Journal of Computer Security 28 (2020) 635–653
DOI 10.3233/JCS-200002
IOS Press

A systematic review of security threats and countermeasures in SaaS

Miguel Ángel Díaz de León Guillén [*], Víctor Morales-Rocha and
Luis Felipe Fernández Martínez
National Laboratory of Information Technologies, Autonomous University of Ciudad Juárez, México

Abstract. Among the service models provided by the cloud, the software as a service (SaaS) model has had the greatest growth. This service model is an attractive option for organizations, as they can transfer part or all of their IT functions to a cloud service provider. However, there is still some uncertainty about deciding to carry out a migration of all data to the cloud, mainly due to security concerns. The SaaS model not only inherits the security problems of a traditional application, but there are unique attacks and vulnerabilities for a SaaS architecture. Additionally, some of the attacks in this environment are more devastating due to nature of shared resources in the SaaS model. Some of these attacks and vulnerabilities are not yet well known to software designers and developers. This lack of knowledge has negative consequences as it can expose sensitive data of users and organizations. This paper presents a rigorous systematic review using the SALSA framework to know the threats, attacks and countermeasures to mitigate the security problems that occur in a SaaS environment. As part of the results of this review, a classification of threats, attacks and countermeasures in the SaaS environment is presented.

Keywords: Software as a service, cloud security, threats, countermeasures, systematic review

1. Introduction

Cloud computing (CC) allows clients connected through the Internet to share resources on demand in order to offer availability, scalability and low cost [16,28]. Some benefits that CC provides is elasticity, easy access, monitoring and some free services [15,45]. Therefore, CC reduces expenses and facilitates system administration [20].

There are four possible cloud deployment models and each of them has some particular security issues. The first are public clouds, which provide services through the internet. These types of clouds are considered less secure because it is more difficult to protect the data against malicious attacks. The second are private clouds, which are managed by the owner or by a third party. In this way it is possible to adjust the security levels according to the needs of the company. The third type is the community cloud, which is one implemented by several organizations sharing the responsibility for configuration. This can cause the lack of application of appropriate security protocols, management and mitigation. The fourth is the hybrid cloud, which is a combination of different models. Unfortunately, this type of cloud inherits all the security issues of the other models [3,16].

On the other hand, the cloud service delivery models are, firstly, infrastructure as a service (IaaS), in which the Internet media infrastructure is delivered to the client. The user has a provision of processing and storage resources [6,28,39]. Secondly, platform as a service (PaaS), in which the necessary means

[*] Corresponding author. E-mail: al183080@alumnos.uacj.mx.

are offered so that software developers can deploy their applications [6,28,31]. Thirdly, software as a service (SaaS), which provides access to applications through the use of the Internet [6,28,30]. Unlike an onsite application, SaaS runs under the PaaS model [43].

Unlike a traditional application in which the software belongs to the client, in the SaaS model the application is managed by the cloud provider. However, the number of applications that are being implemented in SaaS is increasing [25]. In addition, the IT industry can change the application sales model with the use of cloud equipment with the provision of SaaS services, reducing marketing costs [46]. Unfortunately CC inherits security issues from on-site systems, networks, and in addition, vulnerabilities in web services, particularly for the SaaS model, in which the software is accessed through the cloud. The main obstacle that an organization faces in migrating its software to an SaaS architecture is the uncertainty of the security of its data due to the fear of information leakage [8]. Cloud customers do not have the management of the cloud infrastructure. Thus the user should know the security measures that the cloud provider has implemented [3].

The remainder of this paper is organized as follows. Section 2 describes the SALSA framework and its phases and the method used to carry out this review using the SALSA framework. In Section 3, the results are presented, answering the research questions. Finally, the conclusions of this research are presented in Section 4.

2. Methodology

This research has used the reference framework called SALSA, described in [12]. This is a method for carrying out a systematic review of a specific topic. The phases of SALSA will now be presented. This section presents the steps taken to carry out the systematic review from the proposal of questions to define the research approach to the analysis of the information collected.

2.1. Protocol

In this phase the scope of the research is explained. To fulfill this objective, the PICOC framework (Population, Intervention, Comparison, Outcome and Context) [12] has been used. The research objective is defined in the form of questions, which are the following:

(1) What are the threats that affect SaaS?
(2) What are the most common attacks in SaaS?
(3) Do the same threats apply to cloud applications and on-site applications?
(4) Will it be possible to classify the threats?
(5) What security countermeasures can be applied in SaaS?

2.2. Search

In this step, the databases are selected, and a chain is created to carry out a search in these databases in order to obtain documents relevant to the investigation. Databases related to the research area were used. Table 1 shows the search strings and databases, as well as the number of papers that each result yielded.

Table 1
Search results classified by database

String	Database	Date	Results
(SaaS OR “Software as a Service”) AND (Vulnerabilities OR Security OR “Security Issues” OR “Security Challenges” OR “Security measures”)	ScienceDirect	03/05/2019	4771
(SaaS OR “Software as a Service”) AND (Vulnerabilities OR Security OR “Security Issues” OR “Security Challenges” OR “Security Measures”)	IEEE	03/05/2019	672
(SaaS OR “Software as a Service”) AND (Vulnerabilities OR Security OR “Security Issues” OR “Security Challenges” OR “Security Measures”)	ACM	03/05/2019	91
(SaaS OR “Software as a Service”) AND (Vulnerabilities OR Security OR “Security Issues” OR “Security Challenges” OR “Security Measures”)	SCOPUS	03/05/2019	1043
(SaaS OR “software as a service”) AND (vulnerabilities OR security OR “Security Issues” OR “Security Challenges” OR “Security Measures”)	WOS	03/05/2019	146
Total			6723

Table 2
Inclusion and exclusion criteria

ID	Statement
I1	Papers dealing with threats in SaaS
I2	Papers dealing with security in SaaS
I3	Papers dealing with attacks in SaaS
E1	Research not written in English
E2	Research published before 2013
E3	Duplicate Papers
E4	Research that does not match the search objectives
E5	Papers that are not applicable to security in SaaS
E6	Investigations without access

2.3. Appraisal

In this step, an evaluation is carried out to select the documents that will be useful for the investigation. For this purpose, a study selection and quality evaluation were made [12]. Table 2 shows the criteria that were used to include and exclude papers. Table 3 shows the filters applied to a total of 6723 articles. At the end of the process, there were 47 articles for the systematic review.

2.4. Synthesis

In this phase, relevant information was subsequently obtained. A classification of the papers was carried out in order to create an information base for its analysis. This method is divided into two phases: data extraction and categorization. The data extraction phase involves obtaining important data in the papers selected in the previous step of the SALSA framework. Categorization involves the data classification and processing for the analysis [12]. To carry out this step with the selected papers, three main themes were identified to cover the objective of the investigation: threats, attacks, and security countermeasures. An example of the data extracted from the first three papers is shown in Table 4.

Table 3
Results of the selection of studies

Step	Criteria	Total papers	Included	Excluded
Search results	Search string	6723	6723	0
Papers not written in English	E1	6723	6705	18
Remove papers before 2013	E2	6705	4329	2376
Duplicated papers	E3	4329	3651	678
Title reading	E1 to E6	3651	437	3214
Abstract reading	E1 to E6	437	97	340
Introduction-conclusion reading	E1 to E6	97	59	38
Full-text reading	I1 to I3 and E1 to E8	59	47	12

Table 4
Example of data extracted from different papers

Theme	Paper 1	Paper 2	Paper 3
Threats	Malicious SysAdmin, Data loss / manipulation, DoS attacks, EDoS attack	Insecure implementation of the OAuth protocol	Different service delivery, insecure interface and API, malicious insiders, data loss and leakage, account hijacking, risk profiling, identify
Security countermeasures	Multi-factor authentication, Auditing and logging, IDS/IPS, DDoS mitigation, Firewall	Cloud-based DoS-resistant protocol	None
Attacks	None	None	Zombie attack, user root attacks, port scanning, man in middle attack, metadata spoofing attack, phishing attack

2.5. *Analysis*

In the analysis, the data obtained from the synthesis is used to provide sufficient elements to answer the research questions. The objective is to map the relations between the themes of each document. In this step, each of the previously selected papers was analyzed, based on what was captured during the synthesis phase. The objective to be fulfilled is that this analysis manages to answer the research questions raised in Section 2.1. First, papers related to the issue of threats that occur in SaaS will be analyzed in Section 2.5.1. Subsequently, papers related to attacks and their countermeasures that may occur in SaaS in will be analyzed in Section 2.5.2. Lastly, Section 2.5.3 treats papers related to security measures that can be applied to avoid threats and attacks.

2.5.1. *Threats*

In the papers presented in [3] and [8] it is mentioned that the cloud inherits the security problems of on-site applications, particularly the SaaS service model. Even exploiting a vulnerability in SaaS can be more devastating than a traditional Web application.

A total of six papers deal with the threats that exist in SaaS. Threats can adversely affect the system in the cloud. Confidentiality, integrity, and the availability of resources in the cloud are important pillars of the cloud security software guarantee [38]. Threats are known for potential damage to the system [1]. The most important threats are those of malicious employees and password stealing. The types of threats that may occur in SaaS are the following [38]:

- The threat agents can carry out an attack. This kind of threat can originate from a user or software.
- An application or an attacker that makes attacks on the network from the Internet.
- An application of malicious logic.
- An attacker being a cloud client who has shared resources of an infrastructure.
- Malicious experts are human threats that act on behalf of cloud providers.

Based on the information collected from the six documents corresponding to the topic of threats, it was possible to create a list of threats that affect SaaS. In Section 3.4.1, the authors present an elaborate classification based on the types of threats presented above and the security objectives that are affected. Below is the list of threats found in the different papers:

(1) Loss of control over resources. In the cloud, customers give their data to a provider. This may cause the provider to not report how the information has been managed. Therefore, customers must sensitively manage migration to the cloud and pay special attention to the contracts with the provider [1].

(2) Misuse of cloud computing resources. Having a simple access interface could allow malicious users or employees to attack the cloud infrastructure. To reduce the risk of attack, encryption must be implemented, and background checks for employees must be performed. In addition, it is necessary to use authentication controls [1].

(3) Different service delivery/receiving models. The cloud can change how it provides its services. User data can change location through different servers, so they can be governed by different security laws due to their location. As a solution, the use of point to point encryption is proposed as well as standardized security laws [38].

(4) Insecure interface and API. The cloud grants APIs for the communication with its services. Therefore, cloud security depends largely on the APIs. If an API is attacked, this can affect the availability of the service in the cloud. As a security countermeasure, robust security mechanisms and a secure interface must be implemented [38].

(5) Malicious insiders. In general, the employees of the cloud provider have a high level of access, so they are likely to be able to access sensitive customer information [1]. They are well trained and well-versed in the infrastructure, tools, and equipment to carry out their tasks. However, they can suddenly become adversaries when they are dissatisfied with the organization's decision-making, their claims are not met, they are not rewarded, and the organization does not treat them well [49]. The provider should have tools that can track their employees in order to detect malicious activity [1]. As a prevention countermeasure the use of agreement reports and non-compliance notifications is recommended. In addition, the security and management process must be transparent [38].

(6) Data Scavenging. User data can be hosted in the same storage segment. On the other hand, several backup copies of the information hosted in different locations can be made. Unfortunately, this hinders requests for complete data deletion, so an attacker could steal the data from an organization. As a security measure it is recommended that the user point out the confidentiality of their data [1]. Attackers can recover deleted data, because the information can still exist in the storage medium unless it is destroyed [22,22].

(7) Data Loss or Leakage. Data loss occurs when information is transferred or stored incorrectly [22]. Data loss is caused by different factors, such as weak encryption, simple passwords, and the lack

of backups [38]. Data leakage is a major concern, so strict controls are required [42]. As security measures, it is proposed to use secure passwords and encryption methods, periodic backups, and the use of secure APIs [38].

(8) Service/account hijacking. This occurs when an attacker manages to steal a user's access credentials. When this is achieved, an attacker can use the user's password to make new attacks [1,7,38]. This type of attack was classified as the third highest risk in the cloud, according to a report made by the cloud security alliance [2]. As security measures, robust authentication, the use of security policies, and the use of encryption in communication channels are proposed [38].

(9) Risk profiling. Usually the cloud grants hardware and software maintenance to a third party. This can be beneficial; however, the cloud can ignore the procedures, leading to greater risks and threats. As a security measure, there should be a knowledge of the records, as well as of the aspects of the data and infrastructure, to ensure that data usage and alterations of the system are monitored. To reduce this threat, the cloud must take into consideration the details of the infrastructure, data, and records. In addition, the cloud must have a monitoring system [38].

(10) Identity theft. This occurs when an attacker pretends to be another user, using that user's privileges, credits, and other resources, causing the victim to lose confidence. This can happen due to different factors, such as keyloggers, phishing, and weak authentication methods. As a security measure, the use of robust authentication and secure password recovery methods are proposed [38].

