INFORMATION
SYSTEMS IN
MANAGEMENT

Information Systems in Management (2013) Vol. 2 (4) 289–299

SECURE DATA EXCHANGE IN PROVISION OF WEB SERVICES

ZYGMUNT MAZUR, TERESA MENDYK-KRAJEWSKA, HANNA MAZUR

Institute of Informatics, Faculty of Computer Science and Management, Wrocław University of Technology

ICT systems commonly used in communication and web services may be successfully attacked, which may result in unauthorized access to data or taking control of the attacked system. One of the sources of the problem is faulty software, and the faults arise from programming errors or incorrect configuration of applications used. There are a number of threats to secure provision of web services. Already known methods of counteracting undesired phenomena do not offer permanent solutions; however, there are ways allowing for increasing our security on the Internet. This paper is aimed at presentation of significant threats to the security of web services and of possible improvements to be made in this area, which is important when e-services are ever more popular and when we are facing more and more successful attacks on IT systems.

Keywords: soft errors, threats to web applications, security of web services

1. Introduction

The provision of web services using dedicated applications is not entirely secure as ICT systems can be successfully attacked. The main cause is software faults that arise from programming errors, insufficient software testing and incorrect configuration. Faulty applications may allow for access to data or taking control of a system. The scale of the phenomenon we are observing indicates a very serious problem. Unfortunately, the provision of improved software versions or patches for vulnerabilities already discovered offers only a temporary solution.

New faults are being discovered all the time and there are a lot of users who just do not care enough about updating their systems. Sometimes new patches remove the discovered software vulnerabilities but at the same time introduce new vulnerabilities to the system.

Among popular languages used to create websites we can name HTML (to improve the functionality of websites developers also use JavaScript vulnerable to attacks) and an easy and flexible but faulty PHP, which communicates well with database systems (MySQL, PostgreSQL, Oracle). Certainly, there are also numerous threats independent from the programming language. Websites are often created using a free package called XAMPP (Cross-platform, Apache, MySQL, PHP, Perl). A popular server platform for dynamic websites is also LAMP (Linux, Apache, MySQL, PHP).

The security of a web service (including data transfer) depends, among others, on the configuration of a web application, configuration of a web server, protection of the database server and the transfer protocol used for communication. For security reasons the data sent from the user, database or web server files are filtered. The application of special cryptographic mechanisms makes it possible to guarantee their confidentiality and conformity with the original data. Unfortunately, sometimes even advanced protection does not guarantee maximum security. What is worse, tools, trainings and instructions dedicated for IT specialists (system developers, software developers and administrators) may be used also to attack a system.

Web applications available on the server are provided for users using web browsers, but apart from providing information they also allow for interaction between the user and the system (among others, to enter data). They are the basis of e-administration, e-learning, e-banking, e-trade and many other. Web applications include the electronic Platform of Public Administration Services, Wikipedia, auction websites, e-banking systems, systems allowing for booking tickets or hotel rooms, survey and voting systems.

2. Web services

According to a definition formulated by the World Wide Web Consortium (W3C) *a web service* is software providing relevant functionality independent from a hardware platform and implementation, while data are usually transferred with the use of HTTP (Hypertext Transfer Protocol) and XML (Extensible Markup Language) [1].

The use of web applications that provide various services online is often connected with transfer of important data such as personal, financial or medical ones. With the development of digital economy the range of web services is constantly expanded and paper documents are replaced with electronic ones. For example,

universities have replaced traditional student record books and examination minutes with electronic solutions. Another fast-developing area are systems for medical services. For example, once we log onto medicover.pl, we can view a record of our appointments, prescriptions and treatment provided. Since 20 May 2013 we can visit www.krok.org.pl, which allows for entering data about surgeries performed in the National Register of Heart Surgeries. Not to mention money transfer and online shopping, which have become a routine for many of us too.

Common ICT systems must offer high quality, which pertains also to security and accessibility of data that are stored, transferred and processed. The most popular international standard defining the requirements set for the quality management system is ISO 9001:2009.

