IDENTIFICATION AND INTEGRATION OF INFORMATION SECURITY TOPICS

Applied in a Web Application Programming Course

Justin Brown
KAP

Abstract: This paper discusses a unit of university study running within an Internet Computing degree aimed at second year undergraduate students. The unit is preparing for an update of the teaching material. Recent rises in security holes in web technologies has prompted the unit author to identify the critical security concepts that should be taught to web application developers and how such material can be integrated into a 12 week teaching schedule. Current course materials are examined, as are issues in teaching both theoretical and practical concepts of information security within a lecture/workshop arrangement.

Key words: Internet, World Wide Web, Application Development, Programming, Security, Hacking, Hackers, Vulnerability, Education, University

1. INTRODUCTION

Though the hype and financial impetus behind the World Wide Web have slowed in the wake of the 'dot com' collapse of 2000/2001, the requirement for tertiary level courses addressing this medium has not. Since 2000, as a part of an Internet Computing course, the School of Computer and Information Science at Edith Cowan University in Perth, Western Australia, has been running a unit of study called Interactive Web Development. This unit deals with both client and server side technologies, and requires students to exit the semester with identifiable skills in web application development.

Given the ever-changing nature of the Web and the technologies that drive it, this unit is obviously under constant revision, and though operating within its niche of web programming, the tools and technologies have

C. Irvine et al. (eds.), *Security Education and Critical Infrastructures*

changed markedly in only three years. As the unit is preparing for yet another curriculum update for semester 2, 2003, the issues of theory, and subsequent application of that theory must be addressed. Recently, while a list of new technologies for inclusion in the unit was being drawn up, it became apparent that there has been a more fundamental change in the environmental aspect of the web phenomenon, rather than a technical one. This change comes in the form of constant security threats against the web-servers that drive the web and the database-driven applications that provide the content. In an article discussing how to build more secure web sites, Microsoft authors Bollefer, et al, (2002, paragraph 1) highlight some of the most common vulnerabilities, including "cross-site scripting attacks, dynamic Web page source code disclosure, Web page defacement, posting malicious SQL commands to the databases, and theft of credit card data from the databases used". These are just some of the techniques used to exploit weaknesses in server configuration and application code, to say nothing of virus attacks and attempts to gain administrative access to server systems.

Considered the 'whipping boy' of lax web security in their server systems, Microsoft is now taking great pains to illustrate that their systems can be highly secure, but that "Every project is unique and will require work on the part of both the developers and the administrators to figure out the potential vectors of attack and how to guard against them" (Bollefer, et al, 2002, last paragraph). In the last quarter of 2002, the eWeek online technology news site ran a competition whereby Microsoft Corp and Oracle Corp were challenged to re-engineer a web application of eWeek's choice so as to make it as secure as possible using any platform and application development system the two companies preferred. Though the results were very detailed, as were descriptions of the test systems used, the interesting conclusion from this challenge revealed that;

> "Indeed, it's the little things that really matter in security: Each of the two successful cross-site scripting attacks was made possible by a single mistake on a single line of code in the test application" (Dyck, 2002. paragraph 5).

So, even though the servers and operating systems had been rigorously secured, a single mistake in coding practice allowed for an entry point into the system. It is exactly these types of security oversights that will, and must be, highlighted in the revamp of the Interactive Web Development unit. This author has been running web-servers in various fashions on various platforms since 1996, and until the year 2000/2001 experienced little or no problems with security or hacking. From early 2001, particularly with the coming of trojan attacks against insecure web-servers, the propensity for being 'hacked' in one way or another via the web has grown markedly, with

Trembly (2002, pg 18) stating that "security vulnerabilities have risen by 124 percent over the past two years".

Over the previous 18 months we have gone from a situation where regular updating of security patches to web-servers was an adequate defense to now requiring highly configured firewall solutions working in conjunction with patches to offer a modicum of protection. Though different operating systems and web server applications vary in their levels of vulnerability, the fact remains that web application developers must now realize that they need to be aware of these vulnerabilities, as they may be the person building and maintaining such servers.

2. UNIT STRUCTURE AND CONTENT

Currently, the unit provides instruction in interface design and implementation using client side technologies, such as HTML/XHTML/JavaScript/Macromedia Flash while focusing in the second half the semester on server-side technologies, such as scripting languages, web servers and databases. In the lecture program students are introduced to the evolution of the Internet and its' core technologies, such as HTML and web browsers. As the lectures progress the content becomes far more technical and covers server side coding solutions. Emphasis is given to how the scripting code works in conjunction with the web server in order to produce dynamically generated web pages. In the associated workshop or lab program, students actually develop client side code, then server side code in order to complete their assessments.

To date, there is a single lecture covering technologies involved in secure transactions, looking mainly at Secure Sockets Layer and Digital Certificates. Issues of public/private key management and basic concepts of cryptography are covered as well. Issues of web server security and possible vulnerabilities have not been covered in the program to date, aside from anecdotally, as the ability to perform 'hands on' web server construction in the workshop has not been available. The computers used by students in this unit are common across all courses within the School of Computer and Information Science, thus return to a default state upon rebooting. Along with the shared computing issue, the problem of security arises in that students in a web-connected workshop environment are not going to be given administrative rights over the machines, rights which are required to perform almost any change to the web servers' settings.

A solution of sorts has presented itself in the form of students downloading server software to their computers at home and running the systems there, with most students being able to dedicate a machine to that

task alone. The current server-side scripting languages taught within the unit are PHP and Perl, both of which currently run across all the major operating systems and with almost any web server. Students can download Apache server and PHP/Perl and run them on almost any version of Microsoft Windows, or use them built into most distributions of Linux. The fact that students now have access to powerful server and development tools within their own homes creates the opportunity to attack web development security issues in a practical, hands-on way. It should be stated that the requirement for running web development systems at home is not forced upon students, but has come as a result of students doing this independently, so that they can work at home and on campus. Additionally, it is a requirement of all students within the Internet Computing and/or Computer Science degree to have a modern personal computer available to them at home.

In fact, this focus on issues of web security is more a necessity than an opportunity as many students install the servers/scripting environments with their default settings, which usually means the web server starts during the system boot process. If the students or other members of their household connect to the Internet using this machine and are unaware of the security risks, they may inadvertently bring an attack upon themselves, perhaps without even realizing such.

3. POSITIONING OF WEB SECURITY CONTENT

As the teaching structure for Interactive Web Development currently stands, a 12-week teaching semester consists of 12 weeks worth of lectures, and associated workshops. In the first half of the semester, the lecture topics do not necessarily correlate to a direct workshop on that topic, though the lecture/workshop correlation increases markedly in the second part of semester. The structure as it currently stands looks like that shown in table 1.

Table 1: Current unit teaching structure

Week	Lecture	Workshop
1	Unit Introduction: Background to Internet and World Wide Web 1	Cascading Style Sheets and Macromedia Flash
2	Internet and World Wide Web 2	JavaScript Cookies and Flash
3	HCI and Dynamic HTML	Calculations and arrays in Java Script
4	Linking browsers to applications	HTML Forms
5	Linking browsers to Perl	CGI (Perl) interface
6	Manipulating browser data with Perl	Processing HTML with the Perl CGI module
7	Persistent data and Perl	Using the Perl DBI Module for RDBMS access
8	PHP – Linking browsers	Sessions and authentication with Perl CGI
9	Manipulating browser data with PHP	PHP basics
10	PHP persistent data and sessions	PHP and MySQL
11	Web Dev and Project Management	Marinating persistent data using PHP
12	Secure Online Transactions	Assignment work in labs

As the structure in Table 1 illustrates, the only current concession to security is that given in the lecture of week 12, with no associated workshop activity. When the new structure is created, the lectures dealing with web server security will be associated with a practical workshop where the student is tasked with performing some kind of security setup or auditing on their server. It is expected that the workshop element for the various security lectures will involve working on-campus in the computer labs, locating service and security patches for the web-server that the student runs at home, while also looking for known security weaknesses in their server/operating system environment. Students will take this material home and apply the updates/fixes as required, evidence of which can be brought to class in the form of screen shots or config files.

Making room in the current schedule for the new security content will be a challenge in itself, as those who have taught such units can testify. Most students would like to have nothing but 12 weeks of hands-on coding so that they can fully immerse themselves in the technology and become readily conversant in the development tools. However, as a more 'big-picture' approach is needed at a tertiary level, a balance must be identified between pure practical experience and an understanding of why the systems actually work as they do. Without a broader understanding of the underlying technology and the possibilities it presents, students are left with powerful construction tools but no idea what to build with them.

