Autonomic Intrusion Detection System in Cloud Computing with Big Data

Conference Paper · July 2014

CITATIONS
5

READS
5,977

5 authors, including:

Kleber Vieira
Federal University of Santa Catarina
21 PUBLICATIONS **189** CITATIONS
SEE PROFILE

Fernando Schubert
Federal University of Santa Catarina
6 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Guilherme Arthur Geronimo
Federal University of Santa Catarina
28 PUBLICATIONS **124** CITATIONS
SEE PROFILE

Carlos Becker Westphall
Federal University of Santa Catarina
275 PUBLICATIONS **1,243** CITATIONS
SEE PROFILE

Autonomic Intrusion Detection System in Cloud Computing with Big Data

Kleber M.M. Vieira, Fernando Schubert, Guilherme A. Geronimo,
Rafael de Souza Mendes, Carlos B. Westphall
{kleber, schubert, r2, mendes, westphal}@inf.ufsc.br
LRG - INE - UFSC - Florianopolis - SC - Brazil

***Abstract*—This paper analyzes real-time intrusion response systems in order to mitigate attacks that compromise integrity, confidentiality and availability in cloud computing platforms. Our work proposes an autonomic intrusion response technique enabling self-awareness, self-optimization and self-healing properties. To achieve this goal, we propose IRAS, an Intrusion Response Autonomic System, using Big Data techniques for data analytics and expected utility function for decision taking.**

I. Introduction

The quickly expansion in the volume of data generated in the Internet, with the consequent diffusion of personal, financial, legal and other data in the Web, has created a very valuable content for hackers, crackers and other cybercriminals.

In this context, the need for a highly effective and quickly reactive security system gains importance. The growing number of attacks and vulnerabilities exploitation techniques requires preventive measures by system administrators. These measures are getting more complex with the growth of data heterogeneity and the increasing complexity of the attacks. In addition, slow reaction time from human agents and the huge amount of data and information generated makes the decision making process an arduous task. In response to this, there is an increase in the usage of IDS (Intrusion Detection Systems) [1], as a way to identify attacks patterns, malicious actions and unauthorized access to an environment [2].

The need for IDS is growing due to limitations from IPS (Intrusion Preventing Systems) - which focus on alerting administrators when vulnerability is detected, connectivity and threat evolution, as well as the financial appeal of cybercrime [3].

Despite its growing importance, current IDS solutions available have limited response mechanisms. While the researchs focus is on better intrusion detection techniques, response and effective reaction to threats are still mostly manual and rely on human agents to take effect [4].

Recently, some intrusion detection tools started to provide some limited set of automated responses, but with the intrusions growing complexity, the need for more effective response system strategies have raised. Due to implementation limitations, research works on intrusion detection techniques advance in a faster rate than intrusion response systems [2].

The development of reliable and quickly responsive systems is even more important for cloud computing, where the elasticity increases the risk and costs of an attack [5].

A. Motivation

The number of computer attacks has grown in quantity and complexity, making defense an increasingly arduous task. Each computer that suffers an attack has very limited information on who initiated the attack, and the origin of it. Current systems for intrusion detection and response do not follow the growing number of threats [4].

The focus on manual processes creates a delay between detection and response, leaving a time window for attackers [6]. Research findings by [4] indicate that if a skilled attacker has a range of 10 hours between intrusion and response, the attack has 80% chance of being successful, if the attacker has 20 hours, the attack has 95% of chances of being successful, and if the attacker has 30 hours the attack becomes virtually foolproof. In this situation, the system administrators skills become irrelevant. On the other hand, if the response is instant to the intrusion, the chance of a successful attack is almost zero. [4] says that statistics have shown that the number of pro-rated intrusion is growing. The high cost of the contract indicates serious financial commitment made by the Pentagon to prevent and secure their infrastructure from being attacked by another country.

An automated intrusion response system that combines the best techniques of intrusion detection would provide the best defense possible in short time, giving more time to the system administrator to develop a permanent solution to avoid further attacks or fix the vulnerability exploited [6][4].

According to Buyya, the Cloud [7] is complex, large-scale and heterogeneous and its management is challenging. This environment requires an automated and intelligent system to provide security services with efficient cost. Thus, cloud systems represent a distinct structure with several layers of abstraction that requires specific IDS and response techniques to address its complexity.

