Communication papers of the Federated Conference on Computer Science and Information Systems, pp. 213–221 DOI: 10.15439/2017F463 ISSN 2300-5963 ACSIS, Vol. 13

Load-balanced Integrated Information Security Monitoring System

Tomasz Klasa
TK Systems Security
Email: tklasa@tksystemsecurity.pl
West Pomeranian Business School,
Szczecin, Poland
Email: tklasa@zpsb.pl

Imed El Fray
Warsaw University of Life Sciences, Faculty of Applied
Informatics and Mathematics, Warsaw, Poland,
West Pomeranian University of Technology,
Faculty of Computer Science, Szczecin, Poland
Email: imed_el_fray@sggw.pl, ielfray@wi.zut.edu.pl

Abstract— Monitoring is the last step of the information security management process. It is intended to evaluate not the state of security itself, but rather the accuracy and quality of prior security evaluation and risk treatment applied. In other words, it is supposed to provide the answer, whether chosen countermeasures and all other decisions based on the security assessment and evaluation results were accurate, proper and sufficient. If during this phase of the security management process, any significant anomaly is found within the system, it means that either one of the accepted 'as is' risks occurred, or that the applied countermeasures did not provide assumed protection in some point of the system. In such a case it is necessary to identify all the areas that require security audit repeat. As information systems grow in complexity, an integrated solution for security monitoring that will prevent system overload caused by monitoring is proposed in this paper.

I. Introduction

ONE of the final results of the information security audit (or risk analysis) is a list of vulnerabilities that were identified, but not covered sufficiently, causing a necessity for further monitoring of a chosen part of the information system. As a consequence, a list of parameters to follow shall be determined and proper control actions taken. This, however, does not mean that applied monitoring solutions eliminated any risks remaining. This action, or rather plan of activities, will not prevent threat from happening. The most important goal for all those actions it to detect any such anomalies and evaluate their influence on the system, as this says, whether the prior risk evaluation was correct. If a severe anomaly is detected by the monitoring mechanisms, it means that some of the countermeasures applied were unsuccessful in preventing it from happening or at least limiting its scale to some acceptable level. As a consequence, parts of the system, that were touched by that detected anomaly, should be reevaluated, e.g. in a form of the audit repeat.

The problem is, while adding precision to security monitoring, one has to accept quickly rising costs of data acquisition and processing. The more frequently the status of specific elements of the system is to be verified or the more parameters of a similar nature are added to allow more complex and multilevel analysis, the bigger quantity of data must be collected, exchanged and processed within the information system. Because company's assets are always limited, the more of them are assigned to the information security monitoring process, the less are left to provide everyday key services of the organization. This may lead even to limiting basic activity of the organization, due to insufficient efficiency of the system as a whole.

To avoid that, it is necessary to determine, for a given set of monitoring parameters, how to control each of them and how frequently to refresh the information about the state of each of the monitored components of the system. In other words, it is necessary to decide which parameters are more important than others as their status has more significant impact of the state of security of the whole information system, not one of its elements alone. To solve the problem, a model of information security monitoring plan adjustment to known technical and organizational limits was proposed.

II. Related work

Research projects related to security monitoring can be divided into two major directions. One of them focuses on anomaly detection, especially increasing its accuracy, in some chosen security domain or part of the system. As an example, there are works on improvement of IDS/IPS [24], [26] or anomaly detection on the basis of network traffic analysis, e.g. [5], [12], [15]. Even if activity of users and individual system components is analyzed, it is done only on the basis of data gathered from networking appliances and the rest of the infrastructure [21]. As can be easily noted, each of them stands for a part of the image only – they provide the answer whether in some chosen part of the system an anomaly occurred, or not. Because of that, each of them can be seen as an improvement of the quality of the source data gathered from individual system components, but not as a verification of the applied set of risk treatment.

A second major direction of research aims at a definition or a model of a secure system. Often this takes a form of a dedicated security framework used to enforce specific scheme of security monitoring, like in [18] or a specific architecture of the whole system as in [8]. A different solution is to use existing classification of threats and vulnerabilities, e.g. for critical infrastructure systems a partition into four groups was proposed, including dedicated risk treatment techniques [13]. One of other attempts is project FORISK, based on a three-step process: once formal requirements and knowledge about the company are collected (phase 1), a formalized risk assessment is undergone (phase 2), and on that basis an automatic or semi-automatic choice of further treatment is performed (phase 3). The whole solution was designed to be easy to apply even for a not fully trained management [7]. Division into steps or layers is a common approach. For instance, in [16] a smart-grid system security was modelled with the help of three layers (social, software and hardware) and a set of allowed interactions between them. Another example of multi-layered approach, relies on game theory and impact assessment based on AHP [10]. The model of a secure system can be created also with the help of a specific descriptive notation or language, like SecBPMN [4], which defines expected behavior of the system and policies that can be used in monitoring. The common feature of security frameworks is that they introduce description of the expected form of the system (either its structure, activity, or both), so it can be easily said if there are any deviations from that state. This leads to quite simple anomaly detection and assumption, that if the system is not compliant with the framework definition, it is not secure. This, however, is still insufficient to verify applied set of countermeasures properly as usually there is no reference to the results of the security audit, which should be a basis for all further decisions and applied risk treatment. Thus, it is not clear whether introduced description of the system (framework) is adjusted to requirements that emerged from the results of the security audit, or if it was proposed regardless of risk analysis.

