A Machine Learning Based Intrusion Impact Analysis Scheme for Clouds

JUNAID ARSHAD*, IMRAN ALI JOKHIO**, AND PAUL TOWNEND***

RECEIVED ON 16.08.2011 ACCEPTED ON 01.12.2011

ABSTRACT

Clouds represent a major paradigm shift, inspiring the contemporary approach to computing. They present fascinating opportunities to address dynamic user requirements with the provision of on demand expandable computing infrastructures. However, Clouds introduce novel security challenges which need to be addressed to facilitate widespread adoption. This paper is focused on one such challenge - intrusion impact analysis. In particular, we highlight the significance of intrusion impact analysis for the overall security of Clouds. Additionally, we present a machine learning based scheme to address this challenge in accordance with the specific requirements of Clouds for intrusion impact analysis. We also present rigorous evaluation performed to assess the effectiveness and feasibility of the proposed method to address this challenge for Clouds. The evaluation results demonstrate high degree of effectiveness to correctly determine the impact of an intrusion along with significant reduction with respect to the intrusion response time.

Key Words: Cloud computing, Cloud security, Intrusion severity analysis, Intrusion Detection, Intrusion Response.

1. INTRODUCTION

The advent of Internet technologies such as: SOAs (Service Oriented Architectures) has considerably influenced the methods used in e-Science. In order to facilitate e-Science research, the role of SOA's is even more with the appearance of new computing ways and paradigms. Beside others, Cloud computing is one of the rising paradigms that exploits contemporaneous virtual machine technology. The collaboration of Internet and virtual machines allows Cloud computing to materialize as a potential candidate paradigm to development of highly scalable and flexible computing infrastructures. In order to meet the requirements of computational hungry applications such as e-Science application, fast growing online applications etc. these infrastructures can be available in an on-demand fashion. VM (Virtual Machine) technology [1] has important role in this, as it enables a single physical machine to host multiple computing environments. These multiple computing environments can also be created, migrated, and deleted in real-time.

Cloud computing is defined and explained in various ways by various sources however, in the context of research presented in this paper, we have defined Clouds as: "a high performance computing infrastructure based on system virtual machines to provide on-demand

* Ph.D. Student, School of Computing, University of Leeds, Leeds, UK.
** Assistant Professor, Department of Software Engineering, Mehran University of Engineering & Technology, Jamshoro.
*** Senior Research Associate, School of Computing, University of Leeds, Leeds, UK.

resource provision according to the service level agreements established between a consumer and a resource provider".

A Cloud computing setup representing the above definition has been presented in Fig. 1. System virtual machines, as per this definition, are the primary units for the apprehension of a Cloud infrastructure and to emulate a discrete and operating system environment independently. In the scope of this paper, the Cloud based platforms persist to fulfilling computation needs of exhaustive compute workloads, and have been defined as Compute Clouds. Whereas those aimed at large scale data storage as Storage or Data Clouds. In the rest of this paper, Cloud computing and Clouds terms are used interchangeably to refer to our definition of compute Clouds.

As with any other emerging paradigm, different models of Cloud computing have been proposed to harvest its benefits. These are IaaS (Infrastructure as a Service), SaaS (Software as a Service) and PaaS (Platform as a Service) [2]. With regards to these models, the Cloud computing system defined earlier and illustrated in Fig. 1 resembles IaaS and therefore, inherits the characteristics of this model of Clouds. From the definition of Cloud computing presented earlier, the term Cloud computing has been used to refer to IaaS model of Cloud computing for the rest of this paper.

However, security underpins extensive adoption of Clouds as illustrated by [3-4]. In particular, Clouds introduce novel challenges with respect to security such as access control, privacy, and availability [5-7] that need keen efforts to take up them. In relation with this, our research is focused