The most common protocol applied in communication with web services is SOAP (Simple Object Access Protocol) approved by W3C, implemented among others in Apache SOAP, .NET, gSOAP [2] or WCF (Windows Communication Foundation) [3]. WCF facilitates programming of distributed applications and implementation of SOA systems. WCF has the best features applied in other Microsoft technologies for distributed applications and communication plus the latest solutions: .NET Remoting (technology used for communication between application domains), MSMQ (Microsoft Message Queuing), COM+ (Component Object Model) and WebServices. When we create a WCF website we pay no attention to transfer protocols or encryption as it is already defined in the configuration file and does not affect the implementation of the service.

The key aspect of functioning of IT systems is continuity of their operation and data security (current databases, their backup and archives).

Disruptions of a stock exchange or air traffic control systems lead to completely different but very serious consequences. In May 2012 the International Organization for Standardization published a standard called ISO 22301:2012 – Societal security – Business continuity management systems – Requirements. So far business continuity management has been governed by a British standard BS 25999. BS 25999 certificates will expire in May 2014. ISO 22301:2012 is more focused on risk analysis and defining aims related to the continuity of operation, monitoring of effectiveness and assessment of correctness of actions taken.

All disruptions of IT systems, press releases about attacks on strategic systems and data leaks make a lot of people approach online financial transactions with reserve, avoid sending important documents and data using ICT networks and some people are afraid of using web browsers and search engines.

An extremely important aspect of web services and electronic data transfer is archives. It also pertains to Polish state archives. In 2010 there was published a document entitled “State archives strategy for 2010-2020”. Services connected with the provision of common, secure and constant access to the past require a well-thought-out approach to the needs of the public, to sharing archive resources

and their online records (among others, registry office records) as well as the related great responsibility for their security [4].

3. Security of web applications

Data are sent to web applications using HTTP methods: GET, POST or cookies (created by a client contain various data sent by a web server). GET allows for sending data, among others, by adding keys (e.g. index.php?page=contact) and their values in URI (Uniform Resource Identifier – a web standard that makes it possible to identify web resources: people, objects, events). POST on the other hand consists in sending data provided on forms available on websites to web applications.

The verification of data sent online may cover the correctness of their type, scope of values, length, format compliance (e.g. for e-mail addresses or IP), credibility, etc. and shall be performed by both the user and the server. Other aspects to be controlled include the existence of an address on the Internet or the integrity of important data (such as a bank account number or PESEL) using for this purpose special algorithms [5].

Forms that send data often feature certain restrictions, such as 'read only', maximum length of the text or choice limited to only one value. Certainly, each box in the form must be controlled. However, those methods of protection are not enough as a webpage of a form (HTML code) can be saved on a hard drive and next its parameters may be modified (e.g. 'read only' option may be deleted).

The security of services may be threatened in a number of ways. For instance, one can attack using HTTP as it allows for manipulating client's requests and analyzing server's responses.

If forms are sent using POST in which data saved are transferred, for example, to payment gateways and their correctness in not verified, the data may be changed by a website user – and such cases have already been reported.

Attacks on web applications are ever more common and more serious. The problem of software security has been dealt with by the OWASP (Open Web Application Security Project) for many years now. Based on research conducted for several years, in 2010 they published a list of ten most critical threats to web applications [6]. These are as follows:

- injection of a code as a result of errors in data filtering (SQL Injection, untrustworthy data may be sent as part of a command or enquiry),
- XSS attack (Cross Site Scripting) as a result of downloading malicious data to a browser without correct validation and filtering,

- Broken Authentication and Session Management, which may allow for acquisition of access passwords or execution of a command from the level of an authorized user.
- Insecure Direct Object References, which pose a threat of manipulating references in order to access data,
- CSRF (Cross Site Request Forgery) – taking advantage of the privilege of a logged user to send a request to an application in order to perform a specific action,
- errors regarding the configuration of an application, web server, database server,
- encryption errors,
- no restrictions regarding accessing a URL,
- insufficient protection of data sent (no encryption, invalid certificates, poor cryptographic algorithms),
- no validation of redirection.

We are observing more and more cases of squatting, i.e. taking advantage of the similarity between fake and genuine URLs and websites. Users often receive fake or infected (with exploits, Trojans, etc.) websites – according to Kaspersky Lab in Q3 2011 malicious URLs were the most often discovered online threat (more than 75%), while in Q2 2012 the threat increased to 85.8% [8]. In the period from January 2012 to May 2013 Kaspersky Lab experts reported 200 thousand attempts at opening fake websites on Apple computers while in 2011 there was only 1 thousand of such attempts [9].

A document entitled "Online Fraud Report" produced by RSA shows that in 2012 it identified ca. 37 thousand instances of phishing a month. Losses arising from the attacks amounted to ca. 1.5 billion dollars in total. In May 2013 most such attacks were reported in the US (50%) and Great Britain (11%) [7].

Web applications should not feature typical vulnerabilities allowing for attacks, such as:

- disclosure of information by the system (e.g. causes of an error), which may facilitate an attack (for example – information about SQL enquiries which caused an error may reveal the structure of a database – names of tables, columns, data types, etc.),
- disclosure of the path to the place where website files are stored on a server in web application messages,
- path traversal in a web application [10, 11],
- unicode uncoding – UTF-8 encoding allows for entering characters in a number of ways, which causes problems [12],

- forced browsing – navigating in an application not using available references but directly through URLs, which when there is no sufficient control allows for logging onto someone else's account or unauthorized access to data.
- lack of control of data transferred as parameters of an application created in PHP, which may allow for entering and execution of a code,
- parameter delimiter – a possibility to change a separator in a text database, which may lead to modification of privileges.

The security of an application used depends on the environment in which it is launched. Therefore, it is important to configure it correctly and update the entire software (including libraries used by the application). PHP functions which are not used and may pose a threat (e.g. may enable execution of a command from the operating system level) should be disabled (disable_functions). In order to increase protection we can use an extra software. For instance, an advanced system that protects the PHP code interpreter on the server – Suhosin (ensures protection on the kernel level and protects an application against errors), and a mod_security module – a firewall in the application layer (it can monitor HTTP and restrict requests as pre-defined) and the mod_suPHP mode – software that allows for execution of PHP scripts as an authorized user (instead of the default nobody), i.e. restricts access to users' files. A good idea is to define limits for each PHP process (for RAM, process execution time, size of data sent to the server), and enable the display of messages concerning errors in a website code and of information about PHP (among others, about the version number) in response headlines.

These are just a few possibilities of improving the security of web applications. We should also remember to remove programs we no longer use (or update) from the system as they also may become the target.

Certainly, the system should be made available only to those who went though the authentication process successfully, and in the event of failure to log in, the system should not inform which information provided, user name or password, is wrong. In the case of a web application when the name is already known, the password may be hacked in a dictionary attack or a brute force attack. In order to make the attack more difficult we should not provide the user name in a drop-down list while the logging mechanism should minimize the number of possible attempts.

Data must be protected also against programs that browse web resources automatically. However, the act of placing names of important folders that are not to be browsed by robots in a robots.txt file (in the server main catalogue) already indicates where important data are stored [13].

4. Threats to secure provision of services

One of basic threats to the security of web applications is lack of filtering of data transferred or control errors. Attackers may take advantage of various forms (registration form, add-your-post forms, etc.) which applications use to transfer data.

Forms completed incorrectly should be rejected by applications while users should be informed about the fact. However, if JavaScript is disabled in a web browser (using NoScript add-on) – data will not be verified and it will be possible to enter any data in the form, including a malicious code (in HTML or JavaScript). This vulnerability may be used to carry out XSS and SQL Injection, which may result in illegal actions (e.g. execution of a forbidden command, gaining unauthorized access to data, session hijacking and redirecting a client to an infected website).

An attack is also possible when there is no length limitation in a text box (e.g. a comment) that must contain a string (which is controlled). In such an event any code can be entered.

Another threat is the possibility to send forms from external (fake) addresses. The headline Referer (which informs where we were redirected from) may be modified for example using Firefox add-ons [14]. In the case of a completed form with the method parameter set to GET, it is possible to modify the URL in a browser. When the form is set to POST it is possible to send modified data as GET if the application uses the $_REQUEST table instead of global ones.

