International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 07
March 2018

Systematically recognizing potential process-related threats in SCBDA

G.V.S.S.P.Raju & M. Narasimha Raju

[1]Assistant Professor, Department of CSE, Shri Vishnu Engineering College for Women (A),
Vishnupur, West Godavari District, Bhimavaram, Andhra Pradesh-534202
[2]Assistant Professor, Department of CSE, Shri Vishnu Engineering College for Women (A),
Vishnupur, West Godavari District, Bhimavaram, Andhra Pradesh-534202

Abstract: *SCBDA (Semi-trust control Based data acquisition) systems are used for controlling and monitoring industrial processes. We propose a methodology to systematically identify potential process-related threats in SCBDA. Process-related threats take place when an attacker gains user access rights and performs actions, which look legitimate, but which are intended to disrupt the SCBDA process. To detect such threats, we propose a semi-automated approach of log processing. We conduct experiments on a real-life water treatment facility. A preliminary case study suggests that our approach is effective in detecting anomalous events that might alter the regular process workflow.*

Keywords:- ICS · SCBDA · Security · SCBDA log · Log analysis · Frequent pattern mining · Process related threat · HAZOP · PHEA · MELISSA

INTRODUCTION

SCBDA systems can be found in critical infrastructures such as power plants and power grid systems, water, oil and gas distribution systems, building monitoring (e.g., airports, railway stations), production systems for food, cars, ships and other products. Although failures in the security or safety of critical infrastructures could impact people and produce damages to industrial facilities, recent reports state that current critical infrastructures are not sufficiently protected against cyber threats. For example, according to Rental, around 2,700 organizations dealing with critical infrastructures in the U.S. detected 13 million cybercrime incidents, suffered $288 million of monetary loss and experienced around 150,000 h of system downtime in 2005.

Also, in a security study of 291 utility and energy companies in the U.S., 67% of the companies report that they are not using state of the art security technologies. Besides, 76% of the companies report that they suffered one or more data breaches during the past 12 months. The increasing number of security incidents in SCBDA facilities is mainly due to the combination of technological and organizational weaknesses. In the past, SCBDA facilities were separated from public networks, used proprietary software architectures and communication protocols. Built on the “security by obscurity” paradigm, the systems were less vulnerable to cyber attacks. Although keeping a segment of communication proprietary, SCBDA vendors nowadays increasingly use common communication protocols and commercial off-the-shelf software. Also, it is common to deploy remote connection mechanisms to ease the management during off-duty hours and achieve nearly unmanned operation. Unfortunately, the stakeholders seldom enforce strong security policies. User credentials are often shared among users to ease day-to-day operations and are seldom updated, resulting in a lack of accountability. An example of such practice is the incident in Australia when a disgruntled (former) employee used valid credentials to cause a havoc.

Due to these reasons, SCBDA facilities became more vulnerable to internal and external cyber attacks. Although companies reluctantly disclose incidents, there are several published cases where safety and security of SCBDA were seriously endangered.

Like a “regular” computer system, a SCBDA system is susceptible to threats exploiting software vulnerabilities (e.g., protocol implementation, OS vulnerabilities). However, a SCBDA system is also prone to process-related threats. These threats take place when an attacker uses valid credentials and performs legitimate actions, which can disrupt the industrial process(es). Process-related threats also include situations when system users make an operational mistake, for example, when a user inputs a wrong value (e.g., a highly oversized value) for a given device parameter and causes the failure of the process. In general, process-related threat scenarios do not include any exploit of a software implementation vulnerability (e.g., protocol implementation).

