Cross-Site Request Forgery vulnerability scanner

Florent Robineau
University of California, Berkeley
Department of IEOR
Berkeley, CA 94720

robineau@berkerley.edu

ABSTRACT
Cross-Site Request Forgery (abbreviated CSRF) is a form of attack on websites that exploits the trust that said website has in a user; CSRF has been a growing concern in the security world for two reasons: security scanners don't recognize them for now; and unless specific steps are taken to protect the application, virtually any web application is vulnerable to CSRF.

In this paper, we present a project that will aim at:

- Defining the problem at hand
- Analyzing current solutions, their effectiveness and their risks
- Writing a plug-in for Nessus to detect CSRF vulnerabilities in a website

Categories and Subject Descriptors
K.6.5 [**Security and Protection**]: Unauthorized access

General Terms
Security, Design

Keywords
CSRF, Cross-Site Request Forgery

1. INTRODUCTION
Cross-Site Request Forgery is a technique of attack on websites that trust its users ("remember me" authentication, etc). The goal is to induce the target under attack to execute a link to the vulnerable website; the link will trigger some action that will have undesirable side effects like creating a post, deleting a user's account, passing orders like transferring money or buying a book ([1], [2], [3], [4], [5], [6]).

These attacks have been called "rampant" because – unless specific counter-measures are used – virtually any web application is vulnerable. This is all the more so as pernicious because it is very hard for a computer to distinguish between genuine user interaction and a CSRF attack; CSRF vulnerabilities are inherent to the way the Web works.

Additionally, security scanners so far do not include modules to perform security checks for CSRF vulnerabilities. The project I propose wishes to address this need.

2. PROJECT PROPOSAL
I propose a four (maybe five) stages project:

- Milestone 1 (October 28th): investigate more about CSRF
- Milestone 2 (November 4th): analyze the solutions that have been proposed to avoid being vulnerable to CSRF attacks, their strengths / advantages, weaknesses / shortcomings, and overall effectiverness.
- Milestone 3 (November 18th): devise a way to automatically scan a web site for CSRF vulnerabilities
- Milestone 4 (December 2nd): implement the strategy developed during milestone 3 as a plug-in to the Nessus security scanner
- Milestone 5 (time permitting): develop an interceptor for a Java MVC framework such as Struts 2 to detect and avoid CSRF attacks on the server side.

3. ACKNOWLEDGMENTS
My thanks to Professor Wagner for the project idea.

4. REFERENCES
[1] Cross-Site Request Forgery (CSRF) http://en.wikipedia.org/wiki/Cross-site_request_forgery

[2] Mail from Peter Watkins, http://www.tux.org/~peterw/csrf.txt

[3] Chris Shiflett, Security Corner: Cross-Site Request Forgeries, http://shiflett.org/articles/cross-site-request-forgeries

[4] Cgisecurity, CSRF, http://www.cgisecurity.com/articles/csrf-faq.shtml

[5] Information Security Partners, Cross Site Reference Forgery, http://www.isecpartners.com/files/XSRF_Paper_0.pdf

[6] Confused Deputy, http://en.wikipedia.org/wiki/Confused_Deputy