Modelling expected structure and behavior of system components means that a security framework usually does not apply to all security domains or is adjusted to a specific type of the organization. As a consequence, it is necessary to combine various techniques and solutions to address technical and organizational or immaterial layers of the information system at the same time. This causes further difficulties connected with dependencies between solutions (the same asset, e.g. computer network, may be defined in a different way, depending on chosen framework) resulting in unexpected and unnecessary data redundancy or even inconsistency.

Issues with efficiency caused by quantity of gathered and transmitted data are also typical, however this problem is not unique to information security monitoring. A similar issue was a subject of research e.g. in the area of securing supply chain with numerous dependencies between network elements, where system performance was controlled after implementation of treatment perceived as not influencing performance [20]. Amount of transmitted data was an issue also in the case of research under a smart-grid system, where calculations show that petabytes of data must be gathered to analyze the system of a whole country. When compared the number of data sources in that case with a number of individual data sources in a more sophisticated organization, it can be seen that they are quite comparable. Because the size of data transmission depends mostly on the number of unique messages, and it is impossible to limit the number of data sources without decreasing analysis quality, it is necessary to focus on the sampling frequency [1].

Generally, current solutions either focus on some chosen parts of the information system or attempt to model an enforced, secure state of the system. However, little is done to standardize and process the reverse feedback from the system, to identify weaknesses in applied security plan and improve it appropriately. The monitoring plan is usually used to detect problems and counter them instantly with the help of one of the predefined actions, not to induce long-term improvement of the plan itself, e.g. by the replacement of insufficient countermeasures. Current solutions focus on the best accuracy, it comes at a price of high load of the information system, caused by gathering and processing loads of data intended to maximize the level of details. This approach, however, cannot be implemented in many types of organizations, including virtual organizations, as excessive system load may even block basic operation of such an organization, due to limited resources. At the same time, little attention is paid to solutions that adapt to the requirements and limits of the examined organization in terms of operational costs, that may cover multiple functional areas of the organization at the same time.

III. BACKGROUND

There are two major elements of the whole integrated approach to security monitoring. First of all, the initial evaluation of information system's security is the basis for all further decisions and choice of countermeasures. Because of that, this step shall be done in a standardized way. Based on the analysis of the issues in widely adapted risk assessment methods, a more formalized, however still flexible, approach was proposed [6].

Another very important prerequisite is data description unification. The whole information system consists of multiple elements that, from the perspective of monitoring, can be treated as individual data sources. As they are incompatible with each other and cannot provide output data in a single, chosen arbitrarily, format, it necessary to convert their output into a chosen scheme, as is usually done in numerous research projects aimed at some specific part or type of the system, e.g. [3], [9], [25], [27]. To address that problem, a general solution of data scheme conversion, based on the meaning of individual data fields, was proposed in [14].

IV. INTEGRATED APPROACH TO INFORMATION SECURITY MONITORING

Verification of adequacy of countermeasures taken on the basis of the results of security audit, that were supposed to guarantee some expected security level of the information system is complex. That evaluation is influenced not only by the fact of occurrence (or not) of some local anomalies on the level of individual system components, but most importantly the influence of such events on proper operation of the organization and its ability to proceed required business processes.

The main source of data for parametrization of the security monitoring system is the security audit report, which includes references to identified business processes and utilized assets, defines known threats and specifies for which of them there are uncovered vulnerabilities of used assets. This document explains therefore, which areas or components of the system should be monitored, to assure assumed security objectives. Elements of the system selected during the planning phase are supposed to periodically provide reports describing recent operational history. Among the data sources there are systems like SIEM, firewalls, other network devices, hardware and software sensors responsible for physical and environmental security, software telemetrics and other organizational solutions allowing governance of the implemented procedures.

The list of the data sources presented above shows clearly that they do not form a unified, common group. Also the way of storing and processing data by individual sources is different. Because of that it is reasonable to divide the system into a number of logical layers (ref. fig.1). The first of them is data source layer, gathering individual system components, the state or activity of which is to be monitored. Each element of this layer can be questioned – as a response to such a response it should reply with a status report.

Because of the diversity of data sources, it is necessary to convert messages, or at least convert metadata describing the data. It can be done before the actual communication with the central repository or after the message is received by the repository. Unification of data format before wrapping them into the message structure provides an opportunity for a more detailed control mechanisms and data verification on repository input.

Once the data description format is converted, and if necessary also the data format itself, the data is ready to be wrapped into the structure of the message, which is done at the level of the communication layer.

Received messages go to the data verification layer, which is responsible for the control of data structure, integrity and source. Positively verified data is then sent to the data archive layer, which is responsible for the repository update.