2.5.2. Attacks and countermeasures

An attack is a method of exploiting a vulnerability [1]. Regarding this issue, a total of 30 attacks that affect SaaS were found. It should be clarified that not all attacks can occur in this service model. A total of 25 papers describe the 30 attacks. On the other hand, 21 papers describe the countermeasures that can be applied to avoid the attacks mentioned above.

Table 5 shows a summary of the list of attacks and their possible countermeasures. In the Section 3.4.2, the authors develop a classification of attacks based on the STRIDE model and the security objectives involved in each attack. Below is the list of attacks and their possible countermeasures according to the information collected from the different papers that were analyzed.

(1) **ARP Spoofing**. This attack is when a malicious person sends an altered ARP message to redirect to a malicious host. Because ARP does not require checking the source, the attacker can plan an ARP attack by deriving connections to a specific host. As security measures, detection, encryption, filtering in ARP tables should be used [1,16].

(2) **Backdoor and debug options**. This type of attack occurs when developers leave debugging enabled in their applications. With this, the attacker can easily make changes to the application [9,41]. To counteract this attack, the debug option, periodic scans, and directory review must be disabled [9,41].

(3) **Broken authentication**. This happens because of the incorrect implementation of authentication management. Some threats are user credentials which can be discovered due to weak account management functions, unencrypted credentials, session data being presented in the URL, etc. This can enable a malicious person to steal the identity of a legitimate user; usually an attacker searches for an account with limited permissions [24]. The main attacks of this type are brute force attacks, phishing attacks, account hijacking attacks, internal attacks, and keylogger attacks [17]. To counteract these attacks, there are prevention methods such as strong authentication implementation, shielding against Cross Site Scripting by avoiding script failures, access control, use of indirect

Table 5

A summary of the list of attacks and their possible countermeasures

ARP Spoofing	Reliable ARP table, detection, encryption, filtering
Backdoor and debug options	The developer must disable the debugging option
Broken authentication	Strong authentication and session management controls
	Apply automation to verify
	Avoid cross-site scripting (XSS) failures
Buffer overflow	Instruction set randomization
Code injection	Active content filtering
	Web application vulnerability detection
Cookie poisoning	Regularly cleaning cookies
CSRF	Secret token, referrer header and origin header
DNS poisoning	Encryption and filtering
Dumpster diving	Define and implement a policy on disposal of confidential documents
Eavesdropping	Implement Internet Protocol Security (IP sec)
	Implement security policies and antivirus
EDoS	EDoS Shield and Alosaimi graphical Turing tests
Google hacking	Not sharing confidential information
	Use of tools to scan vulnerabilities
Hash value manipulation	A strong protocol for probable data possession is needed
Hidden field manipulation	Avoid placing parameters in a hidden chain
Malware injection and steganography	Schemas such as StegAD
Man-in-the-middle attack	Enforcing conventional security measures
	Tools like Dsniff, Cain, Ettercap, Wsniff, Airjack can help
Meta data spoofing attack	Keep the functionality of the service and other details encrypted
	Strong authentication
Phishing attack	Self-adaptive and self-adjusting encryption and decryption algorithms
	Implement TLS
	HTTPS using certificates
Port scanning	Port blocking
Race condition	Predicate refresh technique
Replay attack	stochastic coding scheme is proposed in [37]
Reused IP address	Cache removal
Service injection attack	Strong isolation
	Strong identification mechanism
	Implementing service integrity
Shared architectures	Analyze the binary code of the application
Sniffing	SSL and TLS
	IPsec
Social engineering	Focus mainly on security policies and staff training
Sybil attack	A solution based on symmetric key cryptography
User to root attack	Strong password
	Strong authentication mechanism
XML signature wrapping attack	Automatic scanners and manual verification
Zombie attack	Robust authentication
	Use of IDS/IPS
	A periodic thorough system scan
	Filter ICMP and SYN packets
	Filter private IP addresses

references of objects either by user or by session, automatic verification implementation, mechanisms for Multilevel authentication, and the use of digital fingerprint signatures (this may involve high costs) [17].

(4) **Buffer overflow**. This happens when an application tries to store more data in a buffer than it supports. Buffers must contain a static amount of data. When an overflow of the buffer occurs, the content is corrupted or overwritten [18]. Because of this, an attacker can execute malicious code and obtain administrator privileges [47]. A countermeasure for this attack is the randomization of instructions [16].

(5) **Code Injection**. The nature of the cloud is to handle shared environments. Thus, attackers are able to insert malicious code into applications to obtain sensitive data from users. If the user clicks on an infected URL, this can execute code on the use's machine, which can cause the attacker to access the user's information. SQL injection is one of the most common techniques of this type of attack. It allows an attacker to insert SQL commands from web formulas to access a database. Active content filtering is used to detect this type of attack and provide the use with dynamically generated SQL in the code [9,17,18,47].

(6) **Cookie Poisoning**. This attack occurs when an attacker manages to obtain a victim's cookies in order to gain access to an application. Because cookies contain information for logging into applications or websites, the attacker can falsify them to be authenticated as an authorized user. As a precautionary measure, cookies stored on the user's computer must be periodically deleted [9,41].

(7) **Cross-Site Request Forgery (CSRF)**. This type of attack is when a user's browser is made to send a false HTTP request with sensitive data, such as cookies and authentication credentials. While the legitimate user is connected to an application and visits a malicious site, this site can inject code into the client's browser, which can lead to identity or data theft, such as credit card data [7,47]. As prevention measures, the use of secret tokens and reference and origin headings [16] is recommended.

(8) **DNS Poisoning**. DNS servers relate IP addresses to a domain name. If this relation becomes corrupted, the attacker can redirect a user to a malicious website. To prevent this attack, the application of encryption and filtering is recommended [1,9,16].

(9) **Dumpster Diving**. This technique consists in the attempt to recover information from data that was insecurely deleted. The malicious user manages to restore the data that the user deleted. In this way an attacker can focus on a specific user to obtain relevant information. Deleted data may contain credentials, cookies, and credit card numbers, among other things. To reduce the possibility of suffering this attack, the implementation of a security policy for the elimination of both physical and digital documents is recommended [26,32].

(10) **Eavesdropping**. This attack happens when an attacker listens to network transmissions without being detected, causing a confidentiality failure. Transmissions can be messages, calls, and videoconferences, among other transmissions [13,22]. A security method is the implementation of IPsec [9].

(11) **EDoS**. This attack is focused on customer billing. It consists of inflating the costs of the services granted to users. DoS attacks on payment-for-use services will result in an increase in the use of bandwidth, CPU, and storage [9]. As a security measure, a firewall must be implemented for the detection of EDoS [27]. In addition, graphical tests of EDoS Shield and Alosaimi Turing can be performed to avoid such attacks [40].

(12) **Google Hacking**. An attacker can use search engines, such as Google, because this is a good option for obtaining confidential data from a user or organization. With this method they can discover

security breaches in applications and carry out other attacks [9,41]. The security measures proposed are to avoid sharing sensitive information on websites and to use tools for vulnerability scanning [9].

(13) **Hash Value Manipulation**. This type of attack occurs when a malicious person alters the hash value of a message in order to gain access to a file. If the altered hash value is hosted in the database, the server links the file with the hash value. On the other hand, if the altered hash value is not found in the database, the server requests a file from the user. This vulnerability can occur in cases where the server uses OpenSSL with the Ncrypto class. As a security measure, the implementation of strong communication protocols using encryption and probabilistic tests is required [26,29].

(14) **Hidden field manipulation**. This attack occurs when the application developer uses hidden fields for the user. For example, it can be used to save prices, but an attacker can take advantage of this to make purchases with altered prices. So it is necessary to avoid placing hidden fields to mitigate this type of attack [9].

(15) **Malware injection and steganography attacks**. This attack happens when it is possible to inject malicious code into an application. In this way, an attacker can insert code into the files that pass through the network. Because it seems that a normal file is being sent, security tools can ignore this attack [21]. In [21], the use of schemes such as StegAD for the detection of such attacks is proposed.

(16) **Man-in-the-middle Attack**. This happens when a malicious person intercepts a communication between two users. The hacker can just listen to the message and forward or modify it. In this way, the attacker obtains sensitive data from the users. This attack can occur due to an Internet protocol failure, weak password management, the use of insecure wireless networks, and weak authentication. As security measures we recommend the implementation of secure channels through SSL and applying authentication and authentication measures to the nodes. Some useful tools to prevent these attacks are Airjack, Cain, Dsniff, and Ettercap [1,9,13,17,38].

(17) **Meta Data Spoofing Attack**. This occurs when an attacker alters the information about the services hosted in Web Services Description Language when delivered. In this way, the attacker gains access to sensitive applications and data. As a security measure, the functionality of the service must be encrypted and robust authentication mechanisms required to access this service [1,38].

(18) **Phishing Attack**. Phishing is a technique that consists in redirecting a legitimate user to a fake website [2]. As a result, users believe that it is a reliable website, and so enter their credentials, thus compromising the user account [2,38]. To improve cloud security, encryption must be implemented, as well as the use of TLS for applications. Finally, the use of certificates through HTTPS is essential [17,38].

(19) **Port scanning**. An attacker uses port scanning to find out if they are open, closed, or filtered. A malicious person uses the open ports to obtain information from the network. If a port is configured to accept traffic without any filters, this port will be affected by a port scan. As a security measure, the ports must be filtered and when a scan is detected, the scan must be blocked [9,38].

(20) **Race Condition**. This attack happens when several processes access the same data simultaneously. A malicious person can have administrator permissions while an application is in administrator mode [47]. The work presented in [19] proposes a technique called predicate update to detect this attack.

(21) **Replay attack**. This type of attack occurs when an attacker reproduces a message obtained by the recipients [13] in order to gain access to unauthorized data [9]. For its detection without affect-

ing the performance of the system, in [48] the implementation of a stochastic coding scheme is proposed

(22) **Reused IP address**. If a user changes networks and another one is assigned the same IP address, this leads to a security problem. This is because the user assumes that their resources are not accessible when leaving the network, but if a new user obtains the first user's IP address, the new user can have access to these resources, something which violates the privacy of the first user. To avoid this security problem, the elimination of cache in ARP tables is proposed [9].

(23) **Service Injection Attack**. This happens when an attacker manages to inject a malicious service which causes users' requests to be automatically redirected to malicious services. As a result, the integrity of the data is lost, and the theft of accounts or services is enabled. As a security measure, we propose the use of a robust asylum mechanism, mechanisms for identifying virtual machines, and using integration services [1].

(24) **Shared architectures**. Because SaaS runs in a shared architecture, it is possible to detect the application execution path. In this way, the attacker can get enough information for the theft of user accounts. As a countermeasure, the application's binary code should be reviewed [21].

(25) **Sniffing**. Sniffing is a technique that uses a software tool to capture packets that are travelling through the network [2]. An attacker could steal sensitive data, such as credentials and credit card data [10]. As a precautionary measure, the use of cryptographic protocols such as SSL and TLS, the implementation of IPsec, and the encryption of each IP packet is recommended [10].

(26) **Social Engineering**. These attacks occur when a malicious person has legitimate users reveal their sensitive data, such as credentials, emails, and credit card numbers, among other things, through deception. This is achieved through the use of web pages, telephone contact, and fake emails, among other techniques [17,47]. The main approaches to reduce this type of attacks are to implement security policies and user training [23].

(27) **Sybil attack**. The attacker steals the identity of a user to create a relationship with a legitimate user which can lead to the attacker raising their privileges within the system [9]. As a solution, the implementation of a symmetric encryption algorithm is recommended [44].

(28) **User to Root Attack**. This is when an attacker obtains the privileges of an administrator user. This is achieved by overflowing data in an application. To reduce the risk of this attack, which can have serious implications for confidentiality and integrity, we recommend using strong passwords and a better authentication mechanism [38]. Also, it is necessary to adopt a privilege separation mechanism to manage privilege access control between different platforms [47].

(29) **XML signature wrapping attack**. This attack happens when there is a vulnerability in SOAP messages. When the user sends a request through the browser, the server generates a SOAP message. This message can find the information used to establish communication between the client and the server. If the attacker manages to compromise the message, the attacker can be authenticated as a legitimate user. As a preventative measure for this attack, the use of tools for vulnerability scanning and manual verification is recommended [17].

(30) **Zombie Attack**. Also known as a distributed DoS attack, this type of attack is harder to detect than a DoS attack. The attacker has several infected machines, called zombies, execute attacks remotely [38]. As security measures, the implementation of IDS / IPS systems, load controls, ICMP and SYN packet limitation, IP address filtering, detailed detection analysis for intruder detection, and better authentication and authorization controls are recommended [1,9,10,17].

2.5.3. Additional countermeasures

This section presents some additional security measures. This will be useful to improve the security of applications hosted in an SaaS environment. The authors classified security countermeasures into five groups: which are security measures in charge of the cloud provider, security in tenant environments, cloud-based secure authentication (CSA), antivirus and IDS and Methods for mitigating Authentication and Access Control threats. This classification describes the recommended methods to make the multi-tenant environment secure, the use of an antivirus program, and the use of an IDS. In addition, control measures are presented for the authentication and authorization of users. A summary of these security measures is shown in the section.

(1) **Countermeasures by the provider**. For security reasons, the cloud provider must provide the user with encrypted credentials and guarantee the integrity and authorization of the services. In addition, one must carry out constant risk assessments, training of employees on computer security, comply with the minimum privilege for users, implement security policies in credential administration, monitor the online movements of employees, implement defenses against malicious code, implement a layered defense against remote attacks, monitor suspicious behavior and act as necessary, deactivate access once the session is over, keep records to facilitate investigation, implement backup and recovery mechanisms, and document internal threats [37]. To mitigate the vulnerabilities that exist in the cloud, provider can also implement the following countermeasures: end-to-end encryption, malicious activity scanning, implementation of secure APIs, business continuity plans, evaluation of employees and contractors to avoid internal attacks, and validation of cloud consumption to avoid EDoS attacks [34].

(2) **Security in SaaS Multi-Tenant Environment**. There are three security measures that can be implemented to improve a multitenant environment in SaaS. First, database-based segmentation, so that only certain columns are accessible to each tenant. Encryption: encrypting the information is usually useful in case it is compromised or stolen by a malicious user. In this way, the attacker will not be able to read the information. Finally, the tenant must know the existing protections to guarantee data isolation between the tenants of the cloud [30].

(3) **CSA**. This protocol is a set of three protocols. The first is used for registration. The second is an adaptation-based identification protocol, which is very useful for countering DoS attacks. The third protocol is used for authentication. The advantages of this protocol are that the cloud can confirm the identity of the client for secure authentication, and also it can detect and prevent DoS attacks [11].

(4) **Antivirus and IDS**. The implementation of an antivirus program in the cloud can improve security because it monitors and blocks any malicious code that affects the system in the cloud. The antivirus program can analyze the files that are transferred in the cloud. In this way, it is possible to detect threats and stop them [21]. In addition, the implementation of an Intrusion Detection System in the cloud increases the level of security because it detects anomalies within the network [36].

(5) **Methods for mitigating Authentication and Access Control threats**. Some important measures to improve security in authentication and access control are: applying single sign-on policies, implementing multifactor authentication, implementing biometric authentication, implementing RSA encryption, implementing an intrusion and firewall detection system, implementing open standards for Exchange authentication, and authorizing data between security domains that allows users to share resources with the use of tokens instead of passwords [34].

2.6. Report

The final step of the SALSA framework is to make a report in the form of a paper in which the results obtained from the systematic review are presented [12]. The results of this work are shown in the following section.

3. Results

This section will answer the questions raised in Section 2.1, which have been possible to answer by completing the systematic review.

3.1. What are the threats that affect SaaS?

Section 2.5.1 presented in detail the threats that may occur in SaaS. In summary, threats may originate from software, an attacker, or a malicious employee. We consider that the threats that represent the greatest risk are identity theft, account theft, and malicious employees, because they may affect the confidentiality, integrity, and availability of the information.

3.2. What are the most common attacks in SaaS?

In Section 2.5.2, a total of 30 attacks were presented: they can be considered the most common ones that can occur in an SaaS. Figure 1 shows the most mentioned attacks in the papers that have been investigated.

3.3. Do the same threats apply to cloud applications and on-site applications?

Section 2.5.1 mentioned the fact that the SaaS service model inherits the security problems of traditional applications. Below is a series of security issues that occur in SaaS.

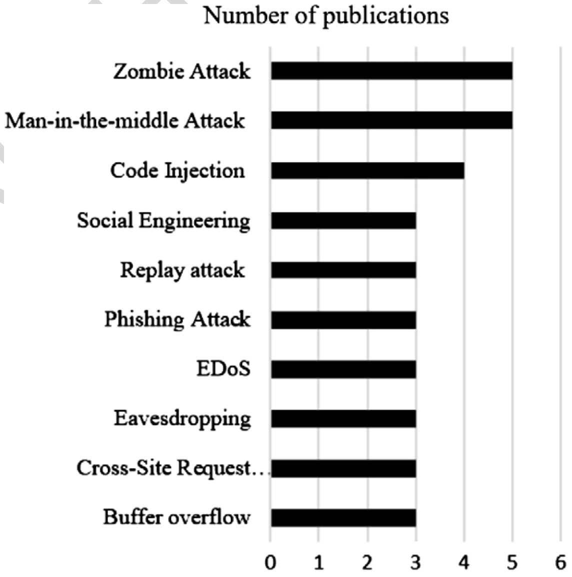

Fig. 1. Attacks with more publications.

3.3.1. Security issues

Attacks, such as denial of services, are more devastating in cloud environments due to the nature of infrastructure sharing. Traditional firewalls and network intrusion detection and prevention (IDP) system are not effective in counteracting DDoS, XML-DoS and HTTP-DoS attacks. Applications hosted under the SaaS scheme require another defense level, at the application level, to reduce the possibility of the occurrence of these attacks [8]. SaaS has specific tasks, so to avoid risks, it is necessary to protect the repositories of information and stored data [22]. To improve security in the cloud, authentication, authorization and access control must be granted to users [35]. To guarantee security in a cloud application, it is necessary to have confidentiality, integrity and availability, which is known as CIA. Some methods to improve data security involving CIA are the following: encryption applied to data at rest and in transit, implementation of hash functions, to validate the integration it is possible to use the third-party audit service (TPA), avoidance of storing the credentials and encryption keys in the same place, application of robust authentication, for availability it is recommended to make periodic backups and duplication.

3.3.2. Data security

To guarantee security in a cloud application, it is necessary to have confidentiality, integrity and availability, which is known as CIA [6,18]. Some methods to improve data security involving the CIA are the following: encryption applied to data at rest and transit, implementation of hash functions, to validate the integration it is possible to use the third-party audit service (TPA), not Store in the same place the credentials and encryption keys, application of robust authentication, for availability it is recommended to make periodic backups, redundancy and duplication [24].

3.3.3. Software security

Software security is used to improve application security. This is in order to avoid vulnerabilities such as a buffer overflow [39]. Creating a vulnerable application can enable its exploitation by malicious users. There are currently many security threats that affect even cloud applications, some of which are not detected by security tools. Therefore, good control is required in the development process [37]. Software security is the main problem that cloud systems and application professionals may face. Data owners may be concerned that data and software are not under their control but are owned by the cloud. In addition, the data owner may not know where the data is geographically at any time [4].

3.3.4. Multitenant security

A feature of SaaS is multitenant [2]. The use of multitenant allows several clients to connect to the same logic of the application, ensuring that each one has their personalized application and does not have access to the data of another tenant [38]. In addition, it allows saving resources because efficiency is improved when using a shared infrastructure [18]. The use of this technology increases the fear of the service user, because their data may be in the same database as their competitor or a malicious user [5]. Because the data of the different tenants are in the same infrastructure, it can happen that if one tenant is attacked, this can affect the others [38]. In addition, forensic analysis is difficult [33].

3.4. Will it be possible to classify threats, attacks and countermeasures?

This section presents the elaborated classifications for threats, attacks and additional security measures. The threats were classified based on their type and the security objective involved. In the case of the attacks, they were classified according to the type of STRIDE threat that they represent and their respective security objectives involved. Finally, the security measures were classified according to the security group they represent.

Table 6
Classification of threats and the security objectives affected

Threat	Threat type	Security objectives
Loss of control over resources	4	Confidentiality
Misuse of cloud computing resources	1,5	Confidentiality
Different service delivery/receiving models	5	Confidentiality
Insecure interface and API	2,3	Confidentiality, integrity, availability
Malicious insiders	1,5	Confidentiality, integrity, availability
Data scavenging	4	Confidentiality
Data loss or leakage	2,4	Availability
Service/account hijacking	2	Confidentiality
Risk profiling	5	Confidentiality
Identity theft	2	Confidentiality, integrity

3.4.1. *Threat classification*

In this work, a classification of the threats described in Section 2.5.1 was presented. This classification was made based on the type of the threats. Furthermore, the authors also present what kind of security objective is involved in each threat. Table 6 shows the classification carried out by the authors. The types of threats are represented by a number, according to the following list:

(1) The threat agents can carry out an attack. This Type of threat can be originated by a user or by software.
(2) An application or an attacker that makes attacks on the network from the Internet.
(3) An application of malicious logic.
(4) An attacker being a cloud client who has shared resources of an infrastructure.
(5) Malicious experts are human threats that act on behalf of cloud providers.

3.4.2. *Attack classification*

Created by Microsoft, STRIDE is a way to classify threats. It employs the following six categories. The category of spoofing contains the threats that lead to identity theft and authentication information. Manipulation is when the data is intentionally modified. Repudiation is when another user of the system performs actions outside the control of the administration. Disclosure of information refers to unauthorized access to information. Denial of service involves attacking the availability of the system. Lastly, elevation of privileges occurs when a user manages to increase their level of privileges within the system [14,16].

Table 7 shows a new classification regarding the types of threats caused by different attacks. Furthermore, the security objectives involved in each attack are presented in the same table.

3.4.3. *Classification of additional security measures*

Another contribution in this work has been the classification of the countermeasures presented in Section 2.5.3. We classified such countermeasures into five groups: security measures in charge of the cloud provider, security in tenant environments, cloud-based secure authentication, antivirus and IDS and Methods for mitigating Authentication and Access Control threats. The groups of security measures are presented in Table 8.

Table 7

Classification of attacks in the STRIDE model and the security objectives affected

Attack	STRIDE type	Security objectives
ARP Spoofing	Spoofing	Authentication
Broken authentication	Spoofing	Authentication
Backdoor and debug options	Tampering	Integrity
Buffer overflow	Elevation of privileges	Authorization
Code injection	Spoofing, tampering	Authentication, integrity
Cookie poisoning	Spoofing	Authentication
Cross-site request forgery (CSRF)	Tampering	Integrity
DNS poisoning	Spoofing	Authentication
Dumpster diving	Information disclosure	Confidentiality
Eavesdropping	Information disclosure	Confidentiality
EDoS	Denial of service	Availability
Google hacking	Information disclosure	Confidentiality
Hash value manipulation	Tampering	Integrity
Hidden field manipulation	Tampering	Integrity
Malware injection and steganography attacks	Tampering	Integrity
Man-in-the-middle attack	Information disclosure	Confidentiality
Meta data spoofing attack	Spoofing	Authentication
Phishing attack	Spoofing	Authentication
Port scanning	Information disclosure	Confidentiality
Race condition	Elevation of privileges	Authorization
Replay attack	Tampering	Integrity
Reused IP address	Information disclosure	Confidentiality
Service injection attack	Spoofing, tampering	Authentication, integrity
Shared architectures	Spoofing	Authentication
Sniffing	Information disclosure	Confidentiality
Social engineering	Spoofing, information disclosure	Authentication
Sybil attack	Elevation of privileges	Authorization
User to root attack	Elevation of privileges	Authorization
XML signature wrapping attack	Elevation of privileges	Authorization
Zombie Attack	Denial of service	Availability

3.5. *What security measures can be applied in SaaS?*

In Section 2.5.2, some forms of mitigation were also presented at the end of each attack. A summary of all attacks and their countermeasures was presented in Table 5.

Section 2.5.3 presented a series of general recommendations to be applied by both the provider and the SaaS customer.

4. Conclusions and future work

In this paper, a systematic review of 47 selected papers was made in order to describe the threats, attacks and countermeasures for SaaS. The systematic review was carried out based on the SALSA Framework inspired by [12]. This was extremely useful because it was only necessary to read a few articles completely. From a total of 6723 papers, a complete reading was only necessary for 47 articles,

Table 8
Additional security measures by group

Group	Security measures
Countermeasures by the provider	Encrypted credentials Guarantee CIA Training of employees Monitor the online movements of employees End-to-end encryption Malicious activity scanning Implementation of secure APIs Business continuity plans
Securing the SaaS multi-tenant environment	Database-based segmentation Encryption The tenant must know the existing protections
CSA	Countering DoS attacks Strong authentication
Antivirus and IDS	Monitors and blocks any malicious code Analyze the files that are transferred in the cloud Detect threats and stop them Detects anomalies within the network
Mitigating authentication control threats	Applying single sign-on policies Implementing multifactor authentication Implementing biometric authentication Implementing RSA encryption Implementing an intrusion and firewall detection system

which results in a saving of time. In addition, using this technique, only papers relevant for this research were selected. Therefore, the systematic review allowed time savings and the use of quality articles for this investigation.

In addition to the security issues that affect the cloud, applications in SaaS inherit the problems of the traditional model, such as Web Services. This study aimed to find out what are the threats facing organizations that are planning to migrate their applications to the cloud, as well as the security measures to reduce the likelihood of their occurrence. Currently there are few papers that describe safety aspects exclusively in SaaS, so this study aimed to remedy this bias in the literature.

The threats that represent the greatest risk are identity theft, account theft, and malicious employees. These could affect the confidentiality, integrity, and availability of the information. Moreover, a total of 30 attacks were presented, which can be considered as the most common that can occur in SaaS. The Zombie attack, Man-in-the-middle Attack, Code Injection, and Social Engineering are some examples. Therefore, the SaaS service model inherits the security problems of traditional applications which can here be even more devastating, such as the zombie attack, due to the architecture of the cloud.

The implementation of the security measures presented in this investigation can prevent the loss of information and money in an SaaS environment due to the prevention of computer attacks.By carrying out this work, a contribution is made to the knowledge of security problems in SaaS as well as how to mitigate them, which is useful for scholars and professionals in the areas of software, security and cloud computing. Thus, the next step in this research will be to create a security framework to assess the security of an application in SaaS.

We consider that some of the current security challenges in SaaS are directly related to the users of the service, such as: knowing their responsibilities for a safe use of the cloud, having measures to detect insecure APIs and being alerted when there is a vulnerability in their computer equipment. In addition, another aspect to consider is the employees of the cloud provider, because their activities can be difficult to trace. Therefore, future research works are invited to address the security issues surrounding SaaS users and employees of the cloud provider. Human behavior related to computer security would be a very interesting topic for research. For example, knowing the most common actions performed by a disgruntled employee would help to be more prepared in the event of any such incident. In the case of users, it would be useful to know what are the most common mistakes they make that allow an attacker to perform a malicious action. Finally, another topic to deal with may be knowing the loss of resources that these human behaviors represent.

References

[1] S. Alam, M. Muqeem and S.A. Khan, Review on security aspects for cloud architecture, *International Journal of Electrical and Computer Engineering* **8**(5) (2018), 3129–3139.

[2] A. Aldaej, Ravichandran, M.G. Ahamad and M.R.A. Dhivakar, A study on state of the art in security and privacy issues on cloud computing, *Journal of Engineering and Applied Sciences* **12**(11) (2017), 9220–9226.

[3] M. Ali, S.U. Khan and A.V. Vasilakos, Security in cloud computing: Opportunities and challenges, *Information Sciences* **305** (2015), 357–383. doi:10.1016/j.ins.2015.01.025.

[4] S.A. Aljawarneh and M.B. Yassein, A conceptual security framework for cloud computing issues, *International Journal of Intelligent Information Technologies* **12**(2) (2016), 12–24. doi:10.4018/IJIIT.2016040102.

[5] M. Almorsy, J. Grundy and A.S. Ibrahim, Adaptable, model-driven security engineering for SaaS cloud-based applications, *Automated Software Engineering* **21**(2) (2014), 187–224. doi:10.1007/s10515-013-0133-z.

[6] L. Ben Arfa Rabai, M. Jouini, A. Ben Aissa and A. Mili, A cybersecurity model in cloud computing environments, *Journal of King Saud University – Computer and Information Sciences* **25**(1) (2013), 63–75.

[7] V. Casola, A. De Benedictis, M. Rak and E. Rios, Security-by-design in clouds: A security-SLA driven methodology to build secure cloud applications, *Procedia Computer Science* **97** (2016), 53–62. doi:10.1016/j.procs.2016.08.280.

[8] G.-Y. Chan, F.-F. Chua and C.-S. Lee, Intrusion detection and prevention of web service attacks for software as a service: Fuzzy association rules vs fuzzy associative patterns, *Journal of INtelligent & Fuzzy Systems* **31**(2) (2016), 749–764. doi:10.3233/JIFS-169007.

[9] R. Charanya, M. Aramudhan, K. Mohan and S. Nithya, Levels of security issues in cloud computing, *International Journal of Engineering and Technology* **5**(2) (2013), 1912–1920.

[10] L. Coppolino, S. D'Antonio, G. Mazzeo and L. Romano, Cloud security: Emerging threats and current solutions, *Computers & Electrical Engineering* **59** (2017), 126–140. doi:10.1016/j.compeleceng.2016.03.004.

[11] M. Darwish, A. Ouda and L.F. Capretz, A cloud-based secure authentication (CSA) protocol suite for defense against denial of service (DoS) attacks, *Journal of Information Security and Applications* **20** (2015), 90–98. doi:10.1016/j.jisa.2014.12.001.

[12] I.F. del Amo, J.A. Erkoyuncu, R. Roy, R. Palmarini and D. Onoufriou, A systematic review of augmented reality content-related techniques for knowledge transfer in maintenance applications, *Computers in Industry* **103** (2018), 47–71. doi:10.1016/j.compind.2018.08.007.

[13] P. Derbeko, S. Dolev, E. Gudes and S. Sharma Security and privacy aspects in MapReduce on clouds: A survey, *Computer Science Review* **20** (2016), 1–28. doi:10.1016/j.cosrev.2016.05.001.

[14] T. Halabi and M. Bellaiche, A broker-based framework for standardization and management of cloud security – SLAs, *Computers and Security* **75** (2018), 59–71. doi:10.1016/j.cose.2018.01.019.

[15] M. Hawedi, C. Talhi and H. Boucheneb, Security as a service for public cloud tenants (SaaS), *Procedia Computer Science* **130** (2018), 1025–1030. doi:10.1016/j.procs.2018.04.143.

[16] J.B. Hong, A. Nhlabatsi, D.S. Kim, A. Hussein, N. Fetais and K.M. Khan, Systematic identification of threats in the cloud: A survey, *Computer Networks* **150** (2019), 46–69. doi:10.1016/j.comnet.2018.12.009.

[17] S. Iqbal, M.L.M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M.K. Khan and K.-K.R. Choo, On cloud security attacks: A taxonomy and intrusion detection and prevention as a service, *Journal of Network and Computer Applications* **74** (2016), 98–120. doi:10.1016/j.jnca.2016.08.016.

[18] J. Jang-Jaccard and S. Nepal, A survey of emerging threats in cybersecurity, *Journal of Computer and System Sciences* **80**(5) (2014), 973–993. doi:10.1016/j.jcss.2014.02.005.

[19] A. Joshi, S.T. King, G.W. Dunlap and P.M. Chen, Detecting past and present intrusions through vulnerability-specific predicates, in: *ACM SIGOPS Operating Systems Review*, Vol. 39, ACM, 2005, pp. 91–104.

[20] S. Khadar, A. Manoj and D.L. Bhaskari, Cloud forensics – A framework for investigating cyber attacks in cloud environment, *Procedia Computer Science* **85** (2016), 149–154. doi:10.1016/j.procs.2016.05.202.

[21] M.A. Khan, A survey of security issues for cloud computing, *Journal of Network and Computer Applications* **71** (2016), 11–29. doi:10.1016/j.jnca.2016.05.010.

[22] N. Khan and A. Al-Yasiri, Identifying cloud security threats to strengthen cloud computing adoption framework, *Procedia Computer Science* **94** (2016), 485–490. doi:10.1016/j.procs.2016.08.075.

[23] K. Krombholz, H. Hobel, M. Huber and E. Weippl, Advanced social engineering attacks, *Journal of Information Security and Applications* **22** (2015), 113–122. doi:10.1016/j.jisa.2014.09.005.

[24] P.R. Kumar, P.H. Raj and P. Jelciana, Exploring data security issues and solutions in cloud computing, *Procedia Computer Science* **125** (2018), 691–697. doi:10.1016/j.procs.2017.12.089.

[25] E. Loukis, M. Janssen and I. Mintchev, Determinants of software-as-a-service benefits and impact on firm performance, *Decision Support Systems* **117** (2019), 38–47. doi:10.1016/j.dss.2018.12.005.

[26] P. Mishra, E.S. Pilli, V. Varadharajan and U. Tupakula, Intrusion detection techniques in cloud environment: A survey, *Journal of Network and Computer Applications* **77** (2017), 18–47. doi:10.1016/j.jnca.2016.10.015.

[27] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel and M. Rajarajan, A survey of intrusion detection techniques in cloud, *Journal of Network and Computer Applications* **36**(1) (2013), 42–57. doi:10.1016/j.jnca.2012.05.003.

[28] M. Muhil, U.H. Krishna, R.K. Kumar and E.A.M. Anita, Securing multi-cloud using secret sharing algorithm, *Procedia Computer Science* **50** (2015), 421–426. doi:10.1016/j.procs.2015.04.011.

[29] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber and E. Weippl, Dark clouds on the horizon: Using cloud storage as attack vector and online slack space, 2011.

[30] Nagarjuna, C.C.K. Srinivas and S. Sajida, Security techniques for multi tenancy applications in cloud, *International Journal of Computer Science and Network Security* **15** (2015), 80–83.

[31] T. Pai and P.S. Aithal, A review on security issues and challenges in cloud computing model of resource management, 2017.

[32] A. Philpott, Identity theft – Dodging the own-goals, *Network Security* **2006**(1) (2006), 11–13. doi:10.1016/S1353-4858(06)70323-3.

[33] A. Pichan, M. Lazarescu and S.T. Soh, Cloud forensics: Technical challenges, solutions and comparative analysis, *Digital Investigation* **13** (2015), 38–57. doi:10.1016/j.diin.2015.03.002.

[34] G. Ramachandra, M. Iftikhar and F.A. Khan, A comprehensive survey on security in cloud computing, *Procedia Computer Science* **110** (2017), 465–472. doi:10.1016/j.procs.2017.06.124.

[35] R.V. Rao and K. Selvamani, Data security challenges and its solutions in cloud computing, *Procedia Computer Science* **48** (2015), 204–209. doi:10.1016/j.procs.2015.04.171.

[36] C. Saadi and H. Chaoui, Cloud computing security using IDS-AM-Clust, Honeyd, Honeywall and Honeycomb, *Procedia Computer Science* **85** (2016), 433–442. doi:10.1016/j.procs.2016.05.189.

[37] A. Shadi, Aljawarneh, A. Alawneh and R. Jaradat, Cloud security engineering: Early stages of SDLC, *Future Generation Computer Systems* **74** (2017), 385–392. doi:10.1016/j.future.2016.10.005.

[38] A. Singh and K. Chatterjee, Cloud security issues and challenges: A survey, *Journal of Network and Computer Applications* **79** (2017), 88–115. doi:10.1016/j.jnca.2016.11.027.

[39] S. Singh, Y.-S. Jeong and J.H. Park, A survey on cloud computing security: Issues, threats, and solutions, *Journal of Network and Computer Applications* **75** (2016), 200–222. doi:10.1016/j.jnca.2016.09.002.

[40] G. Somani, M.S. Gaur, D. Sanghi, M. Conti and R. Buyya, DDoS attacks in cloud computing: Issues, taxonomy, and future directions, *Computer Communications* **107** (2017), 30–48. doi:10.1016/j.comcom.2017.03.010.

[41] N. Srinivasu, O.S. Priyanka, M. Prudhvi and G. Meghana, Multilevel classification of security threats in cloud computing, *International Journal of Engineering and Technology (UAE)* **7**(1.5) (2018), 253–257.

[42] N. Subramanian and A. Jeyaraj, Recent security challenges in cloud computing, *Computers and Electrical Engineering* **71** (2018), 28–42. doi:10.1016/j.compeleceng.2018.06.006.

[43] W.T. Tsai, X.Y. Bai and Y. Huang, Software-as-a-service (SaaS): Perspectives and challenges, *Science China Information Sciences* **57**(5) (2014), 1–15. doi:10.1007/s11432-013-5050-z.

[44] A. Vasudeva and M. Sood, Survey on sybil attack defense mechanisms in wireless ad hoc networks, *Journal of Network and Computer Applications* **120** (2018), 78–118. doi:10.1016/j.jnca.2018.07.006.

[45] N. Vurukonda and B.T. Rao, A study on data storage security issues in cloud computing, *Procedia Computer Science* **92** (2016), 128–135. doi:10.1016/j.procs.2016.07.335.

[46] W. Wang, Data security of saas platform based on blockchain and decentralized technology, in: *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020, pp. 848–851. doi:10.1109/ICICT48043.2020.9112421.

[47] M. Wu and Y.B. Moon, Taxonomy of cross-domain attacks on CyberManufacturing system, *Procedia Computer Science* **114** (2017), 367–374. doi:10.1016/j.procs.2017.09.050.

[48] D. Ye, T.-Y. Zhang and G. Guo, Stochastic coding detection scheme in cyber-physical systems against replay attack, *Information Sciences* **481** (2019), 432–444. doi:10.1016/j.ins.2018.12.091.

[49] Z.M. Yusop and J.H. Abawajy, Analysis of insiders attack mitigation strategies, *Procedia – Social and Behavioral Sciences* **129** (2014), 611–618.