Threat Modeling the Cloud: An Ontology Based Approach

Salman Manzoor[1], Tsvetoslava Vateva-Gurova[1], Ruben Trapero[2], and Neeraj Suri[1]

[1] Department of Computer Science
Technische Universität Darmstadt, Germany
{salman,vateva,suri}@deeds.informatik.tu-darmstadt.de
[2] Atos Research and Innovation, Spain
ruben.trapero@atos.net

Abstract. Critical Infrastructures (CIs) such as e-commerce, energy, transportation, defense, monitoring etc, form the basis of the modern ICT society, and these CI's increasingly utilize ICT services such as the Cloud to provide for scalable, robust and cost-efficient services. Consequently, the resilience of the CI is directly connected with the resilience of the underlying Cloud infrastructure. However, performing a Cloud threat analysis (TA) is a challenging task given the complex interconnection of underlying computing and communication services. Thus, the need is of a comprehensive TA approach that can holistically analyze the relation across system level requirements and Cloud vulnerabilities.
We target achieving such a requirement based threat analysis by developing an ontology depicting the relations among actors involved in the Cloud ecosystem. The ontology comprehensively covers requirement specifications, interaction among the Cloud services and vulnerabilities violating the requirements. By mapping the ontology to a design structure matrix, our approach obtains security assessments from varied actor perspectives. We demonstrate the effectiveness of our approach by assessing the security of OpenStack, an open source Cloud platform, covering user requirements and services involved in Cloud operations.

1 Introduction

Cloud computing delivers on-demand, scalable and shared resources as "utilities" on a pay per use basis. This has led to an increased proliferation of the Cloud in diverse application spaces, and increasingly Critical Infrastructures (CIs) are utilizing it for its potential for large scale device/sensor/system connectivity, scalability, high-availability and cost-efficiency. Consequently, as the CIs security becomes dependent on the Cloud's security, the interest to conduct threat modeling and security evaluation of the Cloud has correspondingly increased and forms the primary target of this research.

One of the advocated methods for generalized (for Cloud or CI) security assurance is by performing threat analysis. Threat Analysis (TA) is an approach to

investigate potential attacks that can undermine the security of the system [16]. However, due to the complex interconnections across the services and also the diverse functional requirements from the Cloud users[3], the current Cloud threat analysis approaches typically focus on a particular service or a particular technology stack [11, 17, 21, 18]. Hence, these schemes lack providing a holistic view of Cloud security. In addition, the schemes from [20, 13] apply very useful graphical security models (e.g., attack graph/tree), but these suffer from scalability issues limiting their usage for Cloud based systems.

To address the aforementioned limitations, a comprehensive threat analysis approach is desired that can analyze the (a) relationship across different actors involved in the Cloud ecosystem, and (b) requirements and threats stemming from the violation of the requirements. Consequently, our research contributions address two main facets. First, we develop an ontology capturing the Cloud actors and relationship among the actors. The actors involved in the ontology are requirements, threats/vulnerabilities and Cloud services. For Cloud services, we consider OpenStack [15], a popular open source Cloud environment to infer the services and their interaction in performing fundamental operations such as launching a Virtual Machine (VM) on behalf of the user. Secondly, the paper models and investigates security threats from varied perspectives of the Cloud actors. For this purpose, we map the ontology to a Design Structure Matrix (DSM) [2]. Among the advantages of the DSM are scalability and applicability of different algorithms (e.g., clustering, sequencing and tearing) to perform a comprehensive threat analysis.

On this background, the specific contributions of this paper are as follows.

1. The development of an ontology based approach to identify the relation between the actors involved in the Cloud ecosystem.
2. The development of a threat analysis approach utilizing the Design Structure Matrix (DSM) to analyze threats to/from Cloud actors.

The remainder of the paper is organized as follows. Section 2 reviews contemporary threat analysis approaches for the Cloud. In Section 3 we detail the insights of the proposed ontology and perform threat analysis using design structure matrix while, Section 4 illustrates a case study on the effectiveness of the proposed approach for analyzing threats in OpenStack.

