FORTIFY® SOFTWARE

The Case for Application Security

How Real Is the Threat and What Are Your Options

Abstract

This report compiles data and research from numerous sources and organizes them into a single, straight-to-the-point, data-driven overview of current threats, the true costs of a data breach and the most effective solutions for securing your applications.

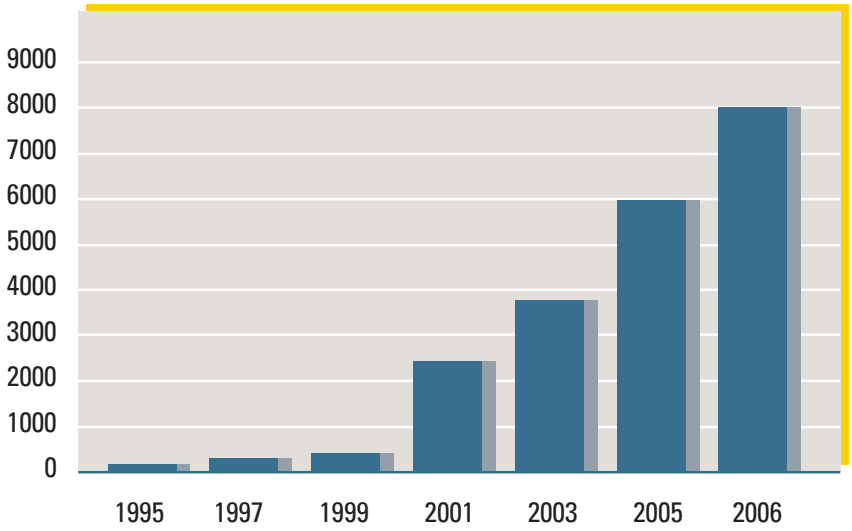
Contents

2300 GENG ROAD, SUITE 102, PALO ALTO, CA 94303 **O** 650 213 5600 **F** 650 843 1424 WWW.FORTIFYSOFTWARE.COM

Part I: How Vulnerable Are Web Applications

The advent of SOA, Ajax and other Web 2.0 technologies has allowed Web applications to become increasingly powerful and complex. With this complexity comes an ever-growing risk that security vulnerabilities will be introduced into applications. Carnegie Mellon University's CERT (Computer Emergency Response Team) tabulates comprehensive data on the number of software vulnerabilities reported each year. Between 1995 and 2006, the data CERT collected and analyzed from numerous sources showed that the number of reported security vulnerabilities increased an average of 42 percent per year.

Vulnerabilities Reported 1995 - 2006



Source: CERT [1]

Even more frightening are the vulnerabilities that are not reported. To gauge this number, a WASC (Web Application Security Consortium) project analyzed 31,373 Web applications for common vulnerabilities. WASC's research shows that these applications contained over 148,000 distinct vulnerabilities and includes the following details about them:

- 7 out of 10 were vulnerable to Cross-Site Scripting
- 1 in 3 aided attackers with Information Leakage
- 1 in 4 was susceptible to Content Spoofing
- 1 in 6 fell prey to SQL Injection
- 1 in 6 employed Insufficient Authentication
- 1 in 6 used Insufficient Authorization

- 1 in 7 allowed Abuse of Functionality
- 1 in 20 permitted Directory Indexing
- 1 in 30 was a victim of XPath Injection

Source: Web Application Security Consortium [2]

Despite compelling data to the contrary, many organizations continue to operate under the misconception that securing their networks will block attacks against vulnerabilities in their applications. An analyst from Gartner, Joseph Feiman, writes "Application developers and their superiors in IT departments too often mistakenly believe that firewalls, intrusion detection systems (IDSs), identity access management (IAM) systems and network traffic encryption are sufficient measures for applications' security. By doing so, they are confusing application security with network security." [3]