The paper is organized as follows: Section 2 describes the proposal underlying concepts and key technologies, section 3 presents an overview of the related work, section 4 details the proposal and section 5 concludes the paper with future directions and open challenges.

II. Background

Autonomic computing can overcome heterogeneity and complexity of computing systems being considered a new and

effective approach to implement complex systems, addressing several issues where humans are losing control due to complexity and slow reactions, such as security systems [8].

The autonomic computing model is based on the so called self properties. The self is inspired by the autonomic nervous system of the human body, which can manage multiple key functions through involuntary control. The autonomic computing system is the adjustment of software and hardware resources to manage its operation, driven by changes in the internal and external demands. It has four key features, including self-configuration, self-healing, self-optimization and self-protection.

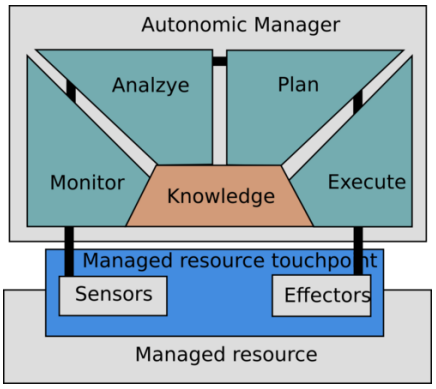

Fig. 1. An autonomous system.

Figure 1 shows the structure of an autonomic system and its MAPE-K cycle [9], composed by the monitoring, analysis, planning and executing modules. All the management of the autonomic component is performed by a meta-management element, which make decisions based on the knowledge-base built.

Sensors are responsible for collecting information from the managed element. Information collected by the sensors is sent to the monitors where they are interpreted, pre-processed, aggregated and presented in a higher level of abstraction. After this, the analysis phase is executed and planning takes place. At this stage, a work plan is resulted, which consists on a set of actions to be performed by the executor. Only the sensors and executors have direct access to the managed element. Through the autonomic management cycle, there may be a need for decision-making, thus it is also necessary the presence of knowledge base [10].

A. Autonomic Systems Properties

The essence of autonomic computing is self- management. To implement it, the system must be self-aware as well as environment-aware. Thus, the system must precisely know its current situation and be aware of the operational environment in which it operates. From a practical standpoint, according to [10], the term autonomic computing has been used to denote systems that have the following properties:

- Self-awareness: the system knows itself: its components, their state and behavior;
- Context-awareness: the system must be aware of the context of its execution environment and be able to react to changes in its environment;
- Self-configuring: the system must dynamically ad- just its resources based on its status and the state of the execution environment;
- Self-optimizing: the system is able to detect performance degradations and functions to perform self-optimization;
- Self-protecting: the system is able to detect and protect its resources from external and internal attackers, keeping its overall security and integrity;
- Self-healing: the system must have the ability to identify potential problems and to reconfigure itself in order to continue operating normally;

III. Related Work

In this section, four related works that we considered important to our research were selected. To evaluate these works, five topics were chosen to analyze them. The chosen topics are: The chosen topics are:

- Does it propose IDS?
- Is it suitable for the Cloud scenario?
- Does it respond against attacks?
- Does it have a Self-Healing method?
- Which kind of algorithm is used?

A. Proposal of Wu 2013

[11] propose an autonomous manager which introduces a mechanism for multi-attribute auction. Its architecture has a layer of managed resources covering generically all physical devices like routers, servers or software applications. These resources should be manageable, observable, and adjustable. The state of resources refers to all data (events) that reflect the state of existing resources, including logging and real-time events. This architecture also has an autonomous agent as a detection engine, optimization strategy, autonomic response, and knowledge base module.

The architecture has agents responsible for MR information capture, preprocessing and redundancy removal before final submission to AM agents.

The multi-attribute auction model is defined as follows:
The auction model: M = A,B,S,V,C,Res
A refers to attributes, each auction has *n* attributes, (A1, ..., An).
B refers to the auction participants that needs to buy (win) an event.
S refers to a seller (auctioneer) which includes *n* buyers.
Buyers can provide events with different attributes. V: V-¿R refers to a buyer function. C refers to seller costs. Res refers to the transaction *Res = P, P* is one of the members of *A*.
The buyer benefit *B* is determined by *U = V(a)-P* and the seller benefit is *Si* ? is *Ui = P-Ci(a)*
The auction process is performed in four steps: 1) the buyer publishes the evaluation V (a). 2) each seller i makes a Bi¿ 0 proposal. 3) the transaction is committed. 4) the transaction is processed.