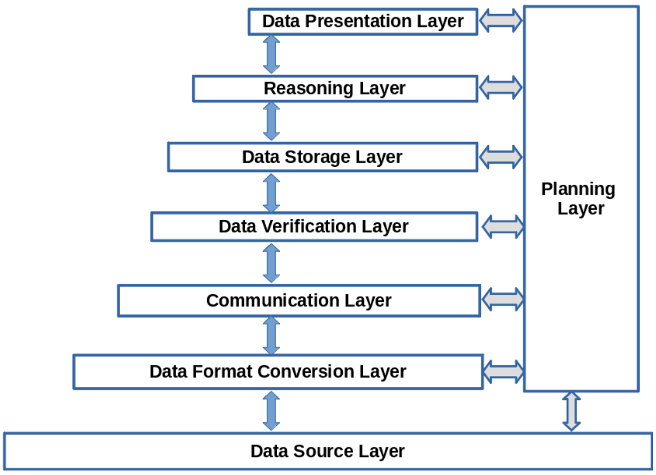

Fig. 1 Multi-layer model of monitoring data integration.

Gathered data are processed and analyzed by the algorithms forming reasoning layer. The result of all operations performed at this stage is an evaluation of effectiveness of implemented countermeasures. the last layer is responsible for mechanisms of results presentation and providing communication between the system and the user.

The only layer which allows communication with any other layer of the system is the planning layer. With its help a functional parametrization is done, as well as technical setup of the components responsible for tasks of the individual layers (e.g. it includes available definitions of data description formats and mappings necessary for their conversion).

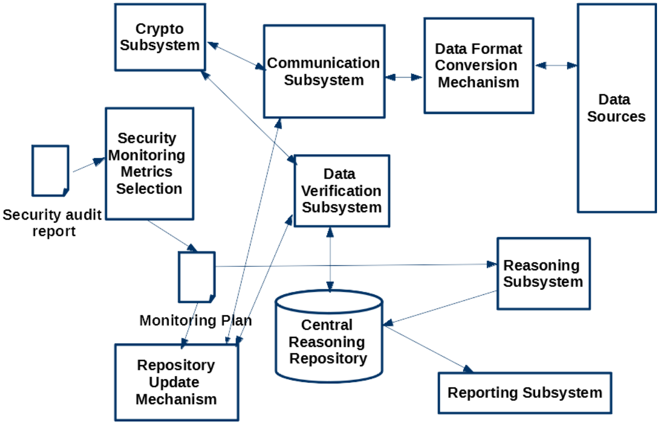
Crypto Subsystem
Communication Subsystem
Data Format Conversion Mechanism
Data Sources
Security Monitoring Metrics Selection
Security audit report
Data Verification Subsystem
Reasoning Subsystem
Monitoring Plan
Central Reasoning Repository
Repository Update Mechanism
Reporting Subsystem

Fig. 2 Security Monitoring Metrics Selection.

This element is a part of the planning layer. Selection of monitoring scope, which means metrics of system components state and means of controlling them, is done on the basis of the security audit results. This provides a Plan of Monitoring, describing which metrics (parameters) should be followed for each of the assets, with the help of which mechanisms and at which frequency their status should be updated at the central repository.

The process consists of two steps. At first, a list of metrics is determined for each of the assets, on the basis of a set of relations described by matrixes, similar to those used by method FoMRA [6]. The results can be shown as an assignment matrix ***RPA***$_{(MxN)}$ of parameters $\boldsymbol{p_i \in P\{p_1,p_2,\ldots,p_m\}}$ to assets $\boldsymbol{a_i \in A\{a_1,a_2,\ldots,a_n\}}$, where ***M*** stands for the number of parameters, and ***N*** number of assets.

$$R_{PA} = \begin{bmatrix} a_1p_1 & a_1p_2 & \cdots & a_1p_m \\ a_2p_1 & a_2p_2 & \cdots & a_2p_m \\ \vdots & \vdots & \ddots & \vdots \\ a_np_1 & a_np_2 & \cdots & a_np_m \end{bmatrix} \quad (1)$$

That scope cannot be modified, if the monitoring process is supposed to cover all assumed security objectives. The only possible way to avoid system overload caused by the monitoring process is to adjust the means of control assigned to each of those parameters in such a way, that the overall cost does not exceed some predefined, acceptable level. Then, for each of those parameters an appropriate way of control is selected, with attention to the overall system load that it will cause. This includes the choice of the right data description format to be used for communication with the data source, which is based on the Triage procedure [23], followed by a comparative analysis of the overhead generated by each of the remaining options (e.g. based on ROI).

Then, the frequency, at which the status of each of the monitored parameters is updated in the central repository is calculated. This is intended to provide as small data granularity as possible, processing and gathering of which will not cause a system load higher than assumed boundary values. Theoretically, the best scenario is to monitor everything in near real-time. Unfortunately, as it was proved by earlier research in various fields, implementation of such an approach is practically near to impossible, e.g. due to the amount of generated network traffic which may interfere with normal, daily operation of the organization. To avoid generating a serious risk of business discontinuity, it is necessary to define a total, acceptable (and safe) system load that can be caused by security monitoring, without significant negative impact on other business processes. Then, the plan of monitoring should be adjusted in such a way, that its implementation does not cause higher load than the one defined as a limit, which may be called a budget B.

A number of weights are used to adapt sampling frequency of various information system elements. All analyzed business processes, assets and parameters are given weights.

Each of the parameters that are subject of monitoring can be connected with only one security objective, but multiple assets and business processes, so they shall not be treated as equal. There are numerous methods for business process assessment, that can be used to evaluate all business processes of the organization, e.g. RAPID RE [17]. Obtained values, after normalization, provide business processes weights $\boldsymbol{WBP_i}$.

All assets that are combined with a given parameter can be treated as equal, so for ***a*** assets connected with parameter ***k*** an asset weight can be calculated as:

$$WA_j = \frac{1}{a},\ WA_j \in \left(0;1\right\rangle \quad (2)$$

As the parameter can be assigned to many assets and business processes, to calculate parameter's weight it is necessary apply proper aggregation. Due to the fact, that relations between business processes, assets and parameters form a tree structure, parameters' weights can be calculated similarly to FTA elements [11], [19], like a total probability of events forming a stochastic tree [2]:

$$WPar_k = \sum_{j=1}^{z}\left(WA_j * \sum_{i=1}^{p} WBP_i\right), WPar_k \in \left\langle 0;1\right\rangle \quad (3)$$

Then, the cost of monitoring plan implementation (which is the system load caused by it) for R parameters and a standardized time period, can be calculated with the help of an adapted formula from process cost calculus [22]:

$$MPC = \sum_{k=1}^{R}\left(Cu_k * A_k * f_k\right) \quad (4)$$

Where:

- *MPC* – monitoring plan cost (additional system load),
- A_k – number of assets connected with parameter *k*,
- Cu_k – load caused by a single update of parameter *k* status (unit cost),
- f_k – parameter k status update frequency.

Then, a following generalization can be assumed:

$$f_k = \mathrm{W}Par_k * f \quad (5)$$

Where:

- $WPar_k$ – weight of parameter *k*,
- *f* – unknown base frequency.

As a result, the following formula is obtained:

$$MPC = \sum_{k=1}^{R}\left(Cu_k * A_k * \mathrm{W}Par_k * f\right) \quad (6)$$

As the maximum acceptable additional load to the system caused by security monitoring was defined as a budget ***B***, the problem got limited to calculation of a maximum base frequency ***f*** for which equation $MPC \leq B$ holds. That boundary can be defined for a number of criteria, the list of which can differ depending on organizational structure and culture. They may include IT infrastructure load and workload of employees.

A set of four criteria was chosen during research:

- calculation power consumed,
- memory consumed,
- quantity of data transferred,
- human workload.

For each of those criteria a separate budget equation should be formed:

$$B_a = \sum_{k=1}^{R} \left(Cu_k * A_k * \mathrm{W}Par_k * f \right) \tag{7}$$

Base frequency ***f*** must fulfill all the criteria of system workload, which means:

$$f = \min \left\{ f_1, \cdots, f_{c-1,} f_c \right\} \tag{8}$$

Calculated frequency ***f*** is used to determine the frequency of each parameter's update, with attention to its weight ***WPar***, calculated earlier:

$$f_k = \mathrm{W}Par_k * f \tag{9}$$

As a consequence, the final monitoring plan ***PM***, providing monitoring technique ***MT*** and status update frequency ***f*** for each of the parameters, takes a form of:

$$PM = \left\{ \left(Par_i, MT_{(Par_i)}, f_{(Par_i)} \right), \cdots \right\} \tag{10}$$

As it was stated at the beginning, a list of parameters was taken as an input and is a result of earlier steps of the security management process. Monitoring techniques were chosen with attention to various requirements and measurable criteria. Finally, update frequency for each of the parameters was established with attention to parameters' significance and within predefined load limit (budget). Changing a set of load criteria will influence the final result in such a way that the update frequencies may be different, but the overall load of the system will always be below predefined limit. Although that limit must be determined separately for every single organization, it is a far simpler task when compared to assessment of all possible monitoring plan combinations.

A. Data Sources

There are many different data sources that may provide data to the system, including SIEM solutions, firewalls and other networking devices, hardware and software sensors for physical and environmental security, software telemetrics mechanisms or other organizational solutions of applied procedures governance.

B. Central Reasoning Repository

It is the main element of the data gathering layer. It collects all status data from individual components of the system, as well as necessary system parametrization data. Depending on the scale of the information system that is subject of monitoring, it may be necessary to implement database architecture designed for the big data environment.

C. Repository Update Mechanism

Another key element of the data gathering layer. It is responsible for periodical overwrite of the oldest records with current, successfully verified valued received from data sources. It is based on the cyclic queue mechanism and makes a use of a typical logistic approach to managing performer operation.

D. Data Verification Subsystem

It is responsible for initial verification of integrity, source and structure of data. Verification of data source and integrity is done with the help of cryptography, implemented in the Crypto module. Verification of data structure is based on the data description format definition that is expected in the message from the agent that is responsible for processing this specific order on data.

E. Data Format Conversion Mechanism

The conversion model relies on assumption that there is some general set of description format fields, from which specific languages are constructed by selection of proper subsets of fields. Such assumption is correct as long as each field that exists in one description format forms a unique element of a set. This means that for instance in the case of field date that appears in n languages to meet that assumption a set of fields LF should contain up to n unique elements that corresponds to date. The mechanism was described in [14].

F. Reasoning Subsystem

To reduce time required to build or adjust the model and to make it easier, classification of local anomalies based on rough sets and linguistic knowledge base was used. It was combined with aggregation on the level of assets, business processes and security objectives.

A number of factors influencing changes in the state of security were chosen based on the widely used methods and standards (ISO2700x and ISO15408-x). They were used to define a model of linguistic base decomposition defined as WL = f(x1, x2, x3, x4, x5, x6, x7), where:

x1 – business processes significance,
x2 – number of corresponding business processes,
x3 – asset recovery time,
x4 – dynamics of changes of asset's parameters state,
x5 – scale of changes of assets parameters,
x6 – available time,
x7 – environment influence.

Then, an aggregation based on a tree structure was done, where level 0 is a tree root that represents security of the whole information system, and nodes on level n described by the *WL* value. The aggregation characteristics are as follows:

- At least one of the child nodes on level ***n+1*** has value equal to WL,
- None of the child nodes on level ***n+1*** has value higher than WL,
- All child nodes on level ***n+1***except of at least one have value lower than WL,
- Value on the level ***n-1*** must not be lower than WL.

Finally, starting with the deepest level of the tree, nodes with the highest value of influence WL are identified and that value is propagated up the tree, carrying the information about the status of the system according to approach “worst case first”. Proposed aggregation model is in this way a combination of analytic side of the FTA and synthetic side of the ETA. In some way it can be perceived as a tree representation of the FMEA model. Because the attention of the administrator or the management is focused on those anomalies that caused the biggest abbreviation from the state perceived as safe, it is easy to prioritize anomaly handling and in this way correct detected weaknesses of the implemented security management system.

V. Case study

Proposed new approach was verified on the basis of data gathered during real life operation of an IT knowledge contest Tik?-Tak! in two different time periods. Tik?-Tak! was a national (Polish) contest organized by the Polish Information Processing Society, addressing primary, secondary and high school students. First set of data comes from its second edition in year 2012, and second set of data was gathered in year 2015, after the contest system was significantly reconstructed. The contest was divided into three phases: school level, regional level and finals. The first two of them were done online, with the help of mentioned contest system. Data used in case study were gathered during the first phase of the contest, in which over ten thousand students attended each of the editions. The contest system was located on a virtual machine provided by Cloudia (in 2015 rebranded to Atende Business Cloud). Users used their own computers to log in to the system.

The following case study presents key steps through the proposed integrated monitoring system. An entry point was a list parameters that came as a result of risk assessment process:

1) Contest system server,
2) Users database,
3) LAN network,
4) Internet connection,
5) Questions database,
6) Answers database,
7) Administrator.

Those assets are used in three main business processes:

- Solving tests,
- Calculating rank lists,
- User and school registration.

Five of the assets used were assessed as secure enough and because of that were not assigned to any monitoring parameter. For the remaining two there were known vulnerabilities that were not covered fully in other way, which caused them to require monitoring. In total, three parameters were defined in this case:

- Answer saved (t),
- Par 65 – Unsuccessful use of the user identification mechanism, including the user identity provided,
- Par 66 – Successful use of the user identification mechanism, including the user identity provided.

Those three parameters, mapped on assets, were taken as input to case study in this paper. The R_{PA} matrix looks like:

$$R_{PA} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{11}$$

For each of the parameters, a unit cost of monitoring was estimated, separately for each of the four conditions: caused system load in seconds, human workload in hours, data transmission in kB and quantity of data stored in kB.

Table I.
Budget definition - load allowance for each of the test scenarios for Tik?-Tak! 2012. Source: own contribution.

test no.	data saved increase	system load	human workload	data transmitted
test 1	1GB	10000s	40h	5GB
test 2	2GB	1000s	40h	5GB
test 3	1GB	10000s	5h	5GB
test 4	1GB	1000s	20h	5GB
test 5	1GB	2000s	8h	5GB

Frequencies of parameter status update were calculated for each of the test scenarios defined by budget limits shown in table1. The results are shown in Tab II.

