Kosovo's websites vulnerabilities and its economic impact

Arbnor HALILI[1], Korab RRMOKU[2], Blerim REXHA[3]
[1, 2, 3] Faculty of Electrical and Computer Engineering, University of Prishtina
arbnor.halili@gmail.com, korab.rrmoku@gmail.com, blerim.rexha@uni-pr.edu

Abstract
We all are witnessing the exponential growth of website sophisticated attacks in different areas and from various sources. This paper presents a research on the vulnerabilities of different web sites that are operational today in Kosovo. It focuses in the process of testing the web sites for different attack possibilities and their weakness in respect to possible "cybernetics" attackers. An advanced software tool, called "Acunetix", is the main apparatus in this whole process of researching and fact gathering. It is worth mentioning that the whole "inspection" is made only for research purposes. First part of this paper presents definitions and approach used for this research. Second part contains the facts and results regarding vulnerabilities and weak points that are gathered. A conclusion about the vulnerabilities that are found on this research, accompanied with statistical records, is presented at the end of the paper.

Keywords: Acunetix, Cybernetic attack, Kosovo, SQL injection, Vulnerability, XSS

1. Introduction

The usage trend of the web applications today is one of the main and important tools in order to provide services on the Internet. These applications are used for tasks that are mission critical and usually for sensitive user data. However, a weak point in this issue remains the level of the developers that deals with these web applications, because very often the applications are developed from people with limited capabilities on web security. As a result the number of web vulnerabilities has risen significantly. This is emphasized on 2013 Internet Security Threat Report [1], where is stated that there was 30% increase in web vulnerabilities in 2012. Recent researches in web vulnerabilities have been focused on the reduction of input validation faults. This vulnerability class is characterized from the fact that an internet application uses an external input, as part of operation without checking its property. Concrete examples of input validations are SQL Injection [3] and Cross Site Scripting (XSS) [2], which will be discussed and analyzed on this paper. Using XSS, an application sends the output to the client without proper syntax checking, thus allowing attackers the possibility of injecting JavaScript code on the output, which after this is executed in client's browser. On the other hand, with SQL injection, attacker is able to insert a series of SQL statement into a query by manipulating data input into an application.

A. Our Approach

Witnessing everyday news and consequences of web site's attack using web vulnerabilities, we were motivated to conduct a research and a survey of the situation of these web vulnerabilities on Kosovo's web sites. In our approach we have decided to conduct the research by using and testing 30 web sites form different categories. As per our best knowledge, there is no similar research and work done earlier in Kosovo regarding this issue; therefore we present the work and results that are made in this paper, as unique and innovative.

The analyzed web sites are divided in three categories: First category is formed by 10 governmental and institutional web sites; Second category is formed from 10 academic and school web sites, and finally, third category contains 10 web sites from informative perspective.

Testing of these web sites is made using an advanced and contemporary web vulnerability software scanner tool, called Acunetix. Based on testing results we concluded that different types of web vulnerabilities were present on these web sites, hence we decided to organize the work and analysis in three perspectives.

On the first perspective we have grouped web site's based on overall and general vulnerabili-

ties, meaning that the statistics are gathered and analyzed without splitting them for each category. There are four levels of vulnerabilities that are counted for each website: high, medium, low and informational vulnerabilities.
On the second perspective we have grouped the websites now more specifically, starting from general and going to individual bases. Now the analysis focuses on the specific types of vulnerabilities. From our research we have found the following vulnerabilities as most common in all testing that we made. These vulnerabilities includes: (i) SQL (Blind) injection, (ii) Cross Site Scripting (XSS), (iii) PHP allow_url_fopen, and (iv) Security vulnerability on database passwords. There were some other (additional) vulnerabilities that were detected, however there were minor vulnerabilities with low probability to cause problems.
Finally, we have made a detailed analysis on specific groups, so we have emphasized most influential and most common vulnerabilities that were detected on certain category, and web site, respectively.
In this way we have gathered statistical results from this research, and in the same time, based on the vulnerabilities found, we give certain ideas for possible improvements.
Knowing that these web sites that we have tested, but also including many others that are used every day from society have very important impact, we have also presented the possible economic impact that malwares and vulnerabilities might cause, if these websites would be possibly attacked.
It is in our responsibility to emphasize that the results found from this research will not be publicly delivered and the findings in these tested websites will be used only for research and academic purposes.

2. Background on web vulnerabilities concepts

Hackers already have a wide repertoire of attacks that they regularly launch against organizations including SQL Injection, Cross Site Scripting, Directory Traversal Attacks, Parameter Manipulation (e.g., URL, Cookie, HTTP headers, web forms), Authentication Attacks, Directory Enumeration and other exploits. Moreover, the hacker community is very close-knit; newly discovered web application intrusions are posted on a number of forums and websites known only to members of that exclusive group [4] [5]. Acunetix, is the software tools used for this research.

Acunetix Web Vulnerability Scanner (WVS)

Acunetix Web Vulnerability Scanner (WVS) is an automated web application security testing tool that audits your web applications by checking for vulnerabilities like SQL Injections, Cross site scripting and other exploitable hacking vulnerabilities [4]. In general, Acunetix WVS scans any website or web application that is accessible via a web browser and uses the HTTP/HTTPS protocol.
Acunetix WVS is suitable for any small, medium sized and large organizations with intranets, extranets, and websites aimed at exchanging and/or delivering information with/to customers, vendors, employees and other stakeholders.

A. SQL Injection Attacks (SQLIAs)

SQL Injection Attacks (SQLIAs)-Structured Query Language (SQL) is an interpreted language used in database driven web applications which construct SQL statements that incorporate user-supplied data or text. If this is done in an unsafe manner, then the web application may be vulnerable to SQL Injection Attack [6] i.e. If user supplied data is not properly validated then user can modify or craft a malicious SQL statements and can execute arbitrary code on the target machine or modify the contents of database.
There are several consequences related directly to SQLIAs, which are grouped in 4 main points, being: *loss of confidentiality*, due to the fact that databases always holds private and critical information*; loss of integrity*, due to external unauthorized modifications; *weak au-*

thentication due to weak SQL queries and *loss of authorization*, where if attacked, attackers may change authorization information.
SQL injection, as described in [3] and [7], is considered to have more severe consequences than XSS, due to the fact that a successful SQL injection can compromise the integrity of a database. A Web application is vulnerable to SQL injection if invalidated user input is used to generate SQL queries.
A typical SQL query used to generate dynamic web pages is as follows:

```
SELECT * FROM articles
WHERE id='<user input>
```

An attacker can control the user input, and e.g. enter:

```
'; DROP ('articles');
```

This adds a second command to the SQL query, which then becomes:

```
SELECT * FROM articles WHERE id='';
DROP ('articles');';
```

These SQL commands will select some data, delete the table “articles” in the database, and then generate an SQL error due to the single quotation mark.
In general, SQL injection gives an attacker the opportunity to manipulate the database and in special cases execute arbitrary code on the database server [8]. It is therefore an effective attack on Web applications. Typical attack vectors are logins, search forms, and the URL of dynamically generated pages.

B. Cross Site Scripting (XSS)

Cross Site Scripting (XSS) is a vulnerability in web applications that allows an attacker to inject HTML, typically including JavaScript code, into a web page. XSS results from the intermingling of server code and user input [9]. If user input is not sanitized correctly, it could contain code that runs along with server code in a client’s browser.
In 2013, XSS was ranked the #3 web application security risk by the Open Web Application Security Project (OWASP) [10] and the #1 software errors by the SANS Institute [11].
There are two primary types of XSS vulnerabilities – *reflected* (non-persistent) and *stored* (persistent) [9]. In a reflected XSS attack, the attacker persuades a victim to click on a specially crafted link that makes a request to a vulnerable web server. This allows the attacker to run arbitrary code in the victim’s web browser. In reflected attacks, the attacker must target each victim individually.
In the following part, we will give an example of possible XSS attack in a fictive web site. Now imagine that you are browsing through auctions on a popular site; let’s call it auctions.example.com. You come across several auctions and would like to see more items that the same person has for sale; let’s assume this person is a “bad guy” and call him BG123. You click on BG123’s website and see a listing of his auctions. So when you click on the link to go to that auction, the webserver informs you for “not finding” that page, which in essence is an XSS attack. So, BG123 offered a link to a web page that looks something like that [12]:

```
<A HREF = http://auction.example.com/<script>alert(‘hello’)</script>">Click
Here</a>
```

The "FILENAME.html" submitted to auction.example.com was:

```
<script>alert(‘hello’)</script>
```

“auction.example.com” then used its ordinary routines to generate an error page to you that read,

```
<HTML>
404 page not found: <script>alert(‘hello’)</script>
....
</HTML>
```

In effect, BG123 managed to "inject" a JavaScript program into the page returned to you by auction.example.com [12].The JavaScript ran as though it originated at auction.example.com, and could therefore process events in that document. It also maintained communication with BG123 by virtue of scripting that BG123 put in the link; this is the way XSS vulnerability can be exploited to "sniff" sensitive data from within a web page, including passwords, credit card numbers, and any other arbitrary information you input.

3. Statistical analysis

Testing has been divided into three main groups, each group containing 10 testing websites. Every site is categorized about its potential security weakness as high, medium and low. Based on this criterion, from total of 30 sites we have 16 sites categorized with high potential of security weakness, 13 with medium and 1 with low, as presented in Figure 1.

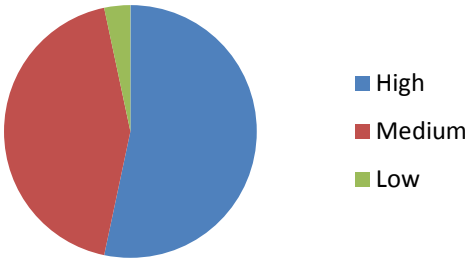

Figure. 1. Security weakness of tested sites

Regarding this type of evaluation, second group is the worst one with 6 sites classified with high potential of vulnerability. Table 1 presents vulnerability distribution among groups.

Table 1. Vulnerability distribution among groups

	High	**Medium**	**Low**
Group 1	5	5	0
Group 2	6	3	1
Group3	5	5	0

This was the general view of main vulnerability categories. There are 2039 vulnerabilities found in all web sites. The leading category is the second category (medium) with 1167 vulnerabilities, followed by first category (high) with 446 and third category (low) with 426 vulnerabilities.

The first group – governmental and institutional web sites, has the smallest number of vulnerabilities. Based on our conducted research there are found 370 possible vulnerabilities with a distribution as presented in Figure 1.

The number of potential vulnerabilities differs a little bit when we talk about third group of sites – informative sites. Here we have found 491 as the number of potential site vulnerabilities. The main characteristic here is that we have a trend of increasing percent of vulnerabilities categorized with high risk.

Second group – academic and school sites were shown to be most preferable for security attacks. Our research has found 1178 possible potential security vulnerabilities. What can be comforting here has to do with percentage of distribution of these vulnerabilities. There is a significant difference for medium vulnerabilities. This difference is reflected in increase of low potential vulnerabilities which is less of a concern regarding other categories. Figure below depicts distribution of founded vulnerabilities regarding each testing groups; first chart present distribution of the vulnerabilities on governmental sites, second chart on academic and educational sites and third chart on informative sites.

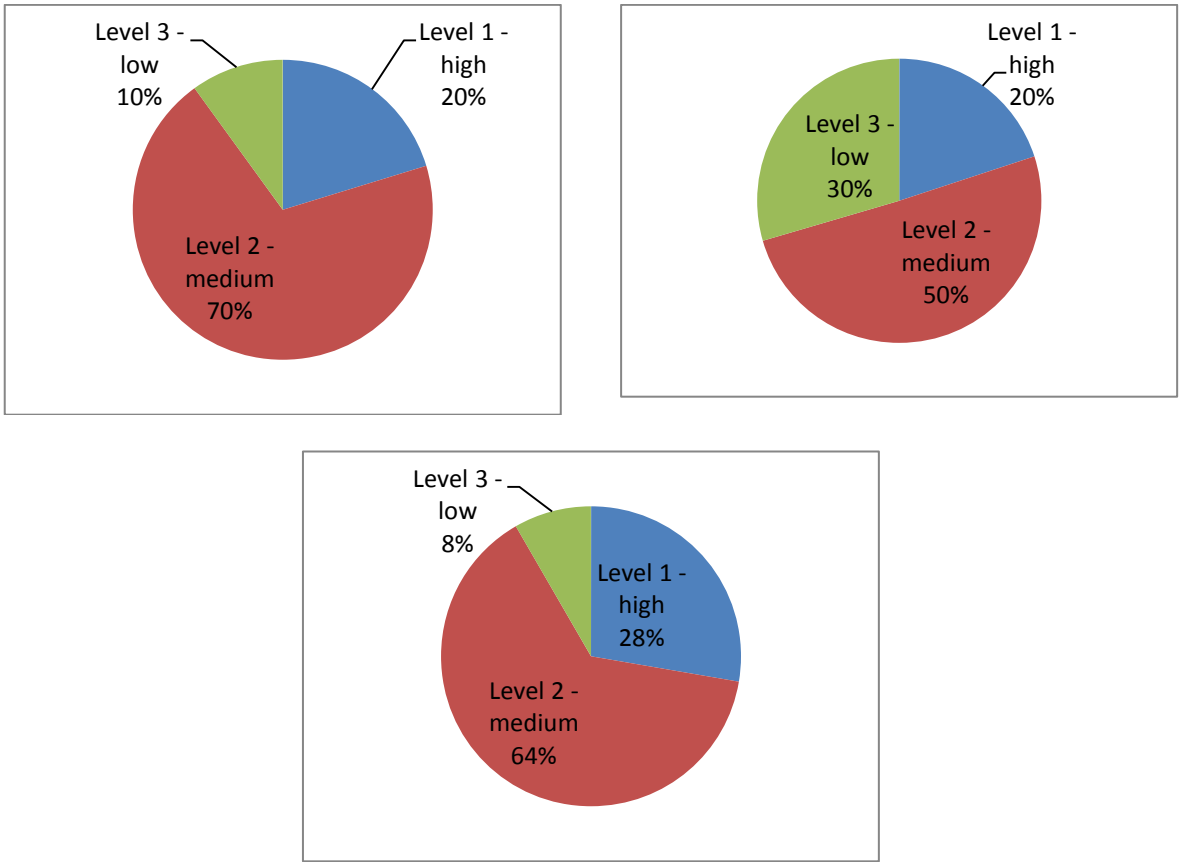

Figure. 2. Vulnerabilities distribution on testing groups: governmental, academic and informative web sites

A. Group 1 details

Basically this group has been resulted to have less vulnerability in comparison with two other groups. Most of them are categorized as XSS attacks and SQL Injection attacks. There are also some other specific vulnerabilities that are detected in most of these sites. One of them has description "HTML form without CSRF protection ". CSRF stands for cross-site request forgery. Typically, CSRF will be used to perform actions of the attacker's choosing using the victim's authenticated session. If a victim has logged into the target site, an attacker can coerce the victim's browser to perform actions on the target website [17].

Another interesting vulnerability is described as "Apache httpd Remote Denial of Service". This is a kind of Denial of Service (DOS) attack. The attack can be done remotely and with a modest number of requests can cause very significant memory and CPU usage on the server [18].

A concerning vulnerability is about passing credentials as clear text. Having in mind that government has its local network and that has a high number of users in same subnet, this can be e potential security breach that can be easily abused. When we talk about user credentials and its potential danger by sending as a clear text, an automatic association goes to the SSL protocol. There is also e possible vulnerability regarding this issue that has been detected. Some sites use SSL 2.0 that is a deprecated protocol. Security problems of this protocol are already known and usage of such protocol can cause possible problems with security.

B. Group 2 details

Group 2 that has in main focus academic and schools sites has been shown to have the highest number of vulnerabilities as it is presented in section above. The two main problems also

here are SQL injection and XSS. SQL injection here comes in variations from blind SQL injection to other types. A characteristic that can be here easily detected is that most of sites have either one or other vulnerability.
Interesting is the fact that in these sites has been found a lot of vulnerabilities categorized as "Application error message". At first sight this seems not to be a dangerous problem or weakness. In developer's world, it is known that application error messages can sometimes be fatal for their security. An error message can describe skeleton of some data or part of a certain class than can result in good information for attacker.
The problem with CSRF protection is present also here and is detected in similar form as it is described above in group 1 details. Another common problem of these two groups is the problem with user credentials. We have found academic sites that have this problem and that is a concerning problem because these sites provides login for students and theirs user credentials can easily be hijacked. There are also a small number of potential weaknesses that can come from usage of depreciated SSL 2.0 protocol.

C. Group 3 details

The main characteristic of vulnerabilities categorizes as high on this group is that they belong just to XSS and SQL injection vulnerabilities. A specific vulnerability that belongs to the group of Denial of Service (DOS) Vulnerabilities, and that is detected in one of these sites is PHP Hash Collision Denial of Service Vulnerability.
Other vulnerabilities presented on sites of this group are similar to vulnerabilities described above for first two groups. They differ only in number they are present on these sites. It must be mentioned that the most presented vulnerability is CSRF protection.

4. Detection and prevention of web vulnerabilities

As we have seen from analysis, two most common web vulnerabilities today are SQL Injection and XSS, hence these two will be elaborated with regard to detection and prevention.
There are many techniques that are used and considered today in order to prevent SQL injection attacks, however according to [6] the main prevention techniques include: content filtering, penetration testing, and defensive coding. However, recently many research works has been done in this direction, therefore now there are several mechanisms that are used in order to prevent and defend against SQL injection vulnerabilities.
In [13], is proposed an authentication scheme using algorithm which uses both Advance Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) to prevent SQL injection attacks. In this method a unique secret key is fixed or assigned for every client or user. On the server side server uses private key and public key combination for RSA encryption.
Another prevention mechanism is proposed in [14], where is suggested a technique for finding vulnerabilities in Web Application such as SQL injection attack by network recording. In this approach network forensic techniques and tools are used to analyze the network packets containing *get* and *post* requests of a web application. This approach uses network based Intrusion Detection System (IDS) to trigger network recording of suspected application attacks.
One of the most important defending techniques for SQL injection is "SQL Server Lockdown" [3]. What is important here is that it is necessary to 'lock down' SQL server, since it is not secure 'out of the box'. The list of things to do when creating a SQL Server build includes: determining the methods of connection to the server, verifying which accounts exist (privileged users), verifying which objects exist, verifying which accounts can access which objects, verifying the patch level of the server, verifying what will be logged, and what will be done with the logs.
Detecting and preventing XSS attacks is a wider field, since XSS it is not defined with a definition and attacks might come from more and different sources. According to NSA there are

three different plans that we might focus in order to detect and prevent XSS attacks: user aspect, developer aspect and network administrator aspect [9]. Users should consider the following measures: restrict untrusted JavaScript, use built-in browser protections, restrict external websites from requesting internal resources, and maintain good system hygiene. For developers on the other side it is important to ensure that they understand the dangers of XSS attacks and have tools that allow them to create secure web applications without hindering their productivity. There are tools that help developers create secure web applications that include: Blacklisting vs. Whitelisting, Enterprise Security API (ESAPI), Microsoft AntiXSS Library and web vulnerability scanners. Lastly, a significant work is also needed from network administrators, who have to make sure that the internal network is secure and ready to prevent attacks. Two main techniques that help in this direction are: Web Proxy Plug-in that accepts scripts only from explicitly trusted domains and Web Application Firewalls (WAFs) a detection/prevention technology that specifically looks at and understands Hyper Text Transfer Protocol (HTTP) traffic.

Another defending approach with regard to XSS is called "Protecting Cookies from Cross Site Script Attacks Using Dynamic Cookies Rewriting Technique" [15]. This approach aims to change the cookies in such a way that they will become useless for XSS attacks. This technique is implemented in a web proxy where it will automatically replace the cookies with the randomized value before sending the cookie to the browser.

Input validation is one of the main mechanisms that are required to be checked in order to prevent attacks. Input can be validated on two different levels: either by the Web application or the Web server. The Web application can implement a function that parses all user input, handling dangerous characters/commands or rejecting the input. At the Web server level a module can be used to validate all input before it is handled over to the Web application [8].

In order to defend successfully a web site from SQL Injection and XSS attacks, a persistent focus and vigilance is needed, due to the fact that every day there is a risk of a new vulnerability to be deployed.

5. Economic impact of possible cybernetics attacks

Even though the empirical results on economic impact that comes from various sources on cybernetic attacks it is hard to be gathered and concluded, there are every day temptations on raising the awareness and preparing the environment for a better defense. Due to these attacks, institutions and businesses might suffer incredible financial loss and loss of credibility. Research results on [16] suggested that information security breaches have minimal long term economic impact. One possible explanation, according to conclusions in the mentioned research, is that the breached firms respond to the breach incident by making additional security investment to prevent from any future breaches. Another explanation is that as the time passes, people forget about what happened earlier and the impact of the breach on financial performance phases out over the long-term. To relate these recommendations to our work, we can emphasize that a possible attack in a financial department from our first testing instance (governmental websites) can cause a state destabilization on micro financial system. In the same way, an attack on academic or informative web sites might lead to credibility loss which in turns influences on the current and future financial situation, including predictions for economic growth.

Having the exact number of vulnerabilities from our research, we can calculate that how much would cost in economic aspect only to fix (deal) with these vulnerabilities. According to a research from WhiteHat Security [19] it is estimated that each SQL Injection or/and any other vulnerability will require approximately 40 developer hours. Considering the situation in Kosovo according to [20] the average wage per hour for ICT engineers is between 3 – 5 € and is usually higher than other sectors [21], from a simple calculation we get that for 2039

possible vulnerabilities (found), there is a need of 81560 working hours only to fix these vulnerabilities, which calculated with 5€/hour, we have an overall of 407800 € spent only on dealing with consequences. In the following table we have shown on details how much these vulnerabilities would cost on certain (tested) instances in our research.

Table 2. Calculations of working hours required to fix vulnerabilities based on each group

Instance/Level of vulnerability	High	Medium	Low	Total vuln.	Hours required per	Wages
Govt. web sites	75	258	37	370	14800	74000€
Academic web sites	235	595	348	1178	47120	235600€
Informative web sites	136	314	41	491	19640	98200€
Total					**81560 hrs.**	**407800€**

From our findings in this research, we propose an extended research work in longer testing periods for economic subjects, therefore we can gather concrete data, and hence more detailed results can be conducted.

6. Conclusions

This work introduces an innovative idea of testing web site vulnerabilities in Kosovo's websites and that in three different target groups. The results gained from research, has identified weak points and potential web sites to be attacked, and unfortunately academic web sites are most prone to suffer these consequences, and in the same time this work can be used as guideline in web defending aspect. Research was not limited only on testing but also on providing solution to these potential vulnerabilities and their economic impact. To the best of our (authors) knowledge, there is not any evidence that such an approach has been ever used, until the date of this paper submission.

To summarize, in the following part are listed the main contributions of this work:

- A novel approach is designed and used in order to identify main web vulnerabilities in three different target groups of web sites, being most frequently used in Kosovo.
- There are concrete results and findings, accompanied with statistical data and visualization that we have gathered from the situation in terrain, which can be used to identify main threat possibilities.
- We have defined and elaborated recommendations related to defending web sites from these potential threats, and we have identified possible economic impacts from these cybernetics attacks.

As future work we believe that a wide research in terms of time and number of sites should be made. A wide research in terms of time means that testing should be done multiple times for a long period than the period presented in this paper; a wide research in terms of number of sites means that a higher number of sites must be considered. This can give a better and more reliable result especially on economic impact.

References

[1] Symantec Global Internet Security Threat Report. Tech. rep., Symantec, April 2013. Volume 18
[2] Amit Klein. Sanctum Security Group, 2002 - Cross Site Scripting Explained
[3] Chris Anley. Advanced SQL Injection In SQL Server Applications Tech. rep., Next Generation Security Software, Ltd, 2002
[4] Acunetix Web Vulnerability Scanner - Web Vulnerability Scanner v8, Acunetix WVS 2004–2012.
[5] E-Spin Professional Book, Web Application Security

[6] Rahul Johari, Pankaj Sharma. A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection
[7] Huseby S. H. Innocent Code: a security wake-up call for Web programmers, Wiley, 2004.
[8] Vebjørn Moen, Andr´ E N. Klingsheim, Kent Inge Fagerland Simonsen, And Kjell Jørgen Hole. Vulnerabilities In E-Governments
[9] The Information Assurance Mission at National Security Agency (NSA), Protect Against Cross Site Scripting (XSS) Attacks, September 2011
[10] Open Web Application Security Project (OWASP) Report, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project (website visited on May, 2013)
[11] Cwe/Sans Top 25 Most Dangerous Software Errors - http://www.sans.org/top25-software-errors/2010/ (website visited on May, 2013)
[12] Jason Rafail, Cert® Coordination Center, Cross-Site Scripting Vulnerabilities, Carnegie Mellon University, 2001
[13] Indrani Balasundaram, E.Ramaraj. "An Authentication Scheme for Preventing SQL Injection Attack Using Hybrid Encryption".
[14] Pomeroy, A Qing Tan. "Effective SQL Injection Attack Reconstruction Using Network Recording" in Computer and Information Technology (CIT).
[15] Rattipong Putthacharoen, Pratheep Bunyatnoparat. "Protecting Cookies from Cross Site Script Attacks Using Dynamic Cookies Rewriting Technique" Feb. 13~16, 2011 ICACT2011
[16] Myung Ko, Carlos Dorantes. "The impact of information security breaches on Financial performance of the breached firms: an empirical investigation" on Journal of Information Technology Management , 2006
[17] Jeremiah Blatz. CSRF: Attack and Defense, McAfee® Foundstone® Professional Services, White Paper
[18] Acunetix – Web Application Security, http://www.acunetix.com/vulnerabilities/apache-httpd-remote-denia/ (website visited on May, 2013)
[19] White Hat Security. The Cost of Fixing Vulnerabilities vs. Antivirus Software, https://blog.whitehatsec.com/the-cost-of-fixing-vulnerabilities-vs-antivirus-software/#.UaDWH7XLrsY (website visited on May, 2013)
[20] Ardit Bejko, Kushtrim Shaipi. ICT in Kosovo – A Sector Decoded, Published by: USAID, 2010
[21] Agjencia e Statistikave të Kosovës, Published by Govt. of Kosovo, Statistikat Ekonomike 2008-2012

Arbnor HALILI has graduated the Faculty of Electrical and Computer Engineering in 2010. He is an MSc candidate and holds a BSc diploma in Computer Engineering. He is the author of several mobile applications and his research interest is closely connected with mobile services and software quality management. His work focuses on the analysis of quality of software applications.

Korab RRMOKU holds a Bachelor of Computer Engineering degree from Faculty of Electrical and Computer Engineering, University of Prishtina. He is currently a Master of Science candidate in Computer Engineering, and working on master dissertation with theme: "Social Network Analysis (SNA) and touristic tour planning". He is focused and interested on research work, and is a co-author of the paper (article) that is related to his master dissertation. Currently he works as a Network and Design Engineer, offering services for world-wide clients. His interest remains in SNA, network and software systems security.

Blerim REXHA has graduated at the Faculty of Electrical and Computer Engineering, in University of Prishtina in 1994. He holds a PhD in Computer Engineering from Vienna University of Technology, Vienna, Austria earned in 2004. He used to work for many years for Siemens AG in Vienna, where he managed several software projects. Since 2007 he works at University of Prishtina. Currently he is associate professor at the Faculty of Electrical and Computer Engineering and he is the head of Computer Engineering department. He is author of more than 35 journal and proceedings in the area of data security, software engineering and energy. His works focuses in data security and privacy, network communication and software engineering.