ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

A SQL Injection Intrusion Detection Framework as a Service for SAAS Providers

Manu Kumar K. S[1] , Kiran G. C[2], Prof. Rajesh V[3]

U.G Student, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, Karnataka, India[1].

U.G Student, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, Karnataka, India[2].

Assistant Professor, Department of Computer Science and Engineering, Sri Krishna Institute of Technology, Bengaluru, Karnataka, India[3]

ABSTRACT: In this project, we propose SQL injection intrusion detection framework as a service for SaaS providers, SQLIIDaaS, which allows a SaaS provider to detect SQLIAs targeting several SaaS applications without reading, analyzing or modifying the source code. To achieve SQL query/HTTP request mapping, we propose an event correlation based on the similarity between literals in SQL queries and parameters in HTTP requests. SQLIIDaaS is integrated and validated in Amazon Web Services (AWS). A SaaS provider can subscribe to this framework and launch its own set of virtual machines, which holds on-demand self-service, resource pooling, rapid elasticity, and measured service properties. Thanks to the outsourcing paradigm, CC is enabling many services. Software as a Service (SaaS) is one of those cloud-based-services. Indeed, SaaS model allows providers to reduce the cost of maintenance and management by transferring traditional on premise deployment to public Cloud. Clients can subscribe, in self-service, to SaaS services based on a pay-per- use model. However, since user data are outsourced to the Cloud, serious security breaches are rising and could harm the reputation of providers and slow down the subscription of clients. SQL injection attack (SQLIA) is one of the most critical SaaS vulnerabilities that allows attackers to violate the availability, confidentiality and integrity of user data.To achieve SQL query/HTTP request mapping, we propose an event correlation based on the similarity between literals in SQL queries and parameters in HTTP requests. SQLIIDaaS is integrated and validated in Amazon Web Services (AWS). A SaaS provider can subscribe to this framework and launch its own set of virtual machines, which holds on-demand self-service, resource pooling, rapid elasticity, and measured service properties.

KEYWORDS: SQLIIDAAS (asql injection intrusion detection framework as a service for saas providers).

I. INTRODUCTION

Nowadays, Cloud computing is one of the top security threats in which attackers can make the use of vulnerabilities and system resources to make attacks. In a cloud environment, millions of users share computing resources so such attacks are very effective. In one system we are giving the solution to multiple attacks, at different stages. This is the client-server based system. Multiple clients will be connected to the server and our aim is to secure server. To protect a system from Brute force attack, if the attacker client enters the wrong password more than three times we will block that client. So, this will protect the attacker from entering into the system. In DoS attack, an attacker will try to flood the targeted resource using multiple computers and multiple Internet connections. In this system, we are using K-means algorithm for finding malicious behavior.

After finding such behavior attacker will be blocked. In SQL injection, an attacker will execute malicious SQL statements to breach the database security. An attacker can access credential data such as passwords stored in a/the

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

database using this attack. In our system, we are providing the solution to such SQL injection attack. There are many security threats that pose serious challenge towards the progress of IT economy.

Cloud Computing (CC) refers to an infrastructure as a Service (IaaS), platform as a Service (PaaS), or Software asa Service (SaaS) offered by a provider through the Internet. Several customers share and use these services according to self-service, on-demand and pay-per-use characteristics .A SaaS is a service-oriented Web application hosted in a CC environment. The platform (PaaS) of anSaaS application consists of Web and database servers that should be installed on virtual machines (IaaS). A SaaS provider can reduce maintenance and infrastructure-related management costs in traditional on-promise environments as well as start-up costsby gradually hiring compute resources in accordance with customer growth. A client or tenant benefits from the SaaS model as a subscription, and not in terms of a license, and pays only in relation to usage (e.g., number of users). A SaaS application inherits the vulnerabilities of Web applications and the service-oriented architecture (SOA) that have increasingly become victims of SQLIA. About 97% of attacks against data are still due to SQLIAs, which can allow

attacker to gain complete control of a database. The SaaS/application (39% of attacks) is the most vulnerable service model, followed by the PaaS/platform/OS. SQLIAs are the most common vulnerabilities in the SaaS layer and are consistently at the top of the OWASP's list of the most critical application security risks.