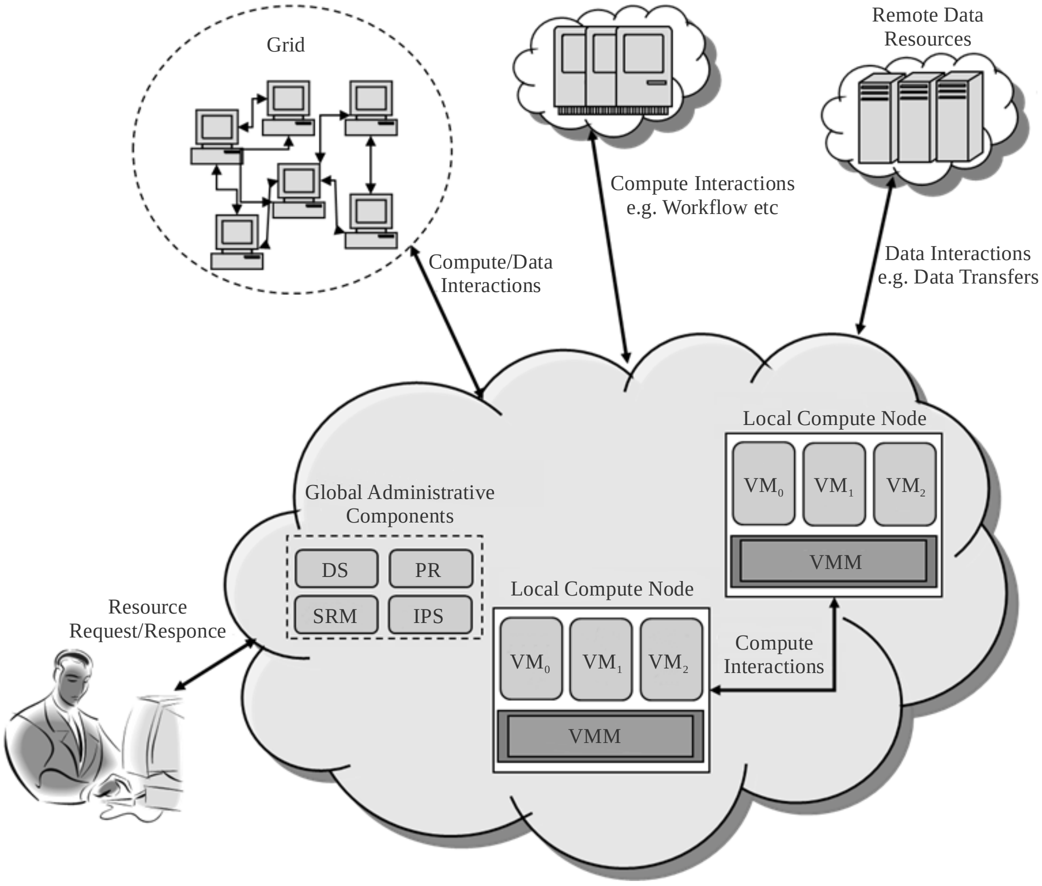
Grid
Remote Data Resources
Compute Interactions e.g. Workflow etc
Compute/Data Interactions
Data Interactions e.g. Data Transfers
Local Compute Node
VM0
VM1
VM2
VMM
Global Administrative Components
DS
PR
SRM
IPS
Local Compute Node
VM0
VM1
VM2
VMM
Compute Interactions
Resource Request/Responce

FIG. 1. A CLOUD COMPUTING SYSTEM

at investigating novel security challenges for Clouds and to devise solutions to address these challenges effectively. This paper is focused at one such challenge i.e. intrusion impact analysis which is paramount to achieving effective response to an intrusion. It summarizes our efforts to address this challenge for Clouds in the form of a machine learning based scheme. The scheme enables VM-specific intrusion impact analysis, achieves protection against zero-day attacks whilst significantly reducing the overall intrusion response time. It also presents results from the evaluation of the proposed scheme demonstrating effectiveness of the scheme to correctly determine the impact of an intrusion along with significant reduction in the overall intrusion response time.

This paper is organised as follows. The Section 2 describes the problem and the state-of-the-art related to it. This is followed by a description of system model for the proposed system in Sections 3. Section 4 describes the proposed methodology to address the intrusion impact problem. This is followed by detailed explanation of the various aspects of evaluation including the experimentation and the respective results. The paper concludes with a mention of the conclusions and potential future work in section 6.

2. PROBLEM DEFINITION AND RELATED WORK

The research presented in this paper is related to impact analysis of intrusions in general and for Cloud computing in particular. Furthermore, similarities can be held with traditional network based systems. Therefore, existing literature in these domains has been explored to draw a comparative analysis of the proposed approach with contemporary approaches.

With respect to Cloud computing, to the best of our knowledge, we are the first to identify the intrusion impact analysis problem for such systems. However, there have been efforts in traditional systems to address this problem. An obvious example of such systems is the NIDS (Network Intrusion Detection Systems) where an intrusion detection system is usually deployed at a border node to look after a whole network of computers. As part of the network, different sub-domains or clusters can have varying security requirements. Existing research suggests that the problem of potential varying impact of an intrusion for such systems is addressed by defining customized security policies for such groups of nodes. However, there are certain defining differences between such systems and the virtual machine based systems such as Clouds. Firstly, the policies for traditional network based systems tend to be static, largely due to the static nature of the monitored systems. This is because the groups of nodes tend to have stable security requirements which have been established overtime based on experience with such systems. However, with Clouds, the monitored virtual machines are added and removed dynamically. Furthermore, the security requirements of individual virtual machines are also envisaged to be diverse, aggravating the problem. Secondly, the security policies in traditional systems are designed and managed by a system security administrator, with some input from the users, who is responsible for the security of the whole system. This human intervention can become the weak link to realize customization and on-demand operation offered by Clouds. Additionally, it also affects the intrusion response time thereby affecting the overall security of the system.

With respect to the use of intrusion impact to select optimal response, intrusion response systems [8] use different metrics to achieve this objective. [9-10] represent two such approaches which use a impact metric. However, this metric is assumed to be calculated by a human administrator through an offline analysis at the policy definition stage. In this case, impact is usually calculated based on the administrator's experience and other resources such as CERT (Community Emergency Response Team) [11].

