INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

NEXT GEN END TO END WEB SECURITY – E-TRACKING SYSTEM

RAJESH M. LOMTE[1], PROF. S. A. BHURA[2]

1. ME (Computer Engg.) SEM-VI, Department of Computer Engineering, BNCOE, Pusad, Amravati University.
2. Assistant professor, Department of Computer Engineering BNCOE, Pusad, Amravati University.

Accepted Date: 15/02/2014 ; Published Date: 01/04/2014

Abstract: Web sites are prone to security risks. A website with weak security opens your network to attack and makes data loss more likely. Hence web security becomes promising task for software builders. Our main motto is a rich web application which provides concrete security for web applications. In this paper we have applied best security protection against very popular web application attacks like SQLI, XSS, DOS and Request Encoding. We have considered to E- applications Online shopping without security & Online shopping with security. With security application will contains security appliances & without security simple one no security solution. Our main aim is to analyse impact of both the applications in the form of Request /Loading (Lt)& Response time (Rt.) which provide positivity towards time factor – negligible difference in the both application of Lt. & Rt.

Keywords: DOS- Denial of Services , XSS- Cross Site Scripting, SQLI- Structured Query Language Injection, Lt- Loading Time , Rt- Response Time

PAPER-QR CODE

Corresponding Author: MR. RAJESH M. LOMTE

Access Online On:

www.ijpret.com

How to Cite This Article:

Rajesh Lomte, IJPRET, 2014; Volume 2 (8): 792-806

INTRODUCTION

Now a day's web security is biggest issue in the corporate world. The world is highly dependent on the Internet .It is considered as main infrastructure of the global information society. Therefore, the availability of Internet is very critical for the socio-economic growth of the society. The "availability" of Internet and its services means that the information, the computing systems, and the security controls are all accessible and operable in committed state at some random point of time However, the inherent vulnerabilities of the Internet architecture provide opportunities for a lot of attacks on its infrastructure and services.[1] XSS , SQL injection, Sniffing, Request Encoding and DOS attacks which poses an immense threat to the availability of the Internet. An occurrence of these attacks on the web degrades or completely disrupt services to legitimate users by expending communication and/or computational resources of the target. Nowadays to achieve security of distributed systems is a dominant task for any organization including the most modest types of e-commerce, banks and even large state systems However, the in3creasing number and a variety of system attacks suggest, between among other things, that the design and realization of these systems are often very poor as far as security is concerned. Web security is essential part of business world. [2] Dos Attack is responsible for attackers direct hundreds or even thousands of compromised hosts called zombies against a single target. XSS attack is responsible for the attacker executes malicious code on the victim's machine by exploiting inadequate validation of data flowing to statements that output HTML. SQL Injection Attack is responsible for the attacker executes malicious database statements by exploiting inadequate validation of data flowing from the user to the database. Sniffing (Request Encoding) attack is responsible for data hacking during data transmission. Previous approaches to identifying these kinds of attacks and preventing them includes defensive coding, static analysis, dynamic monitoring, and test generation. These techniques have their own merits but have some drawback like Defensive coding [6] is error-prone and requires rewriting existing software to use safe libraries. Static analysis tools [13] can produce false warnings and do not create concrete examples of inputs that exploit the vulnerabilities. Because of these attacks Vulnerabilities business market will get hampered and it is headache to the E- business system.[6][15]

This paper will provide the best web security solution for web application attacks and graphical representation of experimental results (comparison chart).

RELATED WORK

Most of the traditional works on network intrusion detection focus on misuse-based or anomaly-based recognition of attack signatures. However, traffic generated from an attack to a web application — except for brute force attacks or similar events — is likely to be very similar to normal traffic because, since HTTP is a text based protocol, it is always possible to encapsulate an attack at application layer without Creating a packet that is anomalous if inspected at network layer. Writing generic network-layer signatures for web-based attacks are thus troublesome, and a source of false positives. On the other hand, host-based IDSs were typically designed to monitor the processes on the protected system (e.g. the web server daemon) rather than the web applications they run. A successful example of protocol-aware anomaly detection based on low level data is presented in[3]. However, nowadays' XSS attacks can perform more sophisticated tasks. Examples are the many attacks against social networking websites, which perform queries on the site without actually moving sensitive information around. The idea of a client side proxy was further exploited in [4]. It must be noted that all these techniques are client-side protections and, as such, they assume user awareness to security. More specialized works have targeted separately XSS and SQL injection attacks. In [5] a client-side proxy was used to detect harmful content (e.g., DOM nodes) supplied by the user and sent back by the server, using a technique similar to the one implemented in web vulnerability scanners [6]. This technology, however, works only on reflected XSS attacks, and not on persistent attacks where the injected malicious code is permanently stored on the server-side and is delivered to the browser at a later time. We are going to provide the best solution to protect the web from various web attacks.[13][14]

