

Why Web Application Security is Important

No one on the Internet is immune from security threats. In the race to develop online services, web applications have been developed and deployed with minimal attention given to security risks, resulting in a surprising number of corporate sites that are vulnerable to hackers. Prominent sites from a number of regulated industries including financial services, government, healthcare, and retail, are probed daily. Some banks have reported being probed as many as 50 times a day.¹ The consequences of a security breach are great: loss of revenues, damage to credibility, legal liability and loss of customer trust.

Web applications are used to perform most major tasks or website functions. They include forms that collect personal, classified and confidential information such as medical history, credit and bank account information as well as user satisfaction feedback. If your organization is legally bound by legislations including COPPA, HIPAA, GLBA and Sarbanes-Oxley to protect the privacy and security of personally identifiable information, and hackers can get at this sensitive information, you run the risk of being found guilty of non-compliance. Gartner has noted that almost 75 percent of attacks are tunneling through web applications.² Web application security is a significant privacy and risk compliance concern that remains largely unaddressed.

Why Web Application Security Should be Part of Your Web Risk Management Program

There are many reasons your organization should identify and address web application security vulnerabilities as part of your web risk management program:

- **Reduce Cost of Recovery and Fixes** -- Computer security attacks cost as much as \$10 billion a year.³
- **Ensure Customer Trust** -- Trust is a key component to customer adoption and retention.
- **Encourage Website Adoption** -- Consumers are still not adopting websites as a preferred channel for doing business. The Tower Group cited that 26 percent of customers don't use online banking for security fears and another 6 percent do not due to privacy issues.⁴

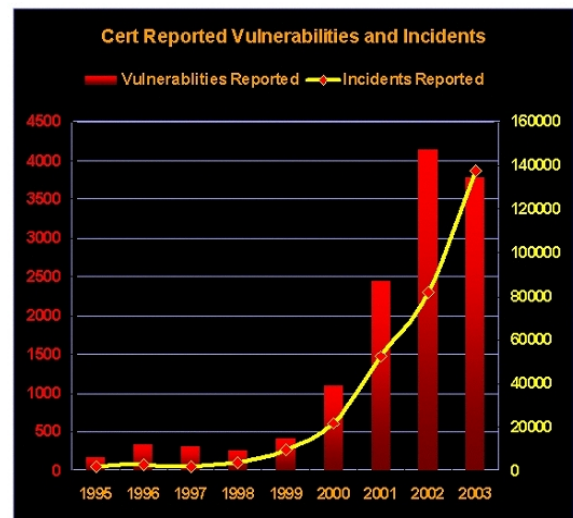
¹ "Risk Assessment: The Foundation for Security Planning", Center for Technology in Government (http://www.ctg.albany.edu/publications/reports/inet_security_seminar?chapter=2)

² "Airline Web Sites Seen As Riddled With Security Holes," *Computerworld*, February 4, 2002. (<http://www.computerworld.com/securitytopics/security/story/0,10801,67973,00.html>)

³ "Internet Security as Part of the Overall Security Plan," (http://www.ctg.albany.edu/publications/reports/inet_security_seminar?chapter=1)

⁴ "Online Banking Opportunities", emarketer, July 2003.

- **Maintain Competitive Advantage** -- Many organizations are using trust as a key competitive advantage and are leveraging customer fears to proactively implement security and privacy programs to ease the uncertainty.
- **Reduce Cost of Manual and Outsourced Security Testing** -- Many organizations today, especially ones in regulated industries, test security using costly manual processes that cannot address all potential website risks. Other organizations spend millions on outsourced security assessment and ethical hacking resources. Consider these statistics:
 - On average, there are anywhere from 5 to 15 defects per 1,000 lines of code.⁵
 - A 5-year Pentagon study concluded that it takes an average of 75 minutes to track down one defect. Fixing one of these defects takes 2 to 9 hours each. That translates to 150 hours, or roughly \$30,000, to clean every 1,000 lines of code.⁶
 - Researching each of the 4,200 vulnerabilities published by CERT last year for just 10 minutes would have required 1 staffer to research for 17.5 full workweeks or 700 hours.⁷
 - Gartner Group estimates that a company with 1,000 servers can spend \$300,000 to test and deploy a patch; most companies deploy several patches a week.⁸



How Hackers Get In

Browser-based attacks use flaws in the web-based application code. Software most vulnerable to these types of attacks includes:

- User interface code -- provides the look and feel of the site
- Web server -- supports the physical communication between the user's browser and the web applications
- Front-end applications -- interfaces directly with the user interface code, and back-end systems

⁵ U.S. Dept. of Defense and the Carnegie Mellon Software Engineering Institute

⁶ "Software Hell: Glitches cost billions of dollars and jeopardize human lives. How can we kill the bugs?" (http://www.businessweek.com/1999/99_49/b3658015.htm)

⁷ Intel whitepaper, CERT, ICSA Labs

⁸ "Attacks Averted", *InformationWeek*, February 3, 2003.

