ARA-Assessor: Application-Aware Runtime Risk Assessment for Cloud-Based Business Continuity

Min Fu[1,2(✉)], Shiping Chen[2,3], Jian Yang[1], Surya Nepal[2,3], and Liming Zhu[2,3]

[1] Department of Computing, Macquarie University, Sydney, Australia
{min.fu, jian.yang}@mq.edu.au
[2] Data61, CSIRO, Sydney, Australia
{Shiping.Chen, Surya.Nepal, Liming.Zhu}@data61.csiro.au
[3] School of Computer Science and Engineering, UNSW, Sydney, Australia

Abstract. Cloud-based systems are prone to be attacked because they share the same cloud infrastructure, where there may exist hackers and malicious users. As a result, cloud system owners need an on-going security risk assessment mechanism to monitor the risk of their systems so that they can be mitigated in a timely manner to ensure the business continuity. Existing methods of cloud system risk assessment usually do not fully consider the dependencies of the system's cloud resources or the conflictions of the threats on the system. In this paper we propose an application-aware cloud system risk assessment method, called ARA-Assessor, for performing security risk assessment for cloud systems. ARA-Assessor includes a cloud system model used to specify the significance value of each system component and their dependencies. With this application-aware model, the cloud system owners are able to continuously assess the risk of their systems. We evaluate ARA-Assessor with three typical cloud systems on AWS. The experimental results show that our method is capable of continuously assessing the runtime risk for multiple types of cloud systems.

Keywords: Cloud security · Cloud risk · Risk management · Risk assessment

1 Introduction

Cloud computing is widely adopted by businesses and governments, and a large number of them prefer to deploy and run their software applications and enterprise systems on the cloud platform [1, 2]. Since the cloud is a multi-tenancy environment shared by multiple users, a significant concern about cloud systems is their security [2, 3]. A survey from the research firm Gartner in 2015 found that around 95% of the consumers of cloud computing reported cloud security issues [4]. A survey conducted

The acronym "ARA" is short for "Application-Aware Risk Assessment".

M. Maximilien et al. (Eds.): ICSOC 2017, LNCS 10601, pp. 511–527, 2017.
https://doi.org/10.1007/978-3-319-69035-3_38

by the Cloud Security Alliance (CSA) in 2016 indicated that at least 35% of the business owners did not trust the security of cloud as much as internal IT systems [5]. Cloud security has become a significant concern for ensuring business continuity [3].

In order to address the security issues of cloud systems and ensure the business continuity despite potential attacks, a useful procedure is to perform security risk assessment for cloud systems at runtime [6, 7]. Existing methods of cloud risk assessment [6, 8–10] largely focus on individual system components. They are concerned with either the application-level attacks that have an impact on service availability [10] or the intrinsic vendor-level risks of the cloud providers themselves [6, 8], and some of them do not fully leverage the risks that result from the on-demand nature of cloud [6, 8, 10]. Another problem with existing cloud risk assessment mechanisms is that they do not consider the complete dependencies of the cloud resources of the cloud system or the full conflictions of the cloud system's threats [9, 10].

As such, in this paper we propose a novel cloud risk assessment framework, called ARA-Assessor, for determining the runtime risk value of the cloud system provided by the system owner. ARA-Assessor is application-aware, which means that the risk assessment leverages the system specification model provided by the system owner. ARA-Assessor relies on the infrastructure-level threats of the cloud system to calculate the risk. We implement ARA-Assessor and evaluate it with three representative types of cloud systems on AWS cloud. The experimental results show that our proposed method is able to continuously and quantitatively assess the runtime risk of cloud systems in an automated way and it is generalizable for multiple cloud systems.

The research contributions of this paper are: (1) we propose a generalizable cloud runtime risk assessment method; (2) we propose a generic cloud system modelling approach and a generic cloud system threats modelling approach; (3) we propose a cloud resource dependencies propagation mechanism and a recursive mechanism for resolving the threats conflictions issue for analyzing cloud system risk.

The remainder of this paper is organized as follows: Sect. 2 describes a motivating example; Sect. 3 discusses cloud system modelling; Sect. 4 discusses cloud system threats modelling; Sect. 5 illustrates our risk assessment method; Sect. 6 presents our experimental evaluation; Sect. 7 discusses the validity and general applicability of our model; Sect. 8 discusses the related work; Sect. 9 provides the conclusion and our future work.

2 A Motivating Example

We use a sample cloud system, as shown in Fig. 1, to discuss the risk management. This cloud system follows the typical 2-tier architecture [11]. The E-Business service and the report generation service run in the web instances. These two services are auto-scaled by the auto-scaling service provided by the Auto Scaling Group (ASG), and the workload requests on these two services are dispatched by the load balancing service provided by the Elastic Load Balancer (ELB). The E-Business service triggers the production database service running inside the production database instance, and the report generation service triggers the report database service running inside the reporting database instance. There is periodical synchronization from the production

database store to the reporting database store. A number of potential cloud infrastructure-level threats could occur to this sample cloud system. We categorize these threats into the following categories: (1) threats on the cloud login credentials; (2) threats on the ASG; (3) threats on the LC; (4) threats on web instances; (5) threats on database instances; (6) threats on the ELB; (7) threats on each instance's Amazon Machine Image (AMI); (8) threats on the security group. To study the occurrence frequency of the threats on the cloud system, we analyzed the cloud security report from Alert Logic [12], and obtained the month-to-month attack spread for real-world cloud systems in 2014, as shown in Fig. 2. How to accurately determine the system's risk is a question.