2 Related Work

Threat analysis enables the systematic identification of threats that can potentially undermine a system. The initial efforts in threat modeling and analysis, albeit at the software level, were led by Microsoft to develop STRIDE [4], a threat modeling approach applicable to data flow diagrams having the potential to explore threats in the system. In schemes [5, 6], Hiller et al. identify propagation of the data errors and their flow in the software. They introduced

[3] At a semantic level, a CI is an instantiation of a user

the concept of error permeability to measure the process of exploring software vulnerability to find software modules that have higher a significance for error propagation across the Software. Their analysis also explore suitable locations for error detection and recovery mechanisms. The scheme presented in [20] developed an attack tree of the Cloud to explore paths that an attacker can use to undermine the security of the Cloud. The authors used a high level abstraction of the Cloud services and even for such a model the scalability of the attack tree suffers. Threat analysis approaches that focus on a particular technology include [14, 17]. In [14], authors performed analysis to classify different types of threats impacting the security of the hypervisor/virtualization layer. They classified threats considering their consequence on the functionality of the hypervisor. Therefore, the threats were classified with respect to different functions of the hypervisor such as virtual CPUs management, symmetric multiprocessing, soft memory management unit, etc. Complementing this classification, authors in [17] explored security issues that could incur over VM hopping, VM mobility and VM diversity. Their assessment comprehensively covered the security threats in the virtualization layer across different Cloud deployments. In [7], the authors proposed vulcan, a vulnerability assessment framework for the Cloud. The framework developed an ontology knowledge base by extracting information from vulnerability reports published in the national vulnerability database [1]. Although, they developed a general ontology though the lack of detailed specification of the Cloud model limits the effective application of their ontology framework. In a similar vein, the authors in [19] proposed an ontology for the vulnerability management. The developed the relation between different components involved in successfully exploiting the vulnerability existing in the system. They considered a single perspective of the ontology, i.e., the relationships in the ontology were aligned towards the attacker and his/her objective of exploiting the system. Thus, the relation among the services and requirements were limited in their ontology.

Our work differs from the existing approaches in that we (a) systematically explore the multi-dimensional relations between the different actors involved in the Cloud ecosystem, and (b) perform threat analysis with respect to the relations and constraints among the actors of the Cloud.

3 Requirements Based Threat Analysis

In order to perform requirements based threat modeling of the Cloud, we develop an ontology depicting the relationships across the different actors involved in the ontology. The primary high-level actors being the abstract user, the Cloud and the threats as shown in Figure 1. In the following sections, we explain the ontology from the perspective of these actors i.e., the constraints and relations across the user, the Cloud and the threats/vulnerabilities followed by mapping the ontology to the DSM for threat analysis.

