against defects with smaller diameters, heavy duty motors with higher speeds, dual processors, error correction and data integrity checks.

Looking closer, the magnetic hard disks are very complicated devices, composed of many high precision mechanical parts, and complicated software (firmware – an erroneous execution of which can provoke the disk failure – e.g. buggy firmware in Seagate's Barracuda 7200.11 disks), and bases on analogue magnetic media. So, the set of possible fault models includes magnetic domain (e.g. platters defects), mechanical (e.g. spindle, heads arm actuators), electronic (e.g. head amplifiers, cache memories, physical interface parts), and software related problems (e.g. bugs in firmware). In fact, a single fault may lead to complex failure scenarios

as mechanical problems may result in magnetic platter damages that may be serious if the vital firmware information is no longer accessible due to the unreadable sector. Some problems are not predictable. However, some of them may be spotted before the catastrophic scenario takes place as their symptoms might slowly arise. For instance, some mechanical problems (with head servo mechanisms, spin engines etc. due to environmental conditions such as dust, humidity, temperature, air pressure, mechanical vibrations etc.) can lead not only to sudden failure but also to some performance degradation. Similarly, the defective sector will be relocated – that also impacts the access time to that sector and is also logged by the disk firmware. In [5] authors report that the disk failure probability is 39x higher within 60 days period after the first scan error. Many other disk operation parameters are also available for the analysis (discussed later on).

It has to be stressed, that due to the complexity of possible defects the early failure prediction or even some error detection is complicated. It can be based on monitoring techniques (section 2). The first idea of disk self-monitoring was introduced in 1992. Recently the SMART (Self-Monitoring, Analysis and Reporting Technology) technique is available in all contemporary disks. The disk firmware monitors a set of attributes related to various aspects. Each attribute (beside its identifier and name) is represented with four values: the current, the worst, the threshold and the raw one. However, the current value is not a direct representation of the attribute's property in most of the attributes. It is rather a calculated value hard to be interpreted directly. The idea is to change the current value (according to some function) – it is assumed that the current value should be higher than the threshold. If it is not true, the BIOS of the computer report the drive to be failing and warn the user. It has to be stressed that attributes reported through the SMART technology and their interpretation can be vendor or even model specific (e.g. different set of attributes, different meaning), and unfortunately, in most of the cases, they are not clearly defined (especially the current and raw values) by the manufactures. That creates problems with building general purpose software to analyze SMART readouts. The goal of the SMART is to prevent from outages/data losses caused by "predictable degradation and/or fault of the device". As some failures have unpredictable nature the user should not be disappointed in such cases. On the other hand, the SMART may warn the user with false alarms. Nevertheless, this technology is definitely not exploited exhaustively. The SMART can be also used as a valuable source of information about the disk to be used in more sophisticated monitoring and analysis systems. Authors in [5] report strong correlation of the disk age (represented in SMART by power-on hours and the number of power cycles) with failure probability. The increase of the operating temperature can originate in ventilation problems (e.g. dust or failure), or reflect the heavy usage of the storage at that time. Higher operating temperatures are reported in [7] as strong failure factor.

It is worth to note that different workloads can impact the failure rate of the HDDs. In [12] authors present that the failure rate of the desktop HDDs can be 2 times higher executing the heavy duty workload. Moreover, it raises another 2x if exposed to the workload based on low-end server pattern. Other important factors relate to the environmental conditions the disk operates in. Among others, they are related to the disk mounting (e.g. firmly mounted with vibration suppression), chassis stability and ventilation. In the enterprise reality this is related to the disk mounting position within the rack and its location within the datacenter [7].