Although, formal methods for evaluation of intrusion impact with respect to applications are presented in [12-13]. But these formal methods are aimed to facilitate a human administrator and are application specific. The impact evaluation method proposed in [12], is function of *Criticality, Lethality, Countermeasures and Net Countermeasures*. The inputs metrics i.e. *Criticality, Lethality, Countermeasures and Net Countermeasures*, to the evaluation function infer from their names that these are subjective and encumber their appropriateness to a user driven system that is flexible and diverse such as Clouds. Furthermore, in the proposed methods the analysis phase is to be carried out by a human that may take it to multifarious problems. Firstly, the metrics in the proposed method are relative hence in order to measure or find the metrics, in-depth or complete working status knowledge of an attacked system, the attack itself, and the system parameters that define the present state of the victim virtual machine. Secondly, only well known intrusions can be anticipated with the metrics that are to be used in the proposed method [12] i.e. Countermeasures and Net Countermeasures. Finally, manual analysis is very cumbersome to response time of the recovery process.

In order to calculate vulnerability impact, a formula with three groups of metric is defined in CVSS (Common Vulnerability Scoring System) [13]. This method is also aimed at manual analysis facilitation of system administration. The manual analysis results in a calculated impact factor of a vulnerability that is assigned to the vulnerability in question. This facilitates the system administration to evaluate the impact of the vulnerability well before its exploitation. On a critical note custom security requirements by the notion of *Environmental Metrics* are not taken into account. These requirements do not participate in the scoring system unless explicitly defined by user to do so. Beside this, the approach [13] has a number of weaknesses and limitations; Firstly, the manual assumption for execution of system wide process. An administrator or representative of a user need to take decisions whilst valuing the impacts of various metrics i.e. Availability, Confidentiality and Integrity. The assigned values to the metrics contribute to analysis and reflect in final impact factor results and metrics. The target system applications or users may have different priorities (values) than what was assigned by the administrators. Secondly, sometimes the metrics are too abstract to clearly specify in the context of an application and this also hampers humans to clearly define or specify different metrics for various applications. For example, the impact of availability, integrity and confidentiality is defined with three levels i.e. *none, partial or complete*. These three levels of impact are overly abstract and vague for an exact expression of the impact. It is also too difficult to model impact of a specific security attribute with these vague terms of impact analysis. Therefore, it has been concluded that the impact analysis is to be carried with the comprehension of user needs and requirements for a more fine-grained analysis.

As compared to the approaches described above, the approach proposed in this paper is fundamentally different in that it is envisaged to form a part of integrated intrusion detection and impact analysis system explained in [14]. However, this paper is focused at description of a specific component within the system explained in [14] i.e. the intrusion impact analysis component. Due to this, our approach also takes into account the effects of this analysis on the intrusion response time. Therefore, an attempt has been made to minimize disruption in the normal execution of system as part of our approach. Furthermore, our approach is envisaged to incorporate user input via SLAs (Service Level Agreements) which enables our method to be VM specific. As this user input is restricted to the resource acquisition phase, it is considered that impact analysis process does not require any human intervention, whereas, the above explained approaches are agnostic of this fact and therefore, are exposed to the limitations described earlier in this section. In addition to these approaches, there are other approaches for alert correlation. However, they are focused on probabilistic models for predicting the likelihood of a successful intrusion by aggregating alerts from different sources of intrusion detection [15].

3. SYSTEM MODEL

In order to facilitate understanding, Fig. 2 presents a lower level model of the proposed system. As illustrated in Fig. 2, the proposed system is envisaged to reside in the domain 0, the most privileged VM, of a virtualized resource in a Cloud. Therefore, in a Cloud infrastructure, each virtualized resource is envisioned to implement this system as part of other local administrative modules in the domain 0. By developing the proposed system as part of domain 0, the system can assume maximum isolation from the monitored VMs [16]. Furthermore, the visibility of activities performed by monitored VMs is enhanced to the system call level which encapsulates all the activities performed by a guest VM. This also has significant implications with respect to the type of intrusions handled by the proposed system. The proposed system is envisaged to deal with intrusions which can be detected at the hypervisor level using system calls executed by malicious processes within a VM. A detailed fault model has been described in [18].

The proposed system consists of the following components:

□ A module to handle system calls executed by VMs System Call Handler.

□ An intrusion detection system - Detection Engine.

□ A database of known attack signatures - Attack DB.

□ An intrusion impact analysis module - Impact Analysis Module.

□ A module to create and manage VM profiles - Profile Engine.

The Hypervisor, as described in Fig. 2, represents the virtualizing software and facilitates the function of multiple independent VMs within a single physical machine as described by [16]. A SCH (System Call Handler) is envisaged to receive system calls executed by monitored VMs via the system call interface and transfers it to the DE (Detection Engine) to detect any possible malicious operations intended to be performed by the system call. The DE can be either an anomaly based or misuse intrusion detection system. In case of a misuse intrusion detection system, the DE can consult the Attack DB for signatures of known attacks. Otherwise, the DE can consult a PE (Profile Engine) to perform anomaly based intrusion detection. The PE is envisioned to create and manage security profiles for monitored virtual machines and can consult external components such as a global resource manager for the cloud. These profiles include security characteristics of a virtual machine along with other attributes. In the event that a system call is identified as malicious by the DE, the system call is transferred to the IAM (Impact Analysis Module) to evaluate impact of the intrusion identified by the DE. As with the DE, the IAM can also consult with a PE to obtain VM specific information such as security characteristics of the victim VM. Finally, IAM is envisioned to evaluate the impact of an intrusion for a malicious event and communicate with an IRS (Intrusion Response System) to convey the result of impact analysis. The intrusion response process is, therefore, completed by the IRS by executing a response in accordance with the impact analysis.

4. CASE FOR THE USE OF CLASSIFICATION TECHNIQUES FOR INTRUSION IMPACT ANALYSIS

Machine learning based classification techniques have been used in contemporary systems to achieve objectives such as effective intrusion detection as demonstrated by [25-26]. Within this context, intrusion detection problem can be generalized as a classification problem where a classifier has to classify a given event into two classes i.e. normal or malicious. The approaches discussed in [25-26] highlight the effectiveness of machine learning based classification approaches to achieve this objective. These schemes and the results from the evaluation of these efforts provide the motivation to adopt machine learning based classification techniques to achieve the objectives of this research.

In relation to this, the problem of intrusion impact analysis can be generalized as a classification problem. At a particular instance in time t, consider an application Z_i which represents a workload hosted within a VM and VM_i, where $Z=\{Z_1,Z_2,Z_3,\ldots,Z_n\}$ represents a set of applications hosted within a VM, and $VV=\{VM_1,VM_2,VM_3,\ldots,VM_k\}$ represents the set of VMs within a Cloud. Clearly, $Z_i \in Z$, and $VM_i \in VV$. Now, if X_i represents the security characteristics of an application Z_i where $XX=\{X_1,X_2,X_3,\ldots,X_n\}$ represents the set of security characteristics for Z, $X_i \in XX$. Also, if M_i represents an intrusion where $M=\{M_1,M_2,M_3,\ldots,M_l\}$ represents a set of potential intrusions for a particular application Z_i where l is an exponentially large number, the impact I of the intrusion M_i on the application Z_i can be defined as a function of M_i and the security characteristics X_i of victim application Z_i i.e. $I=f(M_i,X_i)$.

Now, consider a dataset D containing a record of intrusions on the victim application. Each element d_i in this dataset represents a malicious event M_i which can have a different degree of impact for the victim application. Also, if $C=\{c_1,c_2,c_3,\ldots,c_p\}$ represents a set of classes or levels of impact, then the impact of the intrusion M_i on the victim machine VM_i can be mapped to one of the element of C. Within this context, the output for function I can be defined as under:

$$I = f(M_i,X_i)\{:c_i\}$$

where c_i is an entity in set C representing possible levels of impact.

From the above definitions, the impact analysis of an intrusion results into a classification comprising different levels of impact as represented by C. These levels of impact can be regarded as different classes whereby a class is distinguished by values of different attributes such as; intrusion, state of the SLA, and security characteristics of the victim VM. The argument described above leads to the conclusion that intrusion impact

