Web Security in University Curricula

Cristian OPINCARU

Thales Rail Signalling Solutions
319 Splaiul Independentei, River View House, 060044, Bucharest, ROMANIA
cristian@opincaru.ro
http://www.opincaru.ro

Abstract: While Web applications and Web services gain more and more ground, the academic curriculum is not always keeping the pace. This paper presents the content of a course focused on Web security; as such it starts by defining the goals of the course, then defines the topics for the course units and finally describes the topics and the setup for laboratory units. The paper brings its contribution through the design of a course covering security aspects for both Web applications and Web services and through the detailed description of practical units and laboratory setup.

Key-Words: Security, Curriculum, Web applications, Web services.

1 Introduction

The current trend in the IT industry for some years already is that more and more applications are moving from desktop to the Web. From email clients, to office applications like Google Documents and Google Spreadsheet, to picture editing software like Adobe Photoshop Online, up to internal enterprise applications, Web applications gain more and more user base. At the same time, due to technologies like RSS, the effect of social networks and services like twitter.com, applications that were not possible on the desktop see light on the Web. They incorporate a multitude of technologies and their code runs in different environments (browser, virtual machines, server-side, etc.) and are written in a large amount of languages.

As their usage is so spread and the technologies used so diverse, it is obvious to everybody that the security of Web applications and services is immensely important. It is important for users to be aware of it, it is important for developers to be aware of it and quite naturally it is important for the future IT practitioners to be aware of it. Therefore, the subject of Web security should be covered by the university curricula.

As also recognized in [1] most of the current university curricula is focusing on network security and operating system security rather than focusing on the application security. Given the fact that Web applications operate in a dangerous environment, students should have a solid understanding of the threats and vulnerabilities on the one hand, and on the mitigation possibilities on the other. This is especially important as some of the threats are unique to Web applications (for example cross-site scripting or injection attacks).

Additionally, most of the courses covering Web security, including those described in [1], [7], [8] solely focus on Web applications without touching Web services. As Web services are a major trend pushed by both the enterprise world through the SOAP and WS-* specifications and the Web 2.0 world through the REST approach, it is important that students also learn the security aspects related to Web services.

This paper outlines the topics that need to be covered in a university course focused

This is a post conference paper. Parts of this paper have been published in the Proceedings of the 2nd International Conference on Security for Information Technology and Communications, SECITC 2009 Conference (printed version).

ISSN 2067 – 4074

on Web security, covering both applications and services. It is based on the experiences so far from the Security Masters Program at the Romanian Military Academy in Bucharest [30] and the course content suggested in other articles [1], [2], [3], [4], [5], [6] and course descriptions [7], [8]. Besides the list of topics for the course, the article also presents topics for laboratory, including a list of tools that the students should learn how to use, and advices about the setup of the laboratory.

The reminder of this paper is structured as follows: Section 2 describes the goals and the format of the course, Section 3 presents the content of the course, Section 4 describes the content of the laboratory classes and supporting material, Section 4 details the tools to be used during laboratory classes and Section 5 contains the conclusions.

2 Goals

The primary goal of a Web Security course is for students to learn the unique aspects regarding the security of Web applications and services. The course should be adapted to the level of understanding of the students, and therefore in the development of the course it is important to take into consideration the background and the interests of the students.

The experience so far showed that most of the students own bachelor diplomas in computer science and telecommunications. As far as their interests are concerned, they are quite diverse, ranging from employees of software security companies wanting to acquire deeper knowledge of security aspects, to employees of software companies with the desire to learn concepts that they can apply in their day-to-day activities, and up to non-IT professionals that have interests in the subject but will not (immediately) apply the knowledge in their everyday work. Furthermore, the experience so far also showed that not all students have in-depth knowledge of Web applications and Web services.

Therefore, the aim is to develop a course that contains a refresher of the Web technologies and is not addressed to specialists. As such, it shall insist on the general concepts and ensure that these concepts have been well understood through illustrative laboratory classes. For those students that want to learn more, references shall be provided in the course materials and more difficult exercises will be prepared for them to solve during the laboratory classes.

Upon successful completion of the proposed Web Security course, the students should:

- Have a general understanding of the technologies used by both Web applications as well as Web services;
- Be familiar with the implications of client-side technologies like JavaScript and Java Applets, used in Web applications;
- Have a good understanding of the authentication methods used by both Web applications and Web services;
- Be able to identify, explain and detect common vulnerabilities associated with Web applications;
- Be familiar with the implications of XML security technologies, especially WS-Security, used in Web services;
- Have a general idea about techniques used for secure programming.

