DESCRIPTION OF THE RESEARCH PROJECT FOR THE 2019 SUMMER RET SITE

Project #: Secure Software Development

Area Coordinator:
Dr. Nan Niu
Associate Professor
Department of Electrical Engineering and Computer Science
College of Engineering and Applied Science
University of Cincinnati
Cincinnati, OH 45221-0030
Office: 832 Rhodes Hall
E-Mail: nan.niu@uc.edu
Phone: (513) 556-0051

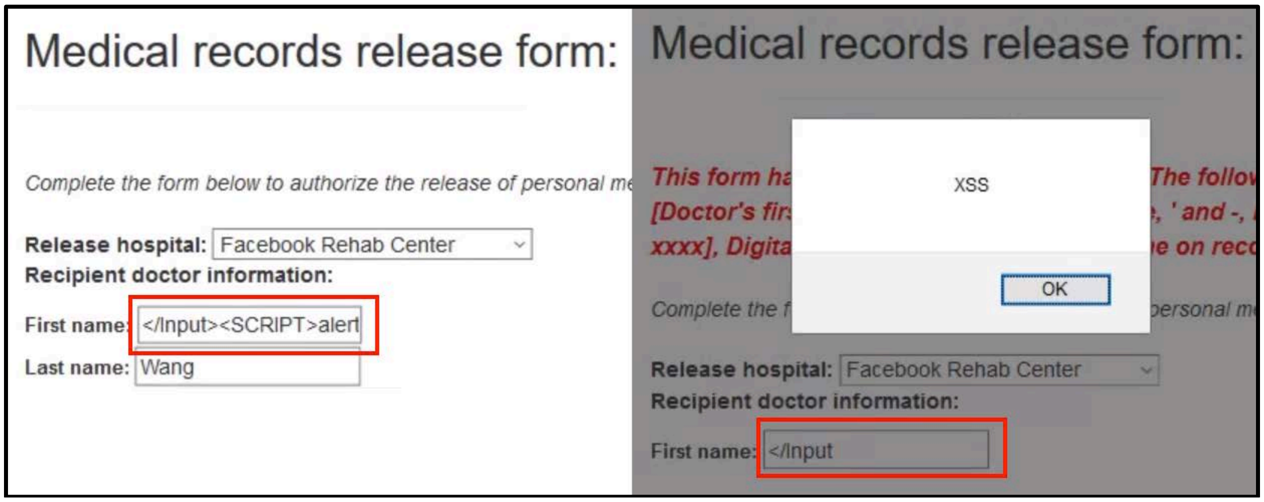
Sub-Area Coordinator:
Dr. Boyang Wang
Assistant Professor
Department of Electrical Engineering and Computer Science
College of Engineering and Applied Science
University of Cincinnati
Cincinnati, OH 45221-0030
Office: 806A Rhodes Hall
E-Mail: boyang.wang@uc.edu
Phone: (513) 556-4785

Graduate Research Assistant:
Mr. Wentao Wang
Ph.D. Candidate in Electrical Engineering and Computer Science
Office: 527 Engineering Research Center
E-Mail: wang2wt@mail.uc.edu
Phone: (513) 302-4797

Project Summary

This research topic is inspired by the serious cross-site scripting (XSS) vulnerabilities observed in web application [1, 2], and linked to the ***big idea*** of securing the confidentiality, integrity, and availability (CIA) of information assets. On eBay alone, a 2014 report shows that nearly 150 million were breached due to attacks like XSS [3]. Due to the vulnerabilities in eBay and other sites, Juniper Research projected that the data breach costs will skyrocket to $2 trillion in 2019 [4].

Figure 1 illustrates the XSS vulnerability which allows attackers to inject client-side scripts into webpages viewed by other users. The application is called iTrust [5], which is a healthcare web application aimed to help patients to keep up with their medical records and to communicate with their doctors. When filling out the medical records, a malicious user input, “</Input><SCRIPT>alert(‘XSS’)</SCRIPT>” entered to the “First name” field shown in the left of Figure 1, is not handled by iTrust. As a result of this security bug (vulnerability), the script causes iTrust to pop up an “alert” window shown in the right of Figure 1, demonstrating a specific XSS attack successfully exploited to iTrust. The example of Figure 1 is known as a *first-order XSS* attack, which typically impacts an individual user session and further requires the attacker to use social engineering to convince a victim to click on a (disguised) link that contains malicious HTML/JavaScript code [6].



