Cloud Computing Security

Lukas Bordak
ALEF Distribution SK, s.r.o., Bratislava, Slovak Republic
lukas.bordak@alef.com

***Abstract*—The aim of this paper was to analyze the issue of the Cloud security solutions as well as to design a secure access for a demonstrative cloud application. Several security items have been chosen for a secure access to the Cloud; moreover, the application has been implemented to Heroku, from Sales force company, with the use of the existing cloud technologies. The created application shows the possibilities of the application's development with the usage of the security items.**

***Keywords*— Authentication, Authorization, Cloud, Heroku, PaaS, Security**

I. INTRODUCTION

Further, the more widespread use of server virtualization [1] has been, a new way has formed for the information technology industry to deliver the complex services we collectively call the *cloud*. Cloud represents one of the most significant trends in the world of information technology today.

Using cloud services is a very attractive option for many organizations to reduce their costs associated with running their own infrastructure. A big plus is certainly reduction in time required to build it. This allows the cloud model on which it is based to shine. The essence of cloud services is their great flexibility. Customers can use computing resources and adapt the use of paid resources according to their needs. A potential user wishing to use such a service must consider the potential risks associated with the transfer of his/her valuables but also sensitive data to the third party application. It is in the personal interest that such data should not be exposed to any possible risks. In the current situation, the cloud platforms are popular as one of the possibilities of using the cloud. Very popular thing about the clouds among the users is the fact that they do not have to be very interested in the clouds internal functioning. For providers of such cloud to-platform solutions, the primary challenge is to convince potential customers that their security measures are of high quality and achieve a high level of security. Service providers aim is to assure the customers that all possible efforts are made to eliminate any security threats that may occur. If there were security incidents on the service provider side, it would mean a loss of trust with customers and the public. Ultimately, that would have a disastrous impact on its platform / solution.

This work focuses on the analysis of cloud security as such. However, the paper is also very interested in the security features used in the PaaS solution, because it has been implemented in our application.

The main goal of this paper is to analyze what security elements we encounter. We mainly focus on user safety, data security, application isolation, backup and open development assurance. For this purpose, the application was developed, which contains some publicly available security features. Subsequently, this application is implemented on the selected platform mentioned earlier.

II. CLOUD DEFINITON

Cloud computing is defined as a model that provides its user with network infrastructure, storage, and computing power. Furthermore, this model enables scalability of these resources while automating the tasks associated with proper service delivery.

III. CLOUD PARTITIONING

The word cloud represents a wide range of technologies, so we often encounter inconsistent definitions of the term. The American federal agency NIST [2] dealt with the exact definition of this term. The agency aims to support US innovation by accelerating technology through ways to increase economic security and quality of life. The cloud can therefore be categorized by multiple characters. The most basic division is to divide the cloud according to who owns the infrastructure, where the cloud platform is located and who has the opportunity to use the platform. Another attribute of the cloud split is the service model. Deploying a cloud describes how a cloud is delivered. This model lists four different types, such as public, private, community and hybrid described in the next section (section IV) in more details.

IV. CLOUD DEPLOYMENT MODEL

Cloud deployment describes how a cloud is provided. This model introduces four different types, such as public, private, community and hybrid clouds [3].

A. Public Cloud

The public cloud can also be called a classic model with a public platform that can be charged depending on usage. This scheme is generally provided to the public and it is accessible via the Internet. The platform itself does not come into contact with users, on the contrary it is managed by providers. Cisco Webex [4] is one of the most popular public cloud computing services.

B. Private Cloud

This type of cloud is characterized as a cloud that is ran only for a particular organization (either by itself or by the third party services). The software packages provide the ability to create a private cloud on their own and their availability is relatively high. Through a private cloud, the user can provide greater security and greater control, as their resources are shared within one organization.

