Journal of Intelligent & Fuzzy Systems 31 (2016) 749–764
DOI:10.3233/JIFS-169007
IOS Press

Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns

Gaik-Yee Chan[a], Fang-Fang Chua[a] and Chien-Sing Lee[b,*,1]
[a]*Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, Cyberjaya, Malaysia*
[b]*Faculty of Science and Technology, Sunway University, Sunway City, Selangor, Malaysia*

Abstract. Cloud computing inherits all the systems, networks as well as Web Services' security vulnerabilities, in particular for software as a service (SaaS), where business applications or services are provided over the Cloud as Web Service (WS). Hence, WS-based applications must be protected against loss of integrity, confidentiality and availability when they are deployed over to the Cloud environment. Many existing IDP systems address only attacks mostly occurring at PaaS and IaaS. In this paper, we present our fuzzy association rule-based (FAR) and fuzzy associative pattern-based (FAP) intrusion detection and prevention (IDP) systems in defending against WS attacks at the SaaS level. Our experimental results have validated the capabilities of these two IDP systems in terms of detection of known attacks and prediction of new variant attacks with accuracy close to 100%. For each transaction transacted over the Cloud platform, detection, prevention or prediction is carried out in less than five seconds. For load and volume testing on the SaaS where the system is under stress (at a work load of 5000 concurrent users submitting normal, suspicious and malicious transactions over a time interval of 300 seconds), the FAR IDP system provides close to 95% service availability to normal transactions. Future work involves determining more quality attributes besides service availability, such as latency, throughput and accountability for a more trustworthy SaaS.

Keywords: Intrusion detection, intrusion prevention, software as a service, fuzzy association rule, web service

1. Introduction

Cloud computing, as defined by the National Institute of Standards and Technology (NIST) of the US Department of Commerce [18], is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage, applications, and services, that can be rapidly provisioned and released with minimal management effort or service provider interaction. This Cloud model is composed of three service models, namely, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

Under this Cloud model, at the PaaS layer, platform is provided for the deployed applications and possibly configuration settings for the application hosting environment. The provision for processing, storage, networks, and other fundamental computing resources for the deployed applications is the capability of the IaaS. What is promised at the SaaS layer is that the consumer is able to use the provider's

[1]The corresponding author was a former faculty at Universiti Tunku Abdul Rahman, Malaysia when the research was conducted.

*Corresponding author. Chien-Sing Lee, Faculty of Science and Technology, Sunway University, Sunway City, 47500 Selangor, Malaysia. Tel.: +60 3 74918622; E-mail: chiensingl@sunway.edu.my.

applications running on a cloud infra-structure on-demand. At all these three layers, consumers are provided with the services on a pay-per-use basis hence benefiting from cost savings in terms of resources being shared and less effort spent on management of these resources.

However, along with these benefits are security issues because Cloud computing inherits all the systems, networks as well as Web Services' security vulnerabilities, in particular for SaaS, where business applications or services are provided over the Cloud as Web Services (WS). As mentioned in [27] that the concerns about data security and trust have become a significant barrier for many organizations to adopt SaaS as a solution. Many organizations or users feel fear of data leakage and loss of privacy through the use of SaaS over the Cloud platform. Hence, Web and WS-based applications, such as e-commerce applications must be protected against loss of integrity, confidentiality and availability when they are deployed over to the Cloud environment, be it over the private, public, community or hybrid Clouds, as Software as a Service (SaaS).

Recently, vigorous research has been carried out to find solutions to counter vulnerabilities and attack found in SaaS, PaaS and IaaS. Due to the distributed nature of the Cloud environment, attacks such as Denial of Service (DoS) which are distributed (DDoS), HTTP and XML-based DDoS are found to be more destructive than the traditional DDoS [23]. These research have found that traditional firewalls, network intrusion detection and prevention (IDP) systems are not adequate to defend against DDoS, XML-DoS and HTTP-DoS attacks. Moreover, Web and WS-based applications are vulnerable to attacks such as SOAP oversized payloads, coercive parsing, SQL injection and XML injections that could not be effectively and efficiently defended against by network intrusion detection and prevention systems. Web and WS-based applications deployed over the Cloud environment as SaaS, therefore, require another line of defense at the application level to counter measure against XML and SOAP related attacks.

According to [22], firewalls are used as entry points for the Cloud and client servers. As such, the number of HTTP or SOAP requests and responses could be tracked. The average response time for each request is then calculated. Experimental results show that as the number of HTTP and XML-DDoS attacks increases, the load balancer has to distribute the load to more instances, thus incurring additional cost for the extra instances.

In our prior research [9], we have developed IDP systems incorporated within an e-commerce application deployed over a network environment, one based on 20 fuzzy association rules (FAR) and the other based on 336 fuzzy associative patterns (FAP), to effectively and efficiently defend against WS attacks in close to real-time with detection rate of greater than 99%. In this paper, we extend our investigation on the use of fuzzy logic, associative pattern matching and association rules for the detection and prevention of existing attacks (signature-based) as well as prediction of new attacks (anomaly-based) to a WS-based e-commerce application deployed over a Public Cloud as SaaS. Our research question is whether our IDP systems, (FAR and FAP), are able to effectively and efficiently protect the SaaS against WS attacks over the Cloud environment? If yes, how will FAR and FAP perform in terms of detection rate, transaction time and load balancing?

This paper is organized as follows: Section 2 gives an overview of related work, Section 3 describes the IDP system and deployment of the IDP system over the Cloud platform, Section 4 presents testing and performance evaluation results, Section 5 compares FAR IDP's performance with other IDP systems and Section 6 concludes and discusses future work.

2. Related work

Many businesses have realized the benefits of Cloud computing technology which allows them to gain fast access to software or deploy applications over the Cloud environment without drastically changing or managing their platform and infrastructure resources with negligible cost. However, security remains the main hurdle for the wide acceptance of Cloud computing technology. Many enterprises still feel reluctant to deploy their business applications and services to the Cloud platform due to complications in protecting the confidentiality, integrity and availability of information and data transmitted across the Clouds. Security vulnerability occurs at three service delivery models, i.e., SaaS, PaaS and IaaS. Therefore in recent years, vigorous research have been carried out to address the security vulnerabilities inherent mainly in PaaS and IaaS independently.

While PaaS and IaaS inherit mainly the systems and networks' vulnerabilities, intrusion detection research has focused on addressing HTTP anomalies, HTTP-DoS and DDoS using host and network-based

ID systems. These ID systems are mostly signature-based (which identify known attacks only), and/or anomaly-based (which is able to identify new attacks) and do not address attacks related to WS and XML for SaaS. Some of these systems for example, make use of approaches such as Heuristic Semi-Global Alignment Approach (HSGAA) to detect DoS attacks; filtering tree and trace-back techniques to identify suspicious IP addresses; *apriori* algorithm to generate possible new signatures of network attacks; multi-thread and pre-defined rule set to detect DDoS; pattern matching and neural network techniques in detecting Trojan Horse and DoS attacks and so on. However, these researches have demonstrated their IDSs' capabilities theoretically without performance validation. Table 1a provides further details [5, 7, 12, 17, 19, 23, 25].

There are other research [2–4, 6, 8, 14] that have demonstrated and validated the capabilities of their Cloud-based ID systems with satisfactory results in detecting mainly network DoS and DDoS. These (Table 1b) however, are not attacks related to WS and XML. Cloud applications are mainly WS-based. Intrusion detection and prevention systems for SaaS should, therefore, provide mechanisms to defend against XML and SOAP-related attacks such as XML injections, XML-DoS or DDoS, SOAP oversized payloads, coercive parsing and so on. As seen from Table 1a and b, traditional firewalls, network and host-based ID systems are not adequate to defend against WS attacks. These intrusion detection and prevention (IDP) systems are network and host-based and address the security vulnerability identified mainly at PaaS and IaaS layers in-dependently from each other.

The above review shows that there is an urgent need to provide solutions to reduce, and if possible to eradicate entirely, the security challenges found in Cloud computing. As the challenge with SaaS security is no different than with any other Web applications [21], SaaS therefore, requires another line of defense at the application level to counter-measure against XML and SOAP-related attacks.

The use of fuzzy logic or fuzzy reasoning in obtaining accurate prediction results have been proven effective in many researches. For example, research in [20] has proposed a fuzzy reasoning and the ensemble method to obtain prediction accuracy as high as 96.35%. In another research in [16], a fuzzy logic based defense mechanism is proposed to dynamically define rules according to network traffic pattern of the cloud environment so as to detect malicious packets to mitigate the DDoS attack with false alarm rate as low as 0.14%. Research in [24] has proposed the use of fuzzy association rules in real time detection of Web Service DoS attack with satisfactory results. We, therefore, put forward our fuzzy association rule intrusion detection and prevention (FAR IDP) system intended for Web and WS-based applications to defense against WS and XML-related attacks for SaaS as well.

Fuzzy logic, unlike Boolean logic which corresponds only to 'true' or 'false' value, is many-valued logic whose 'degree of truth' ranges between 0 and 1. When input data are quantitative, for example in our prior work in [10], input size and SOAP size in bytes, they could be transformed to fuzzy sets through fuzzification to smooth out the change between boundaries. These fuzzified attributes are then mapped to meaningful linguistic labels to carry out the intrusion detection function.

Subsequently, association rule mining is used for discovering interesting relations between sets of fuzzy attributes. In our prior work in [11], detailed analysis and observation of the fuzzified data sets have led to the discovery of associative patterns characterized by seven attributes and then the derivation of fuzzy association rules (refer to Section 3 for further details of these fuzzy patterns and rules).

Association rule mining is based on the market basket analysis's concept of discovering strong rule through measure of interestingness. For example, if a customer buys paper and pencil, most likely he will also buy eraser, thus forming the association rule $\{\text{paper, pencil}\} => \{\text{eraser}\}$. However, how valid or significant is this rule shall depend on the support and confidence imposed. If the item-set $\{\text{paper, eraser, pencil}\}$ has a support of 80%, this means that out of ten purchases, the item-set appears eight times. If the rule $\{\text{paper, pencil}\} => \{\text{eraser}\}$ has a confidence of 99%, this means 99 out of 100 times, the customer who buys paper and pencil shall also buy eraser. Thus, a simple rule-of-thumb to identify strong and interesting rule is to check that the support of its consequent equal to the support of its ante-cedent and achieving the minimal confidence level.

3. The FAR/FAP IDP system and deployment in the cloud platform

Our prior work in [11] had led to 336 fuzzy associative patterns (Table 2) being formed and the derivation of 20 fuzzy association rules (Table 3) for

Table 1a
ID/IDP systems addressing networks and systems' attacks without performance evaluation

No.	Approach & Description	Performance & Remarks	Sourced From
1.	**Signature *apriori* Algorithm + Snort** This NIDS uses Snort for detecting network intrusions, and signature *apriori* algorithm to generate new possible signatures. It is able to detect DoS/DDoS attacks in Cloud offering laaS.	Detection results show low false positive rate and within reasonable computational cost. But there is no experimental results to support these theoretical claims.	[5]
2.	**Boyer-Moore (BM) algorithm + Rules Set** The IDS uses an improved BM algorithm in a high-speed network environment to reduce space complexity by 36% in pattern matching of intrusion signature.	The improved BM algorithm is tested by Snort which captures and matches the packets with 274 rules for Web data among network packets. There is no performance evaluation results for detection rate.	[7]
3.	**Heuristic Semi-Global Alignment Approach(HSGAA)** Cloud IDS and Host-based IDS located at each node to cooperate with each other to identify local security violation events. By exchanging audit data using HSGAA technique to detect attacks at PaaS layer.	The proposed approach is to detect DoS, buffer overflow and masquerade attacks, not XML related attacks. Moreover, there is no performance evaluation or validation results.	[12]
4.	**Multi-Threaded Model + Rule Set** A multi-threaded Cloud NIDS to handle large flow of data packets, analyze them against signature base and a pre-defined rule set for detection of DDOS and XSS.	There is no performance evaluation results to validate its effectiveness and efficiency. Detection not cover XML related attacks and there is no discussion on prevention.	[17]
5.	**Single Controller + Neural Network** The IDS uses a single controller to manage instances. The controller query the Knowledge based (KB) where patterns of user's profile are stored. The KB uses neural network to learn new patterns for detection of access right violation, Trojan Horse and DoS.	A NIDS in a distributed Cloud computing environment for detection of non XML related attacks. There is no performance evaluation results also.	[19]
6.	**Filtering Tree + Trace Back** The approach uses ftltering tree technique to filter suspicious IP addresses. Suspicious IP addresses are stored in a Trace-Back module. A Cloud Defender then detects for HTTP or XML DDoS attacks.	The approach focuses on detection of Cloud API vulnerabilities, e.g. SOAP coercive parsing. HTTP and XML DDoS attacks. No preventive measure being mentioned and no performance evaluation results.	[23]
7.	**Virtual Machine Monitor (VMM) + VM Intrusion deteCTOR (VICTOR)** The proposed technique is to differentiate attack traffic originating from each virtual machine even if multiple virtual machines on a VMM are sharing a single IP address. VICTOR is to identify and isolate the compromised VM that is generating the attack flow.	Demonstrated techniques for securing virtual machines from DDoS and worms attacks in laaS. No performance results to validate the effectiveness and efficiency of the proposed techniques.	[25]

the evaluation of incoming patterns. The capability of the fuzzy association rule-based (FAR) model has been tested and proven to be able to achieve nearly 100% detection and prediction rate with less than 1% false alarm. Experiments conducted using random forest as classifier has shown that the FAR model is able to achieve small Root Mean Square Error (RMSE) of 0.02 with time to build model within 0.02 seconds for each data set based on a sample size of greater than 600 test records.

In our more recent work in [9], we incorporated the 20 fuzzy association rules at one instance and 336 fuzzy associative patterns in another instance, within a WS-based e-commerce application. Thus, two IDP systems, one is the FAR IDP system and the other the FAP IDP system, are developed. Both are deployed over a network environment for performance evaluation of effectiveness and efficiency. Testing results have shown that both IDP systems are able to determine whether to certainly allow access for normal transaction, probably deny access for suspicious transaction or definitely deny access to transactions containing malicious inputs or XML content. It is proven through experiments that both the FAR IDP and FAP IDP systems are able to detect, prevent and predict Web service attacks such as SQL injection, XML injection, DoS and SOAP oversized at close to real-time, detection rate not lower than 99% and a slight difference in terms of transaction time.

In this research, our WS-based e-commerce application incorporated with a fuzzy association rule-based intrusion detection and prevention (FAR IDP) system and a fuzzy associative pattern-based

Table 1b
ID/IDP systems addressing attacks at PaaS/IaaS/SaaS with performance evaluation

No.	Approach & Description	Performance & Remarks	Sourced From
1.	**pre-dEcisioN, advance Decision, lEaRning system (ENDER)** Use a pre-mark decision method to detect attack traffic and label the attack. Added decision making and update method then make another decision about the possibility of the message not being classified correctly. The labeled message is then removed before damage is done.	The IDS detects HTTP-DoS and XML-DoS with 99% detection accuracy and 1% false positive rate. There is preventive measure in protecting the victim. However, technique in the defense against XML injection and SOAP oversized payload is not discussed.	[2]
2.	**Cloud trace back + flexible deterministic packet marking algorithm** The IDS detect DDoS attacks occur at the laaS layer. The IDS is validated using the DARPA (KDD99) data set.	Experimental results have shown a 91% detection accuracy. The IDS does not detect and prevent WS attacks and does not provide preventive measures in countering DDoS attacks.	[3]
3.	**Cooperative Agent + Blocking Rules** This framework consists of IDSs within the Cloud computing regions to exchange their alerts with each other. These cooperative agents compute and determine whether to accept the alerts sent from other agents. By this way, Dos & DDoS attacks could be prevented.	For this IDS, the computation time per packet is 0.00006 seconds more and the detection rate is 0.2% less than that of Snort IDS. However, this cooperative IDS is able to prevent the Cloud service from single point of failure attack.	[4]
4.	**Bayesian algorithm + Snort** This network IDS detects DoS attack and other network level malicious activities in Cloud offering laaS. Bayesian algorithm classifies attack by observing previously stored network events while Snort detects known attacks.	10% of KDD ' 99 intrusion detection dataset is used as training data for Bayesian classifier. The network IDS is able to obtain a detection rate of 96.00% with less than 1. 5% false positive rate	[6]
5.	***n*-grams modeling of web requests** The IDS determines normal behavior of the HTTP requests in training phase. Detection phase identifies anomalies.	Detection accuracy close to 100% with a false alarm rate of close to 1%. The IDS detect HTTP anomalies at the SaaS layer but not for detection of WS attacks.	[8]
6.	**Severity Analysis + C4.5 Decision Tree Algorithm** An intrusion detection and severity analysis system deployed at a border node to monitor multiple virtual machines for the detection of DoS & DDoS attacks occurring at laaS.	VM specific parameter, frequency of attack is used in the analysis together with C4.5 algorithm for classification. Experimental results show detection rate to be over 90%.	[14]