III. SURVEY OF DIFFERENT SECURITY THREAT:

1. 2012 Cost of Cyber Crime Study: UK

Cyber crimes are costly. The Ponemon Institute found that the average annualised cost of cyber crime for the 38 organisations is £2.1 million a year, with a range of £0.4 million to £7.7 million.

2. Cyber criminals target Skype, Facebook and Windows users

Cyber criminals targeted users of Skype, Facebook and Windows using multiple Blackhole exploits in October, according to the latest threat report from security firm GFI Software.

3. Zero-day exploit for Yahoo Mail goes on sale

In November, a hacker was offering a zero-day exploit for Yahoo Mail for $700 that would enable an attacker use a cross-site scripting (XSS) vulnerability to steal cookies and hijack accounts.

The hacker, known as "TheHell", created a video to market the exploit on an underground cyber crime market called Darkode.

4. Six arrested in the UK in worldwide FBI-led credit card data sting

Law enforcement officers arrested six people in the UK and 12 in the US in an FBI-led sting operation in June that netted a total of 24 credit card cyber fraudsters in 13 countries.

The arrests followed a two-year undercover FBI investigation that tracked those buying and selling credit card information through a fake online forum

5. XSS attacks remain top threat to web applications

Cross-site scripting (XSS) attacks remain the top threat to web applications, databases and websites, an analysis of 15 million cyber attacks in the third quarter of 2012 revealed. Other top attack techniques are directory traversals, SQL injections (SQLi), and cross-site request forgery (CSRF), according to a web application attack report by cloud hosting firm FireHost.

6. AT&T takes APTs seriously -

Advanced persistent threats (APTs) are real and all companies should be taking them seriously, says telecommunications company AT&T.

In the past year, the company has set up an IT security team dedicated to researching APTs and making recommendations on how to defend against them.

7. Yahoo user sues over personal data breach

A Yahoo user sued the web portal company for negligence in August for allowing more than 450,000 user names and passwords to be stolen from one of its sites.Jeff Allan of New Hampshire, whose login credentials were posted online after a hacker infiltrated a company database on 11 July, filed a complaint in a federal court in California.

8. Nasa to encrypt data after latest breach

US space agency Nasa is to encrypt all its mobile computers after the loss of a laptop containing personal information about more than 10,000 employees and contractors.

This is the latest in a series of data breaches involving unencrypted laptops at Nasa in recent years and comes just eight months after the theft of a laptop containing personal information of 2,300 employees and students.

9. Cyber attackers increasingly targeting applications, research shows

Web and mobile applications are the new frontiers in the war against cyber attack, according to a top cyber security risks report from Hewlett Packard (HP) published in May.

The report reveals that SQL injection (SQLi) attacks on web applications increased sharply from around 15 million in 2010 to more than 50 million in 2011.

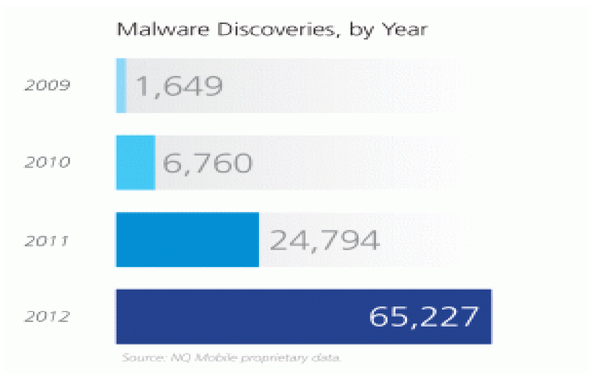

Fig.1 Year wise malware discovery

IV IMPACT OF WEB APPLICATION ATTACKS ON FINANCE:

The Financial Impact of Web Application Threats

Web attacks are the single most dangerous threat facing organizations today. Web attacks are prevalent – striking Websites once every two minutes on average and they inflict enormous damage, bringing down critical applications and causing brand damage, fines, breach notification costs, and customer turnover. To determine the ROSI provided by the SecureSphere Web Application Firewall (WAF),.