Table II.
Test results for Tik?-Tak! 2012. Source: own contribution.

test no.	unsuccessful user identification	successful user identification	answer saved (t_1-t_0)
test 1	103	103	96
test 2	30	30	28
test 3	12	12	12
test 4	30	30	28
test 5	20	20	19

Table IIII.
Budget limit consumption in test scenarios.
Own contribution.

test no.	data saved increase	system load	human workload	data transmitted
test 1	0,010 MB	3348,149 s	39,999 h	0,013 MB
test 2	0,003 MB	999,999 s	11,946 h	0,004 MB
test 3	0,001 MB	418,519 s	4,999 h	0,002 MB
test 4	0,003 MB	999,999 s	11,946 h	0,004 MB
test 5	0,002 MB	669,630 s	8,000 h	0,003 MB

Parameter status update frequency is limited by the most restrictive cost criteria. As can be seen in table 3, in the first, third and fifth test scenario a maximum allowed human workload was reached. In the second and fourth, the limit was caused by the system load allowance.

For each of the parameters, a unit cost of monitoring was estimated, separately for each of the four conditions: caused system load in seconds, human workload in hours, data transmission in kB and quantity of data stored in kB. Then, frequencies of their status update were determined, according to formula 7 and 9, for a number of test scenarios. To provide comparison with results obtained for the 2012 contest system, at first the same test cases were defined (based on the same budget limits).

TABLE IV.
BUDGET DEFINITION - LOAD ALLOWANCE FOR EACH OF THE TEST SCENARIOS FOR TIK?-TAK! 2015. SOURCE: OWN CONTRIBUTION.

test no.	data saved increase	system load	human workload	data transmitted
test 1	1GB	10000s	40h	5GB
test 2	2GB	1000s	40h	5GB
test 3	1GB	10000s	5h	5GB
test 4	1GB	1000s	20h	5GB
test 5	1GB	2000s	8h	5GB

Frequency of parameter status update was calculated again for each of the test scenarios defined by budget limits shown in table 4, while table 5 contains results obtained for the Tik?-Tak! 2015 contest, with the following set of pareameters:

Par1: unsuccessful user identification
Par2: successful user identification
Par3: SSH attack
Par4: Fail2Ban bans
Par5: PHP errors nginx
Par6: PHP sock-fail
Par7: answer saved (t_1-t_0)

TABLE V.
TEST RESULTS FOR TIK?-TAK! 2015. SOURCE: OWN CONTRIBUTION.

test no.	Par1	Par2	Par3	Par4	Par5	Par6	Par7
test 1	33	33	33	33	33	33	33
test 2	7	7	7	7	7	7	7
test 3	4	4	4	4	4	4	4
test 4	7	7	7	7	7	7	7
test 5	6	6	6	6	6	6	6

Similarly to tests for Tik?-Tak! 2012 contest, the first, third and fifth test scenario generated

a maximum allowed human workload value (refer to table 6). The second and fourth were limited by system load criterion. It can be seen as the increase of the parameters quantity, at an unchanged limit of allowed load, caused by monitoring implementation, reduces frequencies of parameters' status update.

TABLE VI.
BUDGET LIMIT CONSUMPTION IN TEST SCENARIOS. OWN CONTRIBUTION.

test no.	data saved increase	system load	human workload	data transmitted
test 1	0,008 MB	4333,334 s	40,000 h	0,014 MB
test 2	0,002 MB	999,999 s	9,228 h	0,003 MB
test 3	0,001 MB	541,666 s	4,999 h	0,002 MB
test 4	0,002 MB	999,999 s	9,228 h	0,003 MB
test 5	0,002 MB	866,666 s	7,999 h	0,003 MB

Two additional test scenarios were added with increased budget limits. The idea was to show the difference in load of the given four criteria, generated by the increased scope of monitoring (when compared to Tik?-Tak! 2012), with the frequencies on comparable level. Table 7 presents budget limits for the additional test scenarios.

TABLE VII.
BUDGET DEFINITION - LOAD ALLOWANCE FOR ADDITIONAL TEST SCENARIOS FOR TIK?-TAK! 2015. OWN CONTRIBUTION.

test no.	data saved increase	system load	human workload	data transmitted
test 6	1GB	10000 s	200 h	5GB
test 7	1GB	20000 s	200 h	5GB

Once again, frequencies were calculated. Results obtained for the new test scenarios were shown in table 8.

TABLE VIII.
RESULTS FOR ADDITIONAL TEST SCENARIOS FOR TIK?-TAK! 2015. OWN CONTRIBUTION.

test no.	Par1	Par2	Par3	Par4	Par5	Par6	Par7
test 6	76	76	76	76	76	76	76
test 7	153	153	153	153	153	153	153

TABLE IX.
BUDGET LIMIT CONSUMPTION IN TEST SCENARIOS 6 AND 7. OWN CONTRIBUTION.

test no.	data saved increase	system load	human workload	data transmitted
test 6	0,019 MB	10000 s	92,309 h	0,031 MB
test 7	0,039 MB	20000 s	184,615 h	0,062 MB

In both cases it was the system load that limited the frequency of parameters' status update. This can be clearly seen in table 9.