Table 2
Fuzzy associative patterns

Transaction Time		T2			T3				
Attributes		User ID	Password		Input values	Input size	SOAP size	XML content	Decision
Patterns #	1.	Valid	Valid	1.	Valid	Normal	Matched	Non malicious	C Allow
				2.	Valid	Normal	Not matched	Malicious	C Deny
				3.	Valid	Normal	Extremely not matched	New	C Deny
				4.	Malicious	Normal	Matched	Non malicious	C Deny
	2.	Valid	Malicious	5.	Malicious	Out-of-range	Matched	Non malicious	C Deny
	3.	Valid	New	6.	Malicious	Extremely out	Matched	Non malicious	C Deny
	4.	Malicious	Valid	7.	Malicious	Normal	Not matched	Malicious	C Deny
	5.	Malicious	Malicious	S.	Malicious	Out-of-range	Not matched	Malicious	C Deny
	6.	Malicious	New	S.	Malicious	Extremely out	Extremely not matched	New	C Deny
	7.	Malicious	Invalid	10.	New	Normal	Matched	Non malicious	C Deny
	8.	New	Valid	11.	New	Out-of-range	Matched	Non malicious	C Deny
	9.	New	Malicious	12.	New	Extremely out	Matched	Non malicious	C Deny
	10	New	New	13.	New	Normal	Not matched	Malicious	C Deny
	11	New	Invalid	14.	New	Out-of-range	Not matched	Malicious	C Deny
	12	Invalid	Malicious	15.	New	Extremely out	Extremely not matched	New	C Deny
	13	Invalid	New	16.	Invalid	Normal	Not matched	Malicious	C Deny
				17.	Invalid	Out-of-range	Not matched	Malicious	C Deny
				18.	Invalid	Extremely out	Extremely not matched	New	C Deny
	14	Invalid	Invalid	19.	Invalid	Normal	Matched	Non malicious	P Deny
	15	Invalid	Valid	20.	Invalid	Out-of-range	Matched	Non malicious	P Deny
	16	Valid	Invalid	21.	Invalid	Extremely out	Matched	Non malicious	P Deny

Table 3
Fuzzy association rules

Rule No.	Attributes						Decision
	UserID	Password	Input Values	Input Size	Soap Size	XML Content	
1.	Valid	Valid	Valid	–	Matched	–	C Allow
2.	Valid	Valid	Valid	–	–	Non Malicious	C Allow
3.	Valid	Valid	Valid	Normal	Matched	–	C Allow
4.	Valid	Valid	Valid	Normal	–	Non Malicious	C Allow
5.	Valid	Valid	Valid	–	Matched	Non Malicious	C Allow
S.	Valid	Valid	Valid	Normal	Matched	Non Malicious	C Allow
7.	Valid	Valid	Invalid	–	Matched	Non Malicious	P Deny
8.	Valid	Valid	Invalid	Normal	Matched	Non Malicious	P Deny
9.	Valid	Valid	Invalid	Out-of-range	Matched	Non Malicious	P Deny
10.	Valid	Valid	Invalid	Extremely-out	Matched	Non Malicious	P Deny
11.	Valid	Invalid				–	P Deny
12.	Invalid	Invalid				–	P Deny
13.	Malicious	–				–	C Deny
14.	New	–				–	C Deny
15.	–	Malicious				–	C Deny
16.	–	New				–	C Deny
17.	–	–	Malicious			–	C Deny
18.			New			–	C Deny
19.	–	–	–		Not-matched	Malicious	C Deny
20.	–	–	–	–	Extremely not-matched	New	C Deny

detection and prevention (FAP IDP) system are deployed independently over to a public Cloud platform with.NET Framework Version 4.5 [26] for testing and performance evaluation of effectiveness and efficiency.

Referring to Fig. 1a, individual desktop, laptop or mobile phone users and users from different private, public, community or hybrid Clouds, can access to our systems through different network mix, such as wired LAN, wireless or mobile and different browser mix such as Internet Explorer, Chrome, Firefox, Pocket IE and Safari. This platform and configuration with different mix of network and browsers form the basis environment for FAR and FAP IDP systems performance and load testing.

Referring to Fig. 1b, the Web service-based e-commerce application with the FAR/FAP IDP system together with the databases are bundled and deployed over to the Public Cloud. Users access the application using Internet through HTTP and the Web service request or XML message is transmitted through SOAP. Under this platform and configuration, the FAR/FAP IDP system is able to perform with very satisfactory results, such as close to 100% real-time (within 5 seconds) detection, prevention and prediction of WS attacks and close to 95% service availability for normal transactions at a work load of 5000 concurrent users over a time interval of 300 seconds.

4. Performance of FAR/FAP IDP systems in the cloud platform

Over in the Cloud platform, both the FAR and FAP IDP systems performed effectively. They detected and prevented all existing known malicious signatures (There were about three hundred (300) existing or known malicious signatures listed in appendix D of [1] that include Web service attacks mentioned earlier), and predicted new or unknown type of attacks on a real-time basis with low false alarm rate. Table 4 shows five representative examples of normal transaction, probably deny access transaction and certainly deny access transactions.

As for efficiency in terms of time performance, experiments are set up to capture the transaction time for each scenario represented in Table 4 for FAR and FAP IDP systems. Experimental results are then tabulated for further analysis. Section 4.1 provides further details.

In order to evaluate the performances of these IDP systems, two different sets of experiments are conducted to observe how the IDP systems behave under stressed conditions in the Cloud environment, yet able to carry out detection, prevention and prediction functions. The first set of experiment is to stress testing the IDP systems with different users' loads, each transacting within a time constraint in fixed length of time. Section 4.2 provides further details. The other set of