1. The Financial Impact of a Web Application Breach

Data breaches are costly, averaging $7.2 million per incident.However, some breaches have proven to be extremely expensive, with one organization alone expected to spend $1 billion to resolve a massive Web application breach in 2011. Web application attacks are one of the most common causes for a data breach. In fact, 89% of all records stolen in data breaches were due to hacking and external threats, and Web-based attacks like SQL injection, XSS, and brute force are hackers' tools of choice to steal data.

2. The Cost of Application DDoS Attacks

In addition to data breaches, Websites are also a target for DDoS attacks. In fact, 74% of businesses reported receiving a DDoS attack in the past year, according to a recent reportand approximately 25% of these attacks are application DDoS attacks. Like data breaches, DDoS attacks are expensive. According to a survey of companies, a successful DDoS attack costs, on average, $1.427 million.

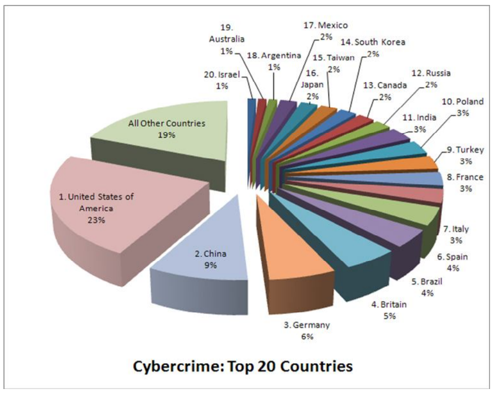

Fig.2 Cyber crime for top ten countries

3. Traditional Network Security: A False Choice for Web Application Security

Almost all enterprises have deployed network firewalls to protect their network infrastructure and their users; most have also provisioned an intrusion prevention system (IPS) or a next generation firewall to detect intrusions and to control user access to applications. While these products may include a handful of Web attack signatures, they do not learn Web application structure or usage and they cannot effectively stop Web attacks.

In fact, a recent report indicates that IPS products configured with their default security policies stop about 15 - 25% of basic Web attacks.

Besides Web attacks like SQL injection and XSS, network security solutions cannot detect or stop:

» Session-based threats like session hijacking and cookie poisoning

» Business logic attacks like site scraping and comment spam

» Web-based fraud

Furthermore, most cannot inspect SSL-encrypted Web traffic.

V PREVENTIVE SECURITY MEASURES:

1. Denial of Services (DOS) Attack :

The motivation for DoS attacks is not to break into a system but to make the target system deny the legitimate user giving service.

Preventive Measures:

1. To handle this attack, each individual request is tracked on custom DB by using **DOSAttack module**.
2. Application determines the threshold value for the time between consecutive requests.
3. Application verifies the timing between the two requests.
4. If the next request is coming from same client within the time less than threshold value then system picks up the Ip address of that client and stores that Ip address into database in table 'blocked IP'.

```
String ip =HttpContext.Current.Request.UserHostAddress;
        _Banned =OnlineShopping.modCommon.GetBlockedIpList();
        if (_Banned.Contains(ip))
        {    HttpContext.Current.Response.StatusCode = 403;
 HttpContext.Current.Response.Write("ERROR:        Access Denied"); }
```

2. SQL Injection Attack:

SQL injection is a code injection technique, used to attack data driven applications, in which malicious SQL statements are inserted into an entry field for execution

We use sql parameters with stored procedures or dynamically constructed SQL command strings. Parameter collections such as

SqlParameterCollection provide type checking and length validation

SqlParameter userIdParam = new SqlParameter("UserId", SqlDbType.NVarChar,50); **Cross site Scripting Attack:**

XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

If an attacker manages to put a script on your page, the victim will not be affected because the browser will not execute the script if it is properly escaped .To implement this we have used the **HttpUtility.HtmlEncode** method to encode output if it contains input from the user or from other sources such as databases. **HtmlEncode** replaces characters that have special meaning in HTML-to-HTML variables that represent those characters. Hence the script from the attacker is encoded and treated as a data instead of the script. So the script from the attacker is not executed on the browser and the user is not affected by the XSS attack.

sql = "INSERT INTO USERDETAILS (USERID, FNAME, LNAME, ADD1, ADD2, CITY, STATE, ZIP, PHONE, EMAIL) VALUES('" + HttpUtility.HtmlEncode(this.UserId) + "','" +

HttpUtility.HtmlEncode(this.FirstName) + "','" + HttpUtility.HtmlEncode(this.LastName) + "','" + "');";

cmd = new SqlCommand(sql, Common.dbConnection);

cmd.ExecuteNonQuery();

3. Cross site Scripting Attack:

XSS enables attackers to inject client-side script into Web pages viewed by other users. A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the same origin policy.

If an attacker manages to put a script on your page, the victim will not be affected because the browser will not execute the script if it is properly escaped .To implement this we have used the **HttpUtility.HtmlEncode** method to encode output if it contains input from the user or from other sources such as databases. **HtmlEncode** replaces characters that have special meaning in HTML-to-HTML variables that represent those characters. Hence the script from the attacker is encoded and treated as a data instead of the script. So the script from the attacker is not executed on the browser and the user is not affected by the XSS attack.

sql = "INSERT INTO USERDETAILS (USERID, FNAME, LNAME, ADD1, ADD2, CITY, STATE, ZIP, PHONE, EMAIL) VALUES('" + HttpUtility.HtmlEncode(this.UserId) + "','" +

HttpUtility.HtmlEncode(this.FirstName) + "','" + HttpUtility.HtmlEncode(this.LastName) + "','" + "');";

cmd = new SqlCommand(sql, Common.dbConnection);

cmd.ExecuteNonQuery();

4. Request Encoding Attack :

In this type of attack, the attacker tries to decode the request which is traversed between client and server. After decoding the request he may track the sensitive data from the application. To prevent this attack, we compress and encode the request before sending to the server by using the compression module. When the request is compressed, it can't be decoded by the hacker in between.

app.Response.Filter = new GZipStream(app.Response.Filter, CompressionMode.Compress);

SetEncoding(GZIP);

private void SetEncoding(string encoding) { HttpContext.Current.Response.AppendHeader("Content-encoding", encoding); }

VI EXPERIMENTAL RESULTS

Performance evaluation of Secured & Unsecured Application. Performance can be evaluated in terms of Request/ Loading & Response Time (Lt. & Rt.)

We are considering simple application with no security means that this application named main don't have any kind of security handling mechanism it done handle any web attack among these four. We are calculating Lt & Rl for this. Similarly we are taking into consideration the application with security which handling DOS, SQLI, XSS and Request Encoding attack. We are also calculating Lt & Rt for this one. Our main task is making comparison of these two application in the terms of Lt. & Rt. And tries to prove that inbuilt security our proposed method is very useful which is having negligible difference in between loading time & response time of both secured & unsecured web application means that if we use any external security solution for securing web application we need large time for loading & responding web requests/data but with security web application (proposed method) will reduce this drawback of existing one. It protect the web application as well it will reduce the loading & response time of the same.

You will get clear idea from following graphical representation.

Loading/request time & response time for modules –

Unsecured web application – Main- Register, Product List, View Card

Secured web application (Proposed E-tracking) - Register, Product List

Web Pgs	**Register**	**Product List**
SQLI Attack		
Comparison in	With Security	Without Security
Loading Time	55 ms	49.8001 ms
	182 ms	110 ms
Response Time	212 ms	64.4002 ms
	128 ms	10 ms
DOS Attack		
Loading Time	67 ms	54.8001 ms
	220 ms	122 ms
Response Time	128 ms	100.001ms
	232 ms	78.0003ms

XSS Attack		
Loading Time	89 ms	98.0202 ms
	199.232ms	265.002 ms
Response Time	1089 ms	40.0001ms
	231 ms	78.0003ms
RE Attack		
Loading Time	16	221.22 ms
	6 ms	
	220.36 ms	352.28 ms
Response Time	1089 ms	40.0001ms
	231 ms	78.0003ms