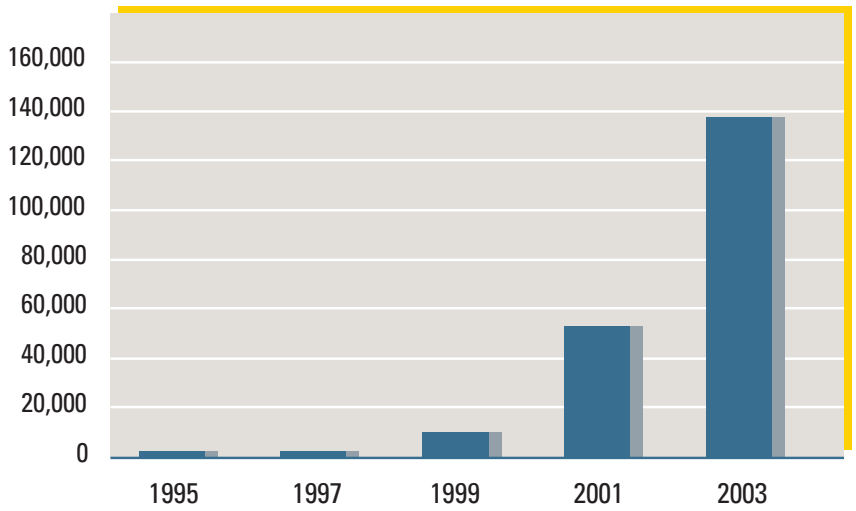
Most of the attackers are aware of this and continue to shift their focus to applications. Many well-respected sources have recognized this change, including:

- NIST, reporting that 92 percent of vulnerabilities are in software. [4]
- Gartner, reporting that 75 percent of breaches are caused by security flaws in software. [5]
- The United States Air Force, reporting that the percentage of attacks directed at their applications (versus their networks) grew from 2 percent to 36 percent between 2004 and 2006. [6]

Part II: The Current Hacking Landscape

Since the mid nineties, the number of attacks directed at the application layer has skyrocketed. CERT measured the growth in number of attacks to be, on average, 66 percent every year between 1997 and 2003, at which point they were forced to stop collecting the necessary data due to the sheer enormity of the endeavor.

Incidents Reported 1995 - 2003



Source: CERT [7]

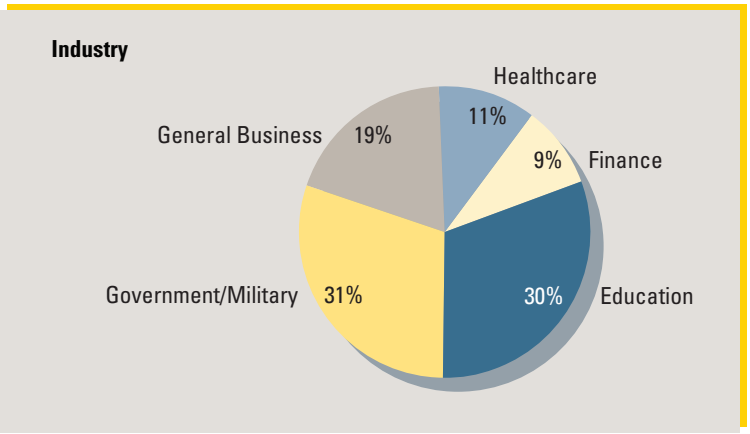
The trend measured by CERT has continued to intensify over the last few years. *InformationWeek* reported that the number of hackers attacking banks jumped by 81 percent between 2005 and 2006, according to figures released at the Black Hat security conference in July, 2007. *InformationWeek* argues this increase is due to the increased availability of hacking toolkits and malware in the online underground. [8] In addition, underground sites, such as *http://www.xssed.com/*, give attackers a blueprint of how to break into enterprise applications.

In 2005 and 2006 alone, over 100 million private records were reported stolen from American businesses; a significant portion (65 percent) of which was compromised as a direct result of a software breach. The following table shows a sample of major breaches that were reported over the last three years.

Company	Records Lost	Date
TJX	45,600,000	1/2007
CardSystems	40,000,000	6/2005
CitiFinancial	3,900,000	6/2005
Scottrade	1,300,000	11/2005
UCLA	800,000	11/2006
Wachovia, BofA, PNC, Commerce Bancorp	676,000	4/2005
Hotels.com	243,000	5/2006
OfficeMax	200,000	2/2006
Ameritrade	200,000	4/2005
Regions Bank	100,000	2/2006
DSW/Retail Ventures	100,000	3/2005
US Air Force	33,300	8/2005

Source: Privacy Rights Clearinghouse [9]

Even from this small sample set, one can see that these attacks occurred in a variety of companies across numerous industries. In 2005, experts have estimated the exact breakout by industry to be:



