Survey of Data Intensive Computing Technologies Application to Security Log Data Management

Anne Tall and Jun Wang

Department of Electrical and Computer Engineering
University of Central Florida, Orlando, Florida
anne.tall@knights.ucf.edu, jun.wang@ucf.edu

Dezhi Han

Shanghai Maritime University
Shanghai, China
dezhihan88@sina.com

Abstract—Data intensive computing research and technology developments offer the potential of providing significant improvements in several security log management challenges. Approaches to address the complexity, timeliness, expense, diversity, and noise issues have been identified. These improvements are motivated by the increasingly important role of analytics. Machine learning and expert systems that incorporate attack patterns are providing greater detection insights. Finding actionable indicators requires the analysis to combine security event log data with other network data such as access control lists, making the big-data problem even bigger. Automation of threat intelligence is recognized as not complete with limited adoption of standards. With limited progress in anomaly signature detection, movement towards using expert systems has been identified as the path forward. Techniques focus on matching behaviors of attackers to patterns of abnormal activity in the network. The need to stream, parse, and analyze large volumes of small, semi-structured data files can be feasibly addressed through a variety of techniques identified by researchers. This report highlights research in key areas, including protection of the data, performance of the systems and network bandwidth utilization.

Keywords—security event log information management; data intensive computing; Hadoop; Spark

I. INTRODUCTION

Security logs are very important to the identification and analysis of attacks in a computer, network enterprise. However the collection, analysis and management of logs is a complex problem that includes many subtle, challenging details that are further complicated when including the logs from a diverse, expansive scope of computers, networking equipment, and mobile devices.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
BDCAT'16, December 06-09, 2016, Shanghai, China
© 2016 ACM. ISBN 978-1-4503-4617-7/16/12...\$15.00
DOI: <http://dx.doi.org/10.1145/3006299.3006336>

The main contribution of this paper is a survey of published, peer-reviewed journal articles on technologies and solutions to security log management. The latest innovations in the area are associated with applying data intensive computing technologies, such as Hadoop^[1] and Spark^[2]. Current research and development in analyzing security log and alert data into actionable security protection measures can be broken down into three key areas: data, analytics, and processing architecture. A variety of research initiatives are underway in each of these areas, as illustrated in the mind map in figure 1 and further discussed in this paper. Collectively these results will need to be integrated together to achieve better detection and implementation of protections. Background motivation on the complexity of this challenge is included in section (II) and the approach using a four tier architecture design is described in section (III). Section IV covers work on extending current solutions, performance optimizations, and other enhancements to security log data management, followed by next steps and conclusion in the final section (V).

Figure 1: Security Log Data Management Research Areas

II. BACKGROUND AND MOTIVATION

The primary motivation for monitoring security log data is increasing dependency on computer communication networks while at the same time the sophistication of cybersecurity-attacks is growing, requiring more advanced, but affordable, approaches to detect and react to these attacks. In today's environment large organizations must develop strategies to be resilient. The focus has moved from adding layers of security mechanisms towards analyzing the data produced by these

layered mechanisms. The recognized goal is to detect and respond to attacks when they occur, rather than trying to apply forensics weeks or months after an incident. Security logs often contain those initial indicators. As a result security log data is becoming increasingly important to achieve continuous monitoring and situational awareness.

Collecting and then producing actionable insights from security log data has been a very difficult problem for the entire industry to achieve even though it has been worked on extensively. The limitations identified^[3] include:

Complexity – system administrators, statisticians, and computer network security experts are typically required to install, configure security event logging systems/software. These systems require people with fairly sophisticated technical skill levels to tune policies and configurations to make them effective. Security logs produced from various types of devices are not standardized making correlation of the data difficult and especially complex when tracking actions from one device to the next.

Time – typically security log management solutions require months to install and configure because every component in the network enterprise has to be configured to send the logs to a centralized data store.

Expense – the price of tools is often based upon the volume of data collected, so as more data is collected to provide higher levels of insight the costs rise dramatically. (The order of magnitude is in the millions of dollars.)

Diversity – traditional solutions are not easily adapted or configured to integrate data from diverse environments that include cloud computing resources, mobile devices and specialized systems.

False Alerts – solutions often generate many alerts or noise that the technical team has to respond to and investigate. This is typical when the capability is first implemented until a configuration and tuning is mature. Failure to recognize and reduce noisy, high-volume false alerts will cause operators to dismiss findings, (even the valid, actionable alerts).