Sometimes system- and process-related threats can be part of the same attack scenario. For example, an attacker can first subvert the access control mechanism to gain control over an engineering work station. This action would use a system-related threat (e.g., exploiting an OS vulnerability). Then, an attacker could use a valid SCBDA control application to perform undesirable actions for the process (e.g., overload pipe system). This part of the attack is performed as a process-related threat scenario. Traditional security countermeasures, such as intrusion detection systems, cannot detect, let alone mitigate, process-related threats. This is because typical intrusion detection systems look for patterns of the behavior known to be malicious (e.g., known payload transfers, TCP header format) or look for anomalies in terms of statistical distributions (e.g., by statistically modeling the content of data packets). The anomalies generated by process-related threats are typically not reflected in communication patterns/data (e.g., injection of executable code to exploit a buffer overflow sent within network traffic data) and can only be detected by analyzing data passed through the system at a higher semantic level.

To understand the higher semantic level from network data, a protocol parser has to be used, such as in Bro. Similarly, for host-based analyses, understanding the specific SCBDA application is crucial. Other approaches for monitoring SCBDA behavior include the usage of field measurements or centralized SCBDA events as information resources. Field measurements represent raw values coming from field devices. Aggregated field measurements can provide information about the current status of the process. However, we argue that the field measurement values are too low level to extract user actions and to evaluate the semantics of the performed actions. SCBDA event logs provide a complete high-level view on the industrial process that is continuous over time and captures information about user activities, system changes in the field as well as system status updates.

Problem Even a SCBDA system used in a small installation generates thousands of potentially alarming log entries per day. Thus, the size (and high dimensionality) of logs make manual inspection practically infeasible. This is a relevant and challenging problem to tackle. It is relevant because process-related threats affect the security and safety of critical infrastructures, which in turn could endanger human life. It is challenging because in the past the analysis of system logs has been applied to other security domains, but failed to deliver convincing results. We propose a semi-automated approach of log processing for the detection of undesirable events that relate to user actions. We acknowledge that the success of a log mining approach depends on the context in which it is applied. Therefore, we perform an extensive analysis of the problem context. In Fig. 1, we show the main steps of the our approach. We group the steps by two means of obtaining context information: (1) system analysis and (2) analysis by a focus group. The system analysis implies the inspection of available documentation and processing of logs. The focus group analysis implies sessions with the stakeholders where we obtain deeper insights about the SCBDA process. Focus groups consist of process engineers who are aware of the semantic implications of specific actions, but typically cannot provide useful information for automatic extraction of log entries. This is due to the fact that engineers do not perform extensive analysis of system outputs and are not experts in data mining. On the other hand, by performing the system analysis, we cannot infer semantic information that is implied in log entries, thus the stakeholders' knowledge is invaluable.