Still another threat to the security of web applications is connected with file transfer on the part of the user. For example, errors discovered by the PHP interpreter during saving a file to a temporary folder are connected with exceeding certain limits. As it is possible to modify the maximum size of files defined in the form and falsify the content of the mime_type box (e.g. set it to image.gif using a special application), there is a threat of saving a malicious code together with the file transferred in the server. Filename extension filtration (based on the so-called black list of rejected types of sets) can also be circumvented.

Skilful manipulation of an address may lead to unauthorized reading of files, and even injection and execution of a PHP code (e.g. if in the HTML code template there are fragments of the PHP code and the functions *include* or *require* are used).

Another threat is the attack on the application session. As HTTP is stateless (no relation between subsequent requests sent), the application itself identifies users and supervises their actions (e.g. clients of online stores or bank systems). For security reasons it should use cookies for that purpose. The session ID (a standard PHP mechanism generates randomly a 32-character alphanumeric string) is saved by the client and sent to the server during each request (the content of the session is stored by the server).

In the PHP language there are ready functions (e.g. session_start()) that facilitate management of the mechanism.However, it is possible to obtain a session ID (PHPSESSID) – session hijacking, session fixation and even guessing the ID, which is tantamount to impersonating the user and gaining access to data. Session fixation is dangerous for applications that allow for sending a session ID using GET or POST parameters. On the other hand, the risk of session hijacking is higher when it is possible to send the ID in an URL (it is saved in the history or the http headline – Referer). The session ID may be also tapped if the data transfer is not encrypted (using SSL – HTTPs).In a number of applications there is a problem with adequate protection of access to data (e.g. due to errors in the authorization mechanism). When we key in the web browser, for example, filetype:pdf cv, we may come across files that lack adequate protection, located in the so-called deep hiding. They can be accessed if we know the access path or by accident. Such a situation has already taken place in 2011 – seven thousand files containing resumes were made accessible on terazpraca.pl [15].

5. Threats to electronic data

An important requirement set for IT systems, in particular those that provide web services, is their reliability. Meanwhile, even large systems of key importance for financial and state institutions do not meet it.

A latest example here is unavailability of the PESEL system in August 2013 (10-11 and 14-18.08), which is of key importance for a number of other systems such as ePUAP and eWUŚ (Electronic Verification of Patient's Privileges) [16]. Another example is eWUŚ implemented on 1 January 2013, which did not work properly in March and July 2013 ([17, 18]). The continuity of system operation is also a problem reported among banks, such as BZ WBK (August 2013) [19, 20], Getin Bank and Multibank (July 2011 [21]) and PKO BP (November 2011 [22], September 2013 [23]). In December 2010 as many as four banks reported system failures in one week. These were mBank, ING Bank Śląski, Kredyt Bank and Bank Pekao [24]. In 2012 servers of the Ministry of Finance failed [25, 26]. The above examples of disrupted system operation prove the significance of the problem.

Another issue is insufficient data protection, no encryption or access control, which leads to data leaks or their availability to unauthorized users. For instance, in July 2012 there occurred a leak of data of around 10 thousand users from a Kielce sports club website [27]. A much more serious data leak was reported in June 2013 from Facebook, which was caused by an error in the Download Your Information software used for downloading archive data by users [28, 29], or a series of leaks from PlayStation Network accounts in 2011.

When testing PCs with MS Windows in 2012, Kaspersky Lab reported 132 million vulnerable applications, including 802 unique vulnerabilities [30].

The most vulnerable were Adobe Shockwave/Flash Player, Apple iTunes/QuickTime and Oracle Java. They also noticed that those programs were not updated (as many as 30% of users did not update Java within seven weeks after the launch of the new version). In 2008 a serious error was discovered in OpenSSL used by Debian and distributions that rely on it (Ubuntu, Knoppix).