The disk performance reflects in several aspects related to storage management. First of all, the storage performance has to be monitored to keep-up with growing demands of the users and systems it handles, letting purposeful planning of storage infrastructure management. In particular, it also impacts data safety (backup and archiving management, dependability of data carriers, etc.). In the simplest case the storage system can be described with capacity and performance measures (e.g. operations per second, read/write throughput, latency of the requests). Such basic properties are commonly available in the performance monitoring applications built-in the nowadays operating systems. Monitoring them, the administrator should not only take care of the temporary values but also correlate that with different period's perspectives: is the hourly average similar to the yesterdays? Even if they differ, is the profile similar to the one observed a month ago? Or maybe one of the departments is just processing the yearly financial reports – so, the storage usage profile should be correlated with the last year data. As stated in section 2 the analysis should also include the average performance trend. The performance degradation and higher error rates can be used as a one of predicates to suspect failing storage or vibration problems [3], e.g. the access times decreases as the servo mechanism cannot correctly position the magnetic heads (due to mechanical problem or the sector markers are hard to be tracked).

One of the most important conclusions from these observations is that to build more accurate failure prediction model of magnetic disks the one should take into account several sources of information: hard disk built-in SMART attributes (the current value and the historical trend), high level (operating system level) profile of the hard disk usage, and environmental conditions (current, changes and dynamics) – operating temperatures, humidity, vibrations. In the monitored systems we have predicted disk problems in relevance to excessive number of 8219 events (excessive file revealing time) in VSS (Volume Shadow Copy Service) log and excessive initialization time od some applications (performance monitoring).

4. MONITORING NETWORK ATTACKS

As was mentioned in section 2 events and performance logs can be used as well for detection of various attacks directed to the monitored system. Although

many dedicated security systems exists, for example, IDS/IPS (Intrusion Detection System/Intrusion Prevention System), WAF (Web Application Firewall), AV (Anti Virus) and UTM (Unified Threat Management), recent studies show that in many cases appropriate logging mechanism can be used as an additional layer of security. In [8] a report concerning security events in large US computing organization states that only 31% of them are detected by deployed IDS solution and 4% by deployed file integrity checks, which can be treated as simplest host based IDS. In contrast a higher percentage of events (37%) can be detected by analyzing the anomalies in various monitored and logged activities. Additionally, archived history logs are very useful in Forensic Analysis (FA) after detection of the successful attack. This is especially important due to the fact that 27% of attacks are not detected by any layered security (e.g. IDS or file integrity mechanisms) but are reported by external sources. In the sequel we present our experience in detecting network attacks basing on two methods: monitoring some performance parameters (CPU, memory, network traffic) using SNMP (Simple Network Management Protocol) architecture and tracing events collected by Linux syslog demon.

SNMP architecture consists of central Network Management System (NMS) and Agents in various network devices and computers. Each agent manages Management Information Bases (MIBs), which are a description of various parameters that it can monitor. Depending on the system various specialized MIBs are supported. What should be emphasized, NMS can monitor any device or software as soon as its developer provide MIB description in ASN.1 (Abstract Syntax Notation no. 1). On the market there exists many NMS, for example, IBM Tivoli, HP OpenView or even open source OpenNMS. However, a simpler solution can be used. In our research we base on MRTG (Multi Router Traffic Grapher) software. It can be used for data collection and its visualization by plotting any parameter that can be accessed via SNMP protocol. This data is helpful in detecting security relating events.

For an illustration we present and comment some MRTG data collected from a real system in our Institute. This system is used for running multiple virtual machines. One machine is running specially configured SSH server and is used for gathering data concerning SSH scanning and brute force passwords guessing – kind of specialized HoneyPot. Activity of some hackers is so high, that it influences performance of host machine. Fig. 1 presents MRTG statistics during SSH brute force scanning. Some increased value between 6 and 7 o clock is visible on all plots. The duration of this increase (associated with all three parameters) can be a symptom of some kind brute force password guessing. In contrast to this anomaly other spike values relate to some maintenance activity of the system (for example defragmentation of hard drive) and they appear only foe one or two parameters. Such activity can be seen, for example in the first plot as CPU spike, just before two o clock. Further analysis of logs in SSH HoneyPot system confirmed that at-

tacker checks more than 120 passwords. These anomalies can be detected in automatic way. We develop this in the real-time monitoring and reporting system. Similar analysis concerning some suspicious activity of DNS is presented in [10].