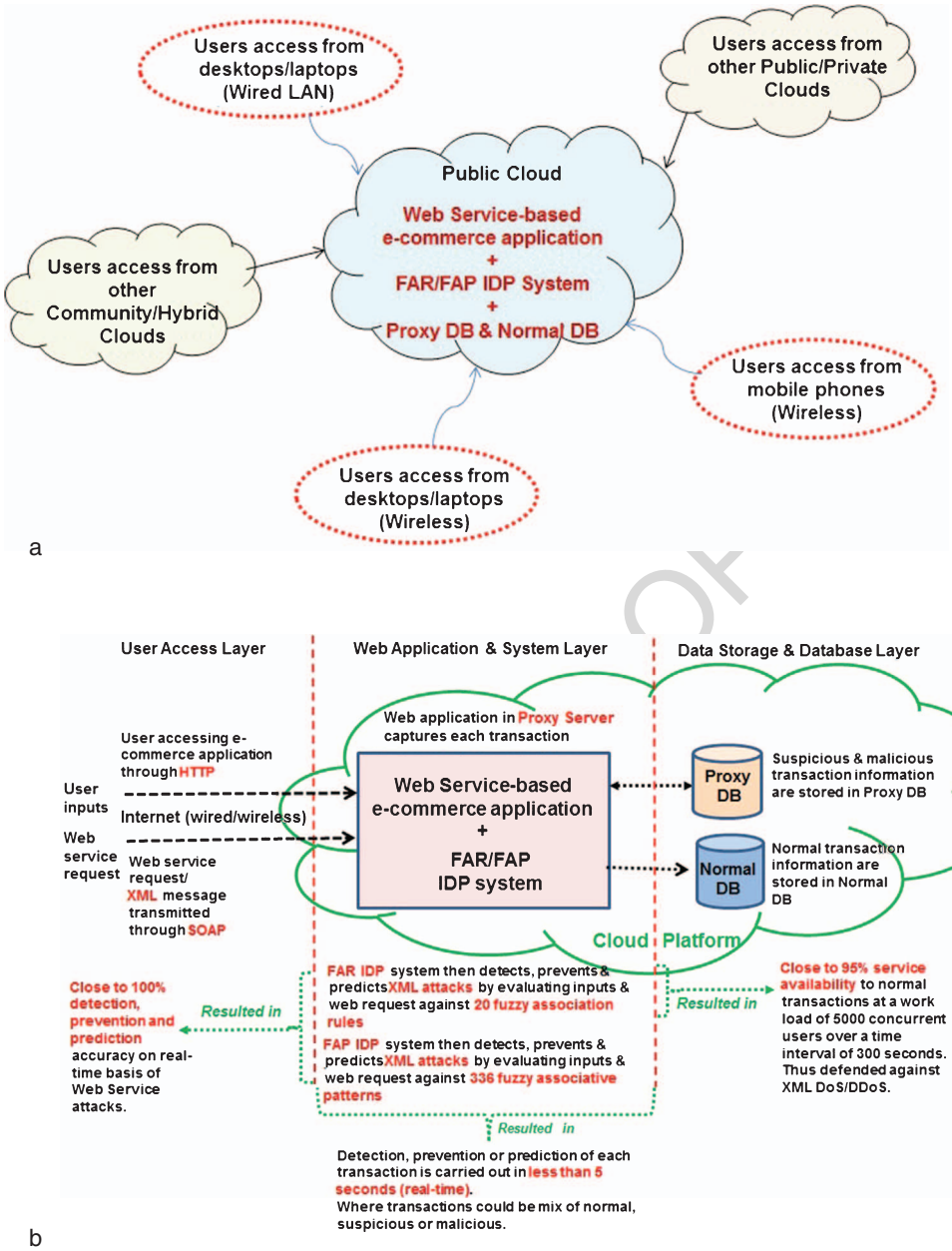


Fig. 1. (a) An overview of access and usage of FAR IDP system deployed over the Cloud platform. (b) FAR/FAP IDP systems deployed over the Cloud platform.

experiment is to stress testing the IDP systems with different users' loads, each transacting with a volume constraint of a fixed number of transactions with no restriction in time. Experimental results obtained are tabulated for further analysis. Section 4.3 provides further details. Based on the experimental results, it is observed that there is a slight difference in time and load performances between the two IDP systems. Explanation on the differences can be obtained from Section 4.4.

4.1. FAR IDP vs FAP IDP in transaction time

To determine the efficiency of FAR IDP and for comparison with the FAP IDP, experiments are set up to test the performances of five scenarios in which

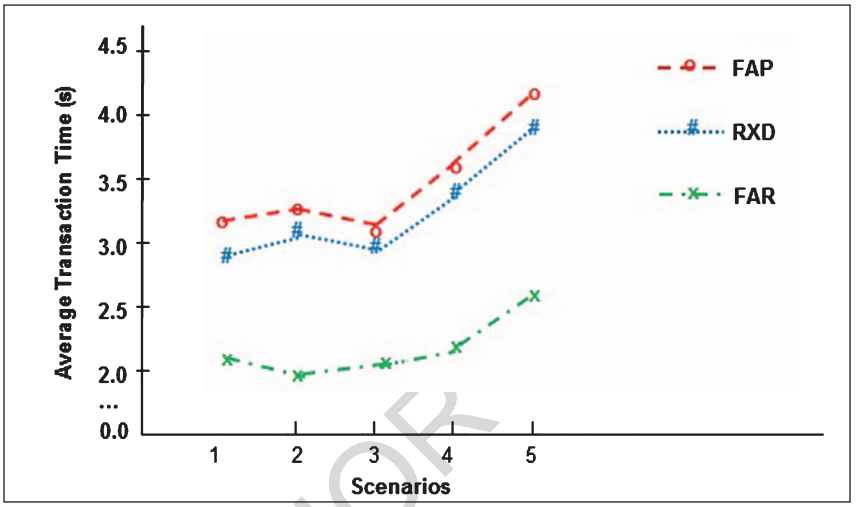
Table 4
The five representative scenarios

Case #	Input values	XML Contents	Input Size /SOAP size/ Decision	[1]Patterns/[2]Rules matched
1	**UserID:** gychan **Password:** gvchan **Credit Card#:** 8642880000825858 **3-digitpin:** 858 **Amount:** 999.90 **Email:** gaikgaik@gmail.com	<soap:Body><InsertPayment xmlns = "lhttp://tempuri.org/"><card>8642880000825858</card><pin>858</pin><payment>999.90</payment><email>gaikgaik@gmait.com</email></InsertPayment></soap:Body></soap:Envelope> **Normal transaction**	Input size: 43 SOAP size: 406 UserID Valid Password Valid Input values Valid Input size Normal Soap Size Matched XML content Non Malicious Decision C allow	Pattern Matched: **T2P1T3P1** Rule Matched: Rules 1–6
2	**UserID:** gvchan **Password:** Gvchan **Credit Card#:** NULL **3-digitpin:** NULL **Amount:** NULL **Email:** NULL	NULL **Invalid password**	Input size: NULL SOAP size: NULL UserID Valid Password Invalid Input values NULL Input size NULL Soap Size NULL XML content NULL Decision P deny	Pattern Matched: **T2P16T3P1** (*Assuming T3P1 transaction is normal*) Rule Matched: Rule 11
3	**UserID:** gychan **Password:** 'hi' or 'x' = 'x'; **Credit Card#:** NULL **3-digitpin:** NULL **Amount:** NULL **Email:** NULL	NULL **SQL Injection**	Input size: NULL SOAP size: NULL UserID Valid Password Malicious Input values NULL Input size NULL Soap Size NULL XML content NULL Decision C deny	Pattern Matched: **T2P2 T3P1** Rule Matched: Rule 15
4	**UserID:** gychan **Password:** gychan **Credit Card#:** 8642880000825858 **3-digitpin:** 858 **Amount:** 999.90 **Email:** abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890@YAHOO.COM	NULL **Input size overly large:** **Buffer Overflow**	Input size: 179 SOAP size: 542 UserID Vaiid Password Valid Input values New Input size Extremely-out Soap Size Matched XML content Non malicious Decision C deny	Pattern Matched: **T2P1 T3P12** Rule Matched: Rule 18
5	**UserID:** gychan **Password:** gychan **Credit Card#:** 8642880000825858 **3-digit pin:** 858 **Amount:** 999.90 **Email:** gaikgaik@gmail.com	soap:Body><InsertPaymentxmIns = "http://tempuri.org/"><card>1234432112344321</card><script>alert<"hi")</script><fool>XSSattack!</fool><pin>123</pin><payment>80.50</payment><email>ggggggg@gmail.com</email></InsertPayment></soap:Body></soap:Envelope> **Soap Oversized (XSS)**	Input size: 41 SOAP size: 462 UserID Valid Password Valid Input values Valid Input size Normal Soap Size Not Matched XML content Malicious Decision C deny	Pattern Matched: **T2P1 T3P2** Rule Matched: Rule 19

[1]For fuzzy associativa patterns, refer to Table 2. [2]For fuzzy association rules, refer to Table 3.

Scenario	Transaction Time (s) (FAR)		Transaction Time (s) (FAP)		Transaction Time (s) (RXD)	
	Range of Values (10 tests)	Average Value (10 tests)	Range of Values (10 tests)	Average Value (10 tests)	Range of Values (10 tests)	Average Value (10 tests)
1	1.84~2.56	2.08	2.8~4.99	3.16	2.45~3.38	2.91
2	1.69~2.12	1.88	2.704~3.60	3.19	2.85~3.31	3.03
3	1.95~2.71	2.03	2.701~3.62	3.05	2.66~3.49	2.98
4	2.01~2.42	2.19	2.95~4.38	3.55	3.19~3.97	3.46
5	2.25~2.74	2.51	3.6~5.00	4.12	3.73~4.05	3.88

a



b

Fig. 2. Performances of FAR and FAP IDP Systems.