978-1-7281-4967-7/19/$31.00 ©2019 IEEE

The private cloud can be categorized according to the nature of the infrastructure, whether it is a custom or deployed structure [5]. With its own infrastructure, all the cloud resources owned by the service organization belong to the cloud. A cloud located on a leased infrastructure is characterized by a more consistent separation of resources that are leased from a third party compared to the public cloud.

C. Community Cloud

Community cloud's task is to share resources in a group of organizations characterized as responsible for running and managing solutions. If there are organizations with similar requirements for cloud services and the public cloud is disadvantageous for them, this cloud becomes a very suitable alternative for them [4].

D. Hybrid Cloud

This is the so-called compromise between public and private clouds. It is characterized by the location of the components of the user systems on its own server and the other parts can be located in the public cloud [6]. At first glance, this cloud acts as a whole, but is deeper interconnected with standardization technologies.

V. Cloud by Service Model

This model gives the user the opportunity to use resources from the provider. We also know it as a distribution model (Service Delivery Model) of cloud computing. It comprises the three basic pillars SaaS, PaaS and IaaS.

A. Platform as a Service (PaaS)

Platform as a Service is primarily for developers implementing their own software solutions. The main purpose of preconfigured environments is to make it easier to create new applications, but also to simplify their production deployment along with maintenance during runtime. These environments also help to eliminate the need to separately purchase middleware or service providing software, or as a component of applications created through it.

Most PaaS, provided with development tools such as plug-ins or command line tools, can mediate interaction between the platform and users. Not every PaaS meets user requirements. Therefore, it is necessary to verify support for technologies that are intended to be used by the user. Platform as a service - provides all the resources necessary to create applications and services exclusively via the Internet, without the need to download or install the necessary software.

1) Positive aspects of PaaS:

- ability to collaborate geographically isolated development teams,
- ability to merge web services from multiple sources,
- opportunity to save money by using integrated infrastructure services for security, scalability and failover, instead of having to be built and tested separately,
- the possibility of financial savings through the use of high-level programming abstractions.

2) Negative aspects of PaaS:

- vendors use proprietary services or development languages. Some developers are worried that they will depend on the only possible provider,
- vendor may allow the application to be transferred to the competition, the cost of transfer is usually higher than transferring applications between common hosts [7].

B. Software as a Service (SaaS)

Software as a Service (SaaS) is a model in which an application is hosted and offered as a service to customers who access the service over the Internet. If the software is hosted externally, the customer does not need to manage or support it. On the other hand, the customer has no influence on when the hosted service provider decides to change the app. The concept is based on the fact that ready-to-use software can be used and does not have to be difficult to integrate with other systems. The provider takes care of all repairs, updates and at the same time keeping the infrastructure running.

1) Other benefits of SaaS:

- **knowledge of the web** - most workers have access to the web on their computer and knowhow to use the web. As a result, they can learn to work with applications delivered externally very quickly,
- **fewer employees** - IT systems are associated with overheads of salaries, benefits, insurance, and space in the building. Thanks to remote access to applications, it is not necessary to employ more IT staff than is really necessary,
- **customization** - legacy applications were difficult to customize and required code manipulation. SaaS-based applications are much easier to edit and can accurately meet organizational requirements,
- **better marketing** - a provider who had previously developed an app for a very narrow market might have had problems marketing it. As part of the SaaS concept, a provider can offer a global solution worldwide,
- **site reliability** - a site can be considered a potential failure site. Occasionally we may encounter unavailability, but in practice the web is relatively reliable,
- **security** - very often, the trusted Secure Sockets Layer (SSL) protocol is used. With this protocol, customers can access their applications securely without having to deploy complex system and costly solutions such as VPN networks,
- **more bandwidth** - in recent months, line capacity has increased significantly, improving the quality of service leading to faster data transfer. Organizations can therefore rely on them to access their applications with low latency and high speed.

2) Disadvantages of SaaS:

- an organization with very specific computing needs may not find an application that would be available in the SaaS,
- vendor dependency (inability to transfer application to another vendor),
- competition in the form of open source software combined with cheap hardware [7].