Wu says that the autonomic response depends on knowledge base of possible actions. It is necessary to form knowledge base with attributes and valuations [11].

B. Proposal of Kholidy 2013

Kholidy approach describes how to extend the current technology and IDS systems . His proposal is based on hierarchical IDS [12]) to experimentally detect DDoS, host-based, network based and masquerade attacks. It provides capabilities for self-resilience preventing illegal security event updates on data storage and avoiding single point of failure across multiple instances of intrusion detection components.

His proposal consists on a hierarchical structure, autonomic and cloud-based, extending his earlier work [12] with features such as autonomic response and prediction. In particular, it assesses vulnerabilities and risks in the system through a mechanism that builds a security model based on risk assessment and security event policies criticality. It also provides the possibility of automatic response to actions based on a set of policies defined by the system administrator. However, a black box format does not clarify possible answers or makes clear how to choose the best answer leaving that decision to a system administrator. Finally, the architecture offers some predictive capabilities based on Holt-Winters algorithm [13], which predicts and detects abnormal behavior of network traffic when the amount of collected network traffic is either too high or too low, compared to normal network traffic. Predictive capabilities improve detection accuracy of both decision making and automated response [14].

C. Proposal of Vollmer 2013

This article describes new architecture that uses concepts of autonomic computing based on SOA and external communication layer to create a network security sensor. This approach simplifies the integration of legacy applications and supports a safe, scalable, self- managed structure.

The contribution of this work is a flexible two level communication layer, based on autonomic computing and SOA. A module uses clustering and fuzzy logic to monitor traffic for abnormal behavior. Another module passively monitors network traffic and deploys deceptive hosts in the virtual network.

This work also presents the possibility of an automatic response but it does not address this topic in detail, leaving it for future works [15].

D. Proposal of Sperotto 2012

It presents an autonomic approach to adjust the parameters of intrusion detection systems based on SSH traffic anomaly.

This paper proposes a procedure which aims to automatically tune system parameters and, in doing so, to optimize system performance. It validates their approach by testing it on a probabilistic-based detection test environment for attack detection on system running SSH [16].

E. About the related works

Related works representing the state of the art attempt to solve the problem of cyber-attacks by proposing intrusion detection mechanisms and increasing detection techniques. Although many of them show the need of automatic responses, none of them go deeper in this direction. The works of [11] and [15] mention the possibility of response to attacks; however, both works leave this point open not going deep into the issue.

Table I shows a brief comparison between the related works, based on the previous described topics.

IV. Proposal

In this work, we propose a model for autonomic intrusion detection system based on the autonomic loop, commonly referenced as MAPE-K (Monitor, Analyze, Plan, Execute and Knowledge Base). To monitor and analyze, we use sensors to collect data from IDS logs, network traffic, system logs, and data communication. For storage and further analytics, a distributed storage is used, for instance we chose Apache Hadoop as storage engine because its performance, scalability and further capabilities to be extended and suffer Map Reduce jobs.

For analysis, planning and execution we present a model based on expected utility function [17].

A. Proposed system: IRAS Intrusion Responsive Autonomic System

The approach of IRAS follows the line of an autonomic system for intrusion response. The sensors collect log data from network IDS and host systems. This information is compiled in a Big Data environment [18], preprocessed and placed on a higher level of abstraction, ready to be sent to analysis and planning cycles of the autonomic loop.

Based on the MAPE-K autonomic loop, the phases of IRAS are:

M data collection from sensors, storage on Big Data infrastructure.

A preprocessing (filtering, aggregation) and analysis.

P calculation of utility.

E execution, this means, based on results of utility function, effective measures will be taken in the system.

K the knowledge base built from the monitored and analyzed data is used to feedback the utility based function, weighting the utilities.

B. Monitoring

The first phase of the MAPE-K autonomic cycle corresponds to monitoring. In this step, sensors are used in order to obtain data reflecting changes in behavior of the managed element or information from the execution environment that are relevant to the self-management process.

The concept of sensor is a little generic, but it is possible to consider that a sensor is a component of the system that makes the connection between the external world and the management system.