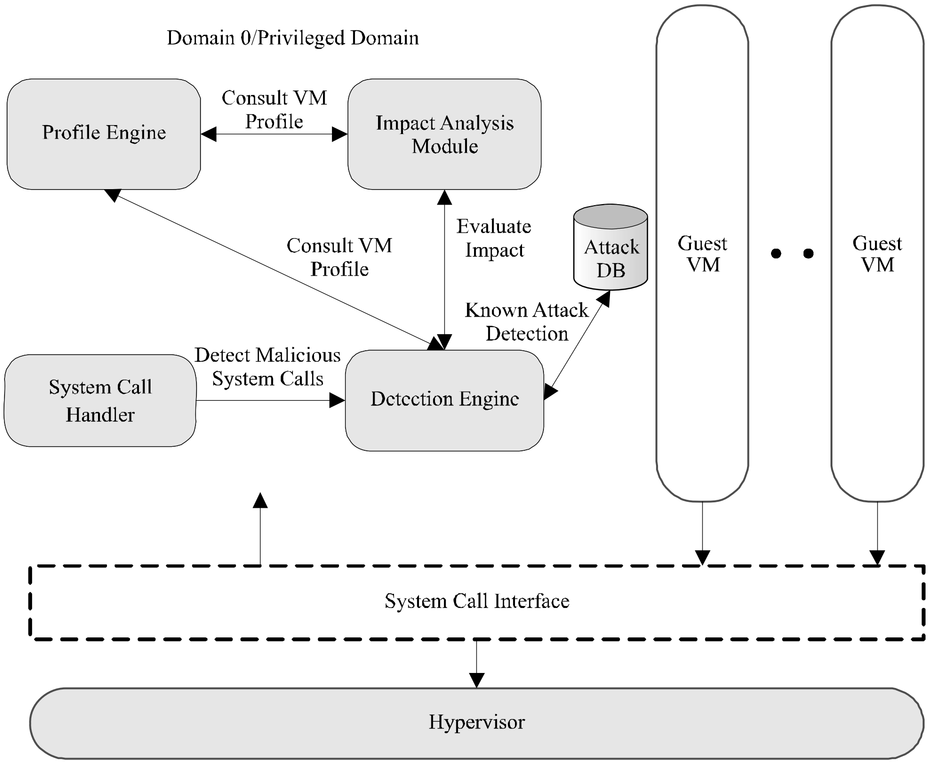

FIG. 2. MODEL FOR THE PROPOSED SYSTEM

analysis problem can be generalized as a classification problem. However, it should be acknowledged that it does require appropriate definition of the levels or classes of impact. Furthermore, as the impact of an intrusion depends on a number of factors described earlier, it increases the complexity of the process and therefore, makes it extremely difficult to evaluate impact for all known intrusions against all possible application types. Machine learning based classification techniques present an opportunity to address this issue efficiently. Therefore, machine learning based classification techniques have been used as part of the proposed scheme to perform intrusion impact analysis.

5. METHODOLOGY

An intrusion can be the result of exploitation of VM vulnerability. Impact of an intrusion can be driven by various parameters. Therefore it is useful to explain the assumptions of this research. The proposed scheme in this paper is established on the belief that the impact of an intrusion on a virtual machine rely on various factors including; security requirements of an application hosted on a virtual machine, state of a negotiated SLA beforehand, and attack frequency on a victim VM. However, there can be more parameters to analyse impact of an intrusion, but it is assumed that these are profound aspects of a VM.

Security requirements and related policy definitions can be included in a guest VM. Management of these requirements and policy definitions can also be handed over to the target VM itself. In order to achieve this, the target VM may have its own policy engine to coordinate with a privileged VM having impact analysis and detection components. This sort of design is appealing due to its simplicity and easiness while implementing the system of VMs. However, in this model, VM its security requirements, policy definitions and management are not isolated. In case of an attack, the attacker has an advantage to temper the policy definitions and launch attacks for its malicious activities. These malicious activities may look as legal actions of the system as the security policy definitions have been modified. The design is contradictory and also supports our assumption i.e. a VM cannot be trusted to delegate a responsibility of protecting policy definitions and management rather all guest VMs may be treated as malicious. While considering these limitations, the simple and easy to implement design may not be applicable to calculate impact of intrusions. Therefore we adopted a design that may have flexibility to customize security definitions and policies whilst providing complete isolation of policies and their management from the guest virtual machines. In the adopted design the security requirements are proposed negotiation with respect to an appropriate SLA in place. This approach of the design provides ability to a user to list and define the SLA related security requirements in resource acquisition process. In order to specify the security requirements of SLA during resource acquisition process, quantification of the security is need. This has been summarized in our previous work [17].

A SLA states play a vital role in building and maintaining trust of a system. Although, our work is not focused to maintain or build trust of the system, but it is useful to define SLA states to calculate the impact factor of an intrusion. Response time of a workload beside other factors is an important entity and the time while executing a business process or workload, is treated as a state of SLA. In the impact analysis the response time of a workload in execution transforms into remaining time to complete the workload. Time to complete the workload has been defined as SLA state because to our understanding the available time in a VM is significantly affected with an intrusion. Preferably, a SLA state is derived from a number of parameters including metrics of quality of service, available resource etc. In order to include all these parameters in SLA state, a full fledge monitoring system is required to establish, that is why this has been excluded from the scope of this research. However, to avoid this limitation an aggregated metric depicting a SLA state is adopted to analyse the impact formally. Finally, an attack attempt occurrences i.e. frequency that are targeted on a specific security requirement highlights either high value assets are under attack or it is likelihood that a successful attack will soon be launched on the security requirement. Hence, an efficient and timely mechanism is need to prevent the high frequency attacks and very same reason holds behind the designation of attack frequency as one of the prime factors for impact of an intrusion analysis.

As stated earlier, it has been proposed to resolve the impact problem by modelling it as a classification or taxonomical problem. In this context, techniques of machine learning are applied to the creation of taxonomy or to perform the classification. For an offline analysis, unsupervised learning techniques are preferred due the variance over the length of datasets in examination. This property of the unsupervised learning techniques leads to unsuitability for systems that need real-time or concurrent classification. The problem we try to address in this paper is of the real-time classification type so unsupervised learning techniques may not be a candidate solution to it. In supervised learning techniques a learning process or initial training is required to be present for the rest of learning and classification process. But the research problem explained here in this paper does not have any historical data regarding the intrusion impacts over virtual machines neither it can be maintained for a real-time process. Therefore, it is very difficult to apply supervised learning techniques to solve the problem. In order to mitigate this challenge, a dynamic weighted scheme was developed to facilitate training phase for the selected supervised classification technique i.e. decision trees. The details of this scheme have been presented in [18].