As can be seen, the monitoring plan is adjusted according to changing load allowance (limit) of the four budget criteria. The final parameters' status update frequencies are calculated in such a way, that it is impossible to exceed a predefined limit, which prevents from information system overload caused by the implementation of security monitoring procedures.

VI. Discussion

It should be noted that data does not necessarily have to be updated in the form of readings from specific points in time. The described mechanism can be applied as well if the data source returns periodically a summary of status changes in time or some aggregated value of status readings from some time period. As a result, this mechanism is connected at least with some latency in status update or even change of data granularity.

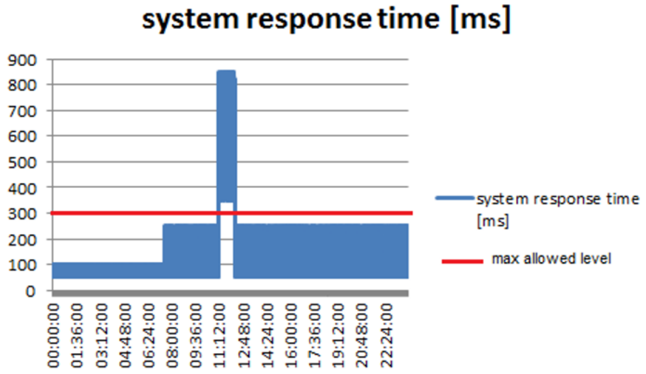

Fig. 3 Value in time of one of the Tik?-Tak! parameters gathered at *f*=1s. Own contribution.

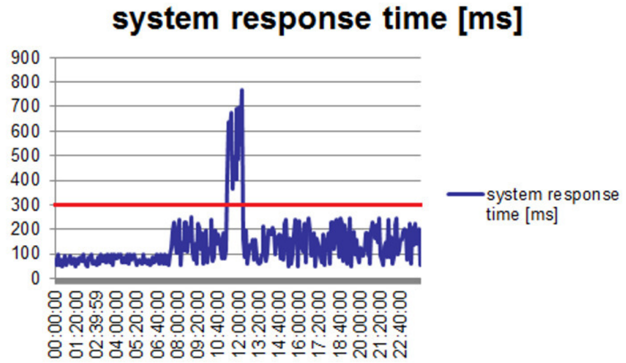

Fig. 4 Values in time of one of Tik?-Tak! parameters gathered with frequency *f*=5min. Own contribution.

In this paper it was assumed that budget criteria values were evaluated according to some rules or methodology, which is irrelevant from the perspective of the proposed mechanism. However, those limits may have impact on the accuracy of further reasoning on the basis of gathered data (due to data granularity and latency in their receive). The method for calculation of those budget limits is not a part of this paper, however. To show, that the mechanism proposed in this paper, if it is not used with too strict budget limits, shall not reduce detectability of security incidents, data for one of the parameters chosen for Tik?-Tak! contest was compared in two scenarios. Figure 5 shows values gathered with a maximum possible frequency (*f*=1s) within a 24h time period. As the system designer defined the maximum comfortable system latency as 300ms, that value was defined as a limit – all incidents when this is exceeded should be treated as a significant system slow down. In analyzed time period such an anomaly occurred around 11:20, and lasted for about 15 minutes.

Figure 6 presents values of the same parameter, but updated/gathered during the same time period at a lower frequency *f*=5min. as can be seen, although figures differ, the overall characteristics remained, so it is still possible to detect this anomaly, however with some latency. The reduction of reasoning precision (latency) caused by data granularity was paired with significant reduction of monitoring costs. As a result, application of a mechanism proposed in this paper should always be combined with proper evaluation of allowed load limits. Standardization of that evaluation is the most significant field of future work. The proposed solution, however, takes budget limits as input arguments, without considering how they were calculated.

VII. Conclusion

The proposed approach to security monitoring, is a complex solution for all steps of the process, starting with planning and finishing with reasoning about the state of security of the whole system. Thanks to many adaptation mechanisms it can be adjusted to requirements and implemented in different organizations. What is important, the selection of monitoring plan is done in a way that is supposed to prevent system overload caused by the monitoring activity itself. Construction of the reasoning process, including knowledge aggregation, was prepared in such a way that it provides a clear answer which areas or elements of the system were affected by significant security incidents, which means that in those areas already implemented countermeasures are probably insufficient. Then, on the basis of such an information, it becomes easy to plan an audit repeat, if required.

References

[1] Aiello, M. and Pagani, G.A. 2014. The Smart Grid's Data Generating Potentials. Annals of Computer Science and Information Systems. 2014, Vol. 2, pp. 9-16., DOI: 10.15439/2014F509

[2] Bean, Michael, A. Probability: The Science of Uncertainty with Applications to Investments, Insurance, and Engineering. Providence : American Mathematical Society, 2001.