AWS offers significant protection against traditional net-work security issues such as IP spoofing and port scanning. However, it does not provide security measures for SaaS applications deployed in its infrastructure. Force.com (PaaSprovider) gives coding tips to protect against SQLIAs with out providing automatic solutions. However, the implementation of IDS for SaaS applications enables SaaS providers to increase their customers by convincing them that the confidentiality, integrity, and availability of data are well insured. An intrusion detection system (IDS) is used to detect suspicious or unauthorized activities within an information processing system

The security of the web application and the databases used in a web application is of major concern these days. The risks of attacks in these are on the increase due to a large number of web applications, web application users and sloppy security mechanisms. SQL injection attacks can be performed in many ways like modifying the data, query manipulation, data extraction etc. The SQL attacker can inject a where condition in the SQL query that always evaluates to true. This can be used to bypass authentication, extract information etc. This type of attack is called a tautology.

SQL injection attack is where the attacker inserts malicious SQL statements which could give him access to the database or the information stored in the database or harm the web application or the web application users' privacy.SQL injection attack is one of the most common and prevalent database attacks. By exploiting the vulnerability of web application and database th eattacker can get unauthorized access to the database and the web application and cause much harm. In this paper an attempt is made to develop a working model for testing and preventing the SQL injection attacks.

Through exploiting SaaS, users can pay a per usage subscription fee withoutinvesting a huge amount of money to install and maintain necessary software andhardware. Users can access the service anywhere and anytime in the world as long as theyhave a device with internet access. Users are guaranteed with the latest version ofsoftware usage without bothering with updates. For SaaS service providers, they aremotivated by an ongoing revenue to focus on improvement of their software. The multi-tenancy and virtualization technologies inherent in SaaS brings the service providers maximized resource utilization and centralized management.Detect and deter attacks can be difficult task for security administrators.Therefore, the use of IDS (Intrusion Detection system) can help both cloud providers andsecurity administrator in order to monitor and analyze network traffic. The goal of usingsuch system is to detect and prevent attacks, using different algorithm of detection.Nevertheless, analyzing and monitoring symptoms produces a multitude alarms with ahuge number of false ones impacting the effectiveness of such systems.

THE PRINCIPLE OF SECOND-ORDER SQL INJECTION

The first-order SQL injection loadV attack payload directly to the SQL query, but the attack process of second-order SQL injection is divided into 2 stages[4], namely the stage of storing attack payload to database or file system and the stage of building the SQL query statement with the attack payload from database or file system. For example, the common function of user registration and user information modification in Web application can carry out an second-order SQL injection attack, the user registration function loads the attack payload to the database, and then the user

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology
An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018
2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"
13th-14th March 2018
Organized by
Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

information modification function extract the attack payload from database to construct a SQL database query, which will cause a second-order SQL injection .
The key codes of user registration function are as follows:

```
$link=new mysqli("localhost", "root", "root")
$query="INSERT INTO Users VALUES('?','?','?')
$cmd=$1ink->prepare($qurey)
 $cmd->bind_param("sss",$usemame,$password,$email)
 $cmd->execute
```

The above codes use precompiled statements to insert a user record to the database instead of dynamicly constructing SQL query. When DBMS processes a precompiled statement, it first compiles the SQL command by placeholders instead of the user input, and then enters the user input to execute. Since the compilation has completed, even if the user input contains attack load, it will not be parsed by the database, this is a method of defense against the first-order SQL injection. For example, now insert a user record with the email field "123' WHERE 1= (updatexml (1, concat (0x5e24 (select password from admin limit 0x5e24, 1, 1)))); -- ",because the user registration function uses precompilation statements to insert a user record, the attack payload will be treated as general data and stored into the database.
However, the attack payload stored in the database will be reused when modifying the user's information. The user data modification function of Web application will verify the username and password of the user before changing the password, and extract the corresponding user data from database to memory, the SQL instructions of verification and extraction of user data list as follows:

```
$query="SELECT * FROM Users WHERE username='? ' AND password='? ' "
 $result = mysql_query($query)
```