6. EXPERIMENTATION AND EVALUATION

As discussed by [19], the effectiveness of an intrusion response is characterised by two factors i.e. (i) the response should be in accordance with the impact of an attack, and (ii) the response should be executed with minimum possible delay. Within this context, the evaluation of the proposed scheme is focused assessment of the scheme against the criteria proposed by [19]. Specifically, the overall evaluation of the proposed system comprised of experiments to evaluate the effectiveness of the intrusion impact analysis scheme with respect to (i) correctly determine the impact of an intrusion and (ii) overall intrusion response time. The overall objective of these experiments is to investigate the effectiveness and feasibility of machine learning techniques for intrusion impact analysis for Clouds. The experiments were conducted on a general purpose workstation with Intel Core 2 Quad CPU 2.83GHz and 4.0GB memory. The machine learning algorithm used for these experiments was C4.5 [20]. The choice of this algorithm is motivated by its splendid success to perform effective intrusion detection as demonstrated by [25-26]. However, the software used to perform these experiments was the Weka machine learning software [21]. Weka is a machine learning software platform which provides the ability to perform rigorous experiments using various machine learning techniques. The software has a very rich library with respect to machine learning techniques and provides variety of tools to support both graphical user interface and programmable API (Application Programming Interface).

In order to evaluate the performance of the proposed impact analysis scheme, two types of experiments were conducted i.e. intrusion impact analysis experiments and experiments for intrusion response time. For these experiments, random subsets of data produced as part of the process described in [18] were used. The two types of experiments performed are described below along with their respective results.

6.1 Experiments for Intrusion Impact Analysis

In order to assess the proposed scheme, a set of experiments were performed using different training and test datasets. Specifically, each round of the experiments involved four different datasets randomly generated using the process described in [21]. Additionally, the elements of these datasets were labelled with appropriate classes using the dynamic weighted scheme described in [21]. For each of these four subsets of data, one subset was used as training data to build a classifier. After the training phase, the classifier was tested against the remaining three subsets of data. This ensures that the test data is essentially unseen for the classifier and is different from the training data. Furthermore, these experiments were repeated four times (four rounds) and results were averaged to preserve the objectivity of the experimental results. This effectively means that the experiments were conducted sixteen different randomly generated datasets and results were averaged.

Table 1 presents the results of these experiments where each row in this table represents one round of experiments and each round of experiments is conducted with four different datasets as described above. The entries in this table present the percentage of correctly classified elements from for each experiment. Furthermore, Fig. 3 presents the results of these experiments with dataset 1 as the training dataset and rest as the test datasets.

As can be seen from both the Table 1 and Fig. 3, the results present a very encouraging picture. Although these results present lower success rates than the results of cross validation, still the lowest average success rate is 84.03% which is very promising with respect to the feasibility of using machine learning based classification techniques for the impact evaluation problem. Furthermore, Fig. 4 presents the average success rates for these experiments. As can be seen from this figure, the lowest average success rate is 84.03% whereas most of the remaining results have success rates higher than 90%. This is reflects a high degree of effectiveness of the scheme and therefore, demonstrates the effectiveness of the proposed approach for intrusion impact analysis for Clouds in general and feasibility of using machine learning based classification approaches for problem addressed in this paper in particular.

TABLE 1. AVERAGED PERCENTAGE SUCCESS RATES OF EVALUATION WITH DIFFERENT TEST AND TRAINING SETS

Training Datasets	Dataset-1	Dataset-2	Dataset-3	Dataset-4
Dataset-1	X	89.959	86.824	92.176
Dataset-2	90.578	X	88.912	94.665
Dataset-3	91.171	84.031	X	90.440
Dataset-4	94.273	94.275	92.242	X