These current limitations can be more efficiently managed or overcome by introducing data intensive computing technologies to security event log processing.

III. ARCHITECTURE DESIGN

As illustrated in figure 2 below, the architecture solutions typically include the following four architecture tiers:

Tier 1 – Log/Event Data Sources: This tier includes all of the devices in the enterprise network that are generating security event logs that can be analyzed to identify anomalies and attack indicators. The types of devices and the logs they generate can be very diverse. This include sources from security mechanisms such as firewalls and intrusion detection systems.

Tier 2 – Collection Servers: Generally placed close to clusters of end points, the collection servers are an intermediate storage of log data. They provide the interface to the end-point-specific operating system or application API and the generic large scale data storage system, providing required

port/protocol/service call translation and integration. This tier provides the services to enable streaming of data from the end points into the large Tier 3 data store. For example, Flume^[4] and LogStash^[5,6] pipeline data from disparate sources, then normalize the data to facilitate storage in the Hadoop Distributed File System (HDFS).

Tier 3 – Storage: In place of traditional relational database management systems (RDBMS) (e.g., SQL databases), HDFS solutions are used due to their ability to handle the characteristics of security log data. For example, HDFS is designed for reading rather than modifying data. Security event logs are lines of information that are not updated, but rather appended over time. In addition, because the data is coming from a diverse set of systems in the enterprise it is not consistently structured (i.e., typically semi-structured). Additionally, moving away from expensive proprietary database systems that are priced based upon volume of data stored to open source solutions provides significant cost savings.

Figure 2: Architecture

Tier 4 – Intelligence and Analytics: The fourth tier of the architecture focusses on the application of threat intelligence (patterns and indicators of cybersecurity attacks) to analyze the threat data. Statistical procedures are applied to identify unexpected patterns in the data. Unexpected data exchanges between IP addresses or clusters of unexpected applications

may be indicators of data exfiltration and malware execution. Layer 4 is particularly challenging in that automated means of identifying these anomalies or unauthorized actions are not in wide use and current activities are either manual^[7], or only automated at very basic levels (e.g., spikes at certain times or clustered around certain address.)

Current Security Information Event Management (SIEM) tool vendors, such as SPLUNK^[8] are making enhancements to Tier 3 capabilities by adding interfaces to HDFS. Work in Tier 4 is acknowledged as a research area in several papers and significant improvements will be enabled with the Tier 3 improvements. The following sections further highlight the challenges and research in security event log management.

IV. EXTENSIONS, OPTIMIZATIONS, ENHANCEMENTS

Current security event log analytical tools are evolving to obtain meaning from data sources from a wide variety of systems. Balancing between the detection, filtering and blocking at each point in an enterprise network and providing overall situational awareness requires focused analysis. Too much information results in a loss of clarity and the generation of too many false positives. However, leaving out key pieces of data that enable the tracking of an attack across multiple points in the enterprise can mean the loss of the key attack indicators, ultimately resulting in the loss of the original log-analysis investment.

Based upon a survey of research in applying data intensive computing technologies to improve the management, identification, and distillation of actionable information, a number of areas are actively being improved, as summarized in Table 1 and expanded in the following paragraphs.

Table 1. Challenges and Research

Current Challenges	Research Areas
Unstructured footprints and indicators of attack	Analytics: machine learning, anomaly detection, signature detection, expert systems Visualization
Massive storage of small data files	Appending and indexing
Parsing semi-structured log data	Index on search Tuples, Log Key
Manual review and parsing of threat reports	Standard semantic characterization of threats, automated management of intelligence
Tampering and corrupting the integrity of log data Privacy or confidentiality of log data	Architecture framework for access control, integrity and confidentiality of log data
Performance	Processing monitoring, optimization, load balancing
Network Bandwidth Utilization	Endpoint Data Filtering Distributed load balancing