Author	IDS	Cloud	Response	Self-healing	BigData	Algorithm
[11]	yes	no	yes	no	no	Auction
[14]	yes	yes	yes	no	no	Holt- Winters
[15]	yes	no	yes	no	no	Fuzzy
[16]	yes	no	no	no	no	Flor-based

TABLE I
RELATED WORKS

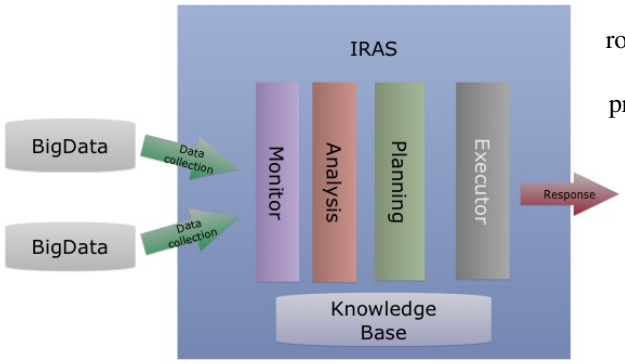

Fig. 2. IRAS Intrusion Responsive Autonomic System

However, the important nuance to observe in data monitoring for security in Cloud Computing, is that the data will be intrinsically temporal. This characteristic impose some peculiarities in the data structure to storage temporal information, as well as, in the queries to be executed in the sensor data base to retrieve useful information.

As defined in [19], Big data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze. [20] defines the three data characteristics of Big Data sets: volume, variety and velocity. We have a large volume of data from various sources like logs, IDS alerts, network traffic scans, where processing and analysis speed is necessary to extract meaningful information from these sources. Based on work by Suthaharan [18] where it was decided to use a structure with BigData tools, in this case Hadoop to organize the collected data in the cloud and perform the monitoring. However, Suthaharan use Machine Learning (ML) to find attacks and in this work we propose to use a technical knowledge based on intrusion detection systems [1] making it possible to detect attacks like Stuxnet or Duqu ones. Thus we make a map & reduce over the collected data to identify signatures of known attacks extracting some significant data such as origin, destination of attack, type, signature and timestamp.

C. Analysis

There is a really resourceful set of analytics methods that correlates data in order to discover causality relationship, or events association. However, it is possible to think in three types of analytical methods which are useful for Cloud Computing security:

diagnostic the method means to synthesize a temporal flow of events arising from sensors in a *security state* of the cloud – it is common represent the state as a dashboard;

root-cause the goal of this type of analysis is to determine what events are the main causes of the actual cloud state;

prediction the prediction methods aims to suggest forecast projections to cloud state.

It is possible to consider that the analysis phase in Cloud Computing security management must have some characteristics:

- there must exist evaluation methods able to supply a set of security metrics for parts and for the whole cloud;
- it must consider the uncertain – uncertainty of the diagnostics provided by analytics methods must carry some fuzzy or probabilistic measure to represent uncertainty;
- it must consider temporality – generally based on time series;
- it must be multi-criteria – may there exists multiples, seemingly uncorrelated, events that articulated, constitute an attack;
- it must be real-time – an fundamental characteristic of the events in Cloud Computing is the need to provide real-time evaluations of the cloud state;
- it must learn – the measures in a real world Cloud changes their statistical distribution, variance and behavior – in this context, an analytical method to security in Cloud Computing must be adaptive to follow this changes.

In this way, our proposal to proceed with the diagnostic is composed by two steps:

- a security state set, where each machine $m \in M$ have a vector $Al_m = (al_1, t_1, al_2, t_2, \cdots, al_n, t_n)$ that contains all security alarms coming from IDs sensors al_x and the time t_x, since the alarm was triggered. The set of all possible cloud security states can be obtained by the product of arbitrary sets $(Al_m)_{m \in M}$, such that $S = \prod_{m \in M} Al_m$, where M is the set of all cloud machines (VMs and PMs);
- two utility functions, to evaluate the security value of machines , and to evaluate the total cloud state, respectively (1) and (2), given:

$$u(m) = -\sum_{k=1}^{n_m} w(al_k) \cdot (t_{now} - t_k) \tag{1}$$

, where n_m is the number of alarms triggered to machine m, $w(al_k)$ is the weight of alarm k, and $(t_{now} - t_k)$ is the amount of time that the alarm k is triggered.

$$\sum_{m \in M} w_m \cdot u(m) \tag{2}$$

, where w_m is the weight that each machine m have in the cloud security, and $u(m)$ is the security evaluation of m.