Web application authenticates user by verifying whether the returned result set is empty, and saves the result set in memory. Web application uses the result set in memory to construct SQL statement for modifying user's passwords, the key code is shown as below.

```
$result=mysql_query($query)
$row=mysql_fetch_array($result)
$usemame=$row[usemame"]
$email=$row["email"]
$password=$_POST["newpwd"]
$query = " UPDATE Users SET username = ' $username '
And Password = ' $password ' , email = ' $email ' WHERE usemame='$username'
 $rs=mysql_query($query)
```

The above code uses the username and email in memory and the new input password of the user to construct the UPDATE instruction. At this point, the attack payload stored in the email field in database is dynamically constructed the SQL query, the new constructed SQL instructions is shown below.
UPDATE Users SET username='XXX',password = 'XXX', email = '123' WHERE l=(updatexml(concat(0x5e24 (select password from admin limit 1)0x5e24)1)-- 'WHERE username='XXX'.
When the DBMS executes the above SQL instruction, a database error is generated because the XPATH string received by the updatexml () function has a syntax error, the error message is shown below: XPATH syntax error:'$admin123$'
From the error prompt message, the attacker can know the password of user admin is admin123. A second - order SQL injection attack is successfully implemented. From the above example, we can see that the second-order SQL injection is divided into 2 stages: storage and trigger. In the storage phase, the attack payload is wrote to the user's input and send to the Web application, because Web application has deployed the defense measure, when the INSERT/UPDATE statement is executed by DBMS, the attack payload will be stored in the database. In the trigger phase, Web application will construct a new SQL instruction with the attack payload stored in the database. When the DBMS execute the newly constructed SQL instruction of the Web application, the attack payload is parsed and executed, and a second-order SQL injection is implemented.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

III. OVERVIEW OF THE LITERATURE

The HTTP analysis-based techniques require human intervention and maintenance and can produce high level of false alerts. The SQL analysisbased approaches are unable to determine the source of malicious SQL queries (Web session, URL, IP address), since they do not deal with HTTP requests. The authors of some paper propose anomaly-based IDS that combine the analysis of HTTP and SQL traffic in order to reduce the errors and false positive during the detection. Recently, many Cloud IDS have been proposed a NIDS is deployed in different regions of the Cloud to reduce the risk of a distributed denial of service attack (DoS) by exchanging alerts. A security model has been proposed to allow an IaaS provider to protect its infrastructure (e.g., VM Monitor) and the VMs of its clients against the DoS and the installation of illegal tools. The authors of some paper propose a high-level architecture of signature-based, distributed, and collaborative IDS to protect the infrastructure (IaaS). Snort (signature-based NIDS) is installed in AWS to detect unauthorized access and suspicious FTP files.Cloud Computing (CC) refers to an infrastructure as a Service (IaaS), platform as a Service (PaaS), or Software as a Service (SaaS) offered by a provider through the Internet. Several customers share and use these services according to self-service, on-demand and pay-per-use characteristics.

A SaaS is a service-oriented Web application hosted in a CC environment. The platform (PaaS) of anSaaS application consists of Web and database servers that should be installed on virtual machines

A SaaS provider can reduce maintenance and infrastructure-related management costs in traditional on-promise environments as well as start-up costs by gradually hiring compute resources in accordance with customer growth. A client or tenant benefits from the SaaS model as a subscription, and not in terms of a license, and pay only in relation to usage (e.g., number of users)A SaaS application inherits the vulnerabilities of Web applications and the service-oriented architecture (SOA) that have increasingly become victims of SQLIA. About 97% of attacks against data are still due to SQLIAs, which can allow an attacker to gain complete control of a database. The SaaS/application (39% of attacks) is the most vulnerable service model, followed by the PaaS/platform/OS.

SQLIAs are the most common vulnerabilities in the SaaS layer and are consistently at the top of the OWASP's list of the most critical application security risks. AWS offers significant protection against traditional network security issues such as IP spoofing and port scanning. However, it does not provide security measures for SaaS applications deployed in its infrastructure. Force.com (PaaS provider) gives coding tips to protect against SQLIAs without providing automatic solutions. However, the implementation of IDS for SaaS applications enables SaaS providers to increase their customers by convincing them that the confidentiality, integrity, and availability of data are well insured.

This undesired condition is utilized by assailants to infuse issues and malignant code into the framework that permits them to run their own code and applications. For better understanding vulnerabilities the production of models that express the set conditions that could lead or begin them is extremely useful; moreover when models are surely knew they could likewise be utilized for counteractive action. Be that as it may, since it is difficult to ensure the nonappearance of vulnerabilities in a bit of code amid its creation, then it is important to have techniques to recognize them.

An intrusion detection system (IDS) is used to detect suspicious or unauthorized activities within an information processing system. Two approaches are used for intrusion detection: misuse and anomaly detection. The misuse approach is to build signatures of known attacks that must be updated regularly by an administrator. Any activity that matches a signature is considered as an attack. The anomaly approach is in two phases: the learning phase that models the normal behavior of the entity to protect (e.g., network, system, application, service) and the detection phase that generates an alert when a deviation from the normal behavior is observed.

Here the attacker closes the current authentication statement and then piggy backs another SQL query which can extract information, delete or drop any other table from the database. Using this attack the attacker can delete the permissions table and gain access to any database tables.Many proposed SQLIA countermeasures analyze the HTTP and/or SQL traffic. However, all these approachesare not available or evaluated in a public Cloud to protecttheSaaS applications. Recently, many Cloud IDS are proposed to allow an IaaS provider to protect the infrastructure IaaS (e.g., VM Monitor) against network attacks (e.g., DoS and DDoS). These Cloud IDS focus on the infrastructure layer without ensuring the security of SaaS applications against SQLIAs, since they do not monitor the input (HTTP) and/or output (SQL) of these applications.

An IDS can be classified according to the source of information that is inspected: host (HIDS), network (NIDS). The HIDS is installed on a physical host or virtual machine (VM) to analyze data from packets. The NIDS is installed on

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

strategic nodes in a network to analyze the header of packets. The most of proposed SQLIA countermeasures require costly analysis and/or modifications of the source code and consequently they are valid for a specific language or environment (e.g., J2EE, PHP).

IV. PROBLEM STATEMENT AND SOLUTION DOMAIN

A SaaS provider can reduce maintenance and infrastructure-related management costs in traditional on-promise environments as well as start-up costs by gradually hiring compute resources in accordance with customer growth. A client or tenant benefits from the SaaS model as a subscription, and not in terms of a license, and pays only in relation to usage (e.g., number of users) A SaaS application inherits the vulnerabilities of Web applications and the service-oriented architecture (SOA) that have increasingly become victims of SQLIA.
About 97% of attacks against data are still due to SQLIAs, which can allow an attacker to gain complete control of a database. The SaaS/application (39% of attacks) is the most vulnerable service model, followed by the PaaS/platform/OS.SQLIAs are the most common vulnerabilities in the SaaS layer and are consistently at the top of the OWASP's list of the most critical application security risks.

Drawbacks
Iit does not provide security measures for SaaS applications deployed in its infrastructureForce.com (PaaS provider) gives coding tips to protect against SQLIAs without providing automatic solutions. Drawback: However, the implementation of IDS for SaaS applications enables SaaS providers to increase their customers by convincing them that the confidentiality, integrity, and availability of data are well insured

PROPOSED SYSTEM
In this project, we propose a framework, SQLIIDaaS, which detects SQL injection targeting SaaS applications deployed in public Clouds (e.g., AWS). SQLIIDaaS provides the HTTP traffic (e.g., HTTP request, Web session), SQL traffic (e.g., SQL query), and the correlation between HTTP requests and SQL queries.

