An Abstract Model for Integrated Intrusion Detection and Severity Analysis for Clouds

Junaid Arshad, Paul Townend, Jie Xu
sc06ja@leeds.ac.uk
School of Computing, University of Leeds, UK LS2 9JT

ABSTRACT

Cloud computing is an emerging computing paradigm which introduces novel opportunities to establish large scale, flexible computing infrastructures. However, security underpins extensive adoption of Cloud computing. In this paper, we present our efforts to address one of the significant issues with respect to security of Clouds i.e. intrusion detection and severity analysis. In particular, we propose an abstract model for integrated intrusion detection and severity analysis for Clouds. With the proposed model, we envision to facilitate minimal intrusion response time whilst preserving the overall security of the Cloud infrastructures. In order to assess the effectiveness of the proposed model, we present detailed architectural evaluation for this model using the Architectural Trade-off Analysis Model (ATAM). Finally, we also present a set of recommendations which can be used as a set of best practice guidelines whilst implementing our proposed architecture.

Keywords: Cloud Computing, Intrusion Detection, Intrusion Severity Analysis, Architectural Evaluation, Virtual machines.

1. INTRODUCTION

The advent of internet technologies has significantly changed the methods used in e-Science along with the emergence of new computing paradigms to facilitate e-Science research. Cloud computing is one of such emerging paradigms which makes use of the contemporary virtual machine technology. The collaboration between internet and virtual machine technologies enable Cloud computing to emerge as a paradigm with promising prospects to facilitate the development of large scale, flexible computing infrastructures, available on-demand to meet the computational requirements of e-Science applications. Cloud computing has witnessed widespread acceptance mainly due to compelling characteristics such as; Live Migration, Isolation, Customization and Portability, thereby increasing the value attached with such infrastructures. The virtual machine technology has profound role in it. Amazon [1], Google [2] and GoGrid [3] represent some of commercial Cloud computing initiatives whereas Nimbus [4] and OpenNebula [5] represent academic efforts to establish a Cloud.

Cloud computing has been defined in different ways by different sources however, for the purpose of research described in this paper, we define Clouds as a high performance computing infrastructure based on system virtual machines to provide on-demand resource provision according to the service level agreements established between a consumer and a resource provider.

A Cloud computing system representing the above definition has been presented in Figure 1. A system virtual machine, as described in this definition, serves as the fundamental unit for
the realization of a Cloud infrastructure and emulates a complete and independent operating environment. Within the scope of this paper, we define the cloud platforms focused at satisfying computation requirements of compute intensive workloads as Compute Clouds whereas those facilitating large scale data storage as Storage or Data Clouds. For the rest of this paper, we use terms Cloud computing and Clouds interchangeably to refer to our definition of compute clouds. As described in the above definition, Cloud computing involves on-demand provision of virtualized resources based on Service Level Agreements (SLA) [6] thereby facilitating the user to acquire resources at runtime by defining the specifications of the resource required. The user and the resource provider are expected to negotiate the terms and conditions of the resource usage through SLAs so as to protect the quality of service being committed at resource acquisition stage.

As with any other technology, different models of Cloud computing have been proposed to harvest its benefits. These are Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS) [7]. Each of these models is focused at achieving specific objectives by introducing novel mechanisms at respective layers of the modern software architecture [8]. With regards to these models, the Cloud computing system presented in Figure 1 resembles IaaS and therefore inherits the characteristics of this model of Clouds. In the remaining sections of this paper, we use the term Cloud computing to refer to this model of Cloud computing.

However, as with any other emerging paradigm, security underpins extensive adoption of Cloud computing. Specifically, we highlight the importance of intrusion detection and severity analysis for Clouds in this paper. We also summarize our efforts to address this problem whilst taking into account unique characteristics of Clouds. Furthermore, specific requirements of Clouds for intrusion severity analysis have been summarized with a detailed

![Figure 1: A Cloud computing system](image-url)
description provided in [9]. In this paper, we focus on the requirement of minimizing overall response time for an intrusion by proposing an abstract model for integrated intrusion detection and severity analysis for Clouds. We also present the architectural evaluation for the proposed solution with the objective to evaluate the effectiveness of the proposed model. To the best of our knowledge, we believe that we are the first to conduct this research for Clouds.

The rest of paper has been organized as follows. The next section introduces the intrusion detection and severity analysis in general and introduces the challenges for intrusion detection and severity analysis for Clouds. In section 3, we present a summary of the requirements of Clouds for intrusion severity analysis followed by a description of our proposed method for automatic intrusion severity analysis for Clouds. A detailed description of this method has been presented in [10]. We describe our proposed architecture in detail in section 5 whereas section 6 contains a detailed description of the evaluation performed for the proposed model. Section 7 describes the recommendations to foster the proposed architecture followed by conclusions in the last section.

2. INTRUSION DETECTION AND SEVERITY ANALYSIS

Intrusion detection is a well established research domain focused at improving the overall security of a system against malicious users. Historically, an Intrusion Detection System (IDS) strives to facilitate a system administrator by raising an alert whenever it detects an intrusion. Contemporary IDSs can be broadly classified as host or network based with respect to their location. As suggested by their names, a host based intrusion detection system is located on the host being monitored and therefore has the benefit of maximum visibility of the monitored system. However, it has the disadvantage of being prone to getting compromised in the event of a successful intrusion taking control of the monitored system. A network based IDS on the other hand, is usually installed at the edge of a network and has the advantage of being isolated from the monitored system. However, it has the disadvantage of reduced visibility of the monitored system. Leveraging the isolation provided by virtual machines, Hypervisor based IDS have been proposed [11,12] which combines the benefits of both host and network based IDS thereby improving the security of the monitored system.

Furthermore, Clouds inherit unique characteristics such as diversity, mobility and flexibility from virtual machines, which present novel security challenges and, therefore, require dedicated efforts to address them [13]. Among these characteristics, diversity provided by virtual machines introduces challenges for intrusion detection and response systems.

![Figure 2: A virtualized resource](image-url)
As described in figure 2, virtual machines provide the ability to host multiple different execution environments on a single physical machine, which enables a Cloud provider to be able to address diverse user requirements with same physical resources. However, it also poses a number of novel security challenges such as: evaluating the impact of an intrusion on the guest virtual machines [14]. From figure 2, a security module residing in the domain 0 of a virtualized resource has to evaluate the impact of an intrusion on the guest virtual machines. This process becomes non-trivial given the potentially different security requirements of the guest virtual machines and the dynamic and flexible nature of Cloud infrastructures. We define the impact of an intrusion on a virtual machine as the Level of Severity (LoS) of the intrusion. Related to this, we define intrusion severity analysis to be the process to evaluate the level of severity of an intrusion for a monitored virtual machine.

Intrusion severity analysis, as defined above, has significant impact on the overall security of a system such as; selecting appropriate response mechanism and the overall intrusion response time. This will enable provision of an intelligent response selection mechanism, which facilitates triggering response mechanisms based on the severity of an intrusion for the victim application. Furthermore, a customized severity analysis also facilitates delivery of virtual machine specific quality of service with respect to security, which is vital for a user-oriented environment of clouds computing. This is envisioned to enable the cloud providers to devise Quality of Service (QoS) based pricing strategies, taking into account the quality of security service delivered in relation with the security requirements of virtual machines. Finally, the dynamic nature of Clouds demands such mechanisms to minimize intrusion response time. Therefore, in order to preserve the flexible and dynamic nature of Clouds, the efficiency of such system is critical. To the best of our knowledge, we are the first to identify the intrusion severity analysis problem, highlight its importance and propose a solution to address it for Clouds [14].

3. RESEARCH CONTEXT
With respect to intrusion severity analysis, Cloud computing has a number of distinct requirements. These requirements are primarily introduced by the unique characteristics of Clouds presented in [13]. In [9], we provide a detailed description of these requirements along with their comparison with existing solutions however, a summary of these has been presented below.

Comprehensive Severity Evaluation: The ability to host multiple execution environments on a single physical machine governs diversity in the infrastructure, which demands a comprehensive severity evaluation.

Real-time Operation: Clouds support flexible infrastructure where guest virtual machines can be created, migrated and deleted at runtime. This requires a security module to be considerate of the dynamic nature of the infrastructure.

Customization: Due to the diversity inherited by Clouds, each guest virtual machine can potentially have different security characteristics. Furthermore, we hold that security characteristics of an application dictate the severity of an intrusion on that application [15]. Therefore, it is necessary for a severity evaluation approach for clouds to enhance customization to a virtual machine level.

Automatic Approach: With regards to flexibility and dynamic nature of clouds, the human intervention for intrusion severity analysis process needs to be minimized thereby eliciting the requirement for an automated approach is mandatory.
Minimized Response Time: Due to the runtime behaviour of a Cloud computing infrastructure, intrusion response time becomes critical. Specifically, the intrusion response time needs to be minimized to facilitate dynamic and runtime nature of Clouds. It is, therefore, required for a severity evaluation method to be integrated with other security modules such as intrusion detection and response systems.

From the above described requirements, our focus in this paper is on Automatic Approach and Minimized Response Time. In order to fulfil these requirements, a solution to address intrusion severity analysis problem must minimize human intervention throughout the process without compromising the overall security of the infrastructure. However, most of the traditional intrusion detection and response systems are independent and isolated from each other [16]. Typically, an intrusion detection system generates alerts for the attention of a human system administrator. The human administrator is then supposed to consult available resources and manually evaluate the severity of the intrusion of the victim application. This process consumes a substantial amount of time between intrusion detection and response and therefore, compromises the security of systems against intrusions with exponential frequency. This has been demonstrated by Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks [17].

Additionally, one of the objectives of Cloud computing is to provide on-demand resources without compromising their security. In order to achieve this objective, an integrated approach for intrusion detection and response systems is mandatory. This is because an integrated approach guarantees minimal response time by eliminating the human intervention. In the following sections of this paper, we present our proposed solution to address these requirements for intrusion detection and severity analysis for Clouds.

4. AN AUTOMATIC APPROACH FOR INTRUSION SEVERITY ANALYSIS FOR CLOUDS

The solution proposed to address intrusion severity analysis problem for Clouds has been described in detail in [9]. In this paper, we present a summary of the proposed solution. As described in [9], the proposed solution is based on the presumption that the severity of an intrusion for a particular virtual machine depends on a number of factors; security requirements of the application hosted by the virtual machine, the state of any Service Level Agreement (SLA) negotiated beforehand, and the frequency of attack on a security requirement. There can be other parameters, however, these are regarded as the most important factors and therefore have been incorporated in the proposed solution. We hold that the severity problem can be treated as a special case of traditional classification problem as it essentially involves segregating intrusion trails into different classes [10]. Finally, both supervised and unsupervised learning algorithms have been used to implement our proposed solution.

With respect to virtual machine specific security requirements, one option can be to render the security policy definition and management a responsibility of the virtual machine itself. This can be achieved by implementing a policy engine within each virtual machine, which will coordinate with detection and severity analysis modules in the privileged virtual machine. This approach is attractive due to the ease of implementation and simplicity of the resultant system. However, it breaks the isolation property as, in the event of a successful attack, an attacker can modify the security policies to facilitate its malicious objectives. Furthermore, a guest virtual machine needs to be trustworthy to be delegated such responsibility which is contradictory to our assumption that, all guest virtual machines are treated as compromised. Due to these limitations of this approach, an approach has been
adopted that guarantees isolation while ensuring customization with respect to security policies. Using service level agreements to negotiate security requirements for a virtual machine has been proposed. Following this approach, a user is envisaged to specify the security requirements as part of the SLA at the resource acquisition phase. However, in order to accomplish this, quantification of security is required. The proposed security quantification is summarized in Table 1.

With respect to SLA state, the time remaining for completion of a job has been designated as the SLA state. This is because of the fact that severity of an intrusion is also affected by the time available for response. Ideally, the SLA state would be calculated by using different parameters such as; quality of service metrics and available resources. This requires establishment of a complete monitoring infrastructure to monitor the status of these parameters and evaluate aggregate SLA state using some mathematical model. Due to these complexities, this has been rendered as out of the scope of this research. However, it is assumed that SLA state is available as an aggregate metric that can be used for formal analysis such as the one described in this paper. Finally, the frequency of attack attempts on a particular security requirement depicts either the value of the target or likelihood of success of attack attempt against the security requirement under attack. This therefore requires relatively immediate and more effective response mechanism to avoid recurrence of such attack attempts. For this reason, the frequency of attacks has been characterised as an important factor to dictate the severity of an intrusion.

Table 1: Proposed Security Requirements

<table>
<thead>
<tr>
<th>Security Attributes</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrity</td>
<td>Workload State Integrity</td>
</tr>
<tr>
<td></td>
<td>Guest OS Integrity</td>
</tr>
<tr>
<td>Availability</td>
<td>Zombie Protection</td>
</tr>
<tr>
<td></td>
<td>Denial of Service Attacks</td>
</tr>
<tr>
<td></td>
<td>Malicious Resource Exhaustion</td>
</tr>
<tr>
<td></td>
<td>Platform Attacks</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>Backdoor Protection</td>
</tr>
</tbody>
</table>

As stated earlier, we propose to address the severity problem by treating it as a classification problem. Related to this, a characteristic of supervised learning techniques is that they involve an initial training or learning phase to serve as a basis for online classification. However, with the problem focused in this research, no previous knowledge of severity of intrusions for applications is maintained which makes it difficult to use supervised learning techniques. Furthermore, most of the unsupervised learning techniques are more suitable for offline analysis as the classifications tend to change over the length of analysis datasets. This characteristic makes them inappropriate for systems that require real-time classification such as the one under consideration in our research. Therefore, we decided to use both supervised and unsupervised classification techniques to achieve our objectives. An unsupervised classification technique i.e. K-means has been used to prepare the training datasets for further
analysis and supervised classification technique i.e. Decision Trees has been used for real-time severity analysis. The experimentation and evaluation of using these techniques for the proposed solution is presented in [10].

5. ABSTRACT MODEL FOR INTEGRATED INTRUSION DETECTION AND DIAGNOSIS FOR CLOUDS

The primary objective of this paper is to present the abstract model proposed to achieve integrated intrusion detection and severity analysis for Clouds. The abstract model is envisioned to fulfill the requirement of Clouds for minimized intrusion response time with respect to intrusion severity analysis. In order to achieve this objective, the time between the point when an intrusion is detected and an appropriate response is activated becomes critical. In order to minimize this time interval, integration between intrusion detection and response systems is required. The integrated architecture presented in figure 3 achieves this objective by defining different components involved in this process and outlining the interactions between these components. Additionally, the response selection mechanism is improved by incorporating rigorous intrusion severity evaluation; this ensures that the selected response is proportionate to the intrusion attempt and the security characteristics of the victim application.

5.1 Assumptions

As described in figure 2, domain 0 is considered to be the most privileged virtual machine within a virtualized resource. Furthermore, this status of domain 0 guarantees that it is entirely isolated from the monitored virtual machines i.e. guest virtual machines. Therefore, a guest virtual machine cannot intrude in the process executed within the domain 0. In order to harvest these benefits, the system represented by the abstract model in figure 3 is envisioned to be incorporated with the domain 0 of a virtualized system. Additionally, the intended system can only interact with the monitored virtual machines via system call interface which implies that the visibility of this system is limited to system calls executed by monitored virtual machines. The hypervisor is assumed to be trustworthy and because of maximum isolation from guest virtual machines, it is not possible for a malicious guest virtual machine to hijack the hypervisor.

It is also assumed that all the guest virtual machines are compromised and therefore, not trustworthy. This is because guest virtual machine can potentially have multiple different owners who, from a resource provider’s perspective, cannot be trusted for the security of the infrastructure. Finally, it is assumed that SLAs are negotiated and agreed upon during the resource acquisition phase.

5.2 Components of the Model

The detailed abstract model for integrated intrusion detection and severity analysis for Clouds has been presented in figure 3. We present the description of different components of the abstract model below. However, evaluation of this model is described in a later section.

System Call Handler: The system call handler is envisaged to be a module to intercept system calls executed by the guest virtual machine to facilitate further analysis on system calls. This module is significant because, as per our assumptions, both the intrusion detection and severity analysis modules are envisioned to use system call data to perform their respective functions. Therefore, this module acts as a pivotal component of the entire architecture. This module is envisaged to intercept all system calls executed by all monitored virtual machines, append virtual machine specific information such as virtual
machine identifier (VMID) and, dispatch it to an intrusion detection system. The virtual machine specific information is used by different components of the model to achieve virtual machine specific behaviour.

Detection Module: The detection module represents a system call based intrusion detection system. However, the type of detection engine i.e. anomaly or misuse based is implementation specific. In case of a misuse based IDS, the Attack DB represents a database containing signatures of known attacks whereas Profile Engine is envisaged to support anomaly based intrusion detection by providing profiles for normal and abnormal behaviours. Although the specific functionality of this module depends on the implementation, the general objective of this component is to detect if a particular system call event represents a malicious behaviour. The event, in this case, may contain a single system call or a sequence of system calls depending upon the type of IDS. In order to achieve flexibility, a Monitor is a dedicated component created at runtime for each virtual machine. A monitor is responsible for handling system call events for a specific virtual machine. The detection module is envisioned to encapsulate information about malicious event as a ISA Request which is sent to the severity analysis module for further processing.

Severity Analysis Module: The severity analysis module represents a component responsible for evaluating the severity of a particular malicious behaviour for a victim virtual machine. This module works in collaboration with the detection module to achieve integrated operation. Upon receiving a ISA Request from the detection module, this module is envisioned to perform rigorous severity evaluation. This involves translation of malicious event from system call to security properties of the victim and is supported by the Profile Engine to provide VM specific security characteristics. By doing this, a customized intrusion severity evaluation is achieved with minimum human intervention. One example implementation of this module has been provided by [9].

Profile Engine: Profile engine is envisaged to generate and manage virtual machine specific profiles. Among other attributes, these profiles are envisaged to contain the prioritized security characteristics for all the guest virtual machines. In order to achieve these objectives, a profile engine is envisioned to interact with global Cloud resource manager. As part of its functionality, profile engine communicates with both the intrusion detection and severity analysis modules to facilitate virtual machine specific operation.

Global Components: with any Cloud infrastructure, there is a need for mechanism to manage the overall infrastructure. More recently, a number of efforts have been made to address this need. These include Nimbus [4], OpenNebulla [5] and iVIC [18]. Global Components represent such mechanisms focused at providing overall management of the Cloud infrastructure. As has been described in the figure 3, these include components such as resource manager, scheduler and profile repositories. Furthermore, it can also include security components such as firewall to enhance overall security of the cloud infrastructure.

Intrusion Response System: The objective of this research is to minimize the intrusion response time and improve the mechanism for response selection. In order to achieve this, the result of intrusion severity evaluation is fed into an intrusion response system. The intrusion response system is envisioned to use this knowledge to select appropriate response mechanism for a particular intrusion. Furthermore, the automated and integrated nature of the proposed architecture facilitates minimizing intrusion response time.
Figure 3: An abstract model for integrated intrusion detection and severity analysis.

The data structures used as part of the abstract model described in figure 3 are defined below.

**Data Structures**

- **Detection Signature**: \{System Call args, VMID, detection policy\}
- **ISA Request**: \{Detection Signature, detection decision, TIMESTAMP, attack info\}
- **ISA Signature**: \{ISA Request, ISA policy\}
- **Alert Message**: \{Detection Signature, detection decision\}
- **Detection Policy Req**: \{VMID, POLICY_IDENTIFIER\}
- **ISA Policy Req**: \{VMID, POLICY_IDENTIFIER\}
The architecture presented in this section provides an abstract model to facilitate virtual machine specific intrusion detection and severity analysis. This approach is envisaged to help fulfill the requirements of Clouds for intrusion severity analysis as described earlier in this paper. Furthermore, it is envisioned to improve the response selection mechanism used by intrusion response systems.

6. EVALUATION OF THE ABSTRACT MODEL

In order to assess the applicability and effectiveness of an architecture, evaluation is mandatory. Architectural evaluation is necessary in order to evaluate if the proposed architecture meets the expected quality of service. Additionally, software architectures represent a key ingredient in its life cycle and an architectural decision is often costly to rectify in the later stages of the SA life cycle. Therefore it is recommended that evaluation should be performed at the design level so as to achieve a robust design which can facilitate achievement of overall objectives of a software life cycle.

A number of different methods have been proposed to evaluate software architectures. These methods differ from each other with respect to the stage of development of the software and the level of information available about the different components of the architecture. Mathematical models represent methods which using mathematical systems to evaluate an architecture. These methods require implementation specific data and therefore, are best suitable for software systems with known implementation data. Additionally, due their requirement for implementation specific data, mathematical models are more suitable for component based system. An alternative approach to mathematical models is scenario based approach for architectural evaluation. Scenario based evaluation methods do not require implementation specific data and rather use *scenarios*. A scenario is a brief description of some anticipated or desired use of a system. Due to this, these methods can be applied to any type of software architectures. As our proposed architectural is novel and does not have any components implemented to gather performance data, we chose scenario based evaluation methods to perform evaluation of the architecture proposed in this paper.

There are a number of methods proposed to perform scenario based evaluation for software architecture. A detailed comparative analysis of these methods has been presented in [19]. However, these methods can be distinguished based on the quality attributes focused by the method and the number of quality attributes considered by the method. Quality attributes represent different parameters which can be used to assess a particular software system. These include security, flexibility, and maintainability. With respect to scenario based methods, Software Architectural Analysis Method (SAAM) represents the pioneer method and is considered parent method for a number of subsequent architectural analysis methods such as Architectural Trade-off Analysis Method (ATAM) [20], SAAM for Complex Scenarios (SAAMCS) [21] and Aspectual SAAM (ASAAM) [22].

For the purpose of this research, we have chosen the Architectural Trade-off Analysis Method (ATAM) to assess the effectiveness of the proposed model for integrated intrusion detection and severity analysis for Clouds. The evaluation is conducted with respect to three different quality attributes i.e. diversity, security, and performance. The choice of ATAM is motivated due to its ability to take into account multiple quality attributes for architectural evaluation whereas other scenario based evaluation methods are mostly focused at one particular quality attribute [19]. For instance, Architectural Level Modifiability Analysis (ALMA) [23] is envisaged to evaluate the modifiability of the software architecture. ATAM is primarily a risk identification mechanism which facilitates identification of scenarios where a desired quality attributes is affected by architectural decisions.
ATAM is envisaged to be conducted by a team of system and software engineers. Related to this, the initial steps of ATAM are focused at introducing the ATAM method to the team, outlining the objectives of the architecture, presenting the candidate architecture and identifying architectural approaches that can be vital to achieve quality attribute goals. As the research presented in this paper does not have a team, these steps have been performed individually. The objective of this research, as described earlier, is to evaluate the architecture for four quality attributes i.e. security, performance, diversity and flexibility. With respect to architectural approaches and critical components, the key architectural components are; virtual machine hypervisor, global resource manager and system call interceptor. These have been considered critical due to their vital role to accomplish the objectives of the architecture. For instance, global resource manager is critical to establish and maintain virtual machine specific profiles which have a profound role to achieve the objectives outline by diversity quality attribute.

Furthermore, ATAM uses different tools such as quality attribute characterization and utility tree to accomplish its goals. And, the output generated as part of ATAM is in the form of risks and non risks and sensitivity and trade-off points. These outputs are then used to fine-tune the architecture for its overall objectives. Each of these tools and their specific outputs for the proposed abstract model has been described in the following subsections.

![Figure 4: Quality attribute characterization for security](image-url)
6.4 Quality Attribute Characterization

In order to evaluate an architecture against a given quality attribute requires understanding of different dimensions of that attribute. For instance, evaluating an architecture for security requires knowledge of how to measure security and understanding how can security be influenced by architectural decisions. This is termed as **quality attribute characterization** for ATAM and forms the basis, along with scenarios, of processes involved in ATAM. A quality attribute characterization for security has been presented in figure 4. The characterization presented in this figure is an extended form of characterization proposed by [24]. The attributes in italic font represent our extensions to include virtual machine specific attributes. As described in this figure, attribute characterization consists of three categories; **stimulus, architectural parameters** and **responses**. The stimuli represent external events which require a response from the architecture. As part of security characterization, these include different types of attacks. The responses represent measures to achieve quality attribute requirements and have been represented by different methods to achieve security in figure 4. Finally, the architectural parameters represent aspects of architecture which facilitate achieving the responses. Among other attributes, these include Hypervisor in figure 4 which represents an architectural component to facilitate security responses such as intrusion detection and severity analysis described in this paper.

For the attribute characterizations for other quality attributes focused in this research i.e. performance, and diversity, the characterizations proposed by [20] have been used.

![Utility tree for the proposed architecture](image)

**Figure 5: Utility tree for the proposed architecture**

6.2 Utility trees

As described earlier, ATAM uses scenarios as the basis of its activities. However, with multiple quality attributes, a number of potentially conflicting scenarios can be generated. For instance, in order to security objectives of the proposed architecture, system call interception is envisaged to be used. However, this approach will result in a performance overhead for the monitored virtual machines. This presents a conflict between two desired quality attributes
i.e. security and performance. In order to comprehend with such conflicts, ATAM introduces utility trees. The objective of utility trees is to identify, prioritize and refine the most important quality attribute goals. A utility tree for the proposed architecture has been presented in figure 5. As shown in figure 5, the high level nodes represent the most important quality goals which in this case are: performance, security, flexibility and diversity. Furthermore, the leaves of this tree represent scenarios to achieve respective quality attribute goals. The result of generating an utility tree is a prioritization and characterization of specific quality attributes as has been demonstrated by figure 5.

6.3 Risks and non-risks

As part of the activities described till now, the architecture has been presented along with the quality attributes, scenarios for these attributes and the utility tree. The next step in ATAM is to analyze this architecture with respect to the quality attributes. A critical output of this process is the identification of risks and non-risks for the proposed architecture with respect to the quality attributes. A risk is an architectural decision with potentially problematic consequences whereas a non-risk is a good architectural decision which is based on assumptions implicit in the architecture. With respect to the proposed architecture, the risks and non-risks identified during the evaluation process are presented in figure 6.

Each risk presented in figure 6 is an outcome of analyzing the proposed architecture with respect to specific quality attribute objectives. For instance, consider the risk: the decision to have a database supported intrusion detection engine is a risk if it is not capable of detecting unknown attacks. This is a risk to the quality attribute security and multiple scenarios described as part of the utility tree in figure 5. In particular, this risk represents a threat to achieving the objectives of the proposed architecture with respect to security. This is because efficient intrusion detection has a profound role to fulfil the purpose the proposed architecture and requires an intrusion detection system capable of detecting all malicious events.

![Figure 6: Risks and non-risks for the proposed architecture](image-url)
As for the non-risks, figure 6 presents a list of non-risks for the proposed architecture. For instance, consider the non-risk; *the integrated operation of different components of the architecture is a non-risk for performance with respect to improved intrusion response time.* This presents an advantage to achieve the quality attribute performance with respect to the scenario described in the utility tree in figure 5 i.e.; *improved efficiency is demonstrated with respect to intrusion response time.* This has been categorized as a non-risk because it is a property of the proposed architecture which has been inherited due to the assumption of integrated operation between different components of the architecture. Additionally, it helps achieve the objective of proposed architecture with respect to performance.

### 6.4 Sensitivity and Trade-off points

As described earlier, one of the reasons to select ATAM for our research was its ability to take into account multiple, potentially conflicting, quality attributes for the evaluation of an architecture. Related to this, ATAM facilitates identifying *sensitivity* and *trade-off points* for the architecture. *Sensitivity points* represent properties of a component that is critical to the success of the overall system whereas *trade-off point* represents a property that affects more than one quality attribute or sensitivity point. The sensitivity and trade-off points for the proposed architecture have been presented by figure 7.

With respect to sensitivity points, each point highlights the importance of a component of the architecture to achieve one or more quality attributes. For instance, consider the sensitivity point; *the performance overhead for a system call is sensitive to the latency due to interception by system call handler.* This highlights the importance of system call interceptor to achieve the performance quality attribute. In particular, it emphasizes the importance of efficiency of system call interceptor to achieve the scenario for performance quality attribute i.e.; *the performance overhead for a particular system call is negligible.* Therefore, in order to minimize the performance overhead for system call execution, an efficient system call interceptor with minimum latency is critical.

As described earlier, trade-off points represent properties of components that can introduce conflicts between two or more quality attribute or quality attributes. For instance, consider the trade-off point; *latency due to system call interception is a trade-off between performance in terms of execution time and security in terms of detecting malicious system call execution.* This highlights the trade-off between two quality attributes i.e. security and performance. More specifically, it emphasizes the importance of an efficient system call interceptor for security and performance quality attributes. Therefore, there is a need to prioritize scenarios with respect to these quality attributes.
As a result of the activities described above, a set of risks, non-risks, sensitivity points and trade-off points has been generated. In accordance with these results, a set of recommendations have been proposed to foster use of the proposed architecture to achieve objectives outlined earlier in this paper. These recommendations are focused at specific components of the architecture and can be used as best practice guidelines to implement the architecture proposed in this paper to achieve efficient integrated intrusion detection and severity analysis for Clouds.

- The efficiency of the system call interceptor is critical to overall response time for an intrusion.

- In order to improve security of the Cloud infrastructure, the IDS should be capable of detecting maximum possible malicious events. Specifically, emphasis should be given to the malicious events represented by the threat model for a system.

- The trade-off between performance and security should be application specific. This highlights the need for appropriate mechanisms to optimize the balance between security and performance according to the security requirements of the application.

- The algorithm used for severity evaluation should take into account all the parameters as proposed in this paper. This is to ensure effective severity evaluation in the case when a malicious event affects multiple security requirements.

- The monitors as part of detection module should be light weight to minimize the performance overhead. Furthermore, the monitors should facilitate minimizing the overall performance overhead due to intrusion detection.

- The implementation of detection module is critical to avoiding it becoming a bottleneck for the overall detection module. Therefore, appropriate mechanisms such as *threads* should be used to foster multitasking within detection module.
• The trade-off between online and offline severity analysis should be application specific. This is because some applications such as mission critical computations are more vulnerable to performance overheads as compared to others. Therefore, quality of service requirements should be consulted whilst implementing the severity analysis module.

The set of recommendations listed above are primarily concerned with the implementation of the proposed architecture. These recommendations are focused at mitigating with the risks identified as part of the evaluation process described earlier in this paper. Furthermore, we believe, that the decisions to comprehend with the trade-off points should be application specific. This is because the impact of attributes involved in risks and trade-off points varies across different applications primarily depending upon the characteristics of the applications and their quality of service requirements. Therefore, we recommend that Cloud infrastructure providers should take into account these factors when implementing the architecture proposed in this paper.

8. CONCLUSIONS
Cloud computing represents an emerging computing paradigm which introduces novel opportunities to establish large scale, flexible computing infrastructures. As with any other emerging paradigm, security underpins the extensive adoption of Clouds computing. In this paper, we have presented our efforts to address one of the significant issues with respect to security of Clouds i.e. intrusion detection and severity analysis. Specifically, we propose an abstract model for integrated intrusion detection and severity analysis for Clouds. With respect to requirements of Clouds for intrusion severity analysis [9], our proposed model is envisaged to fulfil requirements for minimal response time and human intervention. In order to assess the effectiveness of the proposed model, we have also presented architectural evaluation for this model using the ATAM. The evaluation facilitates improved understanding of the architecture and helps identify risks for implementation of the proposed architecture. It also identifies properties of the components to achieve quality objectives along with conflicting properties which require prioritization. Finally, we also present a set of recommendations which can be used as a set of best practice guidelines whilst implementing our proposed architecture.

REFERENCES