The root-cause analysis will not be addressed in this work. But, it may be important to correlate and determine the *what* and *how* of some configuration states (e.g. a blocked ip address in the firewall) influence the occurrence of security incidents. In this way, a sensor component that reads the data from logs, IDS agents, VM and Hypervisor [21] data collectors, network traffic sniffers, SNMP agents and alarms. This analysis will be important to determine and discover possible security actions.

The prediction will be important to establish the consequences of an action $a \in A$ execution, where A is the set of all possible actions, over a state $s \in S$. So, the prediction of action consequences must provide a probability function $p(s^{t+1}|a, s^t)$, read as: the probability of action a, executed over a state s^t in time t conduce to a state s^{t+1} in time $t+1$.

D. Planning

In the planning phase, we will use a simple utility maximization method. However, it is interesting to study the Markov Decision Process (MDP) mathematical framework. It will supply a interesting set of elements to guide our decision function.

MDP is a framework generally described in the follow way:

- a set S of system states – here, product of the diagnostic method of analysis phase;
- a set A of possible actions to be taken in the system;
- a probability transition function $P : S \times A \times S \to \mathbb{R}$ that express the probability of the system in state s, given an action a be conduced to a state s' – here, the probability function will be product of the forecasts provided analysis method;
- a reward function $R : S \times A \times S \to \mathbb{R}$ that evaluate the reward of take the action a in state s and conduce the system to state s'.

There exists another ways to describe an MDP, however, this is the most useful to our objectives, that are use MDP for Cloud Computing security management.

It is important to observe that MDP is known as it does not work under incomplete information systems. To use this approach, we consider that:

1) Cloud Computing security monitoring will provide a big data environment that can supply the information needed by MDP;
2) the set of possible states will be finite and treatable;
3) there is an enough number of analytical methods to supply the forecasting needs to support probability function;
4) there are an enough number of analytical methods to supply the needs of state evaluation to support the reward function;

So, considering that there is an utility and a probability function to evaluate the current state, predict the future state and after execute an action, evaluating the future actions, we will propose to select the action with the maximum reward function since it runs in state s^t (3).

$$r(a|s^t) = \Big(\sum_{s^{t+1} \in S} p(s^{t+1}|a, s^t) \cdot u(s^{t+1})\Big) - u(s^t) \quad (3)$$

The reward function establish the difference between the utility of the each possible future state $p(s^{t+1}|a, s^t) \cdot u(s^{t+1})$ and the utility of the current state $u(s^t)$.

So to choose an action, the function must be (4).

$$a = \arg\max_{a \in A} r(a|s^t) \quad (4)$$

E. Executor

With the calculation of the response with the highest expected utility, it is possible to forward the response to an executing agent in the cloud. The hypervisor is responsible for executing the response getting transparence for each virtual machine.