2.1 Course format

The course contains 12 units (2 hours each) of teaching that are accompanied by 12 units of laboratory (2 hours each). When this course is administered as part of a 14 weeks semester, the first and the last unit are saved for introduction and summary respectively, as well as for other administrative issues.

Each teaching unit shall be followed by a laboratory unit where the students can put in practice what they have learned in the

teaching unit. If this course is administered as part of an executive masters program, the laboratory unit shall be scheduled immediately after the teaching unit, retaining the student's interest and motivation.

3 Content

In designing the content of this course, the content of several other similar courses were analyzed, including [1], [7] and [8]. The content of the course was adapted to the goals described in the previous section.
The course addresses both Web applications and Web services. In order to understand the security of Web applications students need to have a good understanding of the Web technologies and the security models of client-side technologies. This is the reason why two units of the course are dedicated technologies used in developing traditional Web applications. As new Web applications are more and more based on AJAX and JavaScript frameworks, one of the course units is dedicated to Web 2.0 technologies and their security implications.
In contrast to the up-mentioned references where the courses are solely focused on Web applications, this course will also teach students about security aspects related to Web services. For this, a similar approach is followed, in that the student is first presented with the technology basics (SOAP, REST) and only after that, the security topics are presented - XML security, WS-Security, authentication in Web services.
Below you can see the content of the course, with a detailed list of topics for each individual teaching unit:

1. Web security
    a. What are Web applications?
    b. Security principles
    c. What needs to be secured?
2. Web 1.0 technology refresher
    a. HTTP GET / POST
    b. URLs
    c. Cookies
    d. Encodings (Base64, HTML, URL)
    e. SSL
3. Client-side technologies
    a. JavaScript and JavaScript security
    b. Applets and Java security
4. Web 2.0 technologies and frameworks
    a. AJAX
    b. Google Web Toolkit [21]
    c. Dojo [20]
5. Authentication in Web applications
    a. Basic, Digest and Form Authentication
    b. Session management
    c. OpenID [28]
6. Vulnerabilities
    a. Introduction and principles
    b. Web testing proxies
    c. Vulnerability scanners
    d. Brute force attacks
    e. Authentication by-passing
    f. Content spoofing
7. Vulnerabilities
    a. Injection attacks
    b. Cross-site attacks
    c. Logical attacks
8. Web services technologies
    a. Service Oriented Architecture
    b. SOAP
    c. REST
9. XML Security
    a. XML DSIG
    b. XML ENC
10. WS-Security
    a. Username Token, X509 Token
    b. WS-Security extensions
        i. WS-Policy
        ii. WS-Trust
        iii. WS-SecureConversation
11. Authentication and Authorization in Web Services
    a. SAML
    b. Shibboleth

ISSN 2067 – 4074

12. Secure programming
    a. Code review
    b. Automated Web tests

3.1 Supporting material

There is plenty of supporting material to choose from. This includes the *Web application hacker's handbook* [10] and the *OWASP Guide to building secure Web applications and Web services* [17] and Microsoft's *Improving Web Application Security: Threats and Countermeasures* [12] where general topics about security in the context of Web applications and services are described. In addition to this, two OWASP projects - [15] and [16], provide useful information about code reviewing and security testing with the focus of the Web in mind. Last, M. Gregg's book *Build your own security lab* [11], although not specially focused on Web applications and services, provides useful information about security tools.

4 Laboratory

The main purpose of the laboratories is for the students to practice what they learn in the course units. The aim here is to make them as practical as possible and make them as accessible as possible for the students. They should be practical in order to retain the student's interest; the required knowledge to complete the exercises should be low so that all students are actively involved.
The plan for laboratory units is detailed below. In developing the laboratory content one laboratory unit was assigned to each course unit. While the students learn the theoretical part in the course unit they get to apply what they learned in the corresponding laboratory unit.