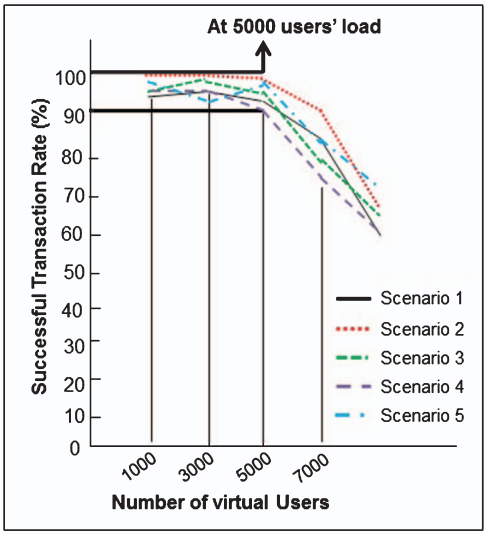
each scenario corresponds to each case of Table 4 (scenarios 1–5 correspond to cases 1–5 in the same order). For example, scenario 1 corresponds to case 1 which represents normal transaction. Scenario 2 corresponds to case 2 which represents probably denying access due to invalid password is entered. Scenario 3 corresponds to case 3 which is a certainly denying access transaction due to SQL injection attack. Scenario 4 corresponds to case 4 where access is certainly denied because the input size is too large and causes a buffer overflow. Scenario 5 corresponds to case 5 which is a certainly denying access transaction due to SOAP is oversized with malicious XML content. Each scenario is tested ten times through the Cloud platform first with fuzzy association rules (FAR) IDP system then followed by the fuzzy associative patterns (FAP) IDP system. These tests are conducted using a machine with Intel x64-based processor (i5-4210 CPU@ 1.7 GHz with 4.00 GB RAM) through the internet with 10Mbps having Windows 8.1 as operating system with Visual Studios Ultimate 2013 as the test tool running under the Microsoft.NET Framework Version 4.5.

The performances of each system based on the five scenarios are shown in Fig. 2 with tabulated results (Fig. 2a) and graphs (Fig. 2b). As seen from Fig. 2a that our FAR IDP system performs more efficiently in which each transaction is completed within three seconds (Fig. 2a Column 2) with average transaction times ranging from 1.88∼2.51 seconds (Fig. 2a Column 3). It can be seen that the FAP IDP system performs less efficiently than our FAR IDP system whereby each transaction is completed within five seconds (Fig. 2a Column 4) and the average transaction times ranging from 3.05∼4.12 seconds (Fig. 2a Column 5). It can be seen from Fig. 2b that the two systems behave in a similar manner (the two graphs show similar curves where the gaps between the curves are the time differences). To further confirm that a transaction model shall display similar behav-

Successful Transaction Rate for Different Users Load (FAR)

Scenario No.	Successful Transaction Rate (%)			
	1000 Users	3000 Users	5000 Users	7000 Users
1	94.11	94.70	94.56	85.29
2	98.31	96.70	96.02	91.74
3	94.94	95.95	92.27	83.48
4	94.85	94.31	91.94	79.80
5	96.49	92.14	93.88	85.07

a

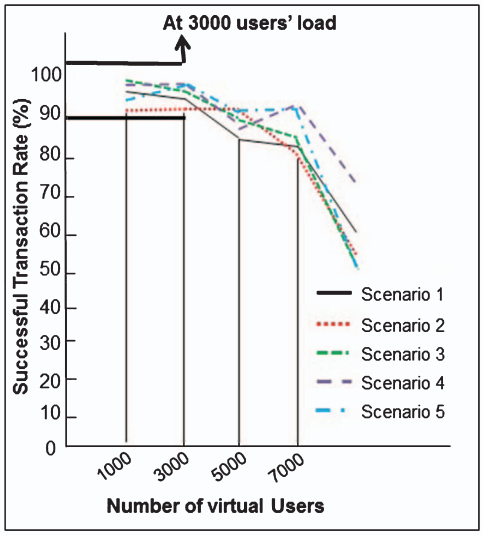


b

Successful Transaction Rate for Different Users Load (FAP)

Scenario No.	Successful Transaction Rate (%)			
	1000 Users	3000 Users	5000 Users	7000 Users
1	94.10	92.66	86.80	86.78
2	90.93	91.69	91.77	84.09
3	94.72	93.41	90.91	86.79
4	94.49	95.05	87.24	91.67
5	93.99	94.78	88.26	88.37

c



d

Fig. 3. Load test results of FAR and FAP IDP systems.

ior, a transaction model (RXD) with rules and patterns excluded is incorporated and its performance evaluated. As seen from Fig. 2 (Fig. 2a Columns 6-7 and Fig. 2b) that RXD's behavior resembles very closely to that of FAP IDP's and it performs slightly better than FAP IDP system but much less efficiently than the FAR IDP system. Explanation on why these systems behave in such a way can be found in Section 4.4.

4.2. *Performance of the IDP systems in the Cloud platform with time constraint*

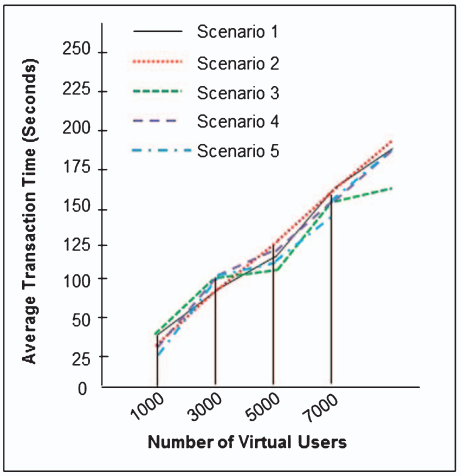
Load or volume testing is conducted with the FAR and FAP IDP systems in order to obtain the optimum performance or limits of the systems when performing under different users load over in the Cloud platform. Experiments are set up with the five scenarios based on the five cases of Table 4 in the ratio of 40:15:15:15:15 for each test run. This means for each user's load, 40% of the transactions represent normal transaction, 15% of the transactions represent invalid user inputs, the next 15% of the transactions represent malicious inputs, the third 15% of the transactions represent inputs that contain long strings and the final 15% of the transactions represent transaction with malicious XML content. For each test run, simulation is performed using the same machine, operating system, platform and test tool that carried out performance testing mentioned in Section 4.1 but with different users' load of 1000, 3000, 5000, and 7000. Additionally, for each test run and each user's load, simulation is carried out in a fixed time frame of 300 seconds only. However in reality, the percentage for normal transactions would be much higher and the malicious transactions would be very low such as 1% or less. Moreover, the time frame for voluminous users transacting over the Cloud platform would not be limited to only 300 seconds in real-life situation. Our experiment is set up in this manner so as to discover how far our FAR and FAP IDP systems can go when under stress (time constraint) in the Cloud platform.

From our experiments, it can be seen that in terms of load or volume under this stressed condition, the system with fuzzy association rules (FAR) and fuzzy associative patterns (FAP) behave similarly. Refer to

Average Transaction Time for Different Users Load (FAR)

Scenario No.	Average Transaction Time in Seconds			
	1000 Users	3000 Users	5000 Users	7000 Users
1	31.97	76.87	120.82	167.83
2	26.64	76.89	125.22	164.65
3	31.97	81.75	111.50	152.97
4	25.85	82.39	126.99	164.07
5	25.09	83.75	118.91	166.19

a

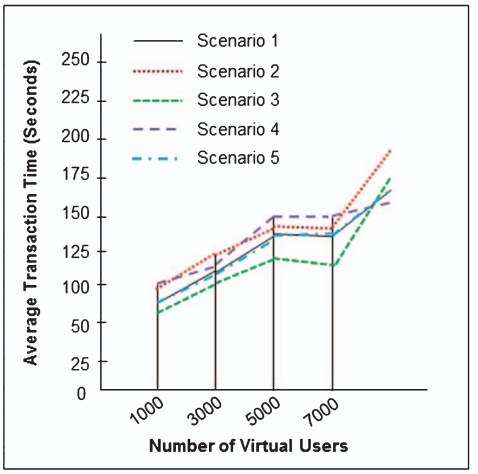


b

Average Transaction Time for Different Users Load (FAP)

Scenario No.	Average Transaction Time in Seconds			
	1000 Users	3000 Users	5000 Users	7000 Users
1	80.39	111.86	133.39	132.09
2	89.87	123.50	136.66	130.57
3	66.68	96.16	116.05	115.32
4	92.53	116.44	142.17	142.19
5	77.67	111.91	134.51	129.99

c



d

Fig. 4. Average transaction time for different users' load for FAR and FAP IDPs.