In a report regarding Q2 2013 Kaspersky Lab analysts mention a malicious package (NetTraveler) used to infect computers of VIPs, the state, embassies, oil and gas industries, research centers and companies cooperating with the army. The attacks were possible, among others, due to failure to update Microsoft software featuring vulnerabilities CVE-2012-0158 and CVE-2010-3333 (although patches were already available) [31]. In July 2013 Microsoft launched an update removing 34 vulnerabilities on its software, including CVE-2013-3172 that allowed access to the system kernel, Internet Explorer errors (16 vulnerabilities allowing for remote code execution and one vulnerability leading to data leak), errors in TrueType font processing (that allowed for execution of Blackhole exploits which redirected users to websites taking control of a system) [32].

Another target of the attacks are computer games producers (in particular online games). In the period from March to June 2013 30 such companies located mainly in South and East Asia, Germany, the USA, Japan, China, Russia, Brazil, Peru and Belarus were reported to have been attacked [30].

A major vulnerability in WordPress v.3.4 (downloaded 3 million times within only two weeks after its launch) led to manipulation of privileges. An update removing this vulnerability was made available in July 2012 [33].

6. Conclusion

Programming problems include incorrect validation and filtering of form boxes and variables transferred in URLs. The software certification procedure should verify not only software functionality but also check if it reveals any data, if it successfully protects against user's errors, prevents transfer of a code or of attempts at disabling protection and if it allows for viewing or modification of the code.

In order to find applications and systems that are faulty or vulnerable to attacks, we can make use of a specialist software. Big corporations (e.g. Google) offer huge amounts for discovering vulnerabilities in their system [34] in order to prevent costs they would have to pay for losses caused as a result of vulnerabilities of their software.

As security is a great responsibility of organizations (companies, institutions) that implement ICT systems in new areas (the threat may concern even our health and lives) – now it is possible to insure IT systems. One of such insurance policies offered (called CyberEdge), which allows an institution to insure against a cyber attack or data leak, was made available in September 2012 by Chartis Europe.

As part of the agreement, the injured party can additionally be provided with legal assistance and technical support ([20, 21]). Introduction of such an insurance offer on the market shows how significant IT security problems are.

Lack of high-quality protection of ICT systems (including web services) may prevent further development of digital economy.

REFERENCES

[1] *W3C Web Service Glossary*, www.w3.org/TR/ws-gloss/#webservice

[2] *The gSOAP Toolkit for SOAP Web Services and XML-Based Applications*, www.cs.fsu.edu/~engelen/soap.html

[3] *Wstęp do technologii WCF* (2007) http://codeguru.geekclub.pl/baza-wiedzy/wstep-do-technologii-wcf,2186

[4] *Strategia archiwów państwowych na lata 2010-2020, Załącznik do Komunikatu Nr 1/2010 Naczelnego Dyrektora Archiwów Państwowych z dnia 23 grudnia 2010 r.*

[5] Hoffman B., Sullivan B. (2009) *Bezpieczeństwo aplikacji tworzonych w technologii Ajax.* Wydawnictwo Helion, Gliwice.

[6] www. owasp.org/index.php/Category:OWASP_Top_Ten_Project.

[7] *Online Fraud Report*, www.rsa.com/phishing_reports.aspx, 11.08.2013.

[8] www.viruslist.pl/analysis.html?/newsid=715

[9] Gliwiński M. (2012) *Ataki na strony internetowe*. Helion, Wydawnictwo CSH, Kwidzyń.

[10] http://blog.gruszka.info/2007/03/05/odpornosc-polskich-stron-na-ataki-typu-path-traversal.

[11] www.future-processing.pl/blog/path-traversal-o-krok-od-bledu.

[12] http://gajdaw.pl/varia/polskie-ogonki-na-www/print.html.

[13] https://support.google.com/webmasters/answer/156449?hl=pl

[14] Hope P., Walther B. (2010) *Testowanie bezpieczeństwa aplikacji internetowych*. Wydawnictwo Helion, Gliwice.

[15] http://niebezpiecznik.pl/post/7-000-cv-w-glebokim-ukryciu.