C. Infrastructure as a Service (IaaS)

The IaaS model is used to provide storage networking and computing resources to users. This model is mainly used to reduce the time required to create the desired environment, eliminate the initial investment in creating your own data center, and also eliminate the common investment associated with normal operations. Each user has a choice in terms of the performance of the servers offered, which can then be used for the functionality of their applications [8].

While SaaS and PaaS provide applications to customers, IaaS offers hardware so organization scan use it in any way they want. Instead of having to buy servers, software, racks and pay for their placement in a data center, customers use IaaS to rent these resources from the service provider [7][9].

VI. CLOUD SECURITY TECHNIQUES

01

Wimalasekera

A. Data Encryption

This technique is used first by the data owner. Prior to cloud storage, data should be encrypted from plain text to separate the ability to store data from the ability to use it, even in the cloud of the provider. For this technique to be effective, the encryption key should be stored in or with the other partner. To ensure that only authorized users can decrypt data, if necessary [10].

Data encryption can be accomplished with many tools to save data to and from the cloud. Either using software, hardware, or a combination thereof. Many dealers offer such solutions. Online Tech offers two methods. The first is full disk encryption, which consists of encrypting the stored data on the hard disk during the entire boot process [10]. The second method is to encrypt the entire drive using the AES (Advanced Encryption Standard) encryption algorithm, so data remains inaccessible even if the hard drive is stolen or lost.

Encryption that uses Secure Socket Layer (SSL), Virtual Private Networks (VPN), and IPsec (IP) security is the best encryption solution for data transfer [11]. SSL is the most widespread cloud infrastructure and is adopted by most CSPs. It consists in building a secure channel between the server located in the cloud infrastructure and the cloud user to ensure communication between the two entities. First, the server and the client authenticate each other with a digital certificate that is created, managed, and authenticated through PKI (Public Key Infrastructure). After successful authentication, the SSL connection is established. Subsequently, the communicating objects use ciphers and a shared key to decrypt the information exchanged during sessions. Newer version of SSL connection is TLS connection, i.e. Transport Layer Security. This protocol was recently designated as TLS / SSL and one open source for its implementation is OpenSSL.

B. Data Access Controls: Physical and Logical

The physical category contains all measures taken to protect the data center where external data is stored to prevent physical attacks, such as controlled physical access, security guards, lockable server racks, integrated alarms, and video surveillance systems [12].

There are several techniques for the logical data access control (authentication, authorization etc.) which ensure that the right cloud users are attempting to access specific stored data

This category includes a set of techniques used across different cloud layers, such as Intrusion Detection Systems, Firewall, Role Based Access Control, Two-Factor Authentication and Audit.

C. Data Isolation

In a public cloud where multiple users use the same cloud, this technique is used to ensure the privacy of users. The goal of controlling access techniques is to achieve a logical separation either between cloud resources or between tenants in the cloud and creating a ”private” environment where users can only access their own data [13].

D. Intrusion Detection Systems

Intrusion Detection Systems (IDS) consist of monitoring nodes in the cloud and auditing user activity data. Therefore, to determine whether the user is behaving as expected or is experiencing unexpected behavior [14]. Based on the analysis of all users and their behavior, IDS determine whether the user is an intruder or not.

IDS use two methods to analyze the checked data: abuse detection and anomaly detection. Reverting both methods uses the IDS kernel to calculate the probability of an action that constitutes an attack if the probability is high enough, sends alerts to other nodes, and notifies the system administrator [15].

E. Role-based Access Control

Role-based access control is a technique used to manage user access to cloud resources. This management consists of permission to grant roles (logical grouping of tasks and permissions) instead of individuals. This requires creating roles based on the individual’s role and responsibility within the company. Subsequently, users are assigned to these roles, so they have a set of minimum access rights based on their role assignment.