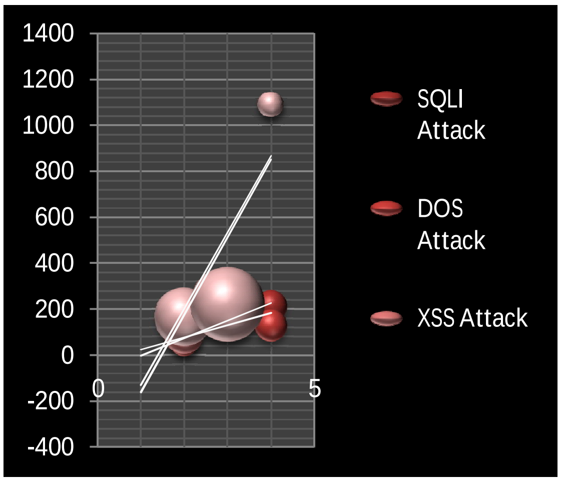

Fig.3 Graphical representation of Experimental results

IV. CONCLUSION & FUTURE SCOPE

CONCLUSION -

The proposed solution will helpful for building rich & secured web application. We can protect the E-business world by using proposed solution. Proposed method is also gives best designing/modeling practices. The heart of the E-tracking system is protection against different web application attacks like DOS, SQL Injection, XSS and Request encoding. By using proposed mitigations for all said attacks we can make our web application secured & efficient which definitely saves our business world. Similarly we are analyzing the impaction of web attacks on secured & unsecured web application in the terms of request time, response time. With the help of results we can say we can that there is no large difference in between execution time in between both application but when we use external security for our web application like IDS etc it will take time to take for execution (Request/loading / response time) so we can apply security measures while designing and coding the web application it is very useful for protecting our web application at initial level hence there is no need to use external security measures. It will reduce the cost because security is maintained by itself.

FUTURE SCOPE -

This can be enhance by providing inbuilt security at primary level for large scale application which contains payment issues similarly by extending it we can secure our web application by protecting it from other web attacks. We can also provide protection against Blind SQL attack by moving in the depth of the same.

We can extend our application by providing web security for web services like share market and we can also provide security as a service for could world.

REFERANCES

1. Monika Sachdeva, Krishan Kumar Gurvinder Singh Kuldip Singh SBS College of Engg. & Technology, Guru Nanak Dev University Indian Institute of Technology Ferozepur, Punjab, India Amritsar, Punjab, India Roorkee, Uttarakhand, Indiamonika.sal(kediffmail.com gzsbawa7 1(yahoo.omkds56fec(&riitr.ernetmin) Performance Analysis of Web Service under DDoS Attacks 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009

2. Diallo Abdoulaye Kindy1,2 and Al-Sakib Khan Pathan2, A Detailed Survey on Various Aspects of SQL Injection in Web Applications: Vulnerabilities, Innovative Attacks, and Remedies, 1CustomWare, Kuala Lumpur, Malaysia 2Department of Computer Science, International Islamic University Malaysia, Kuala Lumpur, Malaysia diallo14@gmail.com and sakib@iium.edu.my , 2012

3. DDoS Attacks in the United Kingdom: 2012 Annual Trends and Impact Survey

4. Joaquin Garcia-Alfaro1 and Guillermo Navarro-Arribas2, Prevention of Cross-Site Scripting Attacks on Current Web Applications_,1 Universitat Oberta de Catalunya,Rambla Poble Nou 156, 08018 Barcelona - Spain, joaquin.garcia-alfaro@acm.org 2 Universitat Autònoma de Barcelona, Edifici Q, Campus de Bellaterra, 08193, Bellaterra - Spain, gnavarro@deic.uab.es

5. William G.J. Halfond, Jeremy Viegas, and Alessandro Orso College of Computing Georgia Institute of Technology {whalfond|jeremyv|orso}@cc.gatech.edu, A Classification of SQL Injection Attacks and Countermeasures, College of Computing

6. Mark Curphey The Open Web Application Security Project David Endler iDefense William Hau Steve Taylor Predictive Solutions Tim Smith The Open Web Application Security Project Alex Russell OWASP Filters project Secure Pipe Inc. netWindows.org Gene McKenna Richard Parke Kevin McLaughlin," A Guide to Building Secure Web Applications The Open Web Application Security Project"

7. Security Threat Report 2013-New Platforms and Changing Threats,SOPHOS

8. Uzi Ben-Artzi Landsmann and Donald Str¨omberg, Web Application Security: A Survey of Prevention Techniques Against SQL Injection, Department of Computer and Systems Sciences Stockholm University / Royal Institute of Technology