[16] *Przerwa w funkcjonowaniu usług systemu PESEL* (2013), http://epuap.gov.pl/wps/portal/E2_Aktualnosci/?WCM_GLOBAL_CONTEXT=/epuap2/ePUAP2/PL/OePUAP/Aktualnosci/2013/Przerwa w funkcjonowaniu uslug systemu PESEL.

[17] www.nfz.gov.pl/new/index.php?katnr=9&dzialnr=4&artnr=5372.

[18] http://wiadomosci.gazeta.pl/wiadomosci/1,114871,14368777,Wielka_awaria_systemu_eWUS__Nie_dziala_w_calej_Polsce_.html,

[19] http://blog.bzwbk.pl/2013/08/zwolnienie-klientow-z-prowizji-za-przelewy-w-oddzialach.

[20] Bolanowski J. (2013) *Awaria w BZ WBK*, http://finanse.wp.pl/kat,6599,title,Awaria-w-BZ-WBK,wid,15857506,wiadomosc.html.

[21] Loda M. (2011) *Awarie systemów bankowych w Getin Banku i Multibanku*, http://wyborcza.biz/finanse/1,105684,9926654,Awarie_systemow_bankowych_w_Getin_Banku_i_Multibanku.html, 11.07.2011.

[22] Ogórek *S.(2011) Awaria w największym polskim banku!*, http://finanse.wp.pl/kat,6602,title,Awaria-w-najwiekszym-polskim-banku,wid,12890230,wiadomosc.html.

[23] Stanisławski P., Joanna Sosnowska J. (2013) *Poważna awaria w PKO BP - z kont Inteligo znikają pieniądze*, http://technologie.gazeta.pl/internet/1,104530,14588869,Powazna_awaria_w_PKO_BP___z_kont_Inteligo_znikaja.html#BoxSlotII3img, 11.09.2013.

[24] Samcik M. (2010) *Czarny tydzień polskich banków, awaria goniła awarię. To nie przypadek!*, http://samcik.blox.pl/2010/12/Czarny-tydzien-polskich-bankow-awaria-gonila.html.

[25] *Wielka awaria serwerów w Ministerstwie Finansów*, http://wyborcza.biz/biznes/1,100896,12924845,Wielka_awaria_serwerow_w_Ministerstwie_Finansow.html, 26.11.2012.

[26] *Ministerstwo Finansów na kolanach*, http://na-plus.blogspot.com/2012/11/ministerstwo-finansow-na-kolanach.html, 26.11.2012.

[27] Wątor J. (2012) *Po ataku hakerów na Koronę Kielce. "Zmieńcie hasła"*, http://kielce.gazeta.pl/kielce/1,35255,12058752,Po_ataku_hakerow_na_Korone_Kielce___Zmiencie_hasla_.html, 02.07.2012.

[28] http://nt.interia.pl/internet/news-jak-sie-uchronic-przed-wyciekiem-danych,nId,986693?utm_source=paste&utm_medium=paste&utm_campaign=msie 30.06.2013.

[29] www.chip.pl/news/bezpieczenstwo/luki-bezpieczenstwa/2013/06/wyciek-danych-6-milionow-uzytkownikow-facebooka, 22.06.2013.

[30] www.securelist.pl/threats/detect/7108,raport_kaspersky_lab_ocena_poziomu_zagrozenia_jakie_stanowia_luki_w_oprogramowaniu.html

[31] www.securelist.pl/analysis/7242,ewolucja_zagrozen_it_ii_kwartal_2013_r.html 29.08.2013.

[32] Baumgartner K. (2013) *Aktualizacje Microsoftu w lipcu 2013 – poważne błędy w IE, DirectShow oraz w funkcji przetwarzania czcionek TrueType*, www.securelist.pl/blog/7233,aktualizacje_microsoftu_w_lipcu_2013_powazne_bledy_w_ie_directshow_oraz_w_funkcji_przetwarzania_czcionek_truetype.html 16.07.2013.

[33] www.instalki.pl/aktualnosci/software/8292-aktualizacja-wordpress-3-4-zamyka-istotna-luke-w-zabezpieczeniach.html 01.07.2012.

[34] www.theregister.co.uk/2012/04/24/google_ups_bug_bounty