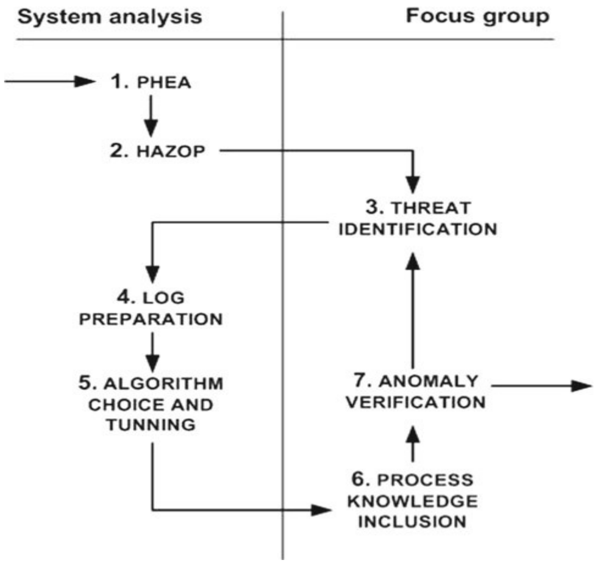


Fig. 1 Steps for mining SCBDA logs

A sequence of actions in Fig. 1 represents the chronological order of the steps that we perform. In steps 1 and 2, we systematically identify process deviations caused by user activity. For this, we adapt two methodologies from the domain of hazard identification (PHEA and HAZOP). We then use the stakeholders knowledge to identify which of the analyzed deviations represent a legitimate threat to the process (step 3). In steps 4 and 5, we perform log transformation and generalization to extract a subset of log attributes suitable for log mining. Also, we discuss the requirements of the mining algorithm that is useful for our context. In steps 6 and 7, we include the stakeholders in the mining process. This implies leveraging the stakeholders' knowledge about the process to improve the semantics of the mined events. The stakeholders analyze the output of the mining process and verify anomalies. Finally, the anomalies are checked for the consistency with the threats identified in step 3. Also, this step is used to revise the list of potential threats and perhaps introduce new threats. To support the proposed analysis, we build a tool that can perform the log processing in search for process-related threats. Our tool leverages a well-known data mining algorithm to enumerate (in) frequent patterns within a given set. Despite being quite simple and straightforward, our benchmarks show that the chosen algorithm is effective in detecting previously overlooked behavioral anomalies.

RELATED WORK

Traditional methodologies for addressing safety problems in process control systems (e.g., FMEA, FTA, HAZOP) do not consider security threats. By introducing a special set of guidewords, Winter et al. show how HAZOP can be extended to identify security threats. Srivantakul et al. combine HAZOP study with UML use case diagrams to identify potential misuse scenarios in computer systems. We take a similar approach to combine PHEA study with HAZOP and analyze user (engineer) behavior in a SCBDA environment.

To detect anomalous behavior in SCBDA systems, authors use approaches based on inspecting network traffic, validating protocol specifications and analyzing data readings. Process-related attacks typically cannot be detected by observing network traffic or protocol specifications in the system. We argue that to detect such attacks one needs to analyze data passing through the system and include a semantic understanding of user actions.

Use periodical snapshots of power load readings in a power grid system to detect if a specific load snapshot significantly varies from expected proportions. This approach is efficient because it reflects the situation in the process in a case of an attack. However, data readings (such as power loads) give a low-level view on the process and do not provide user traceability data.

Authors discuss the difficulties in processing logs with unstructured format. Authors present an approach for failure prediction in an enterprise telephony system. Authors propose to use context knowledge for efficient process visualization and failure prediction. Several researches explore pattern mining of various logs for security purposes. These authors use pattern mining on burst of alarms to build episode rules. However, pattern mining can sometimes produce irrelevant and redundant patterns. We use pattern mining algorithms to extract the most and the least frequent event patterns from SCBDA log.

Authors propose to combine various log resources in a process control environment to detect intrusions. The detection is operator-assisted. To the best of our knowledge, only Bald celli et al. analyze SCBDA logs to detect unusual behavior. There, the authors use case-base reasoning to find sequences of events that do not match sequences of normal behavior (from the database of known cases). The authors analyze sequences of log events that originate from a simulated test bed environment. In contrast, we analyze individual logs from a real SCBDA facility.

SYSTEM ARCHITECTURE

Despite the fact that there are different vendors, the system architectures in various SCBDA systems are similar and the terminology is interchangeable. Figure 3 shows a typical SCBDA layered architecture.

Layer 1 consists of physical field devices, PLCs (programmable logic controllers) and RTUs (remote terminal units). The PLCs and RTUs are responsible for controlling the industrial process, receiving signals from the field devices and sending notifications to upper layers.

Layer 2 consists of SCBDA servers responsible for processing data from Layer 1 and presenting process changes to Layer 3. Connectivity Servers aggregate events received by

PLCs and RTUs and forward them to SCBDA users in the control room.

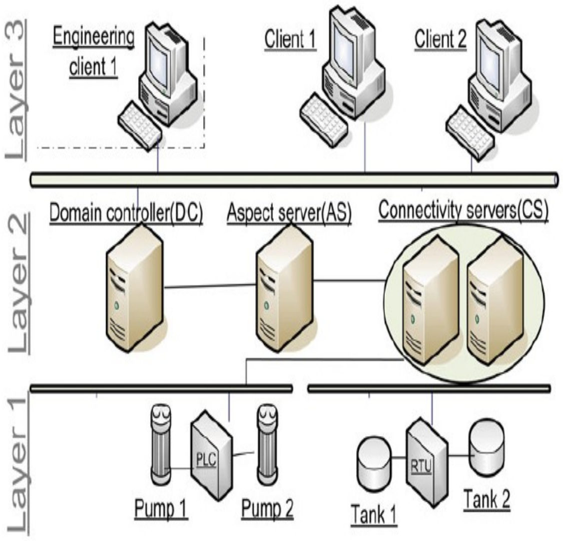


Fig. 2 SCBDA system layered architecture

The Domain Controller in Layer 2 holds local DNS and authentication data for user access. The Aspect Server is responsible for implementing the logic required to automate the industrial process. For example, an Aspect Directory in the Aspect Server holds information about working ranges of the field devices, the device topology, user access rights, etc. Besides, the Aspect Server collects and stores data from the Connectivity Servers into audit and event logs. The various clients in Layer 3 represent SCBDA users.

INPUT DATA FOR ANALYSIS

Input data for analysis The initial, raw, data set consists of 11 attributes. The given attributes can be grouped in four semantic groups:

– time (Timestamp),

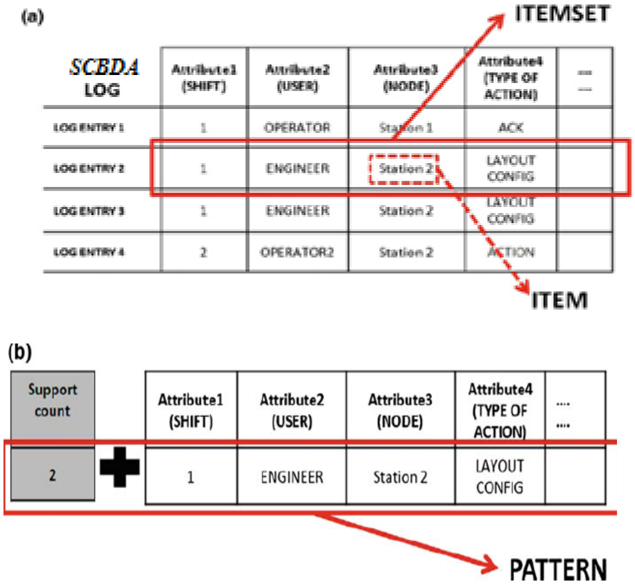
– type of action (Type of action, Aspect of action),

– action details (Message description, Start value, End value),

– user (Username, User full name),

– location (Object path, Source, SCBDA node)

Often the raw data set consists of features that are redundant, irrelevant or can even misguide mining results. This is why we need to perform data preprocessing, analyze the current feature set and select a suitable subset of attributes.



SCBDA LOG	Attribute1 (SHIFT)	Attribute2 (USER)	Attribute3 (NODE)	Attribute4 (TYPE OF ACTION)	...
LOG ENTRY 1	1	OPERATOR	Station 1	ACK	
LOG ENTRY 2	1	ENGINEER	Station 2	LAYOUT CONFIG	
LOG ENTRY 3	1	ENGINEER	Station 2	LAYOUT CONFIG	
LOG ENTRY 4	2	OPERATOR2	Station 2	ACTION	

Support count		Attribute1 (SHIFT)	Attribute2 (USER)	Attribute3 (NODE)	Attribute4 (TYPE OF ACTION)	....
2	✚	1	ENGINEER	Station 2	LAYOUT CONFIG	

Fig. 3 Log translation: a mapping log entries into item sets and items, b mapping item sets into patterns

1. Attribute subset selection :-

Common approaches for attribute selection exploit class labels to estimate information gain of specific attribute (e.g., decision tree induction). Unfortunately, our data set does not consist of class labels (i.e., labels for normal and undesirable behavior), thus we cannot perform the “traditional” attribute evaluation. However, some approaches may evaluate attributes independently. For example, principal component analysis (PCA) searches for k n-dimensional orthogonal vectors that can be used to represent the data. The original data is thus projected into a much smaller space and represented through principal components. The principal components are sorted in the order of decreasing “significance”. Finally, the dimensionality reduction is performed by discarding weaker components, thus those with low variance. By performing the PCA on our data, we discard two low variable attributes (Start Value, End Value) since they only had one value in the whole data set. Also, we identify two redundant attributes (Username and User full name). Thus, we discard one of them As expected, the attribute Timestamp showed the highest variance.

We aggregate this attribute in three working shifts. We describe the details of this aggregation. Now, we try to understand the behavior of the remaining attributes. Due to the fact that the highly variable attributes can produce over fitting, we try to lower the number of distinct values in the most variable attributes (in our context, the ones over 150 distinct values- Object path, Source, Message description). The attribute Object path represents structured text. , we describe the details of generalizing the values of this attribute. The attribute Source represents an ID of the field or network device and consists of around 350 different values. This attribute is highly variable, but does not contribute to the data mining process due to fact that it uniquely identifies a device. For example, a credit card number almost uniquely identifies a customer and thus does not represent a useful attribute to generalize customers behavior thus to be used in the data mining. Thus, we omit this attribute from the analysis.

Similar to this, authors perform de-parameterization of data by replacing IP addresses, memory locations and digits by tokens. The attribute Message description represents unstructured text and consists of 280 values. We perform an in-depth analysis of values to determine means of aggregation. We conclude that a portion of values represents redundant data to other attributes (e.g., information in Message description: “Action A on source B is acknowledged by C” is repeated already in the same entry by the attributes Type of action: A, Source: B, User: C). The rest of messages are presented in an inconsistent way and provide information which, at this moment, we cannot parse and aggregate in a meaningful way. An alternative approach would be unsupervised clustering of messages.

Such clustering, however, does not guarantee semantic similarity of messages. We believe that the remaining attributes can compensate the information loss from this attribute. On the other hand, we are sure that such highly variable attribute does not contribute to the data generalization. Thus, we do not consider this attribute during the analysis. Our final set consists of 6 nominal attributes: Working shift, Aspect of action, Type of action, Object path, User account and SCBDA node. Some attributes are not applicable for all entries. As a result, every entry uses between 3 and 6 attributes.

A SCBDA node represents a computer that sends event details to the log. In our case, there are 8 different nodes. All nodes in the network have a dedicated and predefined role that typically does not change (e.g., there are 2 engineering workstations, 4 operator workstations and 2 connectivity servers). The attribute Type of action takes one out of 12 nominal values. This attribute describes the general type of action, such as: operator action, configuration change, process simple event, network message, etc. For types of action, which are performed by users, the attribute Aspect of action is applicable. This attribute takes one out of 6 nominal values in the log and details the character of the user action, such as: change of workplace layout, change in workplace profile, etc. The attribute Object path provides information about the location of the device, which is the object of the performed action (e.g., plant1/control module/production/cleaning/access settings/groups of devices/tanks). The attribute User account represents the username of the signed on user. Table 4 represents a sample of the analyzed log. Some events in the log are more severe than others. The severity of a SCBDA event depends on the combination of attribute values. Thus, a correct evaluation of specific attribute values can help to detect events that are undesirable for the normal process flow. For example, the value Audit Event Acknowledge of the attribute Type of action is semantically less important than the value Aspect Directory. This is because the first value implies an action where an operator acknowledged an alarm while the latter value implies that a new action was performed on the main configuration directory. Leveraging the stakeholders’ knowledge about the process and the semantics of nominal attribute values can help to distinguish critical and noncritical events in the complete log.

2. Data set validation :-

Our stakeholders argue that, at the time of logging, there were no known security incidents. We investigate the ways of validating this claim. We argue that due to size and high dimensionality of the log, manual inspection is infeasible. Thus, a (semi) automated approach is required. Typically, common log analysis tools imply the usage of predefined rule sets, which filter events out of logs. For example, various rule sets for analyzing logs, such as sys log and ssh log, are maintained. Unfortunately, such rule set for analyzing SCBDA system logs does not exist. Thus, we cannot perform a reliable log analysis to establish the ground truth. An alternative approach for establishing the ground truth would imply the log capture in a controlled environment.

In reality, this means either (1) performing the log capture in a lab setup or (2) performing the log capture in a constrained real environment (e.g., by reducing the number of process components to the ones that are validated to be correctly working). We argue that neither of the cases can compare to the actual real data. We acknowledge that, lacking the notion of the ground truth, we cannot perform an extensive discussion about false negatives. We are aware of this shortcoming in our approach. Nevertheless, the primary goal of our approach is to help operators uncover security-related events from real data, which would be overlooked otherwise.

LIMITATIONS OF THE APPROACH

We now describe the limitations of our approach.

Firstly, there is a threat scenario in which the SCBDA logs could be corrupted. For example, attacks performed on the devices in the field can produce erroneous input data for the SCBDA application and cause the generation of logs (and automated actions), which do not reflect the real situation in the field. Also, an attacker might manage to gain higher privileges (e.g., by exploiting a system-related vulnerability) and then prevent recording or erase some log entries. These attacks cannot be detected by observing SCBDA logs, since the log no longer represents a consistent data resource. For detecting these kinds of attacks, a complementary analysis of network data or field measurement is necessary.

Secondly, an important limitation of our approach is the possibility for an attacker to evade the detection by repeating the same command a number of times. To overcome this, we propose to enlarge the “knowledge window” and so learn what are normal patterns of behavior over a longer period of time. Since our current log capture is limited, we could not have implemented this yet. This also applies to the limitation of the currently manually set output threshold.

Thirdly, our approach for introducing the process knowledge highly depends (and thus can be biased) on the stakeholder’s knowledge about the specific process. We acknowledge that we cannot do anything to overcome this fact (because attribute values are nominative and thus human readable only). Finally, our approach cannot provide reasoning to the operator about the character of a suspicious event (e.g., “This event is suspicious because user A never worked from node B”).

Generally, all anomaly based approaches have the same limitation. This is because the model of normal (i.e., expected) behavior is typically described by a combination of attributes (i.e., implicitly). By inferring rules from the model, this limitation can be partly addressed. For example, by applying the algorithm for mining association rules to the identified patterns, we can compile rules whose interpretation is more readable to humans.

CONCLUSION AND FUTURE WORK

We analyze process-related threats that occur in the computer systems used in critical infrastructures. Such threats take place when an attacker manages to gain valid user credentials and performs actions to alter/disrupt a targeted industrial process, or when a legitimate user makes an operational mistake and causes a process failure. Currently, no control (e.g., monitoring tools) is available to mitigate process-related threats. To detect process-related threats, logs could be analyzed. These logs hold critical information for incident identification, such as user activities and process status. However, system logs are rarely processed due to (1) the large number of entries generated daily by systems and (2) a general lack of the security skills and resources (time). We propose an analysis tool that extracts non-frequent patterns, which are expected to be the result of an anomalous events such a undesirable user actions. We benchmarked the tool with real logs from a water treatment facility. Although no real security incident occurred in the log we took into account, at least five events were labeled by the stakeholders as anomalous. We believe that SCBDA logs represent an interesting data resource which gives a new perspective on SCBDA behavior. We argue that the analysis of SCBDA log represents a complement to the traditional security mitigation strategies.

As future work, we aim at expanding our tool to address anomalous sequences of actions, rather than single events/operations.