[3] Brdys, M.A. 2014. Integrated monitoring, control and security of Critical Infrastructure Systems. Annual Reviews in Control. 2014, 38, pp. 47-70., DOI: 10.1016/j.arcontrol.2014.03.006

[4] Dalpiaz, F., Giorgini, P., Salnitri, M., Designing secure business processes with SecBPMN. Enterprise, Business-Process and Information Systems Modeling, pp.200-214, 2015, DOI: 10.1007/s10270-015-0499-4

[5] Deraison R., Gula R., Ranum M., Unified Security Monitoring (USM), Real-Time Situational Awareness of Network Vulnerabilities, Events and Configurations, Tenable Network Security, 2009

[6] El Fray, I. A Comparative Study of Risk Assessment Methods, MEHARI & CRAMM with a New Formal Model of Risk Assessment (FoMRA). Information Systems. Computer Information Systems and Industrial Management. 2012, Vol. 7564 of the series Lecture Notes in Computer Science., DOI: 10.1007/978-3-642-33260-9_37

[7] Fenz, S., et al. 2013. FORISK: Formalising information security risk and compliance management. 2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W). 2013., DOI: 10.1109/DSNW.2013.6615533

[8] Fernandez E., Monge R., Building a security reference architecture for cloud systems, Requirements Eng (2016) 21:225–249, DOI: 10.1007/s00766-014-0218-7

[9] Han, S., et al. 2014. Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges. IEEE SYSTEMS JOURNAL. 2014, Vol. 8, 4., DOI: 10.1109/JSYST.2013.2257594

[10] Huang S., Zhou C., Yang S., Quin Y., Cyber-physical System Security for Networked Industrial Processes, International Journal of Automation and Computing, 12(6), December 2015, 567-578, DOI: 10.1007/s11633-015-0923-9

[11] IEC. Fault Tree Analysis (FTA), International Technical Commission, IEC Standard, Publication 1025. 1990.

[12] Jung H., Hwang I., Moon J., Park H., A security monitoring method for malicious P2P event detection, Peer-to-Peer Netw. Appl. (2016) 9:498–507, DOI: 10.1007/s12083-015-0369-4

[13] Karabacak, B. and Tatar, U. Strategies to Counter Cyberattacks: Cyberthreats and Critical Infrastructure Protection. NATO Science for Peace and Security Series - E: Human and Societal Dynamics. 2012, Vol. Volume 116: Critical Infrastructure Protection.,

[14] Klasa, T. and El Fray, I. Data Scheme Conversion Proposal for Information Security Monitoring Systems. [book auth.] S. Kobayashi, et al., et al. Hard and Soft Computing for Artificial Intelligence, Multimedia and Security. s.l. : Springer International Publishing, 2017., DOI: 10.1007/978-3-319-48429-7_15

[15] Kondakci, S., A causal model for information security risk assessment, 2010 Sixth International Conference on Information Assurance and Security, Atlanta, GA, 2010, pp. 143-148., DOI: 10.1109/ISIAS.2010.5604039

[16] Li, T., Horkoff, J. & Mylopoulos, J. Softw Syst Model (2016). doi:10.1007/s10270-016-0560-y

[17] Manganelli, R. L., Klein, M. M. The reengineering handbook: a step-by-step guide to business transformation. New York: AMACOM, 1994.,

[18] Martinelli, F. and Matteucci, I. A framework for automatic generation of security controller. Software Testing Verification & Reliability. 2008, pp. 563-582., DOI: 10.1002/stvr.441

[19] NASA. Fault Tree Handbook with Aerospace Applications', Version 1.1, NASA Publication. 2002.

[20] Pero, M. and Sudy, I. 2014. Increasing security and efficiency in supply chains: a five-step approach. International Journal of Shipping and Transport Logistics. 2014, Vol. 6, 3, pp. 257-279., DOI: 10.1504/IJSTL.2014.060785

[21] Shashanka M., Shen M., Wang J., User and Entity Behavior Analytics for Enterprise Security. BigData, 2016. DOI: 10.1109/BigData.2016.7840805

[22] Shim, J. K., Siegel J.G., Modern Cost Management and Analysis, Barron's Business Library, New York, 2009

[23] Stoppler, Melissa, Conrad, MD. 2014. MedicineNet.com. [Online] 12 1, 2014. http://www.medicinenet.com/script/main/art.asp?articlekey=79529.

[24] Thakore U., Weaver G., Sanders W., A Quantitative Methodology for Security Monitor Deployment, 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, 2016, DOI: 10.13140/RG.2.2.17533.56808

[25] Van Tan, V. and Yi, M.J. 2010. Design Issues and Approach to Internet-Based Monitoring and Control Systems. Trends in Applied Intelligent Systems, Pt1, Proceedings, Lecture Notes in Artificial Intelligence. 2010, Vol. 6096, pp. 478-488., DOI: 10.1007/978-3-642-13022-9_48

[26] Vaarandi, R., Real-time Classification of IDS Alerts with Data Mining Techniques, Proceedings of the 2009 IEEE MILCOM Conference, IEEE 2009

[27] Wu, M.Z., et al. 2012. Development and Validation on Integrated Dynamic Security Monitoring Platform. 2012 Sixth International Conference on Genetic and Evolutionary Computing. 2012., DOI: 10.1109/ICGEC.2012.80