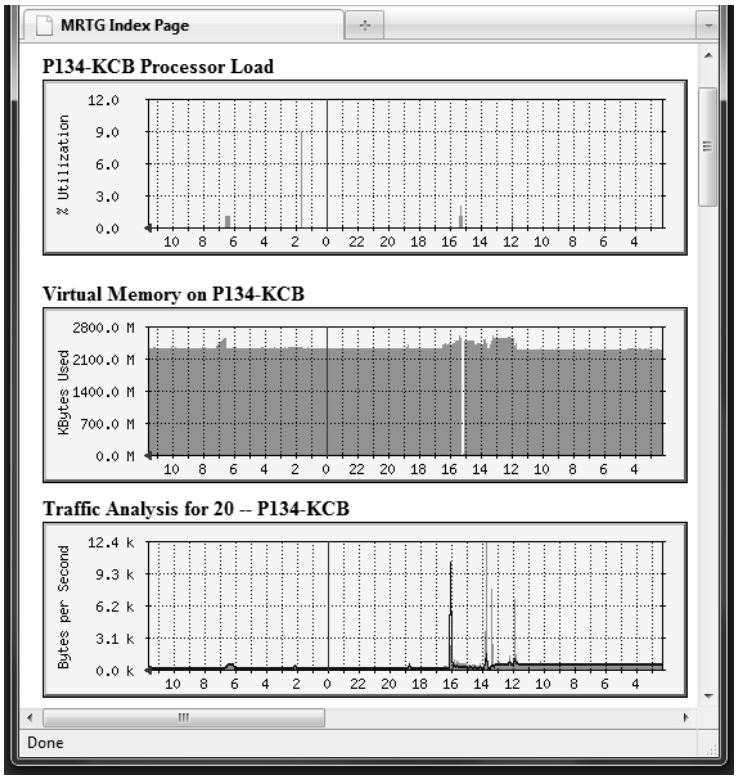

Figure 1. Sample statistics: processor load, used memory, and HoneyPot interface traffic.

The presented analysis based on SNMP and MRTG, can only detect some suspicious activity (in most cases after the actual attack took place). In many cases the attacker activity is discovered after compromising the machine. As an example of event log analysis for security purposes Linux Syslog files can be used. Those logs can be used to determine how the attacker gained the access to this machine.

Syslog is a standard logging protocol in Unix and network devices. Logs generated by applications can be stored locally in the filesystem or sent to a central logging server. Each log has simple header consisting of time and numerical description of this log using two integers – severity and facility. The first number can be used to distinguish level of importance – each log can be marked using some graded labels from critical to debug. The second number (facility) describes the

subsystem that has generated this particular log (e.g. system kernel or a mail service). After the header, the main part of the log comprises some information text.

An example of security related logs is a syslog concerning sshd activity. Two excerpts were logged in the real system connected to the Internet. For security reasons, the real attackers IP address are changed to “aa.bb.cc.dd”. In both cases these logs are related to a common method of attack that is a brute force password guessing. In the first excerpt a connection attempt to ssh using various logins is presented (underlined in the presented excerpts). These login names were not available in the given system. The second excerpt shows an attempt to guess the root password. First three attempts are unsuccessful but the last fourth one (underlined) gives the attacker the full access to this machine.

```
May 28 20:38:04 VM sshd[2910]: Illegal user tomcat from aa.bb.cc.dd
May 28 20:38:05 VM sshd[2912]: Illegal user suporte from aa.bb.cc.dd
May 28 20:38:05 VM sshd[2914]: Illegal user oracle from aa.bb.cc.dd
May 28 20:38:06 VM sshd[2916]: Illegal user test from aa.bb.cc.dd
May 28 20:38:07 VM sshd[2918]: Illegal user admin from aa.bb.cc.dd
May 28 20:38:07 VM sshd[2920]: Illegal user prova from aa.bb.cc.dd
May 28 20:38:08 VM sshd[2922]: Illegal user prueba from aa.bb.cc.dd

May 23 19:01:56 VM sshd(pam_unix)[2348]: authentication failure;
logname= uid=0 euid=0 tty=NODEVssh rhost= aa.bb.cc.dd user=root
May 23 19:02:09 VM sshd(pam_unix)[2354]: authentication failure;
logname= uid=0 euid=0 tty=NODEVssh rhost= aa.bb.cc.dd user=root
May 23 19:02:13 VM sshd(pam_unix)[2356]: authentication failure;
logname= uid=0 euid=0 tty=NODEVssh rhost= aa.bb.cc.dd user=root
May 23 19:02:18 VM sshd(pam_unix)[2358]: session opened for user
root by (uid=0)
```