REFERENCES

1. Agawam, R., Spirant, R.: Fast algorithms for mining association rules in large databases. In: Boca, J.B., Jarke,M., Zanily, C. (eds.) In:

Proceedings of the 20th International Conference on VLDB, pp. 487–499. Morgan Kaufmann (1994)

2. Bald celli, C., Lavelle, L., Viola, G.: Novelty detection and management to safeguard information-intensive critical infrastructures. Int. J. Emerge. Manage. 4(1), 88–103 (2007)

3. Begum, K., Burgess, M.: Principle components and importance ranking of distributed anomalies. Mach. Learn. 58, 217–230 (2005)

4. Bulletin, C., Rush, J.: Vulnerability analysis of SCBDA protocol binaries through detection of memory access taintedness. In: John Hill, L.T.C. (ed.) Proceedings of 8th IEEE SMC Information Assurance Workshop, pp. 341–348. IEEE Press (2007)

5. Bingham J., Games D., Lu, N.: Safeguarding SCBDA systems with anomaly detection. In: Proceedings of 2nd International Workshop on Mathematical Methods, Models and Architectures for Computer Network Security, LNCS 2776, pp. 171–182. Springer (2003)

6. Brigs, T., Guests, K., Wets, G., Van hoof, K.: Profiling high frequency accident locations using association rules. In: Proceedings of 82nd Annual Transportation Research Board, Washington DC (USA), pp. 123–130. Transportation Research Board (2003)

7. Burdick, D., Calmly, M., Flan nick, J., Gherkin, J., You, T.: MAFIA: A maximal frequent item set algorithm. IEEE Trans. Know. Data Eng. 17, 1490–1504 (2005)

8. Burns, L., Heller stein, J.L., Ma, S., Peering, C.S., Rabenhorst, D.A., Taylor, D.J.: Towards discovery of event correlation rules. In: Proceedings of IEEE/IFIP International Symposium on Integrated Network Management, pp. 345–359 (2001)

9. Control Systems Security Program. Common cyber security vulnerabilities in industrial control systems. U.S. Department of Homeland Security (2011)

10. Goethals, B., Saki, M. (eds.): FIMI '03, Frequent item set mining implementations, Florida, USA, vol. 90 of CEUR Workshop Proceedings (2003)

11. Graney, G., Zhu, J.: Fast algorithms for frequent item set mining using FP-Trees. IEEE Trans. Know. Data Eng. 17, 1347– 1362 (2005)

12. Han, J., Camber, M.: Data mining concepts and techniques, 2 pap end. Morgan Kaufmann, San Francisco, CA (2006)

13. Heller stein, J.L., Ma, S., Peering, C.-S.: Discovering actionable patterns in event data. IBM Syst. J. 41, 475–493 (2002)

14. Hebe, J., Graham, J., Guan, J.: An ontology for identifying cyber intrusion induced faults in process control systems. In: Palmer, C., Shensi, S. (eds.) Critical Infrastructure Protection III, vol. 311 of IFIP Advances in Information and Communication Technology, pp. 125–138. Springer, Boston (2009)

15. Jalisco, K., Deicer, M.: Mining intrusion detection alarms for actionable knowledge. In: Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '02, pp. 366–375. ACM, New York, NY, USA (2002)

16. Lee, W., Solo, S.: Data mining approaches for intrusion detection. In: Proceedings of 7th Conference on USENIX Security Symposium—vol. 7, pp. 6–6. Berkeley, CA, USA, USENIX Association (1998)

17. Lim, N., Singh, N., Yank, S.: A log mining approach to failure analysis of enterprise telephony systems. In: Proceedings of the IEEE International Conference on Dependable Systems and Networks with FTCS and DCC, pp. 398–403. (2008)

18. Lees, F.P.: Less' Loss Prevention in the Process Industries. 3rd end. Butterworth-Heinemann, Guildford (2005) 19. Liu, Y., Ming, P., Reiter.: False data injection attacks against state estimation in electric power grids. In: Proceedings of 16th ACM Conference on Computer and Communications Security, CCS '09, pp. 21–32. ACM, New York, NY, USA (2009)

20. Manganaris, S., Christensen, M., Berkley, D., Hermit, K.: A data mining analysis of RTID alarms. Compute. Newt. 34, 571–577 (2000)