# Chapter 1.14
# Security in a Web 2.0 World

**Richard T. Barnes**
*SunGard Higher Education, USA*

## ABSTRACT

Web 2.0 has brought enumerable benefits as well as daunting problems of securing transactions, computers, and identities. Powerful hacker techniques, including cross-site scripting (XSS) and cross-site request forgery (CSRF), are used to exploit applications to reveal and steal, at the worst, confidential information and money, or, at the least, cause trouble and waste time and money for reasons that may be best described as fun or simply possible to do. The people interested in transgressing Web 2.0 applications do so for money, prestige, or for the challenge. An infamous hacker from the early days of the Internet now heads his own Internet security company. A more recent hacker of some infamy has created a stir of concern and consternation as to how pervasive and potentially destructive hacker attacks can be. Securing Web 2.0 applications requires a multifaceted approach involving improved code development standards, organizational policy changes, protected servers and workstations, and aggressive law enforcement.

DOI: 10.4018/978-1-60566-122-3.ch005

## INTRODUCTION

With the multitude of benefits derived from the various Web 2.0 technologies, it is unfortunate that this book needs a chapter on security. Although the collaborations, synergies, and transformations of the collective Web technologies (known as Web 2.0) have immeasurably changed society in a good way, there is a bad element that we must recognize, understand, and defend against.

The relatively open and participative nature of Web 2.0 is, at once, a strength and weakness. Opening sites to user content and comment creates synergies that would not exist had the sites been restricted to a select few. However, it is difficult to restrict user input to only positive discourse; various motivations compel some to poison this well we know as Web 2.0.

Collectively, the responsibility and burden falls on organizations and individuals to share in constraining the enablers to minimize the damage to our 2.0 Web sites. Although it is helpful to understand the motivations behind the various (and growing) attacks, it is more important to follow best practices in code development and security design (Evers,

2007). The adage "the best defense is a good offense" does not apply well to Web 2.0 security. We cannot proactively prosecute and punish someone before they commit a Web attack; we may be on the road to a changing world, but constitutional rights cannot be trampled upon.

It is likely that some are dissuaded by the possibility of punishment if caught; but if only a few carry out Web attacks, our best approach, still, is to mount our best defense. It is of course equally important to prosecute security offenses. The threat of punishment has to be more than theoretical: Offenders must know that if they are caught, there will be consequences.

This chapter will explore the motivations, methods, and defenses against the malicious behaviors that cost time and money, and lessen the positives that can come from these technologies. There have been notable attacks to prominent Web sites; a few of these will be examined for their causes and associated effects. The evolution of the World Wide Web into version 2.0 has had social impacts, too. What are these impacts, and are there trends evident that may help us predict where security attacks and defense strategies will go in the future? Some possibilities are explored here and in subsequent chapters.

There is an old adage that says those who forget the past are condemned to repeat it. This idea cannot be forgotten in Web 2.0 security. We must remember how attacks happened before so we can avoid similar attacks in the future. By examining the trends, analyzing our mistakes, and understanding our needs, we can improve on Web 2.0 and make it better. That is how we got to version 2.0 from 1.0. Perhaps, as the Web evolves into what some in the community are calling Web 3.0, the lessons learned here will not be forgotten.

## BACKGROUND

It is perhaps ironic that the following definition for application security comes from one of the best known wikis, Wikipedia. Application security encompasses measures taken to prevent exceptions in the security policy of an application or the underlying system through flaws in the design, development, or deployment of the application. This definition is an excellent start in addressing a very large problem. However, it does not really tell us why; that is, why is it necessary to prevent exceptions to security policy?

A broader definition may help. There are several definitions of the word *security*: The freedom from danger or the freedom from fear and anxiety are two variants that tell us why application security is so important to Web 2.0 applications. Identity theft, corporate espionage or sabotage, and/or simple maliciousness are certainly enough to give most of us some pause or anxiety. Application security, as it relates to Web 2.0, is now an area of great attention because of our collective need to be free of these dangers.

A confluence of factors has complicated our lives as Web 2.0 becomes a more significant presence. The graphics-rich functionality, collaboration, and opportunities have not only yielded "serendipitous innovation" (Tapscott & Williams, 2006), but less desirable consequences, too.

Consequences such as cross-site scripting (XSS) and cross-site request forgeries (CSRFs) were not anticipated when foundational Web 2.0 technologies were created. Asynchronous JavaScript and XML (extensible markup language), or AJAX, is a set of Web development techniques that enable Web sites to be interactive and rich with features that make the static Web pages of a few years ago seem, well, static. However, it is through AJAX and other technologies