F. User Authentication

Cloud users must prove their identity with the CSP to ensure that the correct identity or person has access to their stored data in the cloud. Knowledge Based Authentication (KBA) consists of verifying a user’s identity through information that is protected by providing answers to knowledge-related questions. These questions are either preset or generated from multiple public and private sources. This technique is mainly used when a user wants to reset their password or prove their identity as part of a multi-factor authentication process.

1) Multifactor Authentication

In the cloud, single-factor authentication is no longer considered secure because it can easily be hacked by password-collecting programs. Multi-factor authentication consists of authenticating a user with two or more factors combined to prove the user's identity before granting access to the cloud resources. These factors come from independent categories:

- something the user knows: PINs, passwords, codes etc.,
- something the user has: tokens, smart cards, USB tokens etc.,
- something personal to the user: biometrics (fingerprints, face, iris, voice etc.).

2) Two-factor Authentication

It is the most advanced model of multifactor authentication and most popular in the cloud systems because it is easy to use. It also provides a highly effective authentication process. The most commonly used combination in a cloud infrastructure is a one-time password (OTP) with a PIN code [16]. OTP is a unique password that can only be used once. It is generated on a secret remote server and sent to the user attempting to log in after successfully entering the username and PIN. OTP is sent either to the user's mobile phone via short message or automatic call, or received directly while using the OTP hardware token.

Two-factor authentication can also be achieved by one of the following combinations: PIN smart card, PIN biometrics, smart card biometrics and bi-factor biometrics.

3) Three-factor authentication

This type of authentication is used only in some cloud infrastructures that store business and government data that require a high degree of security and privacy (armies, national security). It consists of authenticating users with three independent factors using one of the following combinations:

- PIN with bi-factor biometric technologies,
- smart card with PIN and biometry,
- smart card with bi-factor biometric technologies,
- smart card with PIN and biometric technology,
- three-factor biometric technology.

VII. Design and Implementation of our Cloud Security System

Often times it happens that during the developing process of a system with security features most of the time is devoted to the application itself. Its main part is designed and developed consistently, but security features are only designed in the final stages of the process. Therefore, it is important to think and design the security features of the application in the early stages of the development process. The reverse procedure can often lead to a low level of security for the entire application or complicate and prolong the creation process.

One of the points of the work was to create a sample application to which some of the acquired security and security information in the cloud was applied. A simple application with a graphical user interface was chosen that could be expanded and provided at a higher level than SaaS in the future. The application displays a simple, graphical, user interface in an external storage system, where security features are used to secure access or save and download files.

A. Tasks of Users in the Application

The application has three cases of users who can use the application. It is important to explain each user's tasks and capabilities. This analysis can give an idea of the functionality of the created application: *logged in user* (uploads files to the cloud storage, downloads them, categorizes files into folders, changes login data, logs out of the cloud storage), *logged out user* (registers as a new user, takes an advantage of the forgot password and logs in to the cloud storage) and a *system administrator*, who manages user accounts. Its role is to act accordingly when errors occur, is responsible for the correct operation of the system and maintain databases. Next figure (Figure 1) shows the relation among all the users within the system.

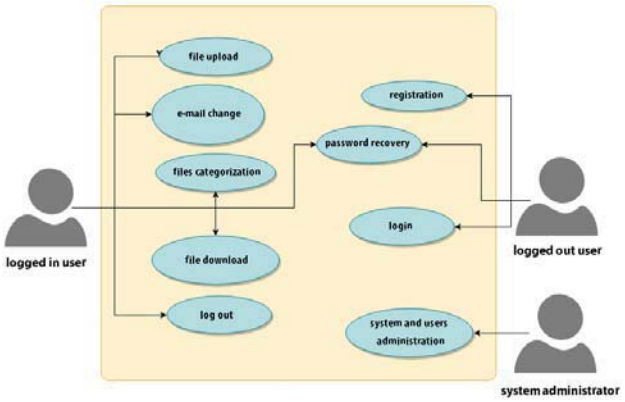

Figure 1. Various tasks of users and their relation within the system

B. System Security Requirements

It is important to define security requirements at an early stage of a system design.