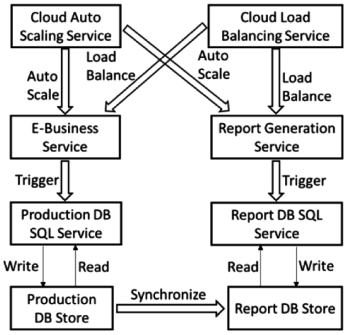

Fig. 1. 2-tier cloud system.

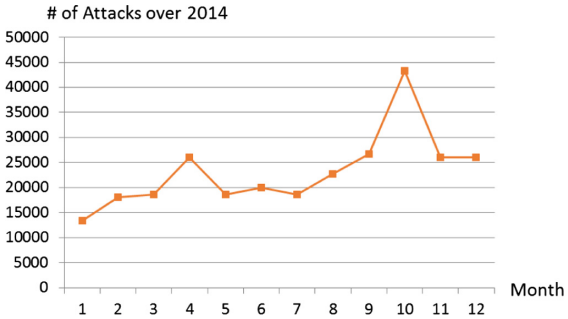

Fig. 2. Attacks for real cloud systems in 2014 [12].

3 Generic Modelling of Cloud Systems

A cloud system is deployed on a set of allocated cloud resources. Each cloud resource has a resource id, belongs to a cloud resource type (e.g. ELB), and has a significance value which reflects the importance of the resource. Some cloud resources in the system have dependencies, i.e. the attacks on such a resource can affect its dependent cloud resources. Hence, we can use DAG (Directed Acyclic Graph) to model the cloud system resources. The cloud system model, denoted as S, is represented as:

$$S = (\boldsymbol{R}, \boldsymbol{E}) \tag{1}$$

where $\boldsymbol{R}$ refers to the set of cloud resources, and $\boldsymbol{E}$ refers to the set of cloud resource dependencies. Each element of $\boldsymbol{R}$, denoted as R_i ($1 \leq i \leq |\boldsymbol{R}|$), is represented as:

$$R_i = (N, V, W) \tag{2}$$

where N denotes resource id, V denotes resource type, and W denotes resource significance value. Each element of $\boldsymbol{E}$, denoted as E_i ($1 \leq i \leq |\boldsymbol{E}|$), is denoted as:

$$E_i = (R_k, R_m), R_k \in \boldsymbol{R}, R_m \in \boldsymbol{R} \tag{3}$$

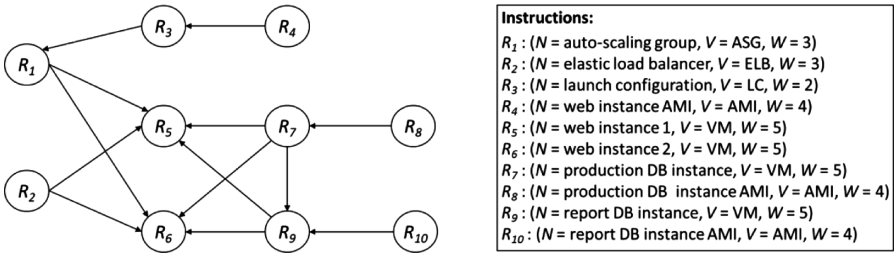

Fig. 3. DAG graph for the sample cloud system.

where R_k refers to any cloud resource that can affect another cloud resource, and R_m refers to R_k's affected cloud resource. Taking the cloud system mentioned in Sect. 2 as an example, its system model is shown in Fig. 3. The resources are represented as R_1 to R_{10}. The id, type and significance value of each cloud resource are presented. The cloud resource dependencies are represented by the directed arrows.

4 Generic Modelling of Cloud System's Threats

A cloud system's threats refer to all the potential infrastructure-level cloud threats that can occur to the system. Each cloud threat consists of the following information: (1) threat name; (2) threat feature which specifies the types of cloud resources that can be directly affected by the threat; (3) threat's directly attacked cloud resources; (4) threat's overall affected cloud resources propagated from the dependencies of the directly attacked resources; (5) threat impact value; (6) threat occurrence probability. Some threats have conflictions with each other, i.e., they are unable to occur at the same time. For example, terminating the database instance and changing database instance type cannot occur simultaneously. Hence, we can model the cloud system's threats as a graph. The cloud system threats model, denoted as T_S, is represented as:

$$T_S = (\boldsymbol{T}, \boldsymbol{E}) \tag{4}$$

where $\boldsymbol{T}$ refers to the set of cloud system threats, and $\boldsymbol{E}$ refers to the set of cloud system threats conflictions. Each element of $\boldsymbol{T}$, denoted as T_i ($1 \leq i \leq |\boldsymbol{T}|$), is represented as:

$$T_i = (N, F, \boldsymbol{R_D}, \boldsymbol{R_A}, I, P) \tag{5}$$

where N denotes each threat's name, F denotes threat's feature, $\boldsymbol{R_D}$ denotes each threat's directly attacked cloud resources set and each element in $\boldsymbol{R_D}$ follows the model defined in Formula (2) in Sect. 3, $\boldsymbol{R_A}$ denotes each threat's overall affected cloud resources set and each element in $\boldsymbol{R_A}$ follows the model defined in Formula (2) in Sect. 3, I denotes each threat's impact value, and P denotes each threat's occurrence probability. Each element of $\boldsymbol{E}$, denoted as E_i ($1 \leq i \leq |\boldsymbol{E}|$), is denoted as:

$$E_i = (T_k, T_m), T_k \in \boldsymbol{T}, T_m \in \boldsymbol{T} \tag{6}$$

where T_k refers to any threat that conflicts with another threat, and T_m refers to T_k's conflicted threat. Taking the sample cloud system mentioned in Sect. 2 as an example, the threats graph of the system is shown in Fig. 4. The threats are T_1 to T_n. The threats conflictions are represented by the undirected edges. T_1 and T_2 cannot occur simultaneously (T_1 conflicts with T_2); T_3 and T_4 cannot occur simultaneously (T_3 conflicts with T_4); T_5 and T_6 cannot occur simultaneously (T_5 conflicts with T_6).