5. CONCLUSION

The monitoring of various event and performance logs in computer systems is a fundamental source of information on appearing problems or forthcoming threats. Moreover it is useful in load balancing, resource tuning and checking system scalability. The scope and accuracy of monitoring is a challenging issue which can base on long term experience with different systems, collected remarks of system users and administrators, etc. In this process we have also to take into account the assumed goals and system specificity (e.g. workload). In particular it is reasonable to extend the classical approaches targeted at detection of anomalies into forecasting possible problems, finding trends and usage profiles to achieve system resilience.

Further research is planned within log filtering and exploring large sets of logs to identify rules and correlations helpful in the defined monitoring goals.

REFERENCES

[1] Chandola V., Banerjee A., Kumar V. (2009) *Anomaly detection; a survey*, ACM Computing Surveys, vol. 41, No.3, 15:1-15:45.

[2] Cinque M., et al.,(2009) *A logging approach for effective dependability evaluation of computer systems*, 2nd Int. Conf. on Dependability, 105-110.

[3] Intel, *Enterprise versus Desktop Systems*, http://download.intel.com/support/motherboards/server/sb/enterprise_class_versus_desktop_class_hard_drives_.pdf

[4] John L. K., Eeckhout L. (2006) *Performance evaluation and benchmarking*, CRC, Taylors&Francis.

[5] Pinheiro E., Weber W.-D., Barroso L.A., (2007) *Failure Trends in a Large Disk Drive Population*. Proc. of the 5th USENIX Conf. on File and Storage Technologies.

[6] Salfiner F., Lenk M., Malek M. (2010) *A survey of failure prediction methods*, ACM Computing Surveys, vol. 42, no. 3, March 10.1-10.42.

[7] Sankar S., Shaw M., Vaid K. (2011) *Impact of Temperature on Hard Disk Drive Reliability in Large Datacenters*, IEEE/IFIP Int'l Conf. on Dep. Systems & Networks, 530-537.

[8] Sharma A., Kalbarczyk Z., Barlow J., Iyer R., (2011) *Analysis of Security Data from a Large Computing Organization*, IEEE/IFIP Int'l Conf. on Dep. Systems &Networks, 506-517.

[9] Simache C., Kaaniche M. (2005) *Availability assessment of SunOS/Solaris Unix systems based on syslogd and wtmpx log files; a case study*, IEEE PRDC Conf., 49-56.

[10] Smith D., *Health or Preformance monitoring to detect security events*, http://www.dshield.org/diary.html?storyid=11227

[11] Sosnowski J., Król M. (2010) *Dependability evaluation based on system monitoring*. Al-Dahoud Ali [ed.]: Computational Intelligence and Modern Heuristics (Ed.), Intech, 331-348.

[12] Whittington W., Mastro J., *SATA in the Enterprise*, MS PowerPoint presentation, http://download.microsoft.com/download/9/8/f/98f3fe47-dfc3-4e74-92a3-088782200fe7/twst05005_winhec05.ppt

[13] Ye, N. (2008) *Secure Computer and Network Systems*, John Wiley& Sons, Ltd.

[14] Yu L., Zheng Z., Lan Z., Coghlan S., (2011) *Practical on-line failure prediction for Blue Gene/P: period based vs event-driven*, IEEE Int'l Conf. on Dependable Systems & Networks, PFARM workshop, 259-264.

Acknowledgement: This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10.