The spheres better described by The Open Web Application Security Project (OWASP) in their studies on the major security threats to Internet applications have been considered to create a list that includes security measures. Security of applications and systems is a very grim theme nowadays, so there is no universal list of security risks, because different systems require different importance of various security features to fully secure a system or parts of the system. The following safety requirements have been developed in this work:

1) Registration, Authorization and Authentication.

- **User registration** - the user enters their email address and their unique password.
- **Authorization** - the user has access only to their account and only to their files, so they don't need to be worried about unauthorized access to their files,
- **Authentication during registration** - the system uses two-factor authentication. So for the registration, the user must be proved to the system as authorized. That is, while logging in for the first time there is a verification link sent to the user-selected email address, which needs to be clicked and accepted.

- **Authentication during login** - the user must enter a unique, automatically generated four-digit code each time he tries to login. The code is sent to the e-mail address.

2) Data Encryption

- **Encrypted System Communication -** the application would run on HTTPS.
- **Database Password Encryption** - user passwords would be encrypted in the database using the selected cryptographic cipher.

3) Additional Security Requirements

- **Input Validation** - Inputs will be validated on the server side. Entry validation is necessary for security against: **Cross-Site Scripting** (XSS) - JavaScript, the programming language used to create web pages, is located on a web page and can manipulate the objects of that page. It can open new browser windows and load additional pages or external objects. The security breach could occur if the script would retrieve any objects from any external location; **Cross-Site Request Forgery** (CSRF) - this is an abuse of the site's trust to the user as opposed to XSS, which abuses the user's trust to the page [17]. The tasks of the page are closely related to the URL and they allow you to act on request. The point of the attack is that the user visits a page that has been compromised and that action is taken without the user's knowledge. Impact of the attack is depended on the role of the compromised user. End data is compromised in the case of a normal user. However, if an attack is made by an administrator user, the attacker gains control of the entire application. In particular, community sites or banks or sites offering payment actions are being attacked. The attack can even be executed on a secure connection page [18]; **SQL Injection** - this is an attack technique that is used by an attacker to exploit an application, resulting in unauthorized access to the database or retrieving information from the database. An attacker exploits the sensitivity of SQL injection remotely without authentication, inserting a malicious string as an input to the application to steal confidential information [19].
- **Database Access Control** - the application will be scanned against unauthorized access to its database.
- **Forgotten Password Recovery** – users could change password for new unique password.
- **Change Login Data** - such as password and e-mail. Logged in users could change their password and e-mail to obtain new login credentials.

VIII. Technologies used for Sample Application Development and Security

Encryption of a client (browser) and server communication - the application runs on secure port 443 (https) and the Lets Encrypt security certificate with automatic extension is implemented.

Password Encryption - user login credentials are SHA1 encrypted in the database.

Cross-Site Request Forgery - by using the authorization token, we are talking about the most efficient but also implementation-complicated method where a user generates a random string before performing each operation. When executed, this string is checked. This method prevents applications from being used in multiple windows at the same time. This problem is solved by generating an authorization token when the previous one has expired or the authorization token is generated for each new request that is made. Instead of the database, it is stored in a session variable, and at the same time access to the application is also possible [20][21].

SQL injection - each of the SQL injection detection techniques requires queries passing from the application to the database server to pass through the interlayer. One of the techniques used is SQL Parse Tree Validation [22]. The syntax tree modeled from the SQL query is used to store user data in sheets. To avoid complications, dangerous query, queries can be detected by comparing the current query tree with the intended query. If there are some differences, the query is evaluated as an attack and is rejected.

PHP - a scripting language which is largely used on the server side and is compiled during the process. That being said, it means that any occurred errors will not be detected until the application would be running. It serves more complex functions and operations, such as processing data from forms, writing and saving data to a database, parsing inputs and outputs. It is used to create web pages from the server side [23]. The application was created with the PHP version 7.1.x.