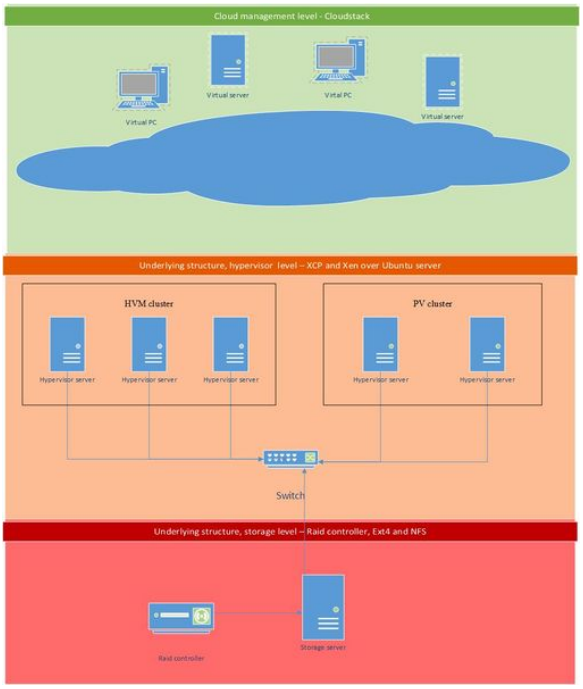

Fig. 3. Cloud environment for validation

F. Discussion

As shown on table 1, our work presents an increment in art when you use Big Data to locate attack occurrences and be able to provide a response that takes into consideration the impacts of the attack across the Cloud environment. Regarding the authors Wu et al. (2013), Vollmer et al. (2013) and Idss et al. (2012) the contribution of our research was to consider the environment Cloud and its peculiarities as the hypervisor, the complexity of providing an answer without being invasive to customers. Our work also considers self-healing and uses statistical function in expected utility to achieve the most efficient response and thereby, block the attacks.

V. Conclusion

This paper suggests the use of autonomic computing to provide response to attacks on cloud computing environments. Thus, it is possible to provide self-awareness, self-configuration and self-healing in the cloud. An architecture that uses the expected utility function for choosing an appropriate response is a statistical model to adjust the answers given in order to provide more results. Furthermore, the work proposes the use of Big Data infrastructure using Hadoop to organize the large volume of data and extract information using the Map- Reduce framework. Thus, we could provide intrusion detection, response and self-healing in cloud environment.

References

[1] K. Vieira, A. Schulter, C. Westphall, and C. M. Westphall, "Intrusion detection for grid and cloud computing," *It Professional*, vol. 12, no. 4, pp. 38–43, 2010.

[2] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security*, vol. 1, no. 1, pp. 169–184, 2007.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, pp. 42–57, Jan. 2013.

[4] K. Lumpur, "An investigation and survey of response options for Intrusion Response Systems (IRSs)," 2010.

[5] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," *NIST special publication*, vol. 800, no. 145, p. 7, 2011.

[6] C. A. Carver, "Intrusion response systems: A survey," *Department of Computer Science, Texas A&M University, College Station, TX*, pp. 77843–3112, 2000.

[7] R. Buyya, R. Calheiros, and X. Li, "Autonomic Cloud computing: Open challenges and architectural elements," *Emerging Applications of ...*, pp. 3–10, Nov. 2012.

[8] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41–50, Jan. 2003.

[9] M. C. Huebscher and J. A. McCann, "A survey of autonomic computingdegrees, models, and applications," *ACM Computing Surveys (CSUR)*, vol. 40, no. 3, p. 7, 2008.

[10] S. Hariri, B. Khargharia, H. Chen, J. Yang, Y. Zhang, M. Parashar, and H. Liu, "The autonomic computing paradigm," *Cluster Computing*, vol. 9, no. 1, pp. 5–17, 2006.

[11] Q. Wu, X. Zhang, R. Zheng, and M. Zhang, "An Autonomic Intrusion Detection Model with Multi-Attribute Auction Mechanism," vol. 10, no. 1, pp. 56–61, 2013.

[12] H. A. Kholidy, A. Erradi, S. Abdelwahed, and F. Baiardi, "Ha-cids: A hierarchical and autonomous ids for cloud systems," in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, pp. 179–184, IEEE, 2013.

[13] C. Chatfield, "The holt-winters forecasting procedure," *Applied Statistics*, pp. 264–279, 1978.

[14] H. Kholidy, A. Erradi, S. Abdelwahed, and F. Baiardi, "A hierarchical, autonomous, and forecasting cloud IDS," pp. 213–220, 2013.

[15] D. Vollmer, M. Manic, and O. Linda, "Autonomic Intelligent Cyber Sensor to Support Industrial Control Network Awareness," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2013.

[16] A.-b. Idss, S. S. H. Case, A. Sperotto, M. Mandjes, R. Sadre, P.-t. D. Boer, A. Pras, and P.-T. de Boer, "Autonomic Parameter Tuning of Anomaly-Based IDSs: an SSH Case Study," *IEEE Transactions on Network and Service Management*, vol. 9, pp. 128–141, June 2012.

[17] R. F. Bordley and S. M. Pollock, "A decision-analytic approach to reliability-based design optimization," *Operations research*, vol. 57, no. 5, pp. 1262–1270, 2009.

[18] S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," in *Big Data Analytics workshop, in conjunction with ACM Sigmetrics*, 2013.

[19] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," May 2011.

[20] P. Zikopoulos, C. Eaton, *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media, 2011.

[21] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42–57, 2013.