Examples of vulnerabilities

Hack attack	What hackers use it for
1. Cookie Poisoning	Identity theft/ Session Hijack
2. Hidden Field Manipulation	eShoplifting
3. Parameter Tampering	Fraud
4. Buffer Overflow	Denial of Service/ Closure of Business
5. Cross-Site Scripting	Hijacking/ Identity Theft
6. Backdoor and Debug Options	Trespassing
7. Forceful Browsing	Breaking and Entering
8. HTTP Response Splitting	Phishing, Identity Theft and eGraffiti
9. Stealth Commanding	Concealing Weapons
10. 3rd Party Misconfiguration	Debilitating a Site
11. Known Vulnerabilities	Taking control of the site
12. XML & Web Services Vulnerabilities	New layers of attack vectors & malicious use
13. SQL Injection	Manipulation of DB information

How do these Vulnerabilities Affect Your Customers?

Your customers can be affected in a variety of ways: from identity theft to session hijacking to the compromise of confidential and private customer data. Cross-Site Scripting (XSS) is one of the leading methods used in identity theft (and an obvious concern to financial and healthcare institutions); it attacks the user via a flaw in the website that enables the attacker to gain access to login and account data from the user. Many of the phishing email-based schemes use cross-site scripting and other application layer attacks to trick users into giving up their credentials.

SQL injection is one of the main attacks used when backend databases are compromised. General consensus has pegged SQL injection as the method used behind the massive compromise of credit card numbers in February of last year.⁹ We still see many cases where cookies aren't properly secured, allowing an attacker to 'poison' the cookie, hijack active sessions or manipulate hidden fields to defraud e-commerce sites. As web applications become more pervasive and more complex, so do the techniques and attacks hackers are using against them. Recent new vulnerabilities and attack methods discovered or reported show an alarming trend toward attacks with multi-faceted damages and even anti-forensics capabilities. This means hackers are using more powerful attacks to cause significantly more damage, while at the same time covering their tracks is becoming easier.

⁹ "Top Five Threats", *ComputerWeekly*, April 24, 2004.
(<http://www.computerweekly.com/Article129980.htm>)

Web Application Security Action Plan

A Web Application Security Process can be implemented using three key guidelines:

1. **Understand:** Perform security audits and defect testing throughout the application lifecycle. Production applications are an obvious first place to implement regular audits and analysis to determine security and compliance risk to an organization. At the same time, one must not forget that the application development lifecycle is the breeding ground for the defects that cause the risks. Performing security testing during the application development lifecycle at key points during the various stages from development to QA to staging will reduce costs and significantly reduce your online risk.
2. **Communicate:** After risks and security defects have been identified, it is imperative to get the right information to the right stakeholder. Development needs to understand what these vulnerabilities and compliance exposures are in development terms. This means providing details around how the attack works and guidance on remediation. There are several good sources both online and in security testing tools for developers. Similarly, QA must be able to perform delta, trend and regression analysis on the security defects just like they do for performance and functionality flaws. Using their existing methods and metrics they, along with Product Management, can properly prioritize and monitor the defect remediation process as well as accurately assess release candidacy. Finally, with the ever-increasing number and scope of Government and internal regulations and policies, teams from Security, Risk, Compliance and R&D need to communicate and validate application risks against these very real business drivers.
3. **Measure:** For any process to be successful, there needs to be criteria by which to measure the successes or failures of the procedures implemented. Organizations use trending and defect remediation analysis metrics to identify areas and issues to focus on (i.e., there may be a certain security defect type that keeps cropping up which can then be identified and dealt with through targeted education and training to recognize repeated risks with a particular infrastructure product or vendor). Ultimately, measuring and analyzing scan results will contribute to a reduction in liability and risk brought about by implementing a web application security plan.

To evaluate Watchfire® AppScan®, please visit:

<https://www.watchfire.com/securearea/appscanauditdownload.aspx>