ISBN: 978-80-245-2125-1
DOI: 10.18267/pr.2015.pav.2125.13

WEB APPLICATION SECURITY: AUDIT TOOLS & LANGUAGES

DOI: 10.18267/pr.2015.pav.2125.13

RNDr. Alexander Galba

Vysoká škola ekonomická v Praze

Katedra systémové analýzy

alexander.galba@vse.cz

ABSTRACT

Testing web application security is an important part of the development web application. Tools and processes for testing the security of web applications and detecting their vulnerabilities experienced development in recent years. Many of these tools and processes depend on the chosen development environment. A crucial factor is the experience of developers, set procedures and control mechanism used in creating web applications. Automated tools are becoming very popular. "Black box" web vulnerability scanners can find security problems such as cross-site scripting, command execution, directory traversal, SQL injection, insecure server configuration and others. Application of these tools requires no wide knowledge about web development technologies. The problem is the interpretation of results and the subsequent repair issue. These tools cannot guarantee the elimination of security risks. Most popular programming languages in web applications are ASP.NET, PHP and JAVA. If we focus on security of the website in terms of programming languages, number of vulnerabilities will be in each language the same. Selecting the programming language does not have a direct impact on security of web applications. Differences in statistics of successful attacks on web applications are caused by a combination of factors for various programming languages.

KEY WORDS

web application, vulnerability, security scanner, security, programming language

INTRODUCTION

Testing web application security is an important part of the development web application. Number of incidents and loosing sensitive information are serious problem. (Open Security Foundation, 2015). Web application security factors are widely tracked and identified. (OWASP, 2013). Tools and processes for testing the security of web applications and detecting their vulnerabilities experienced development in recent years. Many of these tools and processes depend on the chosen development environment. A crucial factor is the experience of developers, set procedures and control mechanisms used in creating web applications. These factors are dependent on the choice of the development environment and the ability of applying guidance. It is interesting to see how the security of web applications depends on the selected programming language, and what progress have automated tools recorded in recent years. If we focus on the security of the website in terms of programming languages, number of vulnerabilities will be in each language the same. Selecting the programming language does not have a direct impact on security of web applications. The differences in statistics of successful attacks on web applications are caused by a combination of factors for various programming languages.

AUTOMATED SECURITY TESTING TOOLS

Tools for safety testing, and identifying weaknesses in web applications can be divided into the following categories: (Galba, 2012)

1) practices recommended by manufacturers and suppliers of web technologies

ISBN: 978-80-245-2125-1

DOI: 10.18267/pr.2015.pav.2125.13

2) procedures based on methods of IS audit

3) checklists

4) automated tools

The first two categories of tools include comprehensive methods for ensuring the security of web applications including validated manuals and procedures related to the entire lifecycle of web applications and technological recommendations for infrastructure. (Microsoft, 2011) (IBM, 2102) Categories 3 and 4 include simple procedures applicable and usable to test the safety of a web application before deciding whether to use comprehensive screening of Category 1 or 2.

Automated tools are becoming very popular. "Black box" web vulnerability scanners can find security problems such as cross-site scripting, command execution, directory traversal, SQL injection, insecure server configuration and others. Application of these tools requires no wide knowledge about web development technologies. The problem is the interpretation of results and the subsequent repair issue. 37 web application vulnerability scanners is listed on website (OWASP, 2013). Eight of these scanners are open source and six are free for use with limited capability.

I have not found any serious rating list or test result success of web vulnerability scanners yet. Four web vulnerability scanners were tested. in research (Jason Bau, 2012). Scanners detect between 21% and 32 % of all testing vulnerabilities. The worst results scanners have in "stored" forms of XSS and SQLI with average detection below 10%. This shows that the use of these tools is useful as security indicators that can highlight weakness of web applications. These tools cannot guarantee the elimination of security risks.

PROGRAMMING LANGUAGES

Finding serious statistic data of programming languages use in web application is not easy (TABLE 1).

	(builtwith.com, 2015)	(Whithstsec, 2014)	(w3techs.com, 2015)
APS.NET	48%	36%	16,2%
PHP	51%	16%	81,5%
Ruby on Rails	1%	-	0,6%
JAVA	-	28%	3%

TABLE 1 (source author)

Differences are probably caused by selecting of the data source. We can say that most popular programming languages in web applications is ASP.NET, PHP and JAVA. If we focus on security of the website in terms of programming languages, number of vulnerabilities will be in each language from 10% to 11% (Whithstsec, 2014). Most successful security attacks was on PHP web application. (builtwith.com, 2015). This suggests that the PHP applications have larger security risk. The reason may be the popularity of PHP at freelancer community. Combination of freelancer and PHP leads to roughly 3x higher security risks. (Jason Bau, 2012).

CONLUSION

Web application security issues become the important part of developing process. Selecting the programming language does not have a direct impact on security of web applications. The differences in statistics of successful attacks on web applications are caused by a combination of factors for various programming languages. Each programming language and developing environment have enough resources for building secure applications. Testing vulnerability and finding weakness of application depend on procedures and tools that developers use. Vulnerability scanners are useful tools which can highlight weakness of web applications but cannot guarantee the elimination of most security risks.

ISBN: 978-80-245-2125-1

DOI: 10.18267/pr.2015.pav.2125.13

REFERENCES

Builtwith.com, 2015. Buildwith.com. [Online] Retrieved November, 2015: Available at: http://trends.builtwith.com/

Galba, A., 2012. Nástroje pro kontrolu bezpečnosti webových aplikací. Systémové přístupy. IBM, 2102. A layered approach to delivering security-rich Web applications. [Online] Retrieved October, 2015: Available at: *ftp://ftp.software.ibm.com/software/rational/web/whitepapers/r_wp_securityrichwebapps.pdf*

Jason Bau, F. W. E. B. P. M. J. C. M., 2012. Vulnerability Factors in New Web Applications:Audit Tools, Developer Selection & Languages. [Online] Retrieved October, 2015: Available at: *http://seclab.stanford.edu/websec/scannerPaper.pdf*

Microsoft, 2011. Security and Service Continuity for Enterprises. [Online] Retrieved April, 2015: Available at: *http://www.microsoft.com/download/en/details.aspx?id=13602*

Open Security Foundation, 2015. Data Loss Statistics. [Online] Retrieved November, 2015: Available at: *http://datalossdb.org/statistics*

OWASP, 2013. OWASP Top Ten Project. [Online] Retrieved November, 2015: Available at: *https://www.owasp.org/index.php/Top10#tab=OWASP_Top_10_for_2013*

w3techs.com, 2015. w3techs. [Online] Retrieved November, 2015:Available at: http://w3techs.com/technologies/overview/programming_language/all

Whithstsec, 2014. 2014 Website Security Statistics Report. [Online] Retrieved November, 2015: Available at: *http://info.whitehatsec.com/rs/whitehatsecurity/images/statsreport2014-20140410.pdf*