This information is required for the detection of SQLIAs that target SaaS applications without modifying or analyzing their source code. SQLIIDaaS is integrated in AWS, which allows a SaaS provider to reduce maintenance and infrastructure costs by pooling computing resources to serve multiple clients or applications simultaneously (resource pooling).
About 97% of attacks against data are still due to SQLIAs, which can allow an attacker to gain complete control of a database. The SaaS/application (39% of attacks) is the most vulnerable service model, followed by the PaaS/platform/OS.SQLIAs are the most common vulnerabilities in the SaaS layer and are consistently at the top of the OWASP's list of the most critical application security risks.SQLIIDaaS is offered as a set of pre-built Amazon Machine Images (AMI). In this way, a SaaS provider can subscribe to this framework on-demand and the resources of framework (VMs) can be scaled (rapid elasticity) and measured (pay-per-use).

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 7, Special Issue 6, May 2018

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

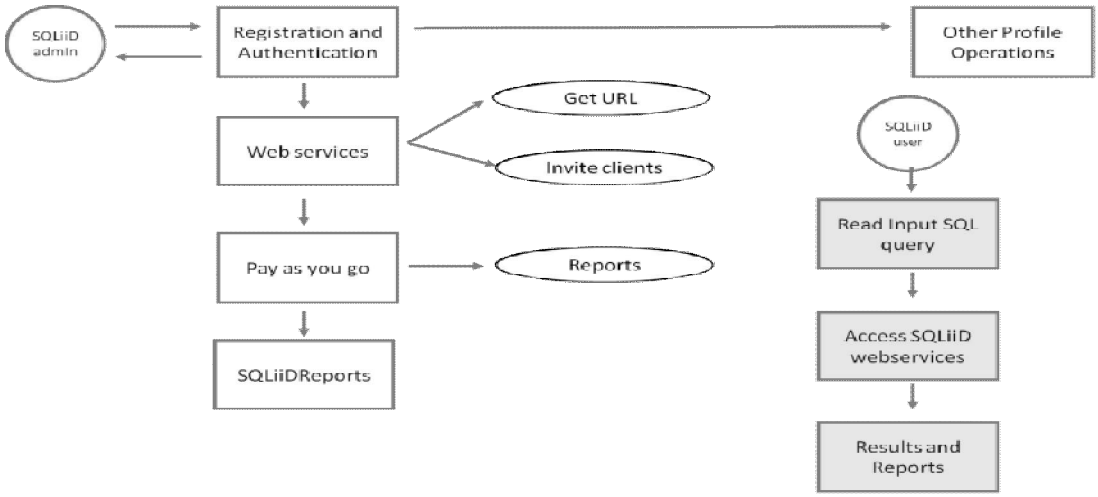

Fig. 1.SQLiD Operations

Advantages of the proposed system

1. The proposed solution can be used by SaaSproviders in a service-based manner with high level of portability.
2. The proposed work will be a Language independent solution.Doesn't requires modification of the source code.