Fig. 3a for FAR IDP and Fig. 3c for FAP IDP, it is found that when the load increases, the percentage for successful transactions for the 5 scenarios decreases. For example, for FAR IDP, it can maintain a range of 91.94∼98.31% (Fig. 3a Columns 2–4) of successful transaction rate at a user's load of up to 5000. Beyond the 5000 users' load, the successful transaction rate decreases sharply (Fig. 3b). Notice also that there is a slight inconsistency in results at the point with user's load of 7000 (Fig. 3a and c, Column 5). The occurrence of these phenomena may be due to network or host's machine resource constraint such as background processes increase for bottle neck or backlogs as the number of user's load increases. As such, comparison of results, are based on user's load of 1000, 3000 and 5000 only. The 5000 user's load is set as the point for optimum performance for FAR IDP. Although behave similarly, the FAP IDP system performs less efficiently under the same stressed condition. For FAP IDP, it can maintain a range of 90.93∼95.05% (Fig. 3c Columns 2–4) of successful transaction rate at a user's load of up to 3000 only. Beyond the 3000 users' load, the successful transaction rate as expected decreases greatly (Fig. 3d) due to resource constraint.

It is mentioned in [23] that Cloud hosted Web application crashes at approximately 3000 concurrent user sessions under a DDoS attack in about 20 minutes. Our experimental results as mentioned above has proven that even with a work load of 5000 concurrent users submitting transactions with some of the transactions being malicious or suspiciously malicious within 300 seconds, our FAR IDP system can still perform and normal transactions are protected against attacks with a success rate of close to 95%. In other words, the WS-based e-commerce application incorporated with our FAR IDP system and deployed in the Cloud platform as SaaS is being protected against DDoS and yet providing 95% availability service for normal transactions.

More experiments have been conducted to further observe the behavior of the system with the fuzzy association rules and fuzzy associative patterns. Refer to Fig. 4, it is observed that the system with FAR and FAP exhibit similar behavior. When the user's load increases, the average transaction time also increases, obviously due to race condition where increasing the load without increasing the processing speed for example. However, the average transaction time for user's load for FAR still perform faster than that of

Successful Transaction Rate for Different Users Load (FAR)
(Max 2,500 transactions with no fixed time frame)

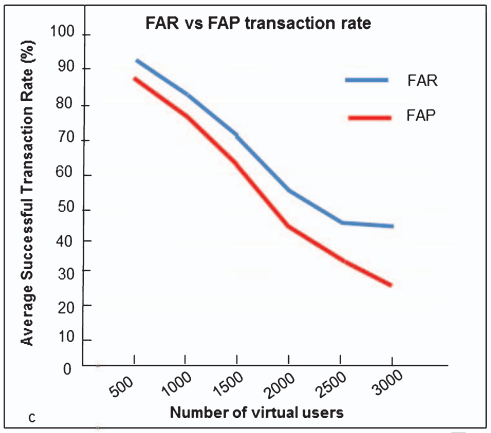
Scenario No.	Successful Transaction Rate (%)					
	500 Users	1000 Users	1500 Users	2000 Users	2500 Users	3000 Users
1	90.82	82.65	69.78	57.49	46.78	45.49
2	93.55	84.08	76.20	58.73	45.18	46.02
3	92.15	81.15	71.69	59.06	48.47	50.14
4	89.42	81.31	65.90	61.63	45.53	41.55
5	92.77	82.14	71.92	55.94	46.58	50.13
Average rate (%)	91.52	82.36	70.72	58.24	46.56	46.40
Time *(min)*	14:11	18:29	20.28	23.12	23.41	23.40

a *Note: all values are rounded off to 2 decimal place from the original values*

Successful Transaction Rate for Different Users Load (FAP)
(Max 2,500 transactions with no fixed time frame)

Scenario No.	Successful Transaction Rate (%)					
	500 Users	1000 Users	1500 Users	2000 Users	2500 Users	3000 Users
1	89.78	76.54	61.42	45.83	34.07	26.71
2	87.47	79.56	59.26	50.94	31.73	29.23
3	91.90	81.28	57.70	44.82	30.14	22.76
4	89.72	79.16	63.19	50.00	30.63	27.75
5	90.50	79.49	62.81	47.79	31.02	22.94
Average rate (%)	89.80	78.56	62.08	47.40	32.20	26.04
Time *(min)*	15:10	19:33	26:48	28:56	28:18	27:10

b *Note: all values are rounded off to 2 decimal place from the original values*



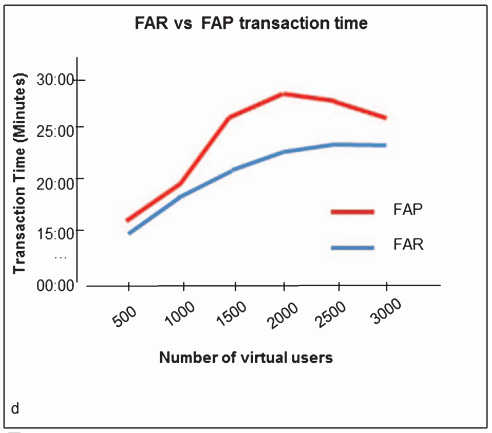
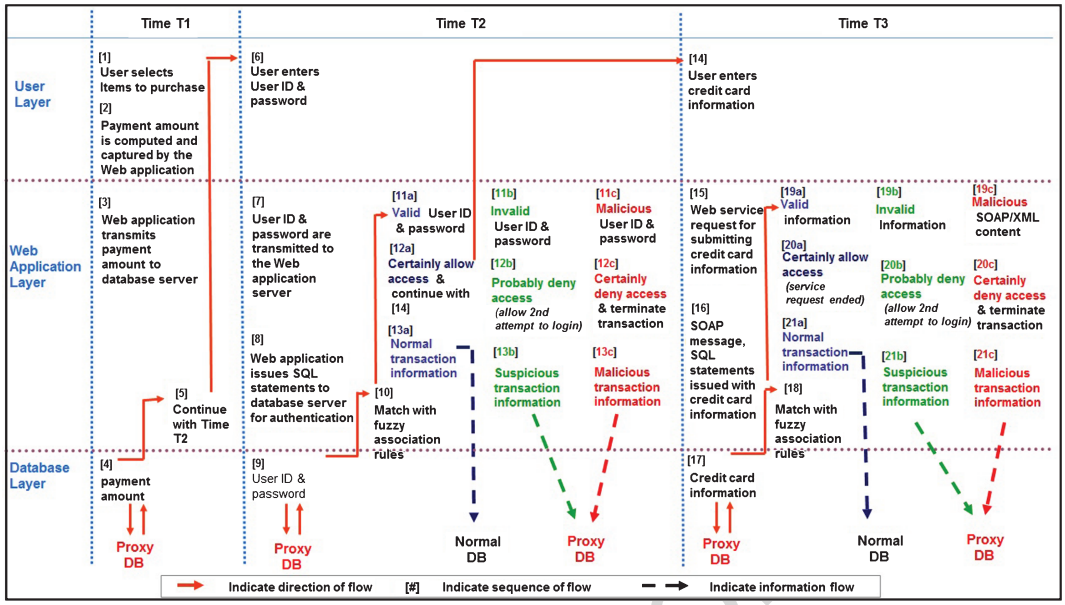


Fig. 5. Successful transaction rates for different users' load for FAR/FAP IDPs.

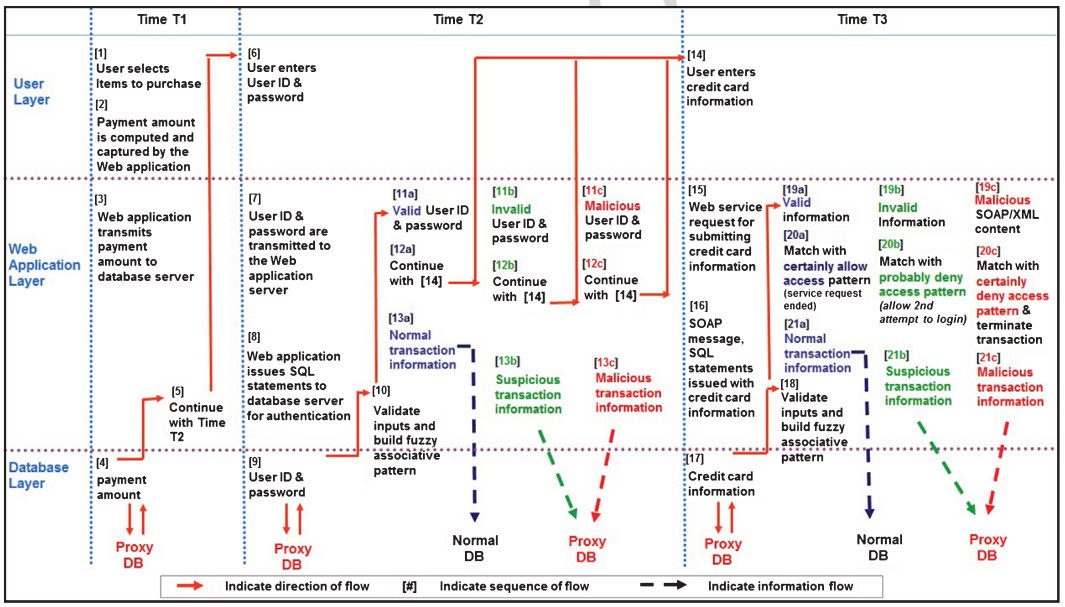
the FAP. For example, for scenario 1 which is the normal transaction, the average transaction time for FAR user's load of 1000, 3000 and 5000 range from 31.97, 76.87 and 120.82 seconds (Fig. 4a, Columns 2–4) whereas for FAP, the range is from 80.39, 111.86 and 133.39 seconds (Fig. 4c, Columns 2–4). Since under this stressed condition, the FAP can only take up a user's load of up to 3000, this means there is a close to 35∼48.5 seconds difference in average transaction time between FAR and FAP for users' load of 3000 and 1000. Similarly for other scenarios, there are significant time gaps between the FAR IDP (Fig. 4a and b) and the FAP IDP (Fig. 4c and d). This further confirms the exhibited behavior as shown in Fig. 2 that the transaction time performance of FAR IDP system exceeds that of the FAP IDP system whether per user's load or per 1000 and 3000 users' load.

4.3. *Performance of the IDP systems in the cloud platform with volume constraint*

Further experiments are designed to test and evaluate the performances of the FAR and FAP IDPs and at the same time checking the consistency of the testing tool in the specific network environment and Cloud platform. Instead of limiting the time frame for different users' loads transacting over the Cloud platform to be 300 seconds only, we use flexible time frame but maximize the transaction volumes to be 2,500 (based on observation from experiments in Section 4.2) for each test run. Again the five representative cases of Table 4 are used in the experiments. Results from these testing are summarized and tabulated as shown in Fig. 5. As shown in Fig. 5 that both FAR and FAP IDP systems behave similarly but with time differences as well. Referring to Fig. 5 (Fig. 5a and b Row 3, Fig. 5c) for both FAR and FAP IDP systems, the average successful transaction rates drop gradually from users' load of 500, 1000 and 1,500 but the drops are significant beyond the user loads of 2000. One obvious reason of cause with 2000 user's load carrying out 2,500 transactions with the same resources as with 500 user's load is four times more stressed, hence the great degradation of successful transaction rate. This may indicate that the performance of FAP and FAR IDP systems are optimum in the load of 500 users carrying out a maximum of 2,500 transactions in more or less than 15 minutes (Fig. 5b Column 2 and Fig. 5a Column 2).



a



b

Fig. 6. (a) Transaction flow of FAR IDP system. (b) Transaction flow of FAP IDP system.

A point to note again is that our FAR IDP system consistently performs better than the FAP IDP system in terms of successful transaction rates for almost all the scenarios and all users' load (Fig. 5a and b, Row 2). In terms of time performance, the FAR IDP system also performs better than the FAP IDP system (Fig. 5a and b Row 4, Fig. 5d) for each user's load.

4.4. *Behaviors of FAR and FAP IDP systems in the cloud environment*

It is observed from this research and from our prior research [9] that both FAR IDP and FAP IDP systems behave similarly, in terms of time performance, whether they are deployed in the Cloud environment

or as standalone systems in the network environment. As indicated from this prior work, the FAR IDP system performs many folds faster, in milliseconds (ms) than the FAP IDP system. This is consistent with our results as shown in Fig. 2 in Section 4.1, where the gap of the curves between FAR IDP system and FAP IDP system has a time difference of about 2 seconds. This indicates that FAP IDP system performs slower than FAR IDP system in all the five representative scenarios under the Cloud environment. This difference is further confirmed by more test results as shown in Sections 4.2-4.3. The difference in efficiency between the FAR IDP and FAP IDP systems can be explained by referring to Fig. 6a, transaction flow of the FAR IDP system and Fig. 6b, transaction flow of the FAP IDP system.

For the FAR IDP system, refers to Fig. 6a, the decision whether to allow access, probably deny or definitely deny access is instantaneous at the point when the rules are matched (Fig. 6a, sequence flows at Time T2:12a, b, or c and at Time T3:20a, b or c). However, for FAP IDP, the decision for allowing access, probably denying or certainly denying access is delayed until the whole transaction flow is completed at Time T3 only (Fig. 6b, sequence flows at Time T3:20a, b or c). This delay is particularly obvious for transaction with invalid password or UserID (Table 4: Case 2 Columns 4-5) and transaction with malicious password or UserID (Table 4: Case 3 Columns 4-5). For these two cases, transaction flows continue from Time T2 to Time T3 even though the inputs are validated to be invalid or malicious due to the fact that the whole fuzzy associative patterns can only be formed and matched with the right decision at Time T3.

Similarly for other transactions where decision is made at Time T3 (Table 4 Cases 1, 4 and 5) when inputs and XML content are validated to be normal, malicious or with extremely large input or SOAP sizes, the FAR IDP still performs better than the FAP IDP in terms of efficiency for the same reason. Additionally for the later three cases, the FAP IDP has to carry out the process of building and forming fuzzy associative patterns throughout Time T2 and Time T3 and when the full pattern is formed then only match with one of the 336 patterns to determine a final decision whether to grant access, probably or certainly deny access for the transaction. This also explains why the transaction model, RXD follows closely the transaction flow of the FAP IDP system but with better efficiency as it does not have to build and form associative patterns in order to make the decision whether to grant or deny access. Thus the 20 fuzzy association rules are more efficient in covering all the 336 fuzzy associative patterns.

5. Comparison of FAR IDP system with other systems

Many existing ID and IDP systems have demonstrated their capabilities in the detection and prevention of attacks occur in Clouds providing SaaS, IaaS, and PaaS. Referring to Table 1b, for example, the network ID system proposed by [6] has demonstrated its capability of detecting DoS attack and other network level malicious activities in Cloud offering IaaS with a detection accuracy of 96% with less than 1.5% false positive rate. Experimental results of the ID system proposed by [3] have shown a 91% detection rate for DDoS attacks occur at the IaaS layer. The IDS proposed by [2] is able to detect HTTP-DoS and XML-DoS with 99% detection accuracy and 1% false positive rate. Research in [14] has proposed an intrusion detection and severity analysis system deployed at a border node to monitor multiple virtual machines for the detection of DoS and DDoS attacks occurring at IaaS with detection rate of over 90%. With a detection accuracy of close to 100% and a false alarm rate of close to 1%, the ID system proposed by [8] is able to detect HTTP anomalies at the SaaS layer. Research in [4] has demonstrated that the proposed IDP system is able to detect and prevent DoS and DDoS with 97% accuracy. Moreover, with a computation time per packet which is less than 0.003 seconds, this proposed IDP system is able to prevent the Cloud service from single point of failure attack.

Nevertheless, these ID and IDP systems are protecting the Cloud services against attacks such as DoS and DDoS only. Our FAR IDP system, on the other hand, not only protects the SaaS against DoS and DDoS attacks, but also detects and prevents known Web and WS-based attacks such as SQL injection, buffer overflow, XML injection, XML-DoS, SOAP oversized payloads and predict new kind of attacks with accuracy close to 100% and false alarm rate of less than 1% on a close to real time basis.

6. Conclusion and future work

As can be seen from our experimental results, both the FAR IDP and FAP IDP systems were able to detect, prevent known Web and WS-based attacks

such as SQL injection, XML injection, buffer overflow, XML content manipulation, XML-DoS, SOAP oversized payloads and predict new variant attacks on a real time basis with detection accuracy close to 100%. However, there is a difference in time performance. The FAR IDP system takes about three seconds while the FAP IDP system takes about five seconds to perform each normal, suspicious or malicious transaction. However, this is still considered to be close to real time. For load and volume testing on the SaaS where the system is under stress, as expected, the FAR IDP system performed better than the FAP IDP system. Particularly with the stress work load of 5000 concurrent users submitting mix of normal, suspicious and malicious transactions over a time interval of 300 seconds (five minutes), our FAR IDP system provided close to 95% service availability to normal transactions, hence protecting the SaaS against DoS and DDoS attacks.

Most recent researches have seen the trend in providing improved quality of services (QoS) for Cloud computing. For example in [15], a double renting scheme is proposed to maximize profits with guaranteed quality of services. This proposed scheme is able to effectively guarantee the quality of service of all requests without great waste of resources. Another research in [28] proposes an efficient mutual verifiable provable data possession scheme to protect data integrity. This proposed scheme is noted for its efficiency as no bilinear operation is required. An adaptive framework is proposed in [13] to dynamically monitor QoS metrics and performance measures in order to ensure compliances to the Service Level Agreement (SLA). This thus imposes accountability for Cloud service providers. Our future work, therefore, is to determine more quality attributes besides service availability, such as latency, throughput and accountability (trust) for a better quality and more trustworthy SaaS over the Cloud platform.

Acknowledgments

The development and implementation of the FAR IDP system over the Cloud platform is supported by a Prototype Research Grant Scheme provided by the Ministry of Education (MOE), Malaysia, under the grant number MMUE/130159. All the three authors who are researchers under this project would like to give special appreciation to Professor Dr. Daniel Wong Kuok-Shoong, Ph.D. (Stanford) and Director of Daniel Wireless Software, Singapore. Dr. Daniel Wong serves as a consultant since the beginning of the project and has provided valuable advice such as industry good practices for the implementation and testing of the system. Another special thank is to be given to Mr. Hassan Mahmood Khan, a research scholar under this project, who has carried out the system implementation and testing professionally.

References

[1] A. Andreu, *Professional Pen Testing for Web Applications*. Indianapolis: Wiley Publishing Inc, 2006.

[2] A. Chonka, Detecting and Mitigating HX-DoS attacks against Cloud Web Services, *The 15th International Conference on Network-Based Information Systems*, Melbourne, Australia, 2012, pp. 429–434.

[3] B. Joshi, A.S. Vijayan and B.K. Joshi, Securing Cloud Computing Environment against DDoS Attacks, *In Proceedings of 2012 International Conference on Computer Communication and Informatics (ICCCI 2012)*, Combatore, India, pp. 1–5.

[4] C.C. Lo, C.C. Huang and J. Ku, A Cooperative Intrusion Detection System Framework for Cloud Computing Networks, *IEEE 39th International Conference on Parallel Processing Workshops*, San Diego, California, USA, 2010, pp. 280–284.

[5] C.N. Modi, D.R. Patel, A. Patel and R. Muttukrishnan, Bayesian Classifier and Snort based Network Intrusion Detection System in Cloud Computing, *In Proceedings of the 2012 Third International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Tamilnadu, India, 2012, pp. 1–7.

[6] C.N. Modi, D.R. Patel, A. Patel and R. Muttukrishnan, Integrating Signature Apriori based Network Intrusion Detection System (NIDS) in Cloud Computing, *In Proceedings of the 2nd International Conference on Communication, Computing & Security (ICCCS-2012)*, Rourkela, India, pp. 906–912.

[7] F. Du, An Effective Pattern Matching Algorithm for Intrusion Detection, *In Proceedings of the IEEE International Conference on Computer Science and Electronic Engineering*, Hangzhou, China, 2012, pp. 34–38.

[8] G. Nascimento and M. Correia, Anomaly-based Intrusion Detection in Software as a Service, *2011 IEEE Dependable Systems and Networks Workshops*, Hong Kong, China, 2011, pp. 19–24.

[9] G.Y. Chan, C.S. Lee and F.F. Chua, Fuzzy Association Rules vs Fuzzy Associative Patterns in Defending against Web Service Attacks, *In Proceedings of the 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2015)*, Zhangjiajie, China, 2015, pp. 558–563.

[10] G.Y. Chan, C.S. Lee and S.H. Heng, Policy-enhanced ANFIS model to counter SOAP-related attacks, *Knowledge-Based System* **35** (2012) 64–76.

[11] G.Y. Chan, C.S. Lee and S.H. Heng, Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks, *Journal of Network and Computer Applications* **36** (2013), 829–842.

[12] H.A. Kholidy and F. Baiardi, CIDS: A Framework for Intrusion Detection in Cloud Systems, *In Proceedings of the 2012 Ninth International Conference on Information Technology:*

New Generations (ITNG), Las Vegas, Nevada, USA, 2012, pp. 379–385.

[13] H.M. Khan, G.Y. Chan and F.F. Chua, An Adaptive Monitoring Framework for Ensuring Accountability and Quality of Services in Cloud Computing, *In Proceedings of the 30th International Conference on Information Networking (ICOIN 2016)*, Kota Kinabalu, Malaysia, 2016, pp. 249–253.

[14] J. Arshad, P. Townend and J. Xu, A novel intrusion severity analysis approach for clouds, *Future Generation Computer Systems, Journal* **29** (2013), 416–428.

[15] J. Mei, K. Li, A. Ouyang and K. Li, A profit maximization scheme with guaranteed quality of service in cloud computing, *IEEE Transaction on Computers* **64**(11) (2015), 3064–3078.

[16] N.Ch.S.N. Iyengar, A. Banerjee and G.Ganapathy, A fuzzy logic based defense mechanism against distributed denial of service attack in cloud computing environment, *International Journal of Communication Networks and Information Security (IJCNIS)* **6**(3) (2014), 233–245.

[17] P.K. Shelke, S. Sontakke and A.D. Gawande, Intrusion detection system for cloud computing, *International Journal of Scientific & Technology Research* **1**(4) (2012), 67–71.

[18] P. Mell and T. Grance, The NIST Definition of Cloud Computing, National Institute of Standards and Technology, US Department of Commerce, Special Publication 800-145, 2011. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[19] S. Dhage, B. Meshram, R. Rawat, S. Padawe, M. Paingaokar and A. Misra, Intrusion Detection System in Cloud Computing Environment, *In Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, Mumbai, India, 2011, pp. 235–239.

[20] S. Park and S.R. Lee, Red tides prediction system using fuzzy reasoning and the ensemble method, *Applied Intelligence, Journal* **40** (2014), 244–255.

[21] S. Subashini and V. Kavitha, A survey on security issues in service delivery models of cloud computing, *Journal of Network and Computer Applications* **34**(1) (2011), 1–11.

[22] S.V. Sandar and S. Shenai, Economic denial of sustainability (EDoS) in cloud services using HTTP and XML based DDoS attacks, *International Journal of Computer Applications* **41**(20) (2012), 11–16.

[23] T. Karnwal, T. Sivakumar and G. Aghila, A Comber Approach to Protect Cloud Computing against XML DDoS and HTTP DDoS Attack, *In Proceedings of the 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*, Bhopal, India, 2012, pp. 1–5.

[24] T. Tuncer and Y. Tatar, Detection DoS Attack on FPGA Using Fuzzy Association Rules, *International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11*, 2011, pp. 1271–1276.

[25] U. Tupakula, V. Varadharajan and N. Akku, Intrusion Detection Techniques for Infrastructure as a Service Cloud, *In Proceedings of the Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing*, Sydney, Australia, 2011, pp. 744–751.

[26] Windows Azure. http://azure.microsoft.com/ (Accessed January – August 2015)

[27] W.W. Wu, Mining significant factors affecting the adoption of SaaS using the rough set approach, *Journal of Systems and Software* **84** (2011), 435–441.

[28] Y. Ren, J. Shen, J. Wang, J. Han and S. Lee, Mutual verifiable provable data auditing in public cloud storage, *Journal of Internet Technology* **16**(2) (2015), 317–324.