9. Sonam Panda, Ramani S," Protection of Web Application against Sql Injection Attacks", International Journal of Modern Engineering Research (IJMER)www.ijmer.com Vol.3, Issue.1, Jan-Feb. 2013 pp-166-168 ISSN: 2249-6645

10. Mihir Gandhi, JwalantBaria," SQL INJECTION Attacks in Web Application", International Journal of Soft Computing and Engineering (IJSCE)ISSN: 2231-2307, Volume-2, Issue-6, January 2013

11. Asha. N M. Varun Kumar Vaidhyanathan. G, Preventing SQL Injection Attacks nternational Journal of Computer Applications 2012 by IJCA JournalVolume 52 - Number 13 Year of Publication: 2012

12. Zhang Chao-yang," DOS Attack Analysis and Study of New Measures to Prevent", Intelligence Science and Information Engineering (ISIE), 2011 International Conference on Date of Conference: 20-21 Aug. 2011

13. Adam Kie˙zun MIT akiezun@csail.mit.eduPhilip J. Guo Stanford University pg@cs.stanford.edu Karthick Jayaraman Syracuse University kjayaram@syr.edu Michael D. Ernst University of Washington mernst@cs.washington.edu

14. Y. Song, S. J. Stolfo, and A. D. Keromytis, "Spectrogram: A mixtureof-markov-chains model for anomaly detection in web traffic," in Proc.of the 16th Annual Network & Distributed System Security Symposium,San Diego, CA, USA, February 2009.

15. C. Criscione, G. Salvaneschi, F. Maggi, S. Zanero Dipartimento di Elettronica e Informazione — Politecnico di Milano 2009 European Conference on Computer Network Defense Integrated Detection of Attacks Against Browsers,Web Applications and Databases.

16. Vipul Patel, Radhesh Mohandas and Alwyn R. Pais Information Security Research Lab, National Institute of Technology Karnataka, Surathkal, India {vip04pat, radhesh, alwyn.pais}@gmail.com ATTACKS ON WEB SERVICES AND MITIGATION SCHEMES

17. Forewords by Mark Curphey, Joel Scambray, and Erik Olson Improving Web Application Security Threats and Countermeasures

18. Encription limited The Stables White Lodge Bevere Worcester WR3 7RQ www.encription.co.uk Campbell Murray encryption limited "The need for secured web development"

19. http://www.linuxtoday.com/infrastructure/2008091100735OSSV

20. Monika Sachdeva, Krishan Kumar Gurvinder Singh Kuldip Singh SBS College of Engg. & Technology," Performance Analysis of Web Service under DDoS Attacks" Guru Nanak Dev University Indian Institute of Technology Ferozepur, Punjab, India Amritsar, Punjab, India Roorkee, Uttarakhand, Indiamonika.sal(kediffmail.com gzsbawa7 1(yahoo.om kds56fec(&riitr.ernetmin) 2009 IEEE International Advance Computing Conference Acknowledgment

21. Adam Kie˙zun MIT akiezun@csail.mit.edu Philip J. Guo Stanford University pg@cs.stanford.edu Karthick Jayaraman Syracuse University kjayaram@syr.edu Michael D. Ernst University of Washingtonmernst@cs.washington.edu

22. Y. Song, S. J. Stolfo, and A. D. Keromytis, "Spectrogram: A mixtureof-markov-chains model for anomaly detection in web traffic," in Proc.of the 16th Annual Network & Distributed System Security Symposium,San Diego, CA, USA, February 2009.

23. Liang Guangmin Computer Engineering Department Shenzhen Polytechnic, Shenzhen 518055, "Modeling Unknown Web Attacks in Network Anomaly Detection "China Email: gmliang@oa.szpt.net Third 2008 International Conference on Convergence and Hybrid Information Technology.

24. Dragan Vidakovic Gimnazija Ivanjica" A Novel Approach to Building Secure Systems" vidakd@ptt.yu Dejan Simic FON Belgrade dsimic@fon.bg.ac.yu .

25. Open Web Application Security Project." The ten most critical Web application security vulnerabilities" http://umn.dl.sourceforge.net/ourceforge/owasp/OWASPTopTen2004.pdf, 2004, visit on 2005/10/05.