Codeigniter Framework - it is a modern PHP library that simplifies work, gives direction and keeps the code clear and transparent. It is based on state-of-the-art MVC (model-view-controller) architecture. The framework works based on root. The URL address is parsed to individual parameters and these are sent to the methods in the controller and then they are processed there. If so, the code can be broken down into a model or even into the third-party libraries. Subsequently, the data is processed and a response is sent.

Technologies such as **HTML**, **CSS** and **JavaScript** were used as well as a **Bootstrap** framework based on those technologies.

For the deployment phase the PaaS **Heroku** was used, which is one of the first Paas cloud platforms. It was created with the aim of providing an environment for programs written in the Ruby programming language, as well as for their functioning. Gradually, Heroku improved and of course expanded. Currently, it has an expanded range of programming languages used for the development on the platform such as Java, PHP, Python or Perl [24].

IX. Conclusion

Various security options have been outlined, as well as many technologies for securing cloud applications. Some of the security options were used and application security technologies described in our sample application. The use of the PaaS cloud service is highly exploited due to the openness of these platforms.

The application has used several technologies and procedures to secure the system. The application was created using Java. The VueJs and Axios JavaScript libraries were used. On the server side, PHP was used along with the Codeigniter framework library. Framework Bootstrap was used to create the design. The application also has user registration and two-factor user authentication. The application is implemented on HTTPS protocol. It uses the SHA algorithm to secure passwords and encrypt them in the database. And against unauthorized inputs, the application uses protection against SQL Injection, Cross-Site Scripting and Cross-Site request forgery.

The platform is certainly an interesting way to use and then deploy different applications, because this form of deployment has multiple uses for certain groups of users. These are user groups that do not have sufficient capital to create an infrastructure with equivalent security features. At the same time, these users must be satisfied that they use a foreign infrastructure and therefore their data is outside their infrastructure. Therefore, trust between the potential user of such a platform cloud service and its provider is very important.

The aim of our work was to analyze the security of clouds, create an architecture to secure the application and then use the knowledge and implement it. Therefore, based on the acquired knowledge, we have created a sample application. We implemented security features on it and then deployed it to the cloud computing environment.

REFERENCES

[1] L. Vokorokos, A. Baláž and B. Madoš, "Application Security through Sandbox Virtualization," in Journal of Applied Sciences, vol. 12, iss. 1, Acta Polytechnica Hungarica, Hungary, 2015, pp. 83-101.

[2] National Institute of Standards and Technology, [online] https://www.nist.gov/standardsgov/government.

[3] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," in National Institute of Standards and Technology, USA, September 2011. DOI: https://doi.org/10.6028/NIST.SP.800-145.

[4] D. Armstrong and K. Djemame, "Towards Quality of Service in the Cloud," in Proc. of the 25th UK Performance Engineering Workshop, UK, 2009.

[5] B. Kezia Rani, B. Padmaja Rani and A. Vinaya Babu, "Cloud Computing and Inter-Clouds – Types, Topologies and Research Issues," in Procedia Computer Science, vol. 50, Elsevier, 2015, pp. 24-29. DOI: https://doi.org/10.1016/j.procs.2015.04.006.

[6] E. Chovancová, N. Ádám, A. Baláž, E. Pietriková, P. Feciľak, S. Šimoňák and M. Chovanec, "Securing Distributed Computer Systems Using an Advanced Sophisticated Hybrid Honeypot Technology," in Computing and Informatics, vol. 36, Slovakia, 2017, pp. 113-139. DOI: https://doi.org/10.4149/cai_2017_1_113.

[7] A.T Velte T.J. Velte and R. Elsenpeter, "Cloud Computing: A Practical Approach," Tata McGraw-Hill, New York, 2011. ISBN 9788025133330.