Medical records release form:

Complete the form below to authorize the release of personal me

Release hospital: Facebook Rehab Center

Recipient doctor information:

First name: </Input><SCRIPT>alert

Last name: Wang

Medical records release form:

XSS

OK

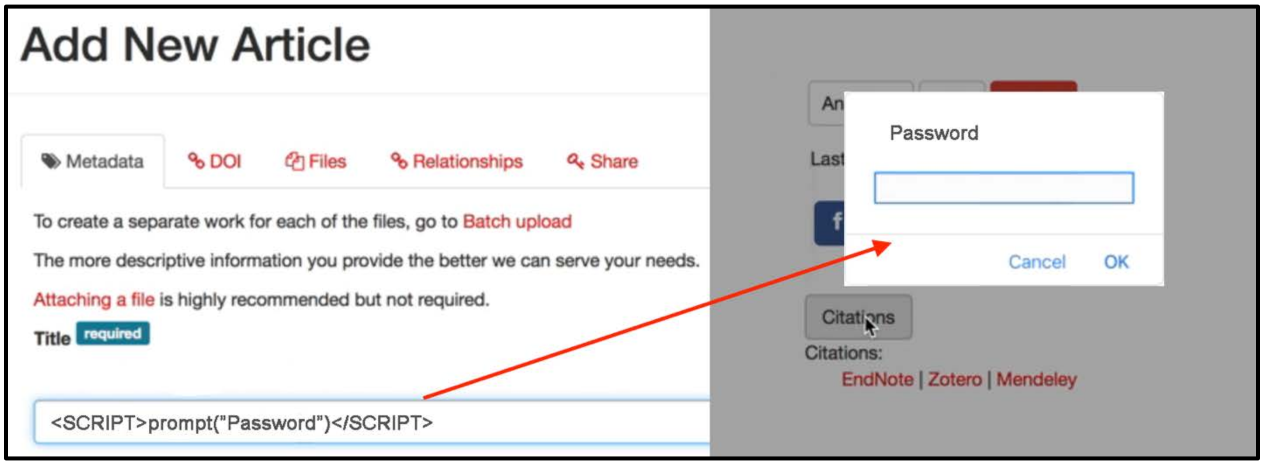
Release hospital: Facebook Rehab Center

Recipient doctor information:

First name: </Input

Figure 1 First-order XSS (cross-site scripting) vulnerability resulting from the iTrust application inserting the user’s input in the next webpage that it renders.

In contrast to the first-order XSS, a *second-order XSS* (also known as persistent or stored) vulnerability results from the application storing (part of) the attacker’s input in a database, and then inserting it in an HTML page that is displayed to multiple victim users [6]. Figure 2 shows an example of a second-order XSS exploited to Scholar@UC (https://scholar.uc.edu/), which is an institution-wide self-submission system enabling the UC community to share research and scholarly works with a worldwide audience.



Add New Article

To create a separate work for each of the files, go to Batch upload

The more descriptive information you provide the better we can serve your needs.

Attaching a file is highly recommended but not required.

Title required

<SCRIPT>prompt("Password")</SCRIPT>

Password

Figure 2 Second-order XSS (cross-site scripting) vulnerability resulting from the Scholar@UC application inserting the user's input into the database.

In the example of Figure 2, the script entered from the "Add New Article" page is inserted into the database of Scholar@UC. This makes the XSS attack persistent. When the user views the citation of this article, the "Password" pop-up is prompted as shown in the right of Figure 2. Second-order XSS is much more damaging than first-order XSS for two reasons: (1) social engineering is not required (the attacker can directly supply the malicious input without tricking users into clicking on a URL), and (2) a single malicious script planted once into a database executes on the browsers of many victim users [6].

The ***guiding questions*** of the research project are to address the main ***challenge*** of discovering vulnerabilities in large-scale and evolving web applications, such as the first-order and second-order XSS vulnerabilities illustrated above. Teachers will engage in ongoing research in modeling test paths based on requirements, user interface interactions, and data flow dependencies. The real-world application of Scholar@UC and their development processes and practices will be used throughout the project to ensure the relevance and usefulness of the research.

Training Provided

Teachers will first learn the state-of-the-practice tools such as Find Security Bugs (a static code analysis tool) and ZAP (a dynamic penetration testing tool). They will also learn different types of software vulnerabilities, their exploits, and the extent to which the practical tools are capable of detecting them. Finally, the teachers will be trained with STRIDE modeling by following Microsoft's security development lifecycle [7] (see Figure 3). It is expected that threat modeling such as STRIDE will improve the efficacy of the contemporary vulnerability detection tools.

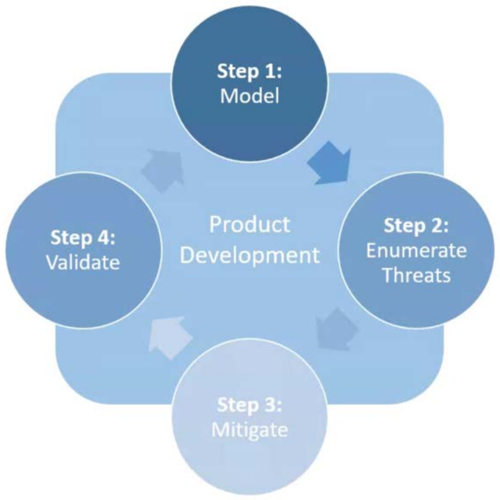


Figure 3 Microsoft's security development lifecycle (SDL)

STRIDE provides a comprehensive set of threats helping developers and testers to list the threats and their mitigations. The six types of STRIDE (and their standard mitigations) are: **S**poofing (cookie authentication), Tampering (digital signatures), **R**epudiation (secure logging and auditing), **I**nformation disclosure (encryption), **D**enial of service (filtering and quotas), and **E**levation of privilege (group or role membership).

Research Facilities

The teachers will be conducted in the Software Engineering Research Laboratory (ERC 527). Currently, the lab has six desktops with 3.4 GHz Intel i3 processor and one server with 2.60 GHz Intel Xeon dual-core processor. RET participants' research will be computational in nature, enhanced with hands-on experience and informed by empirical findings. Consequently, the work will be carried out on computers and whiteboard in our lab. The list of software includes Eclipse, Tomcat 9.0, and MySQL 5.6. Additionally, the RET research will be connected to our ongoing project "Leveraging Ohio Cyber Range at the University of Cincinnati (OCR@UC) to Advance Secure Web Application Development and Education" [8]. In this OCR@UC project, three nodes (virtual machines) built in an on-campus network will be available for the RET researchers to better experiment the tools and to execute the security tests in a distributed setting.

Industrial Partner

Glen Horton, project manager of the Scholar@UC project, has agreed to directly interact with the team. The majority of his work has been in agile and open-source software development. Meanwhile, he has expertise and experience in test-driven development and DevOps. He has been a very close collaborator of our lab for many projects. For example, in the summer of 2018, he helped to transfer our recent research findings in vulnerability discovery to the broader open-source development community (https://github.com/samvera/hyrax/issues/3187). Glen is willing to share the artifacts, processes, and documentations with the RET researchers who are also invited to observe and participate in the scrum meetings and other project meetings of Scholar@UC.

Ideas for Classroom Implementation

The secure software engineering project offers a wide variety of classroom applications built around using the state-of-the-practice tools and the state-of-the-art modeling methods to uncover vulnerabilities in large-scale and evolving web applications. Existing research on Scholar@UC system and the emerging findings ensure that the teachers will build upon valid results for their learning. A classroom unit will constitute the study of Scholar@UC or other safety-critical applications to develop effective and efficient approaches to uncover vulnerabilities such as XSS and SQL injection. The units uncovering the most (severe) vulnerabilities with the least effort will help the teachers to develop optimal strategies and to codify reusable attack patterns to secure the information assets on the web.

References Cited

[1] OWASP (Open Web Application Security Project), "Top Ten 2017 Project," https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project Last accessed: Nov 2018.

[2] National Vulnerability Database, "CWE (Common Weakness Enumeration) Over Time," https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cwe-over-time Last accessed: Nov 2018.

[3] M. Santillan, “Hackers Redirected eBay Shoppers to Phishing Scam,” https://www.tripwire.com/state-of-security/latest-security-news/hackers-redirected-ebay-shoppers-to-phishing-scam/ Last accessed: Nov 2018.

[4] Juniper, “Data breach costs will soar to $2T,” https://news.cuna.org/articles/105948-data-breach-costs-will-soar-to-2t-juniper Last accessed: Nov 2018.

[5] A. Meneely, B. Smith, and L. Williams, “iTrust electronic health care system case study,” in J. Cleland-Huang, O. Gotel, and A. Zisman, editors, *Software and Systems Traceability*, pages 425-438, Springer 2012.

[6] A. Kiezun, P. J. Guo, K. Jayaraman, and M. D. Ernst, “Automatic Creation of SQL Injection and Cross-Site Scripting Attacks,” in International Conference on Software Engineering (ICSE), Vancouver, Canada, 2009, pages 199-209.

[7] Microsoft, “Security Development Lifecycle,” https://www.microsoft.com/en-us/sdl Last accessed: Nov 2018.

[8] N. Niu and W. Wang, “Leveraging OCR@UC to Advance Secure Web Application Development and Education,” https://staging7.uc.edu/research/centers-labs/center-of-academic-excellence-in-cyber-operations/research.html Last accessed: Nov 2018.