A. Analytics and Visualization

Cybersecurity attacks are often characterized by unstructured indicators or footprints. Machine learning techniques are proposed^[9,10] to automate the detection process. Some indicators have characteristics that lend them to this types of analysis. Two areas of analytics are (1) anomaly detection – to identify patterns of authorized activity that reach unacceptable

levels and (2) signature detection – to identify known patterns of unauthorized activity. For example, nearest-neighbor analysis to clustering of activity within an IP address range, examination of workflow patterns associated with time frames to identify unusual off-hour activity, or identification of traffic volume outliers may provide indicators of file exfiltration. A third area proposed is (3) expert systems that upon focusing on attacker behaviors.^[11] Understanding workflow is key to define the execution path of a system (techniques that can be applied include finite state automation, state machines, petri nets) and string trace execution to identify abnormalities (programs spawning unusual processes). Based upon the analyzed workflow, performance measurements of execution time further characterize and provide an understanding of acceptable system behavior. Weighted scores also further identify pattern frequency. Automating the event correlation process facilitates detecting a multi-stage attack across multiple systems, as described in^[12]. Three key areas for analytics are:

1. Anomaly Detection – Applying locality and sensitivity algorithms to cluster log or threat data facilitates the identification of outliers. This facilitates identifying important data for analysis. As applied in recommendation systems, nearest neighbor or cluster data algorithms include^[13, chapter 3]:

- Similarity – Jaccard similarity– the similarity of two sets S and T is, the ratio of the size of S and T to the size of their union. $|S \cap T| / |S \cup T|$ For security event log data character/word-level similarity (not similarity of meaning) could be achieved using this formula to identify entries that match a reference set of events/logs of interest.
- Shingling – construct sets of short strings that appear in the logs for comparison. For K-Singling, K is the length of the sub-string that is used for comparison. K should be picked large enough so that the probability of any given single occurring in any given file is low, i.e., a low occurrence of extraneous or not applicable information. For email K = 5 and for large documents K=9, so for security log data, selecting a value on the order of 5 may be appropriate. Shingling techniques can be optimized by removing frequently occurring stop words or hashing. A challenge with using shingling is that results in large data sets.
- Distance Measures – If the “space” of interest for security event logs is defined, for example based upon a period of time and range of IP addresses in a network, then a comparison or measurement of distance between a given event and the space can be computed. The space under consideration may have multiple dimensions (i.e., n-dimensional Euclidean space) and the distance would be the square the distance in each dimension, sum the squares and take the square root.

2. Signature Detection – Collaborative filtering techniques that apply multiple algorithms, such as similarity and clustering would identify outliers or signatures of potentially unauthorized actions. For example, people in the finance department use certain computers to execute certain applications using specified data, files or folders. Any actions

outside those authorized, normal action would be identified for further analysis. Ultimately if each point where the security log data can be coupled with business process data, greater insights and impacts, in the business context could be derived.

3. Rule-based Expert Systems – Google has applied a method to characterize attacks or deviations (not normal behavior) to identify attacks^[11]. This technique involves programming rules, which requires expert security analysis, for example “Alert if host connects to a suspicious IP shortly after downloading a PDF.” A key characteristics or design goal of the system is to apply rules to small sets of logs and features to make them more easy to understand, less error prone, and directly connect between the rules and the alerts.

The architecture proposed by the NATO researchers^[14] emphasizes the importance of applying metrics to the analysis process to gain meaningful insights on the data. Ideally this action is executed in near-real time. However identifying crisp, easy to gather metrics that are mission or business relevant is complex. Consistent methodologies and processes to the analysis is critical. Graphite is identified as the solution for achieving meaningful syslog and NetFlow metrics based upon its ability to visualize temporal data.

Marty^[15], provides many example approaches to security log data analytics and visualization. Overall the best solutions are designed with users in mind. For example allowing people with deep technical domain knowledge to gain access to event details to take corrective action or alternatively providing managers with higher level status on trends and anomaly detection. Information can be portrayed in a fairly simple display such as a heat map chart using a program such as Excel, or with sophisticated dashboard programs like Tableau and Kibana.

B. Massive Storage of Small Files

In the Apache Hadoop, the default block size is 64 MB and in the Cloudera Hadoop the default is 128 MB. If block size was set to less than 64 MB, there would be a huge number of blocks throughout the cluster, which causes NameNode to manage an enormous amount of metadata. However each individual entry in a security log is typically much smaller, so strategies to store and analyze related data on the same node need to be incorporated to meet performance requirements. Research in this area identifies approaches to manage this challenge^[16]. A proposed approach is to combine small files into a large file and then employ an index for rapid access^[17].

Researchers propose a hierarchical file archiving (HFA) system^[18] where frequently accessed small files are stored in active smaller files, and less frequently accessed, older files are stored in larger archive files. Noticing the performance limitations of Hadoop’s file merging solution, (i.e., Hadoop Archive which facilitates the packing of small files into larger archive files) this scheme applies the criteria of log priority, size, and frequency of use to the archive decision.

C. Processing Semi-Structured Data

Parsing semi structured log data involves storing the data, parsing it for analytics, cleansing or editing it to remove extraneous information, and correlating the results by identifying clusters, folding, compressing and other techniques. Normalization of the massive data set originating from multiple devices can be critical to the quality of the analysis that can be conducted. It may appear trivial but much hidden complexity can appear upon mining the data. For example, timestamps can have many different time zones, be incorrect, or missing and the analysis can be significantly impacted if temporal data is a key dimension.

An effective approach identified by several researchers to conduct analytics involves defining tuples of multiple keys to apply to the lines of log data. An overview of this approach is described in references^[19-21]. The tuples from the logs identify the key information including severity degree, event type, timestamp, and node/source ID.

The focus of the LogMaster solution is based upon the temporal characteristics of the events. For the EASEAndroid solution, threat/attack characteristics are manually correlated. To scale-out to detect a wide variety of attacks, tuples would need to also be identified and applied to the threat and attack characteristics or footprints. Other research^[22-24] provides semantics and potential concepts for identifying tuples to characterize threats.

Efforts to define standards for security log data formats include the Common Event Expressions (CEE), Lumberjack, and Vocabulary of Event Recording and Information Sharing (VERIS). Unfortunately, these efforts have limited support and are not identified as product features of the research efforts analyzed. This leads to the hypothesis that forcing structure on security log data is not going to solve the challenges and so systems must be designed to handle the data in loosely, semi-structured formats.

D. Application of Cyber-threat Intelligence

Manually or automatically characterizing the attack patterns, is an emerging area. Although organizations provide human readable threat reports, automated information typically consists of configuration files for specific devices, for example, lists of IP addresses for firewalls to block. There is a lack of machine readable data that encompasses the full breadth of advanced adversary tactics, techniques, and procedures that include multi-step attacks executed across multiple devices over a longer period of time. Challenges exist with integrating the individual data elements that characterize indicators of compromise, e.g., IP addresses, email, web sites, domain names, file hashes, file paths, communication protocols, and port ranges. Several papers provide examples of standards^[22-24], research, and proposed approaches to this challenge.

E. Protection of the Security Log Data

Researchers have acknowledged and proposed an approach to protect security log data from the end point throughout the architecture, including at the analytics tier.^[25] This is especially important when security data is applied to

investigations requiring legal evidence and chain of custody must be maintained. Authentication, access control and encryption techniques have been proven with past RDBMS, so as new data intensive computing technologies are adopted some of these same techniques can be adjusted and applied.

Not only do the protection techniques need to be thoroughly tested to ensure they cannot be bypassed or otherwise undermined, the performance impact of applying encryption techniques also needs to be understood. The need to anonymize log data when it is stored in a shared cloud computing environment is documented and The Onion Router (TOR) network is proposed as an approach to achieve this objective.^[25] As the volume of security log data and the complexity of threat, attack detection techniques continues to increase, the number of businesses looking to off-load this processing onto cloud service providers would most likely increase. Protecting the logs, especially as they exit the enterprise and go into shared resources will become increasingly important.

F. High Response Time and Near-real-time Performance

To achieve high efficiency in query and analysis in near real-time, including the latest streamed data, on-line processing of large data sets is needed.^[18] During the development of a security system for email spam detection, user pattern mining, and log analysis was developed using Hadoop-off line processes, and then replaced with a more flexible online streaming system.^[26] The original system, and motivation for improvement, was the inability to quickly adapt to bursts of malicious email traffic. The new approach was also applied in memory processing to more rapidly adjust to streaming data and enable real-time analytics.

Spark is identified as the best candidate, in terms of its speed and stability, i.e., repeatedly requiring the same amount of time to perform tasks.^[27] This enables load balancing at scale, with security monitoring data from honeypots, DNS, HTTP and IP-flows. Off-line analysis with Hadoop and on-line analysis with Spark provide the ability to support significant scale-out.