[8] L. Badger, T. Grance, R. Patt-Corner and J. Voas, "Cloud Computing Synopsis and Recommendations," in National Institute of Standards and Technology, USA, May 2012. DOI: https://doi.org/10.6028/NIST.SP.800-146.

[9] A. Pekár, E. Chovancová, P. Fanfara and J. Trelová, "Issues in the Passive Approach of Network Traffics Monitoring," in IEEE 17th International Conference on Intelligent Engineering Systems (INES), Costa Rica, 2013, pp. 327-332. DOI: http://dx.doi.org/10.1109/INES.2013.6632836.

[10] The Top 20 Software as a Service (SaaS) Vendors, 2014 [online] http://www.clouds360.com/saas.php.

[11] L. Vokorokos, A. Baláž and M. Chovanec, "Distributed Detection System of Security Intrusions Based on Partially Ordered Events and Patterns," in Towards Intelligent Engineering and Information Technology, vol. 243, Springer, Germany, 2009, pp. 389-403.

[12] J. Marquilies, S. Pfleeger and C. Pfleeger, "Security in Computing," Prentice Hall, 5th Edition, 2015. ISBN 9780134085074.

[13] A. Pekár, M. Chovanec, L. Vokorokos, E. Chovancová, P. Feciľak and M. Michalko, "Adaptive Aggregation of Flow Records," in Computing and Informatics, vol. 37, iss. 1, Slovakia, 2018, pp. 142-164. DOI: https://doi.org/10.4149/cai_2018_1_142.

[14] L. Hung-Jen, R. L. Chun-Hung, L. Ying-Chih and T.Kuang-Yuan, "Intrusion detection system: A comprehensive review," in Journal of Network and Computer Applications, vol. 36, Elsevier, January 2013, pp. 16-24.

[15] K. Vieira, A. Schulter, C. B. Westphall and C. M. Westphall, "Intrusion Detection for Grid and Cloud Computing," in Cyber Security, Federal University of Santa Catarina, Brazil, 2010.

[16] J. Hurtuk, A. Baláž and N. Ádám, "Security Sandbox Based on RBAC model," in IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI), Romania, 2016, pp. 75-80. DOI: https://doi.org/10.1109/SACI.2016.7507343.

[17] T. Alexenko, M. Jenne, S. D. Roy and W. Zeng, "Cross-Site Request Forgery: Attack and Defense," in 7th IEEE Consumer Communications and Networking Conference, 2010. DOI: 10.1109/CCNC.2010.5421782.

[18] Cross-Site Request Forgery (CSRF) Prevention Cheat, 2009 [online] http://www.clouds360.com/saas.php

[19] N. Patel, F. Mohammed and S. Soni, "SQL Injection Attacks: Techniques and Protection Mechanisms," in International Journal on Computer Science and Engineering, vol. 3, 2011, pp. 199-203.

[20] J. Vrána, "PHP triky," [online] http://php.vrana.cz/cross-site-request-forgery.php.

[21] C. Shiflett, "Foiling Cross-Site Attacks," [online]: http://shiflett.org/articles/foiling-cross-siteattacks.

[22] G. T. Beuhrer, B.W. Weide and P. A. G. Sivilotti, "Using parse tree validation to prevent SQL injection attacks," 2005.

[23] L. Vokorokos, E. Chovancová, J. Radušovský, M. Chovanec, "A Multicore Architecture Focused on Accelerating Computer Vision Computations, " in Journal of Applied Sciences, vol. 10, iss. 5, Acta Polytechnica Hungarica, Hungary, 2013, pp. 29-43.

[24] D. Huang, T. Xing and H. Wu, "Mobile cloud computing service models: a user-centric approach," in IEEE Networks, vol. 27, iss. 5, September-October 2013, pp. 6-11.

Cloud computing security

Bordak, Lukas

01 Dhananjaya Wimalasekera Page 3

23/2/2021 19:31