1. Web security
    - Introduction to the laboratory environment
    - Tools, Web applications
2. Web 1.0 technology refresher
    - Using a web proxy to intercept and analyze HTTP traffic
    - HTTP requests, cookies, encodings
3. Client-side technologies
    - Writing JavaScript code
    - Violating the same origin policy
    - Deploying a Java Applet / certificates / signing
4. Web 2.0 technologies and frameworks
    - Simple AJAX application with GWT / DoJo
    - Investigating XmlHttpRequest security implications
5. Authentication in Web applications
    - WebGoat exercises on HTTP authentication and session management
    - Using a Web proxy to intercept requests and responses
6. Vulnerabilities
    - Assessing a Web application using vulnerability scanners and inspection tools
7. Vulnerabilities
    - WebGoat exercises on vulnerabilities
    - Using a Web proxy to detect and exploit vulnerabilities
8. Web services technologies
    - Using a Web proxy to intercept Web services
    - SOAP-over-HTTP / REST
9. XML Security
    - Exercises on XMLDSIG and XMLENC
10. WS-Security
    - Deploying a Web service and configuring WS-Security
    - Analysis of Web services traffic using a Web proxy
11. Authentication and authorization for Web services
    - Setting up Shibboleth authentication

- Setting up OpenID authentication

12. Secure programming
    - Performing code review on a given application

4.1 Tools

There are numerous tools that can be practiced during laboratory hours. These include Web inspection tools, Web testing proxies, vulnerability scanners, application servers implementing security features and other. For each category, one can find both open-source as well as commercial programs – for the laboratories we tried to focus on open-source tools because they are easier to procure. A good place to start for general-purpose security tools is the top-100 [19], while recommendations for tools specific to Web security are made in [12], [10] and [1].

One of the most important tools a student needs to learn is a Web proxy. They are primarily used for security assessment of Web applications, but they are also used for analyzing the HTTP traffic. Students learn the basics of HTTP GET / POST, different encodings, how parameters are handled, HTTP authentication, cookies and many more through the use of a Web proxy. Similarly, for Web services, the students learn how the requests are handled (SOAP / REST) by investigating the HTTP traffic. The main tool we chose was *WebScarab* [13] which is simple enough for all students to understand, it is open-source and is also very easy to deploy as it runs as a Java Web Start application.

Vulnerability scanners are used by students to detect vulnerabilities in applications. They are usually used by auditors and testers to assess a large number of Web pages in a short amount of time. For this, they usually rely on a spider engine to navigate to all pages of a Web application and make requests to test for well-known vulnerabilities such as file handling errors, injections, cross-site scripting, etc. From the numerous tools existing, we chose *wapiti* [24] and *nikto* [25]. The working principle of the two is slightly different - *nikto* relies on a vulnerability database, while *wapiti* aims to discover unknown vulnerabilities in Web applications; both of them are GNU licensed.

Vulnerabilities are not only discovered through scanners but also manually. One of the main educational tools for the laboratory is *WebGoat* [14] which is a deliberately insecure J2EE application maintained by OWASP and designed to teach Web application security lessons. It consists of various practical modules with themes ranging from the HTTP protocol basics to authentication, cross-site scripting and Web services. Each module consists of several exercises with various difficulty levels, and each exercises contains hints to the solution; the last hint explains the solution. In addition, movies are provided with screen captures of the solution of each exercise. Other deliberately insecure Web applications include *Hacme Bank* [23] and *Bad Store* [22] which are also appropriate for educating students, however they do not contain interactive exercises.

Yet another important tool in the arsenal of Web assessors are Web inspection tools like *Firebug* [26]. With their help you can inspect the content of a Web page, quickly discover hidden input fields, debug and inspect JavaScript and monitor network activity. There are other inspection tools, but Firebug is very comfortable to use as it comes as an extension to the Firefox browser.

For the Web services modules students will have to deploy a simple Web service in an application server and configure WS-Security features. Upon successful configuration, they can use Web proxies to analyze the traffic and see the result of the different configurations. Depending on the programming experience of students, one can choose from *Apache AXIS2* [29] (Java

ISSN 2067 – 4074

based) or *ASP .NET* for students more familiar with .NET technologies. The exercise can be extended to include Web service authentication via one of *Shibboleth* [27] or *OpenID* [28].

4.2 Laboratory set-up

For the setup of the laboratory the most practical way is to use virtual machines – these allow on one hand the creation of an isolated environment where vulnerable Web applications can be deployed and tested, and on the other hand they offer the advantage that it is easy to install them on all the machines in a university laboratory. Furthermore, all students work on the same version of the machine, so it is easy to show solutions to the whole class and it encourages collaboration between participants as it is easier for students to help each other.

So far *VMWare Player* was used as it is well known, available on multiple platforms and also free to use. An *Ubuntu Linux* appliance was set up and packed with all the tools and software required in the laboratory. The appliance was also distributed via Internet to the students so that they can also practice the exercises at home if desired.

5 Conclusions

This paper discussed the content of a course covering both Web application security as well as Web service security. First, the goals of the course were set by establishing a list of abilities that the students should posses upon successful completion of the course and defining the format of the course. Afterwards, the content of the course units has been introduced, by detailing the content of each individual teaching unit. For each of these, a laboratory unit has been designed and its content has been presented. Most importantly, the paper presents a list of free tools that should be used by the students during the laboratory hours.

Further directions for this work include setting the course into practice and getting the feedback from the students. Further topics to be added to the course include Operating System strengthening and operational security covering topics such as how to securely deploy applications and how to configure the host, database and application servers.

6 References

[1] Walden, J. 2008. "Integrating web application security into the IT curriculum." In *Proceedings of the 9th ACM SIGITE Conference on information Technology Education* (Cincinnati, OH, USA, October 16 - 18, 2008). SIGITE '08. ACM, New York, NY, 187-192.

[2] Cynthia E. Irvine, Shiu-Kai Chin, Deborah Frincke, "Integrating Security into the Curriculum," Computer, vol. 31, no. 12, pp. 25-30, December, 1998.

[3] Bogolea, B. and Wijekumar, K. 2004. "Information security curriculum creation: a case study." In *Proceedings of the 1st Annual Conference on information Security Curriculum Development* (Kennesaw, Georgia, October 08 - 08, 2004). InfoSecCD '04. ACM, New York, NY, 59-65.

[4] Yu, H., Liao, W., Yuan, X., and Xu, J. 2006. "Teaching a web security course to practice information assurance." *SIGCSE Bull.* 38, 1 (Mar. 2006), 12-16.

[5] Irvine, C. E. 2006. What Might We Mean By "Secure Code" and How Might We Teach What We Mean?. In *Proceedings of the 19th Conference on Software Engineering Education and Training Workshops* (April 19 - 21, 2006). CSEETW. IEEE Computer Society, Washington, DC, 22.

[6] Irvine, C. E. 2003. Teaching Constructive Security. *IEEE Security and Privacy* 1, 6 (Nov. 2003), 59-61.

[7] "XACS241 - Web Security 2.0", Course Description, Standford University, http://scpd.stanford.edu/search/publicCo

urseSearchDetails.do?method=load&courseId=1284858
[8] "Web Security", Course Description, International Webmasters Association – eclasses.org, http://iwa-hwg.eclasses.org/courseS111/
[9] "Web Security", Google Code University, http://code.google.com/edu/security/index.html
[10] M. Pinto, D. Stuttard, "Web Application Hacker's Handbook", Wiley, 2007
[11] M. Gregg, "Build your own security lab: a field guide for network testing", Wiley, 2008
[12] Microsoft Corporation, "Improving Web Application Security: Threats and Countermeasures", Microsoft Press, 2003
[13] OWASP, "The WebScarab Project", http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
[14] OWASP, "The WebGoat Project", http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
[15] OWASP, "The Code Review Project", http://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
[16] OWASP, "The OWASP Testing Guide", http://www.owasp.org/index.php/Category:OWASP_Testing_Project
[17] OWASP, "A guide to building secure web applications and web services", 2005, http://www.owasp.org/index.php/Category:OWASP_Guide_Project
[18] OWASP, "Top 10 vulnerabilities 2007", http://www.owasp.org/index.php/Top_10_2007
[19] "Top 100 Network Security Tools", http://sectools.org/
[20] "The DoJo Toolkit", http://www.dojotoolkit.org/
[21] Google Inc., "Google Web Toolkit", http://code.google.com/webtoolkit/
[22] Bad Store, http://www.badstore.net/
[23] Foundstone, McAfee, "Hacme Bank", http://www.foundstone.com/us/resources/proddesc/hacmebank.htm
[24] "Wapiti – Web application vulnerability scanner", http://wapiti.sourceforge.net/
[25] CIRT, "Nikto2", http://www.cirt.net/nikto2
[26] "Firebug", http://getfirebug.com/
[27] Internet2, "Shibboleth", http://shibboleth.internet2.edu/
[28] OpenID Foundation, "OpenID", http://openid.net/
[29] The Apache Software Foundation, "Apache AXIS2", http://ws.apache.org/axis2/
[30] Romanian Technical Military Academy, "Security Masters Program", http://www.mta.ro/masterat/masterinfosec/index.html