Overall based upon this survey of the techniques, the primary optimizations enabled by big data computing include:

- Preprocessing data and removing extraneous spacing or (punctuation) marks
- Distributing data between different numbers of data nodes (based upon scalability analysis)
- Optimizing the map phase and reduce phase of the job
- Adding more nodes
- Adjusting the spill ratio, i.e., configuring the memory on the mapper machine so that if it is overloaded and the processing spills out to disk versus being processed in memory
- Setting job thresholds
- Applying a cache for lookup of key words/values and potentially distributing the cache across the nodes
- Configuring the processing to skip bad records

Applying cluster-monitoring tools to gather performance statistics enables these performance tuning options.

G. Network Bandwidth Utilization

The use of high volumes of network bandwidth to exchange log data for distributed event correlation can result in unacceptable performance impacts. End point filters have been proposed.^[10] However end point filtering design needs to address the potential of missing key events and meeting regulatory archive requirements. To alleviate the end point and data node processing burden, approaches applied for other distributed processing challenges, the General Purpose Graphical Processing Unit (GPGPU) have been proposed^[28].

To measure the communication cost of a solution, the size of the input to a task is calculated. The efficiency of implementation can be computed by adding the communication costs of all the tasks, or computation stages of the data processing algorithm^[13, chapter 2]. A cost for a star or multi-way join of log files, (e.g., threat data, log definitions, inventories of equipment, user accounts, permissions) could be computed based upon the size of the data input to the join operation.

Security log analysis shares some of the same characteristics and as a result benefits from the methods of analysis applied to error and performance log analysis such as load-balancing and network optimization. However a distinct difference is that with security logs, malicious processes may be specifically targeting opportunities to undermine the integrity of the data.

V. CONCLUSIONS AND NEXT STEPS

This survey compared and evaluated alternative approaches to security log data management, to formulate an assessment of the value of data intensive computing technologies. The observations formed on the status of security log management are summarized below:

- Proposed architectures are based on a hierarchy of systems that forward data to collection, storage, and analysis nodes. Filtering at the point of origin has been applied to reduce network bandwidth. Other techniques, such as load balancing and addressing the high volumes of small files are also applied to enhance and optimize performance.
- Index on query, rather than index on store (as done with traditional RDBMS) is more effective for security event log data. Log standardization efforts: CEE, Lumberjack, VERIS appear to be stalled and have limited success. Attempts at encouraging software developers to use and adhere to standards when creating software that writes to log files has not succeeded. Log data is diverse and semi-structured and as a result challenging to parse for analysis.
- Security and protection of the security log data is an emerging area. Researchers have proposed encryption and privacy protection techniques to protect the storage and processing in cloud environments.
- By applying analytics and visualization techniques, interactive processing can be achieved to provide early detection and post attack investigative tools. Connecting

related actions across multiple devices using analytic tools and relating it to the business processes provides greater insights.

- Distributed file systems and processing is achieving performance and scale requirements at rates much more affordable than previous RDBMS.

The overall outcome of the analysis is that data intensive computing technologies are effective strategies for collecting (centrally aggregating) and searching (index on search, rather than parse on store) log data from various sources in semi-structured formats (a range of changing structures). Approaches are maturing to address the current challenges previously listed. Further research and investigation in these areas should provide improvements to the analysis by using automated machine learning programs based upon attacker actions. The ability to meet performance requirements using distributed file systems rather than expensive RDBMS have been demonstrated. Solutions to security and network bandwidth utilization issues are proposed. Although there are several commercial and open source product solutions, this remains a challenging problem. Opportunities to apply new computing solutions to large volumes of data, such as applying threat characteristics to search log data using expert system techniques, appears to be the focus area where the most significant gains are expected from further research.

VI. ACKNOWLEDGMENT

This work is supported in part by the US National Science Foundation Grant CCF-1527249, CCF- 1337244 and National Science Foundation Early Career Award 0953946, and the National Science Foundation of China (Grant nos. 61373028 and 61672338).

