OWASP has previously published its risk-based methodology.[1] The following is an overview of the process, and the methods to consider when customizing it to the financial services industry.

Step 1: Identifying a Risk

Using the OWASP guidelines, the first step is to identify your business environment. This includes identification of the following:

1. The **threat agent involved:** OWASP uses the following formula for determining the threat agent: Capabilities + Intentions + Past Activities. Possible threats include employees, organized crime, accidents, and natural disasters.
2. The vulnerability or **attack vector** that permits an attack: a vulnerability is a hole or a weakness found in a software application, such as lack of input validation (where user input is not checked or filtered), a buffer overflow (where an excess of input data is allowed to rewrite programming code), or lack of proper error handling. Vulnerabilities may or may not lead to an attack. An easily exploitable vulnerability is more critical to fix than a vector that is difficult to attack.
    a. What is the **vulnerability prevalence?** If the vulnerability is pervasive, there is a greater chance of exploitation by attackers.
    b. How easily is the **vulnerability detected**? If an attacker can scan and find the vulnerability on your system, then it should be a higher priority than one that is much more difficult to detect.
3. The impact of a successful attack upon your business. There are two aspects:
    a. **Technical impact** of mitigating an attack. For example, you may determine that the costs in mitigating one particular attack significantly outweigh the losses, and so you may choose to accept this risk.
    b. **Business impact.** This is an open-ended definition that should include loss of confidentiality, integrity, or availability (such as a denial of service attack on your company).

New 2010 OWASP Top 10 Guidelines Include Risk Weighting and Impact to Institution

Figure 1: Risk Rating Methodology

Likelihood				Impact	
Agent	Attack Vector	Vulnerability Prevalence	Vulnerability Detection	Technical Impact	Business Impact
Insider	Failure to filter user input	Common coding flaws	Code review	Denial of service	Loss of PII
Malware	Failure to use SSL/TLS		Testing	Access to internal networks	Loss of reputation
Organized crime	Failure to check referrer headers		Application scans	Access to admin controls	Loss of customers
Espionage	Directory listing enabled				Dollar loss
	Leaks in authentication				Fines

5. Cross-Site Request Forgery (CSRF)

Attack Vector	Vulnerability Prevalence	Vulnerability Detection	Technical Impact
Average	Widespread	Easy	Moderate

Related to but different from XSS is cross-site request forgery or CSRF ("cee-surf"). For example, Alice could send Bob what looks like an image file within HTML tags , such as **<img src="">,** however within the quotes is a URL with a scripted request to withdraw money from Bob's bank, such as http://www.bobsbank.com/.

What's at risk here are sites that don't authenticate the user beyond initial login; a withdrawal request should be accompanied with a confirmation that the user is requesting that withdrawal. CSRF attacks often originate from social networking or forum sites that allow users to upload images and other rich content but not JavaScript. CSRF assumes the site doesn't check the referrer. The referrer allows the new page to see where the request came from. Referrer logging should be used to allow web servers to identify where people are visiting them from. To execute this, an attacker must find a form that does something interesting, must input the right values, and then must somehow lure the victim to a compromised page while the victim remains logged in.

For example, Alice is in a session with some financial service and also chatting on a social networking site and reading the latest news at the same time, using different tabs on her browser. Bob sends Alice a malicious link. When Alice opens Bob's link, she's taken to a new page of content, however, in the rendering of that new page there's an embedded request to use her open banking session to initiate a transfer of funds to Bob's account. Alice never suspects.

Mitigation of CSRF

Check the referrer headers, the GET and POST as well as cookie information, and limit the lifetime of the cookie.

- Also, check that there's no cross-domain use of Flash or other multimedia.
- As much as possible, keep the customer on one domain, and if you need to link elsewhere, inform the customer of the change in domain.
- Finally, the use of a secret token for all sensitive requests can stop CSRFs.

6. Security Misconfiguration (New)

Attack Vector	Vulnerability Prevalence	Vulnerability Detection	Technical Impact
Easy	Common	Easy	Moderate

Security is predicated on choosing a secure configuration defined for each application. This extends to the framework, web server, application server, and platform. All these settings should be defined, implemented, and maintained because many are not shipped with insecure default values. OWASP says security misconfiguration was part of the Top 10 in 2004, however, "it was dropped because it wasn't thought of as a software issue. However, from an organizational risk and prevalence perspective, it clearly merits re-inclusion in the Top 10, and so now it's back."