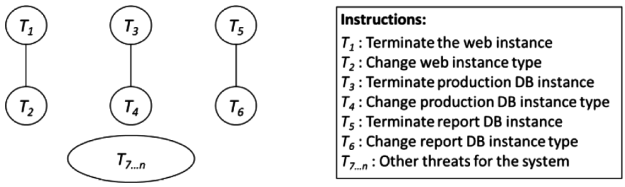

Fig. 4. Threats graph for the sample cloud system.

5 Our Risk Assessment Method

In order for ARA-Assessor to assess the runtime risk for a cloud system, it requires two inputs: (1) the cloud system model S and (2) the cloud full threats model T_F ($T_F = (\boldsymbol{T}, \boldsymbol{E})$). S is manually provided by the cloud system owner who has enough system domain knowledge. The significance value of each cloud resource ranges from 1 to 5. It is determined according to the importance of the internal service. Resource dependencies are determined according to the interactions of the services inside the resources and the dependencies specified in cloud resources documentations [17]. For example, a web instance interacts with a database instance because the web service inside the web instance triggers the database service inside the database instance. For another example, an elastic load balancer (ELB) or an auto-scaling group (ASG) contains multiple web instances. T_F resembles the threats model defined in Sect. 4. We assume T_F is manually prepared by the system owner. The threats set and threats conflictions set of it can be determined by analyzing and understanding the domain knowledge on cloud threats and published dataset [1, 18–21]. The threats in T_F include the threats related to all types of cloud resources, e.g. cloud web instance related threats, cloud database instance related threats, ASG/ELB related threats, etc.

Since attacks on a cloud system are usually unpredictable and can occur at any time, ARA-Assessor periodically assesses the cloud system's risk. We implement ARA-Assessor as a dedicated service, which embodies the concept of "Security as a Service" [16]. Prior to performing a periodical risk assessment, ARA-Assessor first conducts the one-off procedure, which consists of five activities: (1) ARA-Assessor automatically determines the system's threats subset model T_S using the inputs of cloud

system model S and full cloud threats model T_F; (2) ARA-Assessor obtains the initial occurrence probability of each threat $T_S \rightarrow T_i$ $(1 \leq i \leq |T_S \rightarrow \boldsymbol{T}|)$; (3) ARA-Assessor automatically determines the affected cloud resources for each threat $T_S \rightarrow T_i$; (4) the impact value of each threat $T_S \rightarrow T_i$ is calculated; (5) the cloud consumer specifies the frequency of performing risk assessment, e.g. every minute. Then four activities are conducted upon each time tick: (1) ARA-Assessor relies on external attack detection services [13–15] to detect the runtime threats and events that occur to the system, denoted as T_{RT}; (2) we derive a threats sub-model from T_S, denoted as T'_S $(T'_S = (\boldsymbol{T}', \boldsymbol{E}'))$, where $\boldsymbol{T}'$ removes all the conflicted threats of each runtime threat in T_{RT} from $T_S \rightarrow \boldsymbol{T}$, and $\boldsymbol{E}'$ removes all the conflictions with regard to each runtime threat in T_{RT} from $T_S \rightarrow \boldsymbol{E}$); (3) the occurrence probability of each threat in T'_S is updated based on T_{RT}; (4) ARA-Assessor uses the threats sub-model T'_S to calculate the system's risk value for the time tick, denoted as RI_S, as below:

$$RI_S = Max\left(\sum_{j=1}^{|M(T'_S)[i] \rightarrow \boldsymbol{T}|} \left(M(T'_S)[i] \rightarrow T[j] \rightarrow I\right) \times \left(M(T'_S)[i] \rightarrow T[j] \rightarrow P\right)\right) \quad (7)$$

where $M(T'_S)$ refers to an array of threats sub-models derived from T'_S, each threats sub-model $M(T'_S)[i]$ (i ranges from 1 to $|M(T'_S)|$) represents a case of threats model that contains all the threats from T'_S which are independent of each other and do not conflict with each other, and this array enumerates complete cases of such threats models for T'_S. For each case, the risk value is calculated, and the maximum of the calculated risk values is the quantified risk of the cloud system for that time tick.

5.1 Determination of Cloud System's Threats Subset Model

The cloud infrastructure-level threats that can occur to the cloud system are a subset of all cloud infrastructure-level threats that can occur to all cloud resources. When we rely on external attack detection tools to detect threats, we should only subscribe the system's threats subset in order to save cost. Hence, we need to determine the cloud system's threats subset. Using the two inputs of cloud system model S and full cloud threats model T_F, the cloud system's threats subset model T_S is determined as below:

$$T_S = (\boldsymbol{T}, \boldsymbol{E}) \quad (8)$$

where $\boldsymbol{T}$ is a subset of $T_F \rightarrow \boldsymbol{T}$, and $\boldsymbol{E}$ is a subset of $T_F \rightarrow \boldsymbol{E}$. Each threat in T_S, denoted as $T_S \rightarrow T_i$, satisfies such a condition: $(T_S \rightarrow T_i \rightarrow F) \cap V(S \rightarrow \boldsymbol{R}) \neq \emptyset$, where $V(S \rightarrow \boldsymbol{R})$ represents the cloud system's overall resources types set.

