Security and Computer Forensics in Web Engineering Education

William Bradley Glisson, HATII, University of Glasgow, 11 University Gardens, Glasgow, G12 8QQ, United Kingdom b.glisson@hatii.arts.gla.ac.uk

Ray Welland, Department of Computer Science, University of Glasgow, Sir Alwyn Williams Building, Glasgow, G12 8QQ, United Kingdom ray@dcs.gla.ac.uk

L. Milton Glisson, School of Business and Economics, N.C. A&T State University, 1601 East Market Street, Greensboro, NC, 27411, USA glissonm@ncat.edu

Abstract.
The integration of security and forensics into Web Engineering curricula is imperative! Poor security in web-based applications is continuing to cost organizations millions and the losses are still increasing annually. Security is frequently taught as a stand-alone course, assuming that security can be 'bolted on' to a web application at some point. Security issues must be integrated into Web Engineering processes right from the beginning to create secure solutions and therefore security should be an integral part of a Web Engineering curriculum. One aspect of Computer forensics investigates failures in security. Hence, students should be aware of the issues in forensics and how to respond when security failures occur; collecting evidence is particularly difficult for Web-based applications.

Keywords: Web Engineering, Security, Computer Forensics

1 Introduction

The integration of security and forensics into Web applications is imperative! Deloitte's latest survey indicates that organizations are becoming more frugal, more demanding and more cynical through a reduction in spending and reliance on metrics when it comes to the implementation of security [1]. The latest report form PricewaterhouseCoopers (PwC) echoes this message indicating that "security spending is under pressure. Most executives are eyeing strategies to cancel, defer or downsize security-related initiatives"[2]. The Internet Crime Complaint Center (IC^3) Report indicates that complaints increased by 22.3% in 2009 as compared to 2008. This translates into a total dollar loss from referred cases to be $559.7 million, more than doubling the 2008 loss of $246.6 million [3]. A substantial portion of the complaints had to do with some form of fraud [3].

According to the UK Cards Association “Online banking losses totaled £59.7 million in 2009 – a 14 per cent rise on the 2008 figure.” They go on to indicate that “This increase is largely due to criminals using more sophisticated methods to target online banking customers through malware, which targets vulnerabilities in customers’ PCs, rather than the banks’ own systems…” [4]. This indicates that criminals are, possibly, becoming savvier in their attacks and that Web engineering needs to expand its scope to cover the entire transaction process; not just specific systems. Web Engineering has been defined as:

> “the application of systematic, disciplined and quantifiable approaches to development, operation, and maintenance of Web-based applications” [5, 6].

It is important to recognize that previous definitions of Web Engineering do not inherently make any direct references to security or forensics, consequently, today’s Web applications face increased susceptibility to major security problems. This information highlights the need for academic institutions to integrate security and computer forensics’ concepts and practices throughout Web engineering curricula.

2 Curricula

The idea for the implementation of security into systems has been around for a while as witnessed by the creation of a number of Information Assurance academic programs in the United States [7]. Several of these programs are listed on the National Security Agency’s (NSA) Web site for Centers of Academic Excellence [7]. Papers have been published on the implementation of security curricula [8] and the integration of security ethics [9] into education. They have also been published on laboratory based solutions that implement information security knowledge [10] and Internet security [11] into education. However, these approaches offer very broad solutions to the security problem. Universities that are specifically offering curricula in Web Engineering should address business, security and forensics needs throughout all aspects of the Web Engineering curriculum.

From a Web Engineering perspective, an understanding of security needs to be established that includes how security has evolved, the legislation, regulation and certifications that impact security. It includes the implementation of security from a methodology perspective and discussions about how security fits into different application development methodologies. It would also need to address practical lab based implementation scenarios that reinforces concepts put forth in lectures.

There are a limited number of courses being offered in Web engineering and few complete programs to-date. On the postgraduate level, the University of Western Sydney offers a track in a Masters program [12]. Deshpande, from the University of Western Sydney, has put forth ideas for a Web Engineering curriculum and has proposed six levels of complexity that need to be addressed in the curriculum [6]. However, it can be argued that there is a seventh level of complexity that needs to be explicitly addressed and that is Web Engineering Security! Whitehead [13] proposed a curriculum for a masters program in Web Engineering. However, he does not discuss, at any point in the paper, security.

North Carolina A&T State University has proposed a complete undergraduate program in Web Engineering [12]. However, to-date from their Web site this program does not appear to be currently available. The paper that they put forth to propose the program does mention security twice. The first occurrence is in the class description for the Introduction to Web Engineering where they state that students will learn "how to incorporate security feature(s) into web sites" [12]. They also propose a specific course on 'Trust and Security'[12]. Their proposal focuses more on the social aspects of the program and the potential benefits for attracting a diverse student body as well as increasing overall student numbers [12]. There appears to be a lack of in-depth security integration throughout the program. A quick search on the Web reveals a Web Engineering 'suggested' program of study for a Bachelor of Science in Applied Information Technology at Kentucky State University [14]. The program introduces security in the senior year through a course titled 'Information Security' [14]. The lack of information security topics throughout both undergraduate curricula is worrisome at the very least!

The reality in today's increasingly competitive academic environment is that courses need to be utilized as much as possible. Curricula need to be introduced and implemented so that they take advantage of existing infrastructure. A dedicated lab for a forensics or security course can also be utilized by the other programs. These facilities also can be used to implement specific courses in related areas like Web Engineering undergraduate and postgraduate degrees. The topics complement each other very nicely. Course work can be constructed so that students learn how to create viruses, trojans, and worms for the security related courses. The same students can then learn how to develop and integrate code into Web engineering projects that will identify these threats and log actions appropriately for the forensics course.

The trick is to make the learning environment fun for the students and beneficial for employment opportunities once the course is complete. This has to be balanced with University requirements. These requirements would include appropriate security measures to ensure that code developed in the security lab is not allowed to be introduced to the outside world. This could include measures like dedicated labs, swipe card entry, video surveillance, and policies that restricting devices that are brought into the lab. Additional measures would include the removal of USB ports and outwardly facing drives. To help mitigate worst case scenarios, additional software measures could also be introduced that limits the life span of any code developed in the lab.

Regardless of how security is implemented in a specific program, the initial problem with tackling security is the terminology. Terminology in various environments has the potential to have multiple meanings. As Anderson indicated, reality is a complex environment in the real world [15]. Hence, what the terms security and vulnerability mean to one organization, such as a large financial institution, may or may not have the same relevance to another business, such as a newsagent or a small legal firm. Logically, different organizations will require "some combination of user authentication, transaction integrity and accountability, fault-tolerance, message secrecy and covertness” [15]. So what is the definition of security?

3 Security Definition

In this paper, we will define a Web enabled secure system in terms of well established security concepts which consist of confidentiality, integrity and availability [16, 17]. The Web Engineering solution should protect confidentiality by limiting access to the appropriate individuals [18]. This would involve user identification, authentication and authorization. The integrity of the system should be maintained by only allowing modifications to be conducted by the appropriate individuals and within established guidelines [18]. The availability of the system is defined by providing access to the appropriate parties at designated times [18]. It should be noted that there are two additional categories that are commonly included when discussing security and they are ‘non-repudiation’ and ‘accountability’. Non-repudiation is the capability to prevent, in this case, a software user, a system, or an application from denying actions they have performed. Accountability is the recording of the software user’s actions. Since “accountability includes authenticity and non-repudiation” [19] and authenticity is the “property that allows the ability to validate the claimed identity of a system entity” [19], i.e., the authentication aspect, we will consider these topics to be subtopics of confidentiality that are utilized to help ensure integrity.

Vulnerabilities will be defined using The Organization for Internet Safety (OIS) definition. It has been said that “security is about preventing adverse consequences from the intentional and unwarranted actions of others” [20]. OIS publishes Guidelines for Security Vulnerabilities Reporting and Response. In this document, security vulnerability is defined as

> “a flaw within a software system that can cause it to work contrary to its documented design and could be exploited to cause the system to violate its documented security policy” [21].

It should be noted that this statement makes the assumption that a documented security policy exists. The reality of the OIS vulnerability definition is that any flaws in the system design or application coding can potentially lead to security vulnerabilities. The need to improve security in the Web application development is reinforced by testimony from Robert F. Decay, Director, Information Security Issues indicating that patch management is critical in mitigating cyber vulnerabilities [22]. According to the same report, the number of security vulnerabilities reported is increasing and attacks are becoming automated [22]. Software security encompasses more than encryption and password maintenance. The ability to defend against software attacks, in the long run, will need to come from “more rigorous software engineering practices, better tools and technologies” [22].

Using these broad definitions to understand security supports the idea that security means more than implementing encryption, Secure Socket Layer (SSL), firewalls and creating and maintaining secure networks [23, 24]. It is also more than the use of digital certificates, the different technologies used for authentication and authorization or intrusion detection systems [23, 24]. In-depth discussions on these topics and research into their improvement are occurring on a daily basis. However, a system's security is not determined solely by the technology that is implemented. Web security is determined by a number of factors that include legal issues, social issues, technical issues, and Web engineering practices. This expansive perspective on the scope of security was reinforced by Eugene Spafford, a security expert and professor at Purdue University when he stated in an interview that "security is a total-picture issue, not a set of spot problems to patch" [25].

4 Security Literature

In order to incorporate security into a Web Engineering curriculum, it is necessary to appreciate the current state of security methodology research and to acknowledge previous research in the field of information security design methods [17]. Baskerville's analysis separated numerous system methods into three generations [26]. The first generation consisted of check lists and risk analyses. This stage focused on actual physical systems specifications. The second generation engineering methods focused on complex customization through the use of engineering concepts and mechanistic procedures that relied heavily on functional requirements.

Even though Baskerville's analysis of the security design methods did not directly examine the applicability of the security methodologies to Web development, he did make an important point that is applicable to Web Engineering application development. Baskerville's analysis did suggest that

> "systems methods will neither be trustworthy nor successful unless the general research regarding systems methodology incorporates security analysis design as an explicit objective" [26].

Siponen updates and expands on Baskerville's analysis of information security development approaches declaring that there are five information system security generational classifications [27]. Siponen arrives at his conclusion after an examination of the contributing research disciplines and an evaluation of seventeen modern information system security methodologies. Security is a highly diverse research subject that has been an area of interest for a variety of disciplines. Siponen identifies four research communities as contributors to information security research including Management Information Systems (MIS), computer science, software engineering and mathematics.

Siponen's first three generations correspond with Baskerville's generational classifications. Siponen defined the third generation as consisting of structural and object-oriented security methods, information modelling methods, and stepwise security methods. According to Siponen, the fourth generation builds on the third generation by addressing the social and socio-technical aspects of the methods. The fifth generation, of security methodologies, that Siponen discusses [27] is really the next generation of methodologies. This implies that the fifth generation security methodologies do not currently exist, a point which he also articulates in a later article [28]. Siponen's points, regarding the fifth generation, bring us to the heart of the security problem. There have been few industrial attempts to comprehensively address user focused aspects; methodology integration; practitioner malleability and employment of Web engineering security throughout the Web-based application development process via the establishment of a comprehensive security methodology. One industrial solution is the Web Engineering Security methodology (WES). WES is a proactive, flexible, customizable, process neutral security methodology that is based on empirical evidence [17, 29, 30]. The natural question that arises when you are discussing security is what happens when it fails?

5 Web Forensics Information

Security and forensics are two sides of the same coin. Security tries to prevent undesired things from happening while forensics acknowledges that something has happened and attempts to prove it through the evidence that is left on machines, networked devices and/or mobile devices.

Appreciating that security is a broad concept that needs to be covered in its entirety ushers in the concept of educating people to prepare and handle situations when things go wrong. In order to accomplish this integration, forensics needs to be proactively integrated into software development methodologies. Depending on the needs of the individual organizations, this can include the capturing of necessary log information, network packets and mobile device information. It can also include the proper training for personnel on how to handle first responder incidents [31]. It is realistic to perceive a situation where a graduate from a Web Engineering program is a first responder to a potential criminal situation. How do they handle the situation? What can they do or not do that will preserve the evidence? This necessitates a basic understanding of computer forensics principles, techniques, and processes.

In a Web based environment, students should be taught about different operating and file systems and how they store potentially relevant data from the client and the server side. As an example, a Windows registry can store information pertaining to previously visited Web sites, to search queries, and passwords. They should also be informed about the types of information that can potentially be located on networks and networked machines. What are the ethical issues that students need to consider?

To complicate matters, Cisco’s latest prediction is that global IP traffic will get to 667 Exabyte’s by 2013 [32]. Out of all of the data that is being passed around the Internet, what information do you need to keep and for how long? What are the legal implications with maintaining this information? Hence, students need to understand the relevance of this information in a court of law and the legislative issues that complicate jurisdictional rights in a global environment. Relevant legislation has been discussed by Glisson et al. [33]. How does this and other legislation impact the design and implementation of Web engineering systems or the extraction of data in an investigative situation?

The need for security and forensic integration into Web Engineering is highlighted with the emergence and high rate of acceptance currently demonstrated with cloud computing. Hence, a Web Engineering curriculum needs an in-depth exposure to operating and file systems, networks, dynamic memory, legal and ethical implications, mobile devices and basic digital forensics concepts and procedures.

6 Security and Forensics Curriculum Integration

The integration of these concepts into a Web Engineering curriculum is challenging. There are so many aspects of both security and digital forensics that need to be discussed through out the implementation of the program that it makes covering all of them to any depth difficult. This issue is highlighted in Table 1- Curriculum Integration. The courses and the year were taken as an example from the Bachelor of Science in Applied Information Technology at Kentucky State University [14]. The corresponding security and forensic topics are offered as a guide in corresponding classes and are not meant to be a definitive solution.

Table 1. Curriculum Integration

Courses	**Security Topics**
Freshman Year	
Programming Concepts	• Authentication and Authorization • Public Key Cryptography
Computer Hardware	• Computer Architecture & Protection Mechanisms
Sophomore Year	
Advanced Programming Concepts	• Access Control Techniques & Administration • Identification & Authentication Techniques • Practical implementation of these techniques • Application layer security protocols • Building in investigative tools and data capture
Data Communication Technology	• Advanced cryptography concepts • Working with data at the hexadecimal level
Database Management Systems	• Understanding access controls for specific DBMS • Authorization and the need for auditing
Unix Network Programming or Router Theory and Configuration	• Compromising a Unix Host • Investigating a Unix host & understanding protocols
Junior Year	
Network Operating Systems	• Understanding of the different devices in a network and the data captured by each device • Dynamic Host Configuration Protocol configuration & security
Adv. Databases & Data Mining	• Access Control • Types of attacks (SQL injection) • Investigation approached & Data mining for evidence

Internet / Intranet Administration	• Firewalls (Packet Filtering) & DMZs, • Internet Security applications • Security Architecture & Design
Advanced Server Administration	• Operating system security issues
Storage Area Networks	• Security and investigative issues with the cloud
Senior Year	
Intro to Client / Server	• Email Security
Web Engineering	• System development life cycle • Critique the Security of Web Engineering methodologies (SCWAD) • Hardening of Web, Email, FTP, DNS servers etc.
Information Security	• General Security concepts • Security Management Concepts (social engineering, ROI, Patch Management)
Advanced Client / Server	• Understanding the risk to workstation, servers, transmission, and network components.

One solution that has been developed to critique the security of Web Engineering development methodologies is The Security Criteria for Web Application Development (SCWAD) [17, 30]. The goal of any examination should be to highlight the lack of security in the processes and initiate discussions applicable to Web Engineering curricula. In other words, do security methodologies effectively build security into development methodologies? SCWAD attempts to achieve this goal by addressing the following criteria:

1. Active organizational support for security in the Web development process
2. Proper Security Controls in the development environment
3. Security visibility throughout all areas of the development process
4. Delivery of a cohesive system, integrating business requirements, software & security
5. Prompt, rigorous security testing and evaluation
6. Trust and Accountability

7 Conclusion

With the blatant need for improved security, coupled with the increasing implementation of security metrics in industry, the need to address security and forensics throughout Web Engineering curricula is critical. This includes addressing everything from an in-depth understanding of the concept of security, to hardware implications, to secure application development and to the legal, as well as, ethical implications associated with Web Engineering.

As the US Department of Homeland Security has stated “there is nothing inherently ‘security-enhancing’ about most development methodologies”[34].

Developing applications and understanding the interactions from a security and a forensics perspective is critical from a Web Engineering curriculum perspective. These concepts should be integrated throughout a Web Engineering curriculum.

As the digital revolution continues to saturate societies and these devices continually become more networked, the need to address security and forensics in Web engineering curricula will continue to be a critical issue. Future work should explore the impact of cloud computing on practical real-world implementations of security and forensics in the realm of Web Engineering. It should focus on the integration of security and forensics concepts throughout the academic curriculum; not via a single class or set of lectures.

8 References

1. Deloitte (2009) *Losing Ground 2009 TMT Global Security Survey Key findings*. https://www.deloitte.com/.
2. PricewaterhouseCoopers, *Trial by fire*. 2009.
3. Internet Crime Complaint Center, *2009 Internet Crime Report*. 2010.
4. The UK Cards Association. http://www.theukcardsassociation.org.uk/media_centre/press_releases_new/-/page/922/.
5. Deshpande, Y., et al., *Web Engineering*. Journal of Web Engineering, 2002. vol.(No. 1): p. 3-17.
6. Deshpande, Y. *Web Engineering Curriculum: A Case Study of an Evolving Framework*. in *Web Enginering 4th international conference, ICE 2004*. 2004. Munich, Germany.
7. National Security Agency. http://www.nsa.gov/ia/academic_outreach/nat_cae/institutions.shtml.
8. Crowley, E., *Information system security curricula development*, in *Proceedings of the 4th conference on Information technology curriculum*. 2003, ACM: Lafayette, Indiana, USA.
9. Dark, M., et al., *An information security ethics education model*. J. Comput. Small Coll., 2008. vol.(6): p. 82-88.
10. Elitzur, R.,Sai, Y., *A Laboratory Study Designed for Reducing the Gap between Information Security Knowledge and Implementation.* International Journal of Electronic Commerce Studies, 2010. vol.(1): p. 13.
11. Mateti, P., *A laboratory-based course on internet security*, in *Proceedings of the 34th SIGCSE technical symposium on Computer science education*. 2003, ACM: Reno, Navada, USA.
12. Esterline, A. C., Williams, K. A.,Carr, E. C. http://redux.comp.ncat.edu/carr/web_engineering/SIGCSE_Web.pdf.
13. Whitehead, E. J., *A PROPOSED CURRICULUM FOR A MASTERS IN WEB ENGINEERING*. Journal of Web Engineering, 2002. vol.(1): p. 5.
14. Kentucky State University. http://www.kysu.edu/.
15. Anderson, R., *Security Engineering: A Guide to Building Dependable Distributed Systems*. 2001, New York: John Wiley & Sons, Inc.
16. Hansche, S., Berti, J.,Hare, C., *Official (ISC)2 Guide to the CISSP Exam*. 2004, Boca Raton: Auerbach.

17. Glisson, W. B., *The Web Engineering Security (WES) Methodology*, in *Department of Computing Science*. 2008, University of Glasgow: Glasgow. p. 245.
18. Pfleeger, C. P.,Pfleeger, S. L., *Security in Computing*. Third Edition ed. 2003, Upper Saddle River, NJ: Prentice Hall.
19. Krutz, R. L.,Vines, R. D., *The CISSP and CAP Prep Guide*. 2007, Indianapolis, IN: Wiley.
20. Schneier, B., *Beyond Fear: Thinking Sensibly About Security in an Uncertain World*. 2006, New York: Springer-Verlag New York Inc. 303.
21. Organization for Internet Safety. http://www.symantec.com/index.jsp.
22. Dacey, R. F., *INFORMATION SECURITY Effective Patch Management is Critical to Mitigating Software Vulnerabilities*, in *Testimony Before the Subcommittee on Technology, Information Policy, Intergovernmental Relations, and the Census, House Committee on Government Reform*. 2003, United States General Accounting Office.
23. Dickson, J. B., *Web applications have become IT's next security battleground.* San Antonio Business Journal, 2004. vol.
24. Ellis, J.,Speed, T., *The internet security guidebook: from planning to deployment*, ed. E. Carrasco. 2001, San Diego: Academic Press. 1-320.
25. McCormick, J. http://www.baselinemag.com/article2/0,1397,2152093,00.asp.
26. Baskerville, R., *Information systems security design methods: implications for information systems development.* ACM Computing Surveys, 1993. vol.(4): p. 375-414.
27. Siponen, M. T., *Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods*. 2004, Department of Information Processing Science, University of Oulu: Oulu, Finland. p. 37.
28. Siponen, M. T., *Secure-System Design Methods: Evolution and Future Directions.* IT Professional, 2006. vol.(3): p. 40-44.
29. Glisson, W. B.,Welland, R. *Web Engineering Security: Essential Elements*. in *The Second International Conference on Availability, Reliability and Security (ARES)* 2007. Vienna, Austria: IEEE.
30. Glisson, W. B., McDonald, A.,Welland, R. *Web Engineering Security: A Practitioner's Perspective*. in *International Conference on Web Engineering*. 2006. Palo Alto, California: Springer.
31. Hoolachan, S.,Glisson, W. B. *Organizational Handling of Digital Evidence*. in *The 2010 ADFSL Conference on Digital Forensics, Security and Law*. 2010. St. Paul, Minnesota, USA: Association of Digital Forensics, Security and Law.
32. Cisco. http://newsroom.cisco.com/dlls/2009/prod_060909.html.
33. Glisson, W. B., Glisson, L. M.,Welland, R. *Secure Web Application Development and Global Regulation*. in *The Second International Conference on Availability, Reliability and Security (ARES)* 2007. Vienna, Austria: IEEE.
34. Department of Homeland Security, *Security in the Software Lifecycle*. 2006, Department of Homeland Security: Washington, DC.