VII. REFERENCES

1. Apache Hadoop, <https://hadoop.apache.org/>
2. Spark, <https://spark.apache.org>
3. Chuvakin, A.; Schmidt, K.; Phillips, C.; Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management, Elsevier, 2013.
4. Apache Flume, <https://flume.apache.org/>
5. Beaver, D.; Hutchinson, S.; Elasticsearch, Logstash, and Kibana (ELK), Carnegie Mellon University Software Engineering Institute FloCon 2015, January 2015.
<http://www.cert.org/flocon/past-conferences.cfm>
6. Logstash, Elastic, <https://www.elastic.co/products/logstash>
7. Yen, T.; Oprea, A.; Onarlioglu, K.; Leetham, T.; Robertson, W.; Juels, A.; Kirda, E.; "Beehive: Large-Scale Log Analysis for Detecting Suspicious Activity in Enterprise Networks," ACM, ACSAC, December 2013.
8. Goldberg, J., SPLUNK, "Big Data for Security: How Can I Put Big Data To Work For Me?" and "Good Guys Vs Bad Guys: Using Big Data to Counteract Advanced Threats," RSA Conference Europe 2013, October 2013.
9. Fu, Q.; Lou, J.; Wang, Y.; Li, J.; "Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis," 2009 IEEE International Conference on Data Mining, 2009.
10. Myers, J.; Grimaila, M.; Mills, R.; "Log-Based Distributed Security Event Detection Using Simple Event Correlator," Proceedings of the 44th Hawaii International Conference on System Science, 2011.
11. Poletto, M., "Data mining for security at Google," Google Security Team – Stanford CS259D, 28 Oct 2014.
12. Bhatt, P.; Yano Toshiro E.; Gustavsson, P.; "Towards a Framework to Detect Multi-Stage Advanced Persistent Threats Attacks," 2014 IEEE 8th International Symposium on Service Oriented Systems Engineering, 2014.
13. Leskovec, J.; Rajaraman, A.; Ullman, J.; Mining of Massive Datasets, Second Edition, Cambridge University Press, 2014.
14. Vaarandi, R.; Pihelgas, M.; "Using Security Logs for Collecting and Reporting Technical Security Metrics," 2014 IEEE Military Communications Conferences, 2014.
15. Marty, R., PixlCloud – Data Visualization, <https://about.me/raffy>
16. Mackey, G.; Sehrish, S.; Wang, J., "Improving Metadata Management for Small Files in HDFS," Proceedings of IEEE International Conference on Cluster Computing and Workshops, August 2009.
17. Dong, B.; Qiu, J.; Zheng, Q.; Zhong, X.; Li, J.; Li, Y.; "A Novel Approach to Improving the Efficiency of Storing and Accessing Small Files on Hadoop: a Case Study by PowerPoint Files," 2010 IEEE International Conference on Services Computing (SCC), 2010.
18. Yang, F.; Liu, H.; Zhao, Z.; "Research on Cloud-Based Mass Log Data Management Mechanism," Journal of Computers, Vol 9, No 6, June 2014.
19. Zhou, W.; Zhan, J.; Meng, D.; Xu, D.; Zhang, Z., "LogMaster: Mining Event Correlations in Logs of Large scale Cluster Systems," Cornell University Library, January 2013.
20. Ferrera, P.; dePrado, I.; Palacios, E.; Fernandez-Marquez, J.; Seruguendo, G.; "Tuple MapReduce: Beyond classic MapReduce," 2012 IEEE 12 International Conference on Data Mining, 2012.
21. Wang, R.; Enck, W.; Reeves, D.; Zhang, X.; Ning, P.; Xu, D.; Zhou, W.; Azab, A.; "EASEAndroid: Automatic Policy Analysis and Refinement for Security Enhanced Android via Large-Scale Semi Supervised Learning," Samsung Research America and North Carolina state University, 24th UNENIX Security Symposium, 12 August 2015.
22. Adversarial Tactic, Techniques and Common Knowledge (ATT&CK), <https://attack.mitre.org/>
23. Common Attack Pattern Enumeration and Classification (CAPEC), <https://capec.mitre.org/>
24. Structured Threat Information eXpression (STIX), <http://stixproject.github.io/>
25. Ray, I.; Belyaev, K.; Strizhov, M.; Mulamba, D.; Rajaram, M.; "Secure Logging as a Service – Delegating Log Management to the Cloud," IEEE Systems Journal, Vol. 7, No. 2, June 2013.
26. Chen, G.; Wu, S.; Wang, Y.; "The Evolvment of Big Data Systems: From the Perspective of an Information Security Application," Elsevier - Big Data Research, 23 March 2015.
27. Marchal, S.; Jiang, X.; State, R.; Engel, T.; "A Big Data Architecture for Large Scale Security Monitoring," 2014 IEEE International Congress on Big Data, 2014.
28. Bandre, S.; Nandimath, J.; "Design Considerations of Network Intrusion Detection System using Hadoop and GPGPU," IEEE 2015 International Conference on Pervasive Computing (ICPC), 2015.