



The Growing Risk of Web Application – Vulnerabilities in Software as a Service Offerings

[By Lars Ewe, Chief Technology Officer, Cenizic Inc.](#)

Industry adaptation of Software as a Service (SaaS) has gained significant momentum during the past several years. As a result, more and more critical user data is being managed by SaaS offerings, as opposed to the traditional in-house hosting model. In return, this means that SaaS providers are becoming the gatekeeper for sensitive information, whether it is in the form of personal data (credit card or SSN information), or corporate data (customer or business information). Security breaches can therefore be potentially devastating for both users and SaaS providers. Industry data indicates the cost for lost user data to be in the hundreds of dollars per lost data record, with the average cost for documented security breaches being in the millions of dollars. Not to mention, the damage done to the company's brand name.

This article will outline the various security challenges associated with SaaS and Web applications, as well as:

- *Why conventional network security solutions are inadequate*
- *How new Web 2.0 technologies like AJAX further compromise security*
- *Testing strategies for both manual pen testing and automated assessments and*
- *Remediation strategies targeted at SaaS providers in order to help improve their security posture*

So why is the security of Web-based SaaS offerings so challenging? Basically, the challenge with SaaS security isn't any different than from other Web application technology and is theoretically reasonably well understood, at least within the expert community. That being said, one of the problems is that traditional network security solutions, such as network firewalls, network intrusion detection and prevention systems (IDS & IPS), don't adequately address the problem. Web applications introduce new security risks that can't effectively be defended against at the network level, and do require application level defenses.

Why is that (Why are Web apps so risky or the inherent risk of Web apps)?

One of the reasons Web applications are risky is firewalls have to open ports 80 & 443 (for regular HTTP traffic and SSL encrypted HTTP traffic respectively), and by doing so enable a firewall hole for attackers to circumvent most of the conventional network security mechanisms. Fundamentally this problem comes down to how you differentiate between legitimate and malicious HTTP requests – a non-trivial challenge – as this differentiation often requires more than just a simple signature-based detection logic. Depending on the attack vector, successful detection can depend on context-based session state and/or knowledge of the underlying application logic.

So what are some of the more well known Web application security exploits currently deployed 'in the wild'? Two of the most common Web application attack vectors include SQL Injection and Cross-site Scripting (XSS). For more information on common Web application vulnerabilities refer to the Open Web Application Security Project ([OWASP](#)).

In addition to some of the more common Web application attacks described above, there are new attack vectors enabled by Web 2.0 technology that add a complexity layer to the ever-growing application security problem. Examples of these new Web 2.0 technologies include Asynchronous JavaScript and XML (AJAX), RSS, (RESTful) Web Services, Rich Internet Application (RIA) technologies, such as Flash, Silverlight, etc., and the use of so-called 'Mashups.' An example of an AJAX specific attack would be [JSON/JavaScript hijacking](#). AJAX adds additional complexity to the task of detecting malicious requests due to the wide array of various up & down-stream data formats often used by AJAX frameworks and home-grown AJAX implementations, such as XML, JSON, HTML and other custom formats.

'Mashups' – the popular approach of combining content and services from various sources to build new Web sites – are often vulnerable to hacker attacks, as the services and content can have weak security attributes associated with them, often per design, to allow for easier integration between Web applications and services.

New attacks are being discovered at an alarming rate. The [Q1 2008 Application Security Trends Report](#) (compiling data from SecurityFocus, CVE, SANS, US-CERT, SecurityTracker, as well as other third party databases for Web application security issues) revealed that out of the total 1,409 vulnerabilities found in Q1 2008, 70 percent were Web application vulnerabilities. See table 1.



Table 1: According the Q1 2008 Application Security Trends Report, 70% of the total Vulnerabilities were comprised by Web applications in Q1 2008.

We've also seen a troublesome shift in the sophistication and motivation of attackers. So-called 'script kiddies' have been replaced by financially motivated, highly sophisticated, professional hackers that will choose their targets based on anticipated profits and with the primary goal of financial gains, rather than just being 'cool.'

And while we have seen the introduction of various regulatory standards, such as the [Payment Card Industry's Data Security Standard \(PCI DSS\) for Web applications](#), experts fear that the standards are not strong enough to fend off the most sophisticated hackers.

Due to the plethora of security vulnerabilities facing Web applications, SaaS providers need to guard their security posture more than ever and ensure service security via a combination of tactics:

1. Manual pen testing
2. Ongoing, automated testing, as well as

3. Security-related improvements in their Software Development Lifecycle (SDLC)

There are various stages in the SDLC at which developers, QA, InfoSec and Operations should ensure the implementation of up-to-date security tests and assessments and there are a number of software solutions available. These include HP, IBM and CenZic Hailstorm. But with a large majority of existing Web applications in deployment, it's just as important to test production applications – due to the ever-evolving list of known attack vectors – on a regular and continuous basis. The challenge with testing production applications is that any deep testing can result in data corruption or even service failure. It's therefore quite understandable that many SaaS providers are reluctant to test deployed production applications and focus solely on the security of new applications (or applications that are on staging by QA) before deployment.

With the assessments of Web applications in the development or QA phase of the SDLC being addressed by existing security solutions, how does one go about testing already deployed production applications? There are three different approaches. The first is to use very careful manual assessments. The second is to use scaled-back automated attacks. Both approaches try to minimize the risk of corrupting any data or taking down the live production application. But do these two approaches provide enough test coverage for all possible vulnerabilities? The simple answer is no.

After all, hackers don't use scaled-back attacks. And while scaled-back testing is certainly one option for production application testing, it only provides a subset of test coverage and therefore, limited security assessments. This is where the third possible approach comes into play.

The third option is to create a copy of the production application in a staging environment. The challenge here tends to be the associated effort of staging a copy of the application, especially if you want to test the application on a regular basis in order to cover the latest and greatest known attack vectors. This can obviously be a labor and cost-intensive effort.

Lars Ewe is Chief Technology Officer of CenZic, a provider of Web application security, Web application security testing and Web application security assessment. He is a technology executive with a broad background in (Web) application development and security, middleware infrastructure, software development and application/system manageability technologies. Throughout his career, Lars has held key positions in engineering, product management/marketing and sales in a variety of different markets. Prior to CenZic, he was Software Development Director at Advanced Micro Devices, Inc. and was responsible for AMD's overall systems manageability and related security strategy as well as all related engineering efforts. Before AMD, Lars held key positions at Borland Software Corp., Oracle Corporation, and Webgain Inc. For article feedback, contact Lars at Lars@cenZic.com