Taking the sample cloud system mentioned in Sect. 2 as an illustrating example, the determined threats subset model is the one represented in Fig. 4 in Sect. 4.

5.2 Determination of Threats Initial Occurrence Probabilities

The initial occurrence probabilities of the cloud system's threats can be determined by analyzing cloud attacks historical data such as the security reports from Symantec

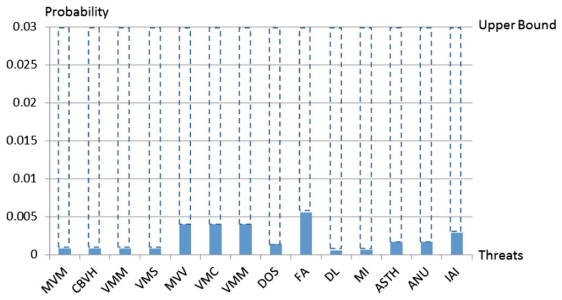

Fig. 5. Threats probabilities for cloud [23].

Corporation [22]. Based on the research on cloud security threats analysis done by the University of Tunis [23], we are able to obtain the threats probabilities as shown in Fig. 5. According to the research, the probability of no cloud threats occurring is 0.97, so the upper bound probability for each cloud threat is 0.03. For simplicity, we assume that the occurrence probabilities of all the threats for the cloud system are 0.03.

5.3 Determination of Cloud Resources Affected by System Threats

The affected cloud resources for a threat refer to the system cloud resources that are affected either directly or indirectly by the threat. The indirectly affected resources are propagated from the directly attacked resources. A challenge with the resource propagation is that the resource dependencies can be multi-layer, which means that a cloud resource's dependent resources can further have dependent resources, and so on. We address this challenge and design the affected resources determination mechanism, as illustrated in Algorithm 1. For each threat, we first determine its directly attacked cloud resources by mapping the feature of the threat with the cloud resources in the system model (DetermineDirectlyAffectedResources($T \rightarrow F$, $S \rightarrow \boldsymbol{R}$)); second, for each directly attacked resource of the threat R, we add it into the threat's affected resources set, and then we use a recursive function to add its overall propagated dependent affected resources into the threat's affected resources set (RecursivelyDetermineAffectedResources(R, $T \rightarrow \boldsymbol{R_A}$)). Inside the recursive function, we first get R's dependent affected cloud resources set (GetAffectedResources(R, $S \rightarrow \boldsymbol{E}$)). If this set is empty, we exit the recursion; otherwise, for each of R's dependent affected resources, R', we add it into the threat's affected resources set, and then we further add its overall propagated dependent resources into the threat's affected resources set.

Algorithm 1: System Threats Affected Cloud Resources Determination

Input: Cloud System Model S, System's Threats Subset Model T_S

Output: The System's Threats Model T_S with Affected Resources Determined

```
1   Function DetermineThreatsAffectedResources(S, T_S) {
2     Foreach (T in T_S → T) {
3       T → R_D = DetermineDirectlyAffectedResources(T→F, S→R);
4       Foreach (R in T → R_D) {
5         T → R_A = R ∪ (T → R_A);
6         RecursivelyDetermineAffectedResources(R, T → R_A); }}
7     Return T_S;
8   }
9
10  Void Function RecursivelyDetermineAffectedResources(R, R_A) {
11    If (GetAffectedResources(R, S → E) == Ø) {
12      Return; }
13    Foreach (R' in GetAffectedResources(R, S → E)) {
14      R_A = R' ∪ R_A;
15      RecursivelyDetermineAffectedResources(R', R_A); }
16  }
```

Taking the sample cloud system mentioned in Sect. 2 as an illustrating example, its system model is represented by Fig. 3 in Sect. 3. One of its threats is "Attack the web instance AMI". This threat's attacking point is R_4. R_4 affects R_3, R_3 further affects R_1, and R_1 further affects R_5 and R_6, and hence the overall affected cloud resources of this threat are R_1, R_3, R_4, R_5 and R_6.

5.4 Calculation of Threats Impact Values

For each threat in the cloud system's infrastructure-level threats set, with its affected cloud resources determined, we are able to compute its impact value based on the significance value of each affected cloud resource. The impact value of each threat in the cloud system's threats subset, denoted as $T_S \rightarrow T_i \rightarrow I$ ($1 \leq i \leq |T_S \rightarrow \boldsymbol{T}|$), is calculated as below:

$$T_S \rightarrow T_i \rightarrow I = \sum_{j=1}^{|T_S \rightarrow T_i \rightarrow \boldsymbol{R_A}|} (T_S \rightarrow T_i \rightarrow R_A[j] \rightarrow W) \tag{9}$$

where $T_S \rightarrow T_i \rightarrow \boldsymbol{R_A}$ denotes the overall affected cloud resources set of each threat, and $T_S \rightarrow T_i \rightarrow R_A[j] \rightarrow W$ denotes the significance value of each affected cloud resource of each threat.

Taking the sample cloud system mentioned in Sect. 2 as an illustrating example, according to Sect. 5.4, the threat of "Attack the web instance AMI" affects the cloud resources of R_1, R_3, R_4, R_5, and R_6, and hence the impact value of this threat is calculated to be 19 (3 + 2 + 4 + 5 + 5).

5.5 Threats Sub-model Derivation

Upon a time tick, the detected the runtime threats and events are denoted as T_{RT}; and the threats sub-model derived from the system's threats set T_S is denoted as T'_S, which removes all the threats in T_{RT} from T_S. If T_{RT} is empty, T'_S is equal to T_S. We define GetConflictedThreats(T, $\boldsymbol{E}$) as the function to get the conflicted threats set of threat T, and define GetConflictions(T, $\boldsymbol{E}$) as the function to get the set of conflictions with regard to threat T. Then, T'_S is determined as below:

$$T'_S = (\boldsymbol{T}', \boldsymbol{E}') \tag{10}$$

$$\boldsymbol{T}' = T_S \to \boldsymbol{T} - \sum\nolimits_{i=1}^{|T_{RT} \to \boldsymbol{T}|} \text{GetConflictedThreats}(T_{RT} \to T_i, T_S \to \boldsymbol{E}) \tag{11}$$

$$\boldsymbol{E}' = T_S \to \boldsymbol{E} - \sum\nolimits_{i=1}^{|T_{RT} \to \boldsymbol{T}|} \text{GetConflictions}(T_{RT} \to T_i, T_S \to \boldsymbol{E}) \tag{12}$$

5.6 Updating of Latest Occurrence Probabilities of Threats

Now, ARA-Assessor needs to perform probability updating for each threat in T'_S based on the detected runtime threats and events, denoted as T_{RT}. The runtime threats detected are those threats that are factually occurring to the cloud system. The runtime events consist of two attacks: (1) CPU-intensive user requests explosion, which means that the attackers send excessive workload requests that significantly affect the CPU utilization of cloud instances to the cloud system; (2) data-intensive user requests explosion, which means that the attackers send excessive workload requests that significantly affect the database to the cloud system. Cloud systems are not necessarily faced with both of the two runtime events. If the cloud system only contains web servers (e.g. web instances running Tomcat service), then it can only have the runtime event of "CPU-intensive user requests explosion"; if the cloud system contains both web servers and database servers, then it can have both runtime events.

For the runtime threats detected upon a time tick, the updated occurrence probability of each detected runtime threat is set to be 1 because it has factually occurred to the cloud system and is causing certain negative consequence on the cloud system.

When either of the two runtime events happens upon a time tick, the probability of the correspondent threat must be updated, and we argue that the updated probability

(denoted as P') is relevant to the number of CPU-intensive user requests or data-intensive user requests at that time tick (denoted as W), the threshold number of CPU-intensive user requests or data-intensive user requests for the cloud system (denoted as $W_{threshold}$), and the initial occurrence probability of the correspondent threat (denoted as P). P' must satisfy three requirements: (1) P' is greater than P; (2) P' increases with W; (3) P' converges to 1. Hence, we calculate P' as below:

$$P' = 1 - \frac{1-p}{a^{(W-W_{threshold})}} \tag{13}$$

where a is a constant greater than 1, in order to make P' an increasing function (i.e. P' increases with W). The value of the constant a ($a > 1$) is determined as below:

$$a = \sqrt[\left(\frac{10000+W_{threshold}}{2}-W_{threshold}\right)]{\frac{1-p}{1-\frac{1+p}{2}}} = \sqrt[\frac{10000-W_{threshold}}{2}]{2} \tag{14}$$

5.7 The System Risk Determination Mechanism

The mechanism of calculating the risk based on T'_S is shown in Algorithm 2. We first derive a list of threats models each of which only contains the threats that do not conflict with each other (GetValidThreatsModelsList(T'_S)), then we calculate the risk value for each threats model in the list, and we return the maximum risk value as the final risk for the system. The function of GetValidThreatsModelsList(T'_S) utilizes the recursive mechanism in order to enumerate all the cases where the threats in the threat model of the system are able to occur simultaneously. Inside this function, we first obtain all the threats that do not conflict with any other threats (GetThreatswithoutConflictions(T'_S)). If the number of such threats is equal to the overall threats number, we return such threats as the output of the function; otherwise, if the number of such threats is greater than 0, we first divide the threats model into threats with conflictions (*TSubModel*) and threats without conflictions (T_W), and then we recursively call the same function (GetValidThreatsModelsList(*TSubModel*)) using *TSubModel* as the input; otherwise, if the number of threats without conflictions is 0, we loop through each threat and divide the threats model into each threat and the threats that do not conflict with it, and then we recursively call the same function to obtain the output (GetValidThreatsModelsList(T_C)) and merge it into the final output (*TMList*).

Algorithm 2: Risk Assessment Algorithm of ARA-Assessor

Input: Cloud System' Threats Model T_S' upon a Time Tick

Output: System's Risk R_S for the Time Tick

```
Function ComputeSystemRisk (T_S') {
    List<ThreatsModel> validTMList = GetValidThreatsModelsList(T_S');
    List<Double> riskValues = new List<Double>();
    Foreach (ThreatsModel T_M in validTMList) {
        riskValues.Add( ∑_{i=1}^{|(T_M → T)|} (T_M → T[i] → I) × (T_M → T[i] → P)); }
    Double R_S = Max(riskValues);
    Return R_S; }

List<ThreatsModel> Function GetValidThreatsModelsList(ThreatsModel T_S') {
    ThreatsModel T_W = GetThreatswithoutConflictions(T_S');
    If (|T_W → T| == | T_S' → T|) {
        List<ThreatsModel> TMList = new List<ThreatsModel>();
        TMList.Add(T_W);
        Return TMList; }
    Else if (|T_W → T| > 0) {
        ThreatsModel TSubModel = T_S' - T_W;
        List<ThreatsModel> TMList = GetValidThreatsModelsList(TSubModel);
        Foreach (ThreatsModel T_M in TMList) {
            T_M += T_W; }
        Return TMList; }
    Else {
        List<ThreatsModel> TMList = new List<ThreatsModel>();
        Foreach (Threat T in T_S' → T) {
            ThreatsModel T_C = T_S' – GetConflictedThreats(T, T_S' → E);
            List<ThreatsModel> tempTMList = GetValidThreatsModelsList(T_C);
            TMList += tempTMList;}
        Return TMList ; }}
```

6 Experimental Evaluation

We implemented the prototype of ARA-Assessor and evaluated it with three typical cloud systems deployed on AWS EC2 [21]. They are: (1) the "all services in one instance" cloud system; (2) the 2-tier cloud system with a production database; (3) the 2-tier cloud system with a production database and a reporting database. Each of them is a simplified version of the real-world cloud system. These three cloud systems are a

good representation of all types of cloud systems because they consider a variety of cloud resources composition scenarios of different cloud systems, so they are complete enough to verify the feasibility and generalizability of ARA-Assessor. The experimental environment is shown in Fig. 6. Since we assume we rely on external attack detection services to detect threats and events, we simulate the detection of runtime threats and events. ARA-Assessor triggers the attack detection services by simulation to obtain the simulated detection results, using the generated cloud system model as the input. The output is the cloud system's quantified ongoing runtime risk.

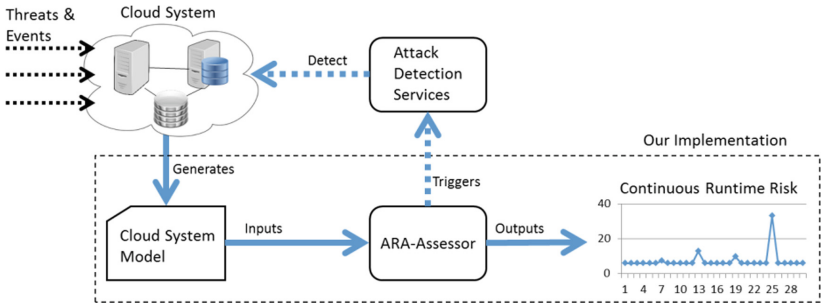

Fig. 6. Experimental environment.

6.1 Experimental Procedure

For each of the three cloud systems, the frequency of performing risk assessment is set to be 1 min. This is in accordance with the monitoring frequency of the CloudWatch function provided by AWS [24]. We simulate the running of each system for 60 min and dynamically inject one or more infrastructure-level threats and events at random time points during each system's running by simulation. The infrastructure-level threats randomly injected are able to occur simultaneously. The events can be injected by using an open source tool named Httperf [25], which is used for generating user workload requests from the client side. Since we use the free-tier cloud resources in our experiments, the allowed maximum number of CPU-intensive workload requests and the allowed maximum number of data-intensive workload requests should follow the requests number threshold of a free-tier instance. Based on our previous empirical study [26], we know that the requests threshold for a free-tier instance is 360 simultaneous requests per machine, including both CPU-intensive requests and data-intensive requests. As such, for the first cloud system, we determine that the CPU-intensive requests threshold is 180 and the data-intensive requests threshold is 180. For the second cloud system, there are initially eight free-tier web instances attached to an ASG and registered with an ELB, and one free-tier database instance shared by the web instances. So, the overall requests number threshold for the system is 2880, so the CPU-intensive requests threshold is 1440 and the data-intensive requests threshold is 1440. For the third cloud system, there are initially six free-tier web instances attached to an ASG and registered with an ELB, six free-tier reporting

instances attached to another ASG and registered with another ELB, one free-tier production database instance shared by the web instances, and one free-tier reporting database instance shared by the reporting instances. So, the overall requests number threshold for the system is 4320, and hence the CPU-intensive requests threshold is 2160 and the data-intensive requests threshold is 2160. In the real-world case, the values of the two thresholds of a real industry system should be determined by the system owner. Since we rely on external attack detection tools, we simulate the detection results of threats and events and perform risk assessment. For each of the three cloud systems we run the experiment 50 times and obtain the ongoing risk values and the average execution time of performing risk assessment. The hardware configuration of the ARA-Assessor server is: CPU-Dual Core 2.6 GHz and RAM-8 GB.

6.2 Experimental Results

The assessed risk values and the average execution time for the three cloud systems are shown in Figs. 7, 8 and 9. The results are based on 50 runs.

The risk threshold for each cloud system is determined to be 10.7, 20.5 and 36.3, respectively. Where the risk value is greater than the threshold, it is considered to be

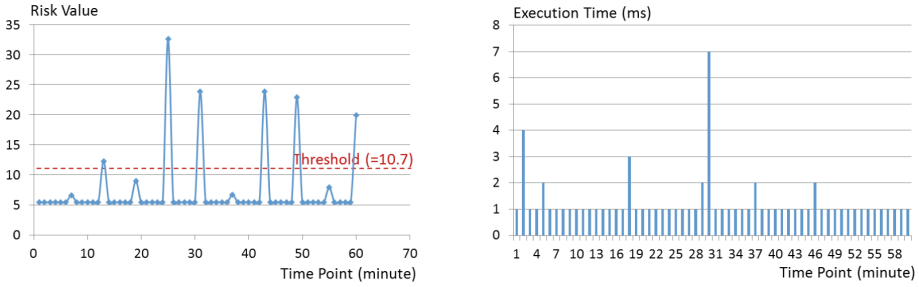

Fig. 7. Risk assessment results for the first cloud system.

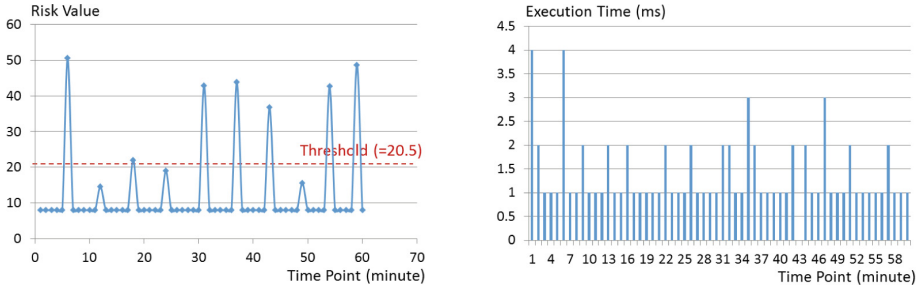

Fig. 8. Risk assessment results for the second cloud system.

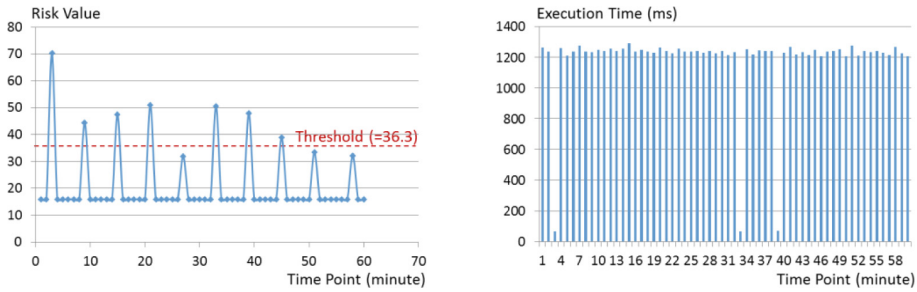

Fig. 9. Risk assessment results for the third cloud system.

high risk and appropriate responses such as performing system recovery should be considered. The average execution time of performing risk assessment for each time point for each cloud system is less than 1300 ms, which is well below the risk assessment time frequency set by the system owner (60000 ms). This also buffers enough time to perform threats and events detection for the systems. The maximum relative standard deviation for the execution time of performing risk assessment for each time point for each cloud system is 1.9%.

7 Validity and Applicability of the Model

First, while the cloud threats concerned by ARA-Assessor are only cloud infrastructure-level, we assume that the business stakeholders and the system owners are capable of figuring out the full set of all infrastructure-level cloud threats. In the implementation of our proposed method, we only consider certain types of infrastructure-level cloud threats when constructing the full cloud threats set, and we argue that by doing this it does not influence our method's validity.

Second, the determination of the initial occurrence probabilities of a cloud system's threats is only based on a limited study of existing research work, and hence the determined probabilities might not be comprehensive enough. We expect the businesses to figure out the probabilities in a more sophisticated manner.

Third, for evaluating the generalizability of ARA-Assessor, we only used three types of cloud systems. Although we argue that these three systems are a good representation of all cloud systems, it is still worthwhile to evaluate our method against more types of cloud systems. Moreover, it would be even better if we could evaluate our method with systems deployed on other cloud platforms.

8 Related Work

8.1 Risks and Threat Models in Cloud Computing

The European Network & Information Systems Agency (ENISA) [18] classifies cloud computing risks into three categories: Organizational, Technical and Legal [1]. The

organizational risks refer to "all the risks that may impact the structure of the organization or the business as an entity", e.g. "loss of business reputation due to the tenants sharing the same resources" [1]. The technical risks refer to "problems or failures associated with the provided services or technologies contacted from the cloud service provider" [1], e.g. "malicious insiders/outsiders attacks on cloud" [1]. The legal risks refer to "issues that surround data being exchanged across multiple countries that have different laws and regulations concerning data traversal, protection requirements and privacy laws" [1]. The Cloud Security Alliance (CSA) [19] lists the following threats as the top cloud computing risks: malicious insiders, data loss/leakage, abuse and nefarious use of cloud computing and shared technology vulnerabilities. From the perspective of cloud infrastructure, the cloud threats include attacks on cloud instances (virtual machines), attacks on cloud data storage and attacks on cloud networking facilities such as elastic load balancers or auto scaling groups [20, 27]. From the perspective of SaaS providers or cloud consumers, cloud threats include attacks on different application functions, attacks on the business workflows of cloud systems, and attacks on the service modules of cloud systems [28].

8.2 Existing Risk Assessment Methods for Cloud Computing

Risk is measured in terms of the consequence (or impact) and the likelihood of the attacking event or threat [29]. Researchers from the University of Leeds proposed a cloud risk assessment framework used by cloud service providers and service consumers to assess risk during service deployment and operation [8]. This framework quantitatively assesses the risks in various stages of the service lifecycle, and it considers the risks of both cloud service providers (the cloud consumers) and infrastructure providers (cloud vendors) [8]. The threat impacts are mainly determined by looking at the seven security criteria (e.g. past SLA performance) of cloud providers and the three performance criteria (e.g. past SLA performance) of cloud consumers [8]. This framework calculates risk by computing the cross-product of the threat impacts vector and the threat probabilities vector [8]. However, the main drawback of this risk assessment framework is that it does not consider the threats resulting from the on-demand nature of cloud since it largely focuses on the seven typical security evaluation criteria of various cloud infrastructure providers and the three performance criteria of the cloud consumers. Researchers from Lincoln Laboratory of MIT proposed another risk assessment tool for cloud services, which can be used for evaluating the runtime risk of particular cloud services [9]. This tool assesses system runtime risk based on analyzing a list of possible runtime threats on cloud services. The impact of each threat is determined by investigating how many virtual machines can be affected by the threat, and the probability of each threat is derived from external historical data [9]. However, the major problem with this tool is that its way of calculating threat impacts does not well capture the real natures of the consequences of security failures on cloud, and it does not fully consider the dependencies of cloud resources of the cloud system or the conflictions of cloud system threats. In comparison, our proposed method addresses all of these drawbacks. To the best of our knowledge, it is the first time that such a cloud risk assessment framework is ever proposed.

9 Conclusion and Future Work

Systems deployed on the cloud are prone to security attacks, which is one of the greatest issues with cloud computing. Cloud system risk assessment is helpful for managing and analyzing the security of cloud systems. Since existing methods of cloud system risk assessment usually do not fully consider cloud resources dependencies or cloud system threats conflictions, we proposed ARA-Assessor to continuously perform risk assessment for cloud systems. ARA-Assessor is application-aware and leverages cloud infrastructure-level threats. We implemented the prototype of ARA-Assessor and evaluated it using three typical cloud systems. Based on the experimental results, we can see that our method is able to automatically assess the runtime risk of cloud systems in a continuous manner, and it is generalizable for multiple types of cloud systems. Our future work includes: (1) include the application-level and service-level threats into our risk assessment method; (2) evaluate the feasibility of our method with more types of cloud systems and more cloud platforms.

Acknowledgement. This work is supported by Macquarie University and Data61, CSIRO. The work is partially funded by ARC DP150102966.

References

1. Dahbur, K., et al.: A survey of risks, threats and vulnerabilities in cloud computing. In: Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications (ISWSA 2011), vol. 12, April 2011
2. NIST: Resource Security. http://csrc.nist.gov/groups/SNS/cloud-computing/
3. Mather, T., et al.: Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. O'Reilly Media, Sebastopol (2009). copyright 2009, ISBN: 0596802765, 9780596802769
4. Gartner: Why private clouds fail. Network world official website. http://www.networkworld.com/article/2881794/cloud-computing/gartner-why-private-clouds-fail.html
5. Cloud Security Alliance (CSA): State of Security 2016. CSA Global Enterprise Advisory Board. https://downloads.cloudsecurityalliance.org/assets/board/CSA-GEAB-State-of-Cloud-Security-2016.pdf
6. Saripalli, P., Walters, B.: QUIRC: a quantitative impact and risk assessment framework for cloud security. In: 3rd IEEE International Conference on Cloud Computing (CLOUD 2010), July 2010
7. Heiser, J., Nicolett, M.: Assessing the security risks of cloud computing. Gartner Research Report 2008, ID no. G00157782, June 2008
8. Djemame, K., et al.: A risk assessment framework for cloud computing. IEEE Trans. Cloud Comput. **4**(3), 265–278 (2016). ISSN: 2168-7161
9. Lippmann, R.P., Riordan, J.F.: Threat-based risk assessment for enterprise networks. Lincoln Lab. J. **22**(1), 33–45 (2016)
10. Kholidy, H.A., et al.: Online risk assessment and prediction models for autonomic cloud intrusion prevention systems. In: AICCSA 2014, November 2014
11. Rahimi, M.R., et al.: MAPCloud: mobile applications on an elastic and scalable 2-tier cloud architecture. In: 5th IEEE International Conference on Utility and Cloud Computing (2012)
12. Alert Logic: The Changing State of Cloud Security. Cloud Security Report 2015 (2015)

13. Nenvani, G., Gupta, H.: A survey on attack detection on cloud using supervised learning techniques. In: IEEE Symposium on Colossal Data Analysis and Networking (CDAN 2016), March 2016
14. Lo, C., Huang, C., Ku, J.: A cooperative intrusion detection system framework for cloud computing networks. In: 39th International Conference on Parallel Processing Workshops (ICPPW 2010), September 2010
15. Zhang, T., et al.: CloudRadar: A Real-time Side-channel Attack Detection System in Clouds. Princeton University publications, Department of Electrical Engineering (2016)
16. Krutz, R.L., Vines, R.D.: Cloud security: a comprehensive guide to secure cloud computing. In: Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Wiley Publishing (2010). ISBN: 0470589876, 9780470589878
17. AWS Cloud Documentation Official Website. http://aws.amazon.com/documentation/. Last access time: 6 Aug 2017, 17:50
18. ENISA: Cloud Computing: Benefits, risks and recommendations for information security (2010)
19. CSA: Top Threats to cloud computing. v1.0 (2010)
20. Chou, T.: Security threats on cloud computing vulnerabilities. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **5**, 79–88 (2013)
21. AWS official Website. http://aws.amazon.com/. Last access time: 6 Aug 2017, 17:55
22. Symantec ISTR: Internet Security Threat Report 2016. Symantec Website, vol. 21, April 2016. https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf
23. Jouini, M., Rabai, L.B.A.: Mean failure cost extension model towards security threats assessment: a cloud computing case study. J. Comput. **10**, 184–194 (2015). doi:10.17706/jcp.10.3.184-194
24. CloudWatch Website. https://aws.amazon.com/cloudwatch/. Last access time: 6 Aug 2017, 18:40
25. Httpref Website. https://linux.die.net/man/1/httperf. Last access time: 6 Aug 2017, 18:50
26. Fu, M., et al.: Runtime recovery actions selection for sporadic operations on cloud. In: ASWEC 2015, Adelaide, Australia, pp. 185–194, September 2015
27. Sabahi, F.: Cloud computing security threats and responses. In: 3rd IEEE International Conference on Communication Software and Networks (ICCSN 2011), May 2011
28. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. **34**(1), 1–11 (2011)
29. Misra, K.: Risk analysis and management: an introduction. In: Misra, K. (ed.) Handbook of Performability Engineering, pp. 667–681. Springer, London (2008)