With this in mind, the unit as it currently stands needs to be re-developed in a way that allows for significant experience with the development tools, but also a non trivial interaction with web server setup, focusing on security weaknesses inherent in the way such systems operate. The order in which this is done is also going to be crucial, as issues of server security must be addressed early in the semester, as students tend to start working with the code at home well ahead of its place in the teaching structure. Logically, the workings of the web server need to be discussed followed by or, in conjunction with, the protocols that operate through the server and across the web. These protocols will be introduced early in the program and will replace the more introductory materials dealing with the history of the Internet.

Working on this basis, a decision has been made to drop three of the workshops pertaining to Perl, leaving only one such workshop as a demonstration of the language. The workshops will be 6 through 8. The lectures associated with these workshops will also be removed in favour of the new content, leaving a three week slice of the teaching schedule in which to deal with the topics of web server and application security. This gap will lead to the need for a program reshuffle, but before this reshuffle can take place, the key topics in web development security will need to be identified.

4. IDENTIFICATION OF KEY TOPICS

As has been touched on earlier, there are many aspects to web application / web server environments, and many vulnerabilities that can be exploited by those with the technical known how. The aims of the current unit re-write, and the focus of this paper, is to find the crucial topics in security that need to be covered in the context of the web developer. As there is a limited amount of teaching space in the lecture and workshop program to cover what is a very broad subject area, the issues in security need to be those that fit into two main criteria:

1. Does the topic fall into the category of required knowledge, one that is essential for the day-to-day development of web applications and management of web server systems?
2. Can the topic be taught in a theoretical and practical situation to the level where students actually synthesize and understand the concepts and the underlying logic?

Given these criteria and the amount of room available in the teaching program, it would seem prudent to concentrate on three broad areas within the subject area, those being;

1. Know the most common attack methods
2. Secure systems and code against these common attacks
3. Audit system regularly and know system environment thoroughly

4.1 Know the Most Common Attack Methods

Currently, one of the most relentless and annoying security concerns with the web is automated IP address and port scanning by hackers with varying motives. According to UK Security Online (2002, paragraphs 7-11) these motivations can vary from industrial spying, thrill-seekers, self-styled gurus seeking respect or out and out criminals.

These scans take the form of networking software automatically ICMP pinging known address ranges, like those used by ISPs, attempting to gain access to the machine at the other end of the connection should a return ping be sent. If the machine receiving the ping is running a web server, identified using a port scan on port 80, the scanning software at the other end can interrogate that server to identify the server application type. Once compromised, a server can be used to serve as a remote hacking platform to other networks, as entry point into a network or as a victim of defacement. Defacement attacks are usually perpetrated by the hacker gaining administrative rights to the server, then uploading new source files in place of the existing, legitimate content, or by altering the content of database driven websites. Jim Wagner gives some indication of the frequency of these attacks when he states:

> "In the past two weeks, Zone-H.org proprietor Roberto Preatoni said defacements have increased to more than 500 separate attacks a day and more than 1,500 over weekends. A year ago, he said, his site got around 30 to 50 defacement notices a day from hackers." (2002, paragraph 2).

4.2 Secure Systems and Code Against These Common Attacks

There are numerous methods available for limiting the ability of hackers to enter a system and perform some of the violations discussed above. As this paper is focusing on issues of security that can be taught and comprehended in a class of developers rather than security experts, the type

of solutions to these problems will be considered the most 'obvious' and 'minimum level'.

Below are the technologies and methodologies identified as being critical to a minimum level of defense for web applications and that also offer the possibility for integration into the learning process.

4.2.1 Firewalls and Protocalls

To begin with, students should be given a thorough understanding of TCP/IP and the most common protocols that sit on top of it, particularly http, FTP and SMTP. Along with a discussion of what these protocols do, a detailed description of the role of ports is also required. The reason for the focus on the protocols and their related ports is to create a foundation for instruction on the role of firewalls, software and hardware applications that can monitor protocols and ports in order to detect possible attacks. Most students understand that a firewall is a device that protects networks connected to the Internet, but little beyond this piece of common knowledge. By understanding the role of protocols in the functioning of the World Wide Web, students are better capable of understanding how firewalls actually work and their critical role in protecting web servers from unwanted attack. It is hoped that students will be able to identify the protocols that are necessary for the application they are developing and those that are not.

4.2.2 Web Server Setups

Anecdotal evidence from previous semesters teaching web technologies indicate that most students install pre-packaged server solutions, especially IIS under Windows 2000/XP Pro. Usually students select the basic install option, which usually includes an active SMTP and FTP server. As students do not understand what these systems do, they just leave them alone, running, with no tweaking of security settings. Given that a student develops an understanding of the role of such protocols, they can decide whether to keep them or not. Should they leave them enabled, they should then learn how to configure firewall applications in order to minimize chances of attack against them. In her example based on protecting university network systems, Olsen (2000, p. A40) eloquently states that

> "Universities should shut down services that they don't need to run. Network operating servers come, by default, with most services turned on when you install them. People need to configure these machines when they install them. If they absolutely don't need to have a particular service

running, they should shut it down, because if a service is running, it can be exploited."

A less is best approach will be instilled into students, along the lines of Olsen's beliefs, whereby http on port 80 is advised as the only open port to the outside world, given that this is the way web content is accessed and transferred. If FTP is essential for updates for web masters and web admins, it should be accessible only through the same domain (as is common with most ISPs), or through a secure connection, such as a VPN. If a web application requires the use of a mailing system using SMTP, that SMTP server should be limited to specific tasks, like only sending email within a set domain or disallowing bulk emails without verified addresses.

4.2.3 DMZs

Though it may be argued to be outside the purview of the budding web developer, the concept of network topology and domain numbering should be introduced for the purposes of discussing Demilitarized Zones (DMZs). A DMZ is used to separate a web server from the rest of the company/agency network, so that if the server is compromised, an access point to the rest of the network is not available. Once students understand that TCP/IP is a routable protocol, and that a method of translation is required when moving from one domain numbering system to another, they should be able to see the value of creating a special zone for servers only. At this point, the concept of using a hardware-based firewall/router can be introduced, so that students can see how a DMZ can be created between the outside world and the internal intranet without having to physically isolate the web server from the local network.

4.2.4 Patches

As well as running defense tools such as hardware/software firewalls, students need to be instructed on the importance of keeping both their operating systems and their web server system constantly up-to-date with security/application patches. When a security problem arises with a web server application, or an operating system or a server-side scripting tool, a patch is usually released by the software house that created the application. This patch will fix the current problem, and no further patches will be required until the next hole is found within the development environment. Students should become ingrained with the habit of checking for any type of update that is available for any of the tools they use in web development, particularly for web server and scripting applications. Checking for,

downloading and installing product updates is considered a tedious use of valuable time, and while once is it was considered more optional than necessary, it is now essential for the protection of any web-facing server system. Students will be instructed that a firewall is one piece of the puzzle for managing server security, but that to cover only one flank in a battle is a perilous way of working indeed. The importance of patching is reiterated by Goldsborough (2001, p 51) when he puts forward "Whatever security approach you take, keep current by installing patches and upgrades as they become available".

4.2.5 Code

The concepts above have dealt more with the systems that reside outside the direct experience of the web developer, as they are software/hardware settings rather than code. However, students must be taught that even though their firewall may be running nicely, and their systems are fully patched, the imperative is upon them to write scripting code that does not open a door for entry into the system. As http is always going to be necessary in order to serve web pages, it is most likely to offer the first point of attack. Students need to identify possible security weak spots in their applications and propose methods of reducing such weakness. Use of scripts which have admin level usernames and passwords is to be highlighted as poor practice, as is the use of file-upload scripts that have full read-write-execute access to the file system. Scripts that do not perform adequate data validation upon submission may cause buffer overrun errors, crash the server or even open a gateway into the server. Code that accesses SMTP applications needs to be configured in such a way that hackers cannot exploit the system to their own gain. For example, a page that executes calls to an SMTP system could be configured to only be accessible from the current ip address of the server machine, receiving the required instructions from another page that the user sees, thus putting a small extra barrier between system and potential attacker.

4.3 Audit Systems Regularly and Know System Environment Thoroughly

Perhaps the most overlooked and unexciting of all the protection methods for web development systems is the monitoring of logfiles, from web server activity logs to operating system logs. Most web servers will detail where each 'hit' has originated from (IP address), what time the hit was made and on what port and using what protocol. This information can be very informative, as it can indicate trends that may be someone trying to attack the system. A classic (which this author has experienced *ad nauseum*) is a

remote host attempting to log onto port 21 utilizing FTP, sending various combinations of usernames and passwords, hoping to find one with administrative rights. If the IP address of this individual is constant, indicating they are using a machine with a fixed IP, that IP number can be placed in an inclusion list of banned addresses, placed in either the web server if it supports this function, or even better, within the firewall's blocking list.

Unexpected changes to files are another indication that perhaps someone has illicitly gained access to the system and are altering content. Most file systems will allow a file listing that can be sorted according to when a file was last changed. If such a change occurs without the web developer actually making the change, investigation is warranted. This is obviously extremely difficult for large websites, especially where the content is dynamically generated from database systems. In order to make this problem more approachable, it may be decided that only certain files need to be closely monitored, such as server-side include type files which contain large number of code functions or configuration information for database connections. Though these files are not supposed to be accessible from sources external to the server, the assumption must be made that they are vulnerable. Software tools can be used to monitor changes to files, similar to the way most virus checkers work, though once again, maintaining change-tracking for large numbers of files on a big site is a time consuming, usually expensive undertaking.

By having a complete understanding of the current setup and structure of the web development environment, such as the operating system, web server, scripting tools, databases, user accounts and number of files/total size of site, the web developer is in a far better position to see where undesirable changes have occurred. If a suspected hack has occurred the web develop should dive straight into the log files for firewall/os/web server, at which point the breach should become apparent. Students should learn that these basic methods of problem monitoring and identification should be as much a part of web application development as writing code. Of course, choosing what to monitor requires some thought, as Posey (2002, paragraph 9) justifiably states

> "It's better to just audit administrative actions such as account creations, password changes, and the assignment of permissions. Whatever you choose to audit, be sure to review the audit logs daily or even more often if you detect any suspicious activity."

5. INTEGRATING THE SELECTED TECHNOLOGIES INTO TEACHING PRACTICE

Having identified the topics considered to be required for teaching in the context of web development, the problem of integrating these technologies into the current course setup still remains. As well as the theoretical underpinnings, students need to be exposed, where possible, to practical examples of this security theory.

In order to achieve this, the unit, which as stated earlier is in need of an update, is to be re-structured quite rigorously, with some content being placed in a different part of the program, while other content, such as bulk of the Perl materials, will be discarded from the current teaching curriculum. As well as introducing the much needed materials on web development security, the course will be changing the base markup language to XHTML, whilst also introducing some materials on database design and analysis, focusing on practical database structures for web applications.

Some content regarding security and database systems will be introduced here, further enhancing the point of thinking about security in all aspects of web design. Table 2 on the following page contains the new-look structure for the unit.

Table 2: Proposed new unit teaching schedule inclusive of security content

Week	Lecture	Workshop
1	Unit Introduction: Background to Internet and World Wide Web	Introduction to XHTML
2	Markup languages (HTML and XHTML) and the Document Object Model	XHTML and Cascading Style Sheets
3	Project Management for Web development	JavaScript Cookies
4	Internet / Web protocols and addressing systems	HTML Forms
5	Dynamic content generation – the role of server-side scripting languages	Processing XHTML with the Perl CGI module
6	Web servers and web server configuration: practice, problems and vulnerabilities	Examination and customization of web server configuration files
7	Enhancing web application security – the role of firewalls, smart coding and system auditing	Identification of security loopholes and possible solutions
8	Database design considerations – efficiency – reliability – scalability and security	Application of database design principles
9	PHP – Linking browsers	PHP basics
10	Manipulating browser data with PHP	PHP and MySQL
11	PHP persistent data and sessions	Marinating persistent data using PHP
12	Efficient coding practice and code reusability	Assignment work in labs

Instead of concentrating the security topics into the newly created three week gap left by the Perl materials, the topics of security will be spread across 5 weeks, being integrated into the theoretical lecture materials, starting with week four. The proposed changed content and teaching methods are as follows;

Week:

1. LECTURE: Will cover the primary protocols used throughout the web and how they work in conjunction with the IP numbering system that allows for information transfer. Concepts such as different domain numbering systems and protocol port numbers will be addressed – WORKSHOP: Will be a normal markup language tutorial with no direct relation to the lecture material
2. LECTURE: Introductory information regarding the concepts of client-side vs server-side technology will be presented, along with in-depth discussion as to why server-side systems offer such functionality, yet such security risks – WORKSHOP: Will be a Perl code dynamically generating an XHTML document exercise, with a focus on how the server-side code produces client-side code for transport via the protocols discussed in week 4. The lab will end with a discussion of the role of the CGI-BIN directory with security.
3. LECTURE: The general functions of common web servers will be discussed, with in-depth analysis of all the key system settings, such as FTP (where the server supports it), SMTP, Sessions, Cookies, Timeouts, Root folder locations, Domain restrictions, Scripting Language support, Log file creation, File Upload capabilities and permissible user accounts – WORKSHOPS: Students will be shown a sample config file from a publicly available web server such as Apache. Students will be asked to change the config file to meet various criteria, such as restricting CGI-BIN to a specific folder, or allowing script-only processing, not executables. Student's work will be examined in class, and where possible, tested on a 'sacrificial' server.
4. LECTURE: This lecture will be a 'nuts and bolts' approach, illustrating to students many of the known attack methods and how they can be defended against. Examples from the press will be highlighted, as will the importance of firewalls, server and OS patching and writing secure code – WORKSHOP: Students will be given a number of scenarios, such as Company A wishes you to develop an application that performs this task using this database back-end and this development environment. Students will need to draw up a plan on the key technologies they will need to develop the application, what protocols they will need, what the level of security will need to be and how to best deliver that level of security.

5. LECTURE: Though this lecture will be based mainly on the concepts of database design and normalization, some mention will be made on issues in database security, such as encryption, access permissions, connections to and possible attack points depending upon the database in question – WORKSHOP: Students will actually begin designing the database system for their assignments, though the ability for them to create security accounts is yet to be seen.

The above breakdown of the new teaching materials should indicate that the course has gone from almost no reference to issues of security, say for a single lecture at the end of semester, through to a relatively thorough and hopefully practical exploration of key security topics.

6. CONCLUSION

Most educators, especially educators of technology related content, wish they could fit into a schedule all the key points that they, from personal and professional experience, feel are essential to the student's understanding of the content. For the most part this is not possible, though it is perhaps possible to break these points down into a list of the absolutely necessary. It is hoped that this paper has shed some light on the thinking involved when re-engineering a tertiary web development unit to focus more heavily on the topic of security, for it is a topic that now falls into the category of 'essential'. In the same way that building architects must consider issues of safety and security, so must the developer of web applications, whatever their level of involvement in the process.

REFERENCES

1. Bollefer, T, Chander, G, Johansson, J, Kass, M & Olson, E. (2002). Building and Configuring More Secure Web Sites. *Microsoft Corporation*. Retrieved December 29, 2002, from http://msdn.microsoft.com/library/en-us/dnnetsec/html/openhack.asp
2. Dyck, T. (2002). OpenHack Wrap. *eWeek*. Retrieved January 5, 2003, from http://www.eweek.com/article2/0,3959,743411,00.asp
3. Goldsborough, R. (2001 Nov 22). Protecting yourself against cyberterrorism. *Black Issues in Higher Education*, 18(20), p52. Retrieved December 19, 2002, from ProQuest Database.
4. Olsen, F. (2000). Logging in with...Thomas J. Talleur. *The Chronicle of Higher Education*, 46(45), p A40. Retrieved December 27, 2002, from ProQuest Database.
5. Posey, B. (2002, August 29). Basic strategies for securing Internet Information Server, CNETAsia. Retrieved December 30, 2002, from http://asia.cnet.com/itmanager/trends/0,39006409,39073946-2,00.htm
6. Trembly, A. (2002 Dec 16). Security woes go from bad to worse. *National Underwriter*, 106(50), p18-21. Retrieved January 4, 2003, from ProQuest Database
7. UK Security Online. (2002). Hacker Threat Analysed. *UK Security Online*. Retrieved January 3, 2003, from http://www.uksecurityonline.com/threat/hackers.php
8. Wagner, J. (2002 Oct 22). Web Vandalism on the Rise. *Internetnews.com*. Retrieved December 15, 2002, from http://www.Internetnews.com/dev-news/article.php/1485601