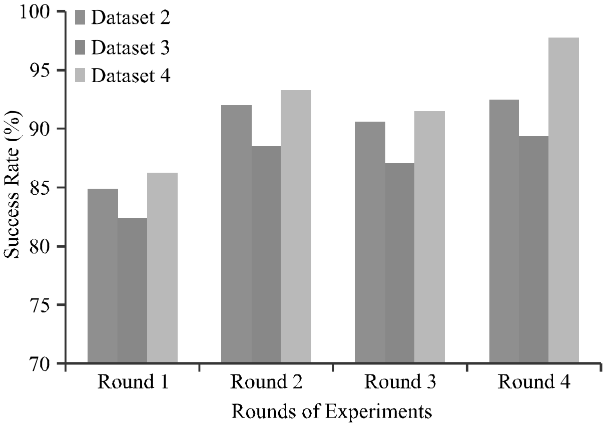

FIG. 3. RESULTS OF EXPERIMENTS WITH DATASET 1 AS TRAINING AND REST AS TEST SETS

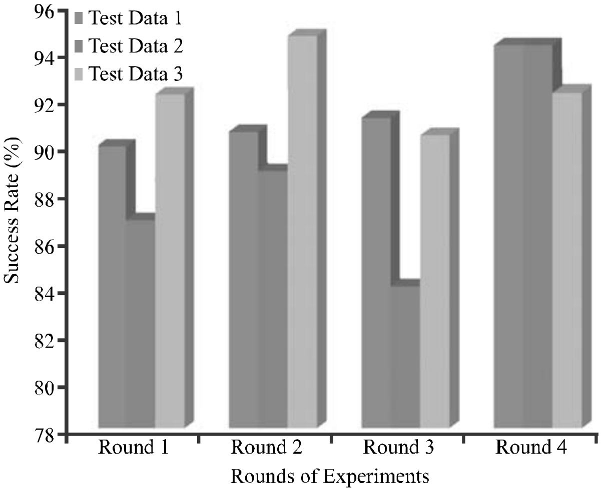

FIG. 4. AVERAGE SUCCESS RATES FOR ALL ROUNDS

6.2 Experiments for Intrusion Impact Analysis Time

For the purpose of the evaluation with respect to impact analysis time, the experimental data is divided into a number of sub-datasets randomly to achieve rigorous evaluation. Additionally, in order to assess the impact analysis time for the proposed method, the time required by a classifier to evaluate severity of one malicious event is calculated using Java's built-in functionalities. This time is then averaged over the entire population of each dataset and therefore represents average impact analysis time for a classifier. These experiments are repeated four times (each represented as a round in Fig. 5) to take into account the repeatability of the experiments. The results of these experiments along with the average impact analysis times for respective rounds of experiments have been presented in Fig. 5 and Table 2.

As evident from Table 2, the average impact analysis time for all the experiments is less than 2µs with a best of 1.463µs which represents an exceptional improvement as compared to the 7seconds required by a highly experienced human administrator [22]. Furthermore, pH [23] represents an effort to develop an automated intrusion response system by simply delaying or aborting abnormal system calls. Although pH only delays or aborts system calls without doing any further processing, it is reported to incur average time overhead of 4.7µs. In comparison with this overhead, the best average impact analysis time from Fig. 5 i.e. 1.463µs represents significant improvement. This improvement is further enhanced considering the fact that the time consumed in pH is for delaying system calls whereas the time consumed in this research is to perform rigorous intrusion impact analysis thereby facilitating rigorous intrusion response. This demonstrates the effectiveness of the proposed method to facilitate effective intrusion response with reduced overall intrusion response time.

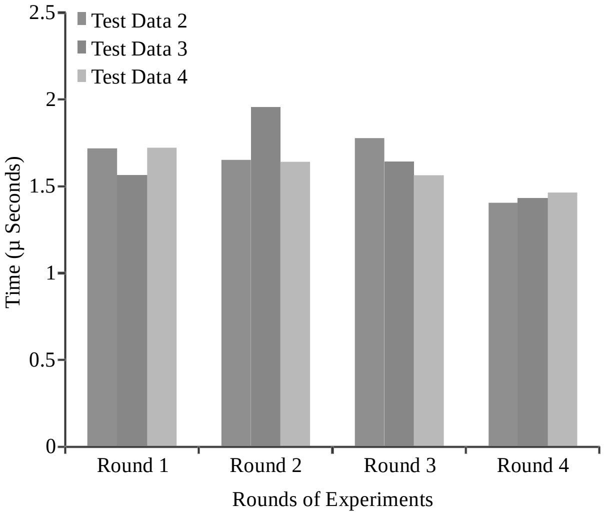

FIG. 5. IMPACT ANALYSIS TIME IN MICROSECONDS FOR PROPOSED INTRUSION IMPACT ANALYSIS METHOD

The overall intrusion response is composed of (i) the intrusion detection time, (ii) the impact analysis time, and (iii) response execution time [22]. Within this context, the scheme proposed in this paper is focused at intrusion impact analysis and therefore envisages to achieve significant reduction in the overall intrusion response time by reducing the impact analysis time. Therefore, the experiments conducted with respect to intrusion impact analysis time represent an effort to evaluate the contribution of the proposed scheme towards the overall intrusion response time. The results presented in Table 2 show the significant reduction in the impact analysis time achieved by the proposed scheme via the use of machine learning technique and remark a significant improvement as compared to the traditional approaches. This clearly

TABLE 2. AVERAGE IMPACT ANALYSIS TIMES IN MICROSECONDS FOR THE PROPOSED METHOD

Training Datasets	Dataset-1	Dataset-2	Dataset-3	Dataset-4
Dataset-1	X	1.638	1.648	1.598
Dataset-2	1.619	X	1.685	1.545
Dataset-3	1.696	1.793	X	1.775
Dataset-4	1.487	1.463	1.754	X
Average	1.600	1.632	1.697	1.639

indicates the contribution of the impact analysis scheme towards significant reduction in the overall intrusion response time. However, the performance of the scheme with respect to the overall intrusion response time requires an implementation of the integrated system presented in [14]. The efforts in this regard are in progress and are an integral part of our future work.

7. CONCLUSIONS

Cloud computing has emerged as a novel paradigm to revolutionize the contemporary approach to computing. However, the extensive adoption of Clouds is threatened by the security challenges faced by it. This paper is focused at one such challenge i.e. intrusion impact analysis which is paramount to achieving effective intrusion response. The paper presents a novel method to address this problem for Clouds in the form of a context-aware scheme which is capable of evaluating impact of an intrusion in accordance with the security characteristics of the victim VM. The paper also presents rigorous evaluation of the method to assess the effectiveness of the scheme to achieve precise intrusion impact analysis. The results of this evaluation show high degree of effectiveness of the method to correctly determine the impact of an intrusion for a victim VM. Furthermore, the experiments also demonstrate the effectiveness and feasibility of the method to significantly reduce the overall intrusion response time. As part of current and future activities, the implementation of the proposed method is in progress with iVIC [24], a real Cloud system.

ACKNOWLEDGEMENTS

Authors are thankful to Mehran University of Engineering & Technology, Jamshoro, Pakistan, Azad Jammu & Kashmir University, and HEC, Pakistan, for support in providing funding to conduct this research work. Authors are also thankful to School of Computing, University of Leeds, facilitating the required infrastructure to conduct this research work.

REFERENCES

[1] Goldberg, R.P., "A Survey of Virtual Machine Research", IEEE Computer, Volume 7, pp. 34-45, 1974.

[2] Mell, P., and Grance, T., "A NIST National Definition of Cloud Computing", available at: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

[3] IT Cloud Services User Survey, Part-2: Top Benefits and Challenges. Available online at: http://blogs.idc.com/ie/?p=210, October 2008.

[4] New IDC IT Cloud Services Survey: Top Benefits and Challenges. Available online at: http://blogs.idc.com/ie/?p=730, December, 2009.

[5] Pearson, S., "Taking Account of Privacy when Designing Cloud Computing Services", Proceedings of CLOUD'09, Vancouver, Canada 978-1-4244-3713-9/09/ May 23, 2009.

[6] Herald, R., "Privacy and Cloud Computing Challenges", Appeared in Infosec, Available online at: https://www.infosecisland.com/blogview/3539-Privacy-and-Cloud-Computing-Challenges.html 16th April 2010.

[7] Dournaee, B., "Taking Control of the Cloud for Your Enterprise", Intel SOA Expressway Cloud Security White Paper, 2010.

[8] Stakhanova, N., Basu, S., and Wong, J., "A Taxonomy of Intrusion Response Systems", International Journal of Information and Security. Inderscience Publishers.

[9] Schnackenberg, D., Holliday, H., Smith, R., et al, "Cooperative Intrusion Traceback and Response Architecture (CITRA)", Proceedings, IEEE DARPA Information Survivability Conference and Exposition (DISCEX I), 2001.

[10] Porras, P., and Neumann, P., "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", Proceedings of the National Information Systems Security Conference, 1997.

[11] Community Emergency Response Team, available at: http:// www.cert.org

[12] Northcutt, S., and Novak, J., "Network Intrusion Detection: An Analyst's Handbook", 3rd edition New Riders Publishing Thousand Oaks, CA, USA ISBN:0735712654

[13] Mell, P., and Scarfone, K., "A Complete Guide to the Common Vulnerability Scoring System" Version 2.0 available at: www.first.org/cvss/cvss-guide.html

[14] Arshad, J., "Integrated Intrusion Detection and Diagnosis for Clouds", Proceedings of Dependable Systems and Networks (DSN), Student Forum 2009.

[15] Porras, P. A., Fong, M. W., and Valdes, A., "A Mission-Impact-Based Approach to INFOSEC Alarm Correlation", Proceedings of RAID pp. 95-114, 2002.

[16] Barham, P., Dragovic, B., Fraser, K., et al, "Xen and the Art of Virtualization", Proceedings of SOSP'03, October 19-22, 2003

[17] Arshad, J., Townend, P., and Xu, J., "Quantification of Security for Compute Intensive workloads in Clouds", Proceedings of the International Conference on Parallel and Distributed Systems (ICPADS), 2009.

[18] Arshad, J., Townend, P., and Xu, J., "A Novel Intrusion Severity Analysis Approach for Clouds", International Journal of Future Generation Computer Systems Special Issue for Clouds, 2011 (To be Published).

[19] Brackney, R., "Cyber-Intrusion Response", Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, West Lafayette, 1998.

[20] Quinlan, J.R., "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993.

[21] Weka-Data Mining with Open Source Machine Learning Software in Java. Available at: http://www.cs.waikato.ac.nz/ml/weka/

[22] Usher, A.T., "The Future of Network Intrusion Detection", The Newsletter for Information Assurance Technology Professionals, Volume 7, 2005.

[23] Somayaji, A., and Forrest, S., "Automated Response Using System Call Delays", Proceedings of the 9th Conference on USENIX Security Symposium, Volume 9, 2000.

[24] Huai, J., Li, Q., and Hu, C., "CIVIC: A Hypervisor Based Computing Environment", Proceedings of the International Conference on Parallel Processing Workshops, pp. 809-820, 2007.

[25] Bouzida, Y., and Cuppens, F., "Neural Networks vs. Decision Trees for Intrusion Detection", Proceedings of IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), Tuebingen, Germany, 28-29 September 2006.

[26] Kang, D-K., Fuller, D., and Honavar, V., "Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls", Proceedings of the IEEE Workshop on Information Assurance and Security, United States Military Academy, West Point, NY, 2005.