V. CONCLUSION

we propose SQL injection intrusion detection, which allows a SaaS provider to detect SQLIAs targeting several SaaS applications without reading, analyzing or modifying the source code. To achieve SQL query/HTTP request mapping, we propose an event correlation based on the similarity between literals in SQL queries and parameters in HTTP requests. SQLIID is integrated and validated in Amazon Web Services (AWS). A SaaS provider can subscribe to this framework and launch its own set of virtual machines, which holds on-demand self-service, resource pooling, rapid elasticity, and measured service properties. The proposed solution can be used by SaaS providers in a service-based manner with high level of portability. The proposed work will be a Language independent solution. Doesn't requires modification of the source code. Reusable by any IaaS providers Proposed solution is language independent Proposed solution doesn't requires any modification to the source code Maintenance cost will be reduced, time saving. SQLIID is offered as a set of pre-built Amazon Machine Images (AMI). In this way, a SaaS provider can subscribe to this framework on-demand and the resources of framework (VMs) can be scaled (rapid elasticity) and measured (pay-per-use). A SaaS provider can reduce maintenance and infrastructure-related management costs in traditional on-promise environments as well as start-up costs by gradually hiring compute resources in accordance with customer growth.Web application advancement and utilizations are expanded radically because of their accessibility and trust over the framework which could be remotely measured. This trust and uses are increment as a result of theirhigh security and lessened load. However, amid the most recent decade this online applications gets influenced with SQL infusion and cross webpage scripting (XSS) assaults.

This is on the grounds that the information situated web applications have the client's information and some place it connects with the content based client fields and orders. This open range is utilized by the vindictive client to get into the framework utilizing the above characterized assault groupings. Customary methodologies which are working towards location are expending substantial framework stack. This paper works towards enhancing the customary assault location system with enhanced exactness and diminished calculation stack. The recognition and assault moderation is totally mechanized with the proposed lead based assault location having specifically fields for keeping the vindictive script to be embedded into the framework. Scientific assessments and future bearing of the work will demonstrate the same with an executed framework.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 7, Special Issue 6, May 2018*

2nd National Conference on "RECENT TRENDS IN COMPUTER SCIENCE & INFORMATION TECHNOLOGY (RTCSIT'18)"

13th-14th March 2018

Organized by

Departments of Computer Science & Information Science Engineering, Sri Krishna Institute of Technology, Bangalore, India

REFERENCES

[1] P. Mell and T. Grance, "The nist definition of cloud computing"National Institute of Standards and Technology.Special Publication 800-145, pp. 1–7, 2011.
[2] W.-T. Tsai, X. Sun, and J. Balasooriya"Service-oriented cloud computing architecture," in Information Technology: New Generations (ITNG),2010 Seventh International Conference on. IEEE, 2010, pp. 684–689.
[3] L. Wu, S. K. Garg, S. Versteeg, and R. Buyya, "Sla-based resource provisioning for hosted software-as-a-service applications in cloud computingenvironments" Services Computing, IEEE Transactions on, vol. 7, no. 3pp. 465–485, 2014.
[4] P. K. Chouhan, F. Yao, and S. Sezer, "Software as a service: Understanding security issues," in Science and Information Conference (SAI),2015. IEEE, 2015, pp. 162–170.
[5] S. Curtis, "Barclays: 97 percent of data breaches still due to sqlinjection,"http://www.techworld.com/news/security/barclays-97-percent-of-data-breaches-still-due-sql-injection-3331283/, 2015.
[6] C. Cerrudo, "Manipulating Microsoft SQL Server Using SQL Injection,"Application Security, INC., Tech. Rep., 2010
[7] S. Subashini and V. Kavitha, "A survey on security issues in servicedelivery models of cloud computing," Journal of Network and ComputerApplications, vol. 34, no. 1, pp. 1–11, 2011.
[8] J. Williams and D. Wichers, "Owasp: Top 10 - 2013, owasp foundation,"2013.
[9] K. Scarfone and P. Mell, "Guide to intrusion detection and preventionsystems (idps)," NIST special publication, vol. 800, p. 94, 2007.
[10] T. Probst, E. Alata, M. Kaaniche, and V. Nicomette, "Automated evaluation of network intrusion detection systems in iaas clouds," inDependable Computing Conference (EDCC), 2015 Eleventh European.IEEE, 2015, pp. 49–60.
[11] W. G. J. Halfond and A. Orso, "Preventing SQL injection attacksusing AMNESIA," Proceeding of the 28th international conference onSoftware engineering - ICSE '06, pp. 1–4, 2006.