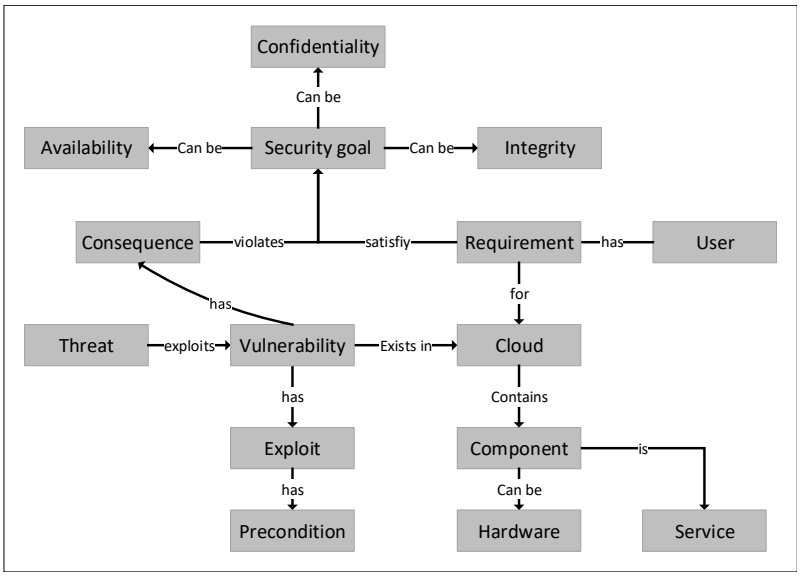

Fig. 1. Correlation among Services, Requirements and Threats

3.1 User and Requirement Capturing

This section captures the requirements of the user. The user specifies his/her requirements for the Cloud and assign weights to each requirement indicating its critically. These weights are qualitatively assigned using linguistic terms such as Highly-Critical (HC), Critical (C), Less-Critical (LC) and Not-Critical (NC). The HC requirements are more important to the user, and therefore, violation of HC requirements or threats compromising these requirements have higher significance than the critical, less-critical and not-critical requirements. As depicted in Figure 1, the requirement satisfies a specific security goal which could be maintaining Confidentiality (C), Integrity (I) and Availability (A).

3.2 Cloud Perspective of the Ontology

The Cloud is an important actor in the ontology and the user requirements describes his/her preference driving the Cloud usage. As shown in Figure 1, the Cloud contains different components including hardware and services that it offers to the user. The ontology offers a minimalistic representation of the Cloud. However, to evaluate the possible violation of the requirement across different services or multi-stage threats, a Cloud model depicting the relation between services is needed. Therefore, in Figure 2, we show essential services in the Cloud and their interactions in launching a Virtual Machine (VM). In order to create this model, we surveyed multiple open source Cloud computing platforms such as [15, 12, 10] and abstracted their essential services to create the model representing the Cloud operations.

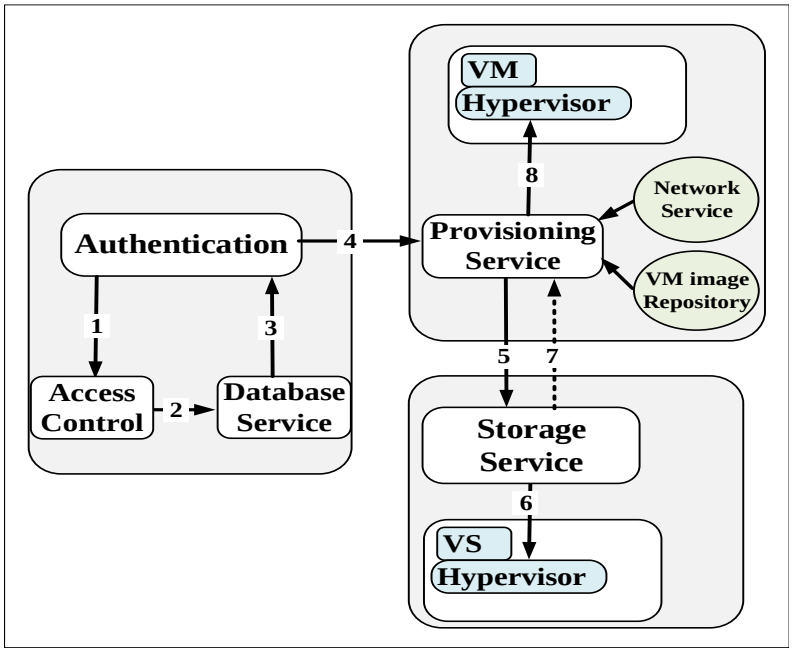

Fig. 2. Services and their interaction in the Cloud

The illustration of service relationships is essential in evaluating the violation of the requirement and propagation of the respective threat across the interacting services. The services interactions of Figure 2 are described in the following.

- Transitions 1, 2, and 3: These transitions are responsible for performing authentication and validation of the access roles of the user. This prevents the user from accessing/utilizing the resources of other tenants residing in the same physical machine.
- Transition 4: After validating user credentials the user can request provisioning service to initiate a new Virtual Machine (VM) or perform various operations on the existing user's VM. These set of operations include restarting a VM, pausing a VM, etc.
- Transition 5, 6, 7: The storage service is responsible for providing virtual storage options to the user's VM. For completeness, we have included network service and repository service in the model as these services respectively perform tasks related to network operations (such as assigning virtual network interface, mapping between the virtual and physical network interfaces, etc) and managing repository of the operating system images that can be readily used by the user for their VMs.
- Transition 8: The provisioning service communicates with the hypervisor to fulfill the user request.

We can also utilize techniques such as the state space analysis of the Cloud model to obtain an in depth analysis on the services interactions. We refer the reader to our previous works [8, 9] that target the Cloud's operational services, and also detail the services interaction using state space analysis. For this work, the abstract model presented in Figure 2 suffices as our primary objective is to perform a requirement based threat analysis and the implication of the threats across the interacting services.

3.3 Vulnerability Perspective of the Ontology

This section describes the utility of an ontology from a vulnerability perspective that exists in a Cloud component or a service. The consequence of the vulnerability exploit is to violate the security goals and cause damage to the user and the Cloud. In order to undermine the system security goals, an attacker tries to exploit the vulnerability by satisfying its pre-conditions. For this work, we restrict ourselves to these characteristics of the vulnerability without exploring the actual exploitation of the vulnerability by the attacker.

3.4 Mapping Ontology to a Design Structure Matrix

In order to perform threat analysis from the perspectives of varied actors, we map the ontology to a Design Structure Matrix (DSM). A DSM offers visualization advantage in illustrating the relations among entities by marking the respective rows and columns. Furthermore, the advantages of a DSM include multi-facet representation and reordering of a DSM to a particular perspective [3]. Thus, the DSM provides a coherent visualization of the ontology and the relationships across the actors and allows restructuring for profiling varied actors. Therefore, in Figure 3, we show the mapping of the ontology, consisting of requirements, services and vulnerabilities as the primary actors and their relationships, to a DSM.

		Requirements			Services			Vulnerabilities		
#		R 1	R 2	R 3	S 1	S 2	S3	V 1	V2	V3
1	R 1				X			X		
2	R 2					X			X	
3	R 3		X				X		X	X
4	S 1	X					X			
5	S 2		X							
6	S 3	X		X						
7	V 1	X			X					
8	V 2		X	X		X				
9	V 3			X			X			

Fig. 3. Interactions among actors of the ontology

Row# 1 in Figure 3 exhibits the relation (marked as X) between requirement $R1$, service $S1$ and vulnerability $V1$. The requirement applies to the service $S1$

and the potential vulnerability $V1$ can be used to violate the requirement by exploiting it on $S1$ and thus, undermining the security goal of $R1$. The DSM can also maintain a transitive relation, for example, row# 4 identifies a transitive relation between $R1$ and $S3$ through service $S1$. Furthermore, the interaction between $S1$ and $S3$ could be utilized in launching a multi-stage attack. Similarly, a requirement can depend on other requirement for its proper functionality. This is shown in row# 3 where requirement $R3$ depends on $R2$ and in case of $R3$ violation, the functionality of $R2$ could also suffer.

The DSM also offers varied options for partitioning and restructuring its data elements as a means of exploring inter-relations. For example, we can restructure the DSM to identify the highest influential actor, i.e., the actor with the highest dependencies or interactions. We can achieve this by reordering the DSM rows and columns to transform the DSM to a matrix that has the highest dependency/interactions at the first row and the least dependent/interactions is placed at the last row. This is achieved using the following two steps:

- Step 1: Actors with highest number of dependencies/interactions have maximum number of values (marked as X) in their respective columns and thus, placed at the top of the DSM.
- Step 2: Actors that are ad-hoc and do not provide information on other actors are placed at the bottom of the DSM. This can be identified by observing the empty columns in the DSM.

Applying these two steps to Figure 3 recursively reorders the DSM with actors having highest dependency/interactions placed at row number 1. This reordered DSM is shown in Figure 4 which shows that requirement $R3$ has the highest number of dependency/interaction among the actors. Thus, the requirement $R3$ is the most influential while the service $S2$ has the least influence on other actors involved in the ontology.

	R 3	V 2	R 1	R 2	S 1	S 3	V 1	V 3	S 2
R 3		X		X		X		X	
V 2	X			X					X
R 1					X		X		
R 2		X							X
S 1			X			X			
S 3	X		X						
V 1			X		X				
V 3	X					X			
S 2				X					

Fig. 4. Most influential actors

The advantages of such an analysis are to prioritize the vulnerability and calculate the cost of patching the vulnerability. For example, a vulnerability affecting a highly critical requirement should have the highest priority for the

patch. Alternatively, the most influential actor can be used to assess the impact of the vulnerability holistically and the system administrator can accordingly calculate the associated risk of the vulnerability.

4 Case Study: Profiling Security of the Cloud

In this section, we elaborate, using a case study, the effectiveness of the proposed ontology and DSM-based approach for profiling Cloud threats from varied perspectives of the involved actors. Table 1 presents an excerpt of the data from the actors that is used to assess the threats holistically considering the relationship among the actors. The requirements field in the table describes the user requirement, its goal and the respective priority assigned by the user. The goal indicates the security purpose of the requirement while, vulnerabilities are exploited by the attacker to undermine this security goal.

Table 1. Excerpt of the actors data for profiling threats in the Cloud

Requirements				Threats			Cloud Services		
ID	Description	Priority	Goal	ID	Impact	Description	Name (ID)	Functionality	Interconnection
R 1	Each user should have a unique user name and password to utilize Cloud services	HC	CIA	V1	CI	Incorrect timestamps comparison for tokens leads to retaining access via an expired token	Keystone (S1)	Identity and Access Management	Database Service (S2)
R 2	The data at rest should be encrypted and only the authorized user should be able to decrypt	C	A	V2	I	improper client connections handling leads to denial of service	Keystone (S1)	Identity and Access Management	Storage (S3)
R 3	The data in transfer should be encrypted	C	C	V3	A	Changing the device owner of the port leads to bypassing IP antispoofing controls.	Neutron (S4)	Network related operations	Hypervisor (S5)
R 4	The Cloud service providers should not be able to delete, modify or access user's data.	HC	CIA	V4	CIA	When using Xen as a hypervisor, attackers can obtain sensitive password information by reading log files	Hypervisor (S5)	Virtualization Management	Keystone, Storage

The vulnerabilities presented in the table are extracted from publicly available database, e.g., NIST's national vulnerability database [1]. The database discloses every vulnerability, its impact and affected products to the public. The Cloud services presented in the table are extracted from the model (cf., Section 3.2). However, we map the respective services of the model to the actual OpenStack service name. Thus, the name field in Table 1 represents the corresponding OpenStack service name performing the designated functionality. For example, *Keystone* service in OpenStack is responsible for identity and access

management. The relations among requirements, threats and services are also indicated in the table. For example, the requirement $R1$ serves multiple security purposes (CIA) for the user while, the associated threat delineate CI of the security purposes by exploiting the vulnerability on the responsible service $S1$. To comprehensively cover different aspects of the threat assessment, we create a DSM, shown in Figure 5, using the data of Table 1. The relationship among the Cloud actors is represented in the DSM by marking X in the respective row and column. For completeness, we included *goal* and *priority* for identifying threats violating a specific goal or assessing influence of the requirement in the Cloud. In the following section, we utilize reordering and restructuring of the DSM to assess influence of different actors in the Cloud.

	Requirements				Priority			Goal			Cloud Services					Vulnerabilities			
	R 1	R 2	R 3	R 4	HC	C	LC	C	I	A	S 1	S 2	S 3	S 4	S 5	V 1	V 2	V 3	V 4
R 1					X			X	X	X	X					X			
R 2						X				X	X		X				X		
R 3						X		X						X				X	
R 4					X			X	X	X					X				X
HC	X			X															
C		X	X																
LC																			
C	X															X			
I	X																X		
A	X	X																	
S 1	X	X										X	X			X	X		
S 2											X								
S 3											X								
S 4			X												X				
S 5											X		X						X
V 1	X							X	X		X								
V 2		X							X		X								
V 3			X							X				X					
V 4				X				X	X	X					X				

Fig. 5. Design structure matrix of the case study data

4.1 Extracting Influential Actors using DSM

We now illustrate how reordering the DSM (steps from Sec 3.4) can help identify the most influential actor. We rearrange the DSM by placing the most interconnected element, marked as X, to the first row and recursively perform this operation. The rearranged DSM is shown in Figure 6 having the most influential actor at the first row. The most influential component is requirement $R1$ a highly critical requirement for the user and also the most influential due to its interactions with most of the actors involved. Thus, violating this requirement or vulnerability affecting this requirement has the potential to propagate across the system due to its high degree of connections. Alternatively, from the threat analysis perspective, the DSM identifies critical aspects of the threat propagation and impact on the system. For example, vulnerability $V1$ can be used to

undermine $R1$ by compromising service $S1$. However, $S1$ can also be compromised by vulnerability $V2$ and due to $S1$ interactions with services $S2$ and $S3$, the likelihood of propagation of threats should also be assessed. The DSM can also be used to lower the number of dependencies to restrict the impact of the respective actor. On the contrary, we can also examine the least influential actor using bottom-up approach in Figure 6.

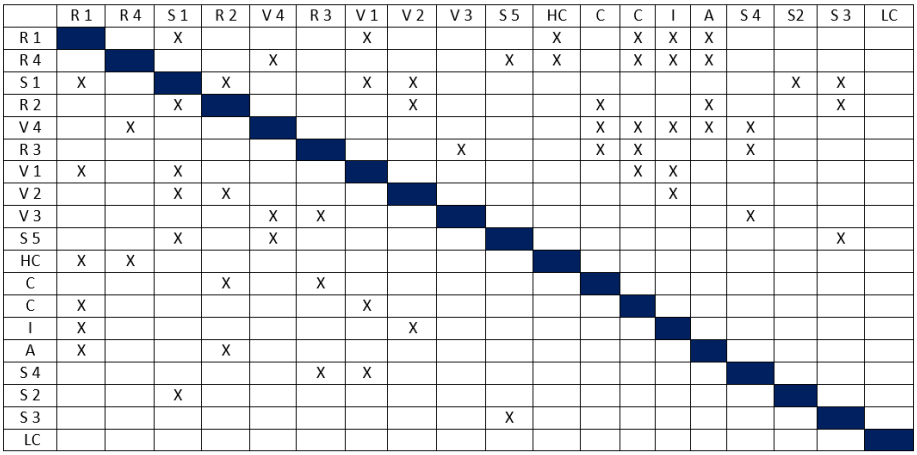

Fig. 6. Reordering to extract most influential actor.

Alternatively we can tear the DSM to analyze it from a specific perspective. For example, Figure 7 shows the perspective of tearing the DSM for analyzing threats that impact confidentiality of the system. As the figure depicts, the critical vulnerability to undermine confidentiality is $V1$ which is exploited on service $S1$. Similarly, $V1$ can be used to undermine the requirements $R1$ and $R2$. Therefore, $V1$ patching should be prioritized in order to maintain the confidentiality of the system.

Beside these two facets, we cam also reorder DSM to identify services exposed to most vulnerabilities. Similarly, the reordering can be utilized to extract threats stemming from a particular requirement.

5 Conclusions and Future Work

In this paper, we have explored the relation among different actors involved in the Cloud ecosystem to develop an ontology. This ontology is further mapped to a design structure matrix for evaluating threats from varied actors perspectives. Our DSM-based threat analysis can be utilized to identify the most critical/influential as well as least critical/influential actor in the Cloud. However,

	C	V1	S1	R1	R2	R3	R4	S4	S5
C		X				X	X		
V1	X		X	X	X				
S1									
R1									
R2									
R3	X							X	
R4	X								X
S4						X			
S5							X		

Fig. 7. from the view point of confidentiality

our DSM-based approach is flexible and thus, it can be used to reveal other critical information such as classifying vulnerabilities that achieve a common goal. We believe that by systematically identifying the Cloud vulnerabilities, the CI based on using the Cloud can consequentially be better protected.

In our future work, we will focus on improving the ontology by including countermeasures, composite vulnerabilities and more refined pre-conditions of the vulnerabilities. We will comprehensively perform threat assessment by applying different algorithms to the DSM. These algorithms include sequencing that can be used to illustrate interactions among the vulnerabilities and their propagation and tearing to limit the DSM structure to a point of interest for exploring a particular pattern/set of vulnerabilities presence in the system.

Acknowledgments

Research supported in part by grants NECS GA# 675320 and CIPSEC GA# 700378

References

1. NIST. National Vulnerability Database, https://nvd.nist.gov/.
2. S. Eppinger and T. Browning. *Design structure matrix methods and applications*. MIT press, 2012.
3. D. Gebala and S. Eppinger. Methods for analyzing design procedures. In *Proc. of Design Theory and Methodology*, pages 227–233, 1991.
4. S. Hernan, S. Lambert, T. Ostwald, and A. Shostack. Uncover security design flaws using the STRIDE approach. *MSDN Magazine*, Nov. 2006.
5. M. Hiller, A. Jhumka, and N. Suri. An approach for analysing the propagation of data errors in software. In *Dependable Systems and Networks, 2001. DSN 2001. International Conference on*, pages 161–170. IEEE, 2001.
6. M. Hiller, A. Jhumka, and N. Suri. Epic: Profiling the propagation and effect of data errors in software. *IEEE Transactions on Computers*, 53(5):512–530, 2004.
7. P. Kamongi and et al. Vulcan: Vulnerability assessment framework for cloud computing. In *Proc. of IEEE Software Security and Reliability (SERE)*, pages 218–226, 2013.

8. S. Manzoor, J. Luna, and N. Suri. Attackdive: Diving deep into the cloud ecosystem to explore attack surfaces. In *Proc. of IEEE Services Computing (SCC)*, pages 499–502, 2017.
9. S. Manzoor, A. Taha, and N. Suri. Trust validation of cloud iaas: A customer-centric approach. In *Proc. of IEEE Conference On Trust, Security And Privacy In Computing And Communications (Trustcom)*, pages 97–104, 2016.
10. D. Milojičić, I. Llorente, and R. Montero. Opennebula: A cloud management tool. *IEEE Internet Computing*, vol 15:pages 11–14, 2011.
11. S. Myagmar, A. Lee, and W. Yurcik. Threat modeling as a basis for security requirements. *Symposium on requirements engineering for information security (SREIS)*, pages 1–8, 2005.
12. D. Nurmi and et al. The eucalyptus open-source cloud-computing system. *Proc. of Cluster Computing and the Grid (CCGRID)*, pages 124–131, 2009.
13. E. Oladimeji, S. Supakkul, and L. Chung. Security threat modeling and analysis: A goal-oriented approach. *Proc. of IEEE International Conference on Software Engineering and Applications (IASTED)*, pages 13–15, 2006.
14. D. Perez-Botero and et al. Characterizing hypervisor vulnerabilities in cloud computing servers. *Proc. of the international workshop on Security in cloud computing*, pages 3–10, 2013.
15. O. Sefraoui, M. Aissaoui, and M. Eleuldj. Openstack: toward an open-source solution for cloud computing. *International Journal of Computer Applications*, 55:38–42, 2012.
16. F. Swiderski and W. Snyder. *Threat modeling*. Microsoft Press, 2004.
17. H. Tsai and et al. Threat as a service?: Virtualization's impact on cloud security. *IT Professional*, 14:32–37, 2012.
18. C. Walter, N. Suri, and M. Hugue. Continual on-line diagnosis of hybrid faults. In *Proc. of Dependable Computing for Critical Applications*, pages 233–249, 1995.
19. J. A. Wang and M. Guo. Security data mining in an ontology for vulnerability management. In *Proc. of IEEE Bioinformatics, Systems Biology and Intelligent Computing (IJCBS)*, pages 597–603, 2009.
20. P. Wang, W.-H. Lin, P.-T. Kuo, H.-T. Lin, and T. C. Wang. Threat risk analysis for cloud security based on attack-defense trees. *Proc. of Computing Technology and Information Management (ICCM)*, pages 106–111, 2012.
21. S. Winter, C. Sârbu, N. Suri, and B. Murphy. The impact of fault models on software robustness evaluations. In *Proc. of International Conference on Software Engineering (ICSE)*, pages 51–60, 2011.