One scenario: A vulnerability that affects a financial service's Web server application gets patched. But before the patch is applied, Bob, an attacker, reverse engineers the vulnerability and exploits it. Bob discovers a vulnerable web server at the financial institution before the IT staff can apply the vendor patch.

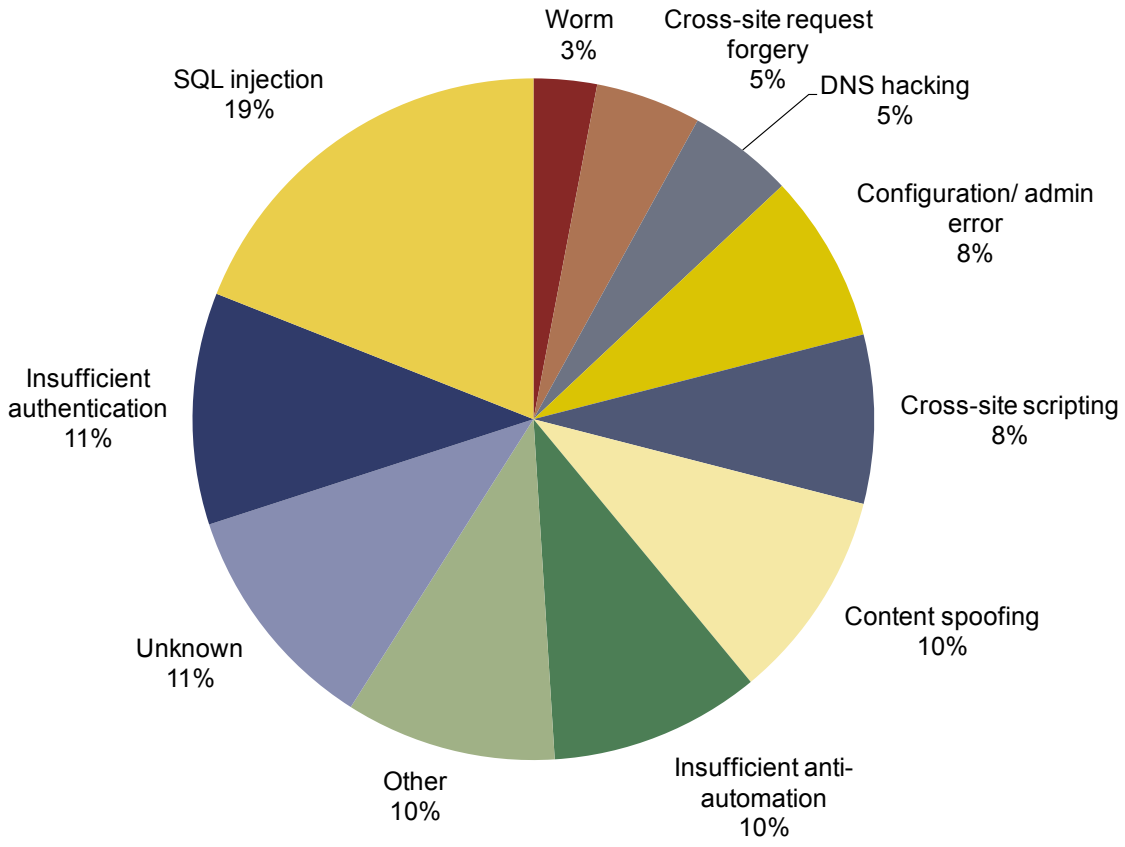
In a second scenario, a default administration console is exposed to the web with default passwords in place. Bob, scanning the Internet, finds a financial institution with default "admin" as password, and is able to gain elevated privileges on the site, which he can exploit to do his mischief.

Looking at the risks of an attack is only one side of the equation. Also important is the prevalence of an attack in the wild. For that, Javelin cited Breach Security's Web Hacking Incidents Database 2009 report.

SQL injection remains the number one attack, with cross-site scripting and cross-site request forgery ranked relatively lower. A top ten bank confirmed similar findings to Javelin. That doesn't mean that XSS and CSRF are no longer problems, only that criminals are focusing more upon SQL and other forms of code injection. As we can see from the vulnerabilities data in the next section, XSS remains a major problem across all industries.

SQL Injection is Also the Most Common Attack

Figure 5: Prevalence of Attacks in 2009



Source: Breach Security THE WEB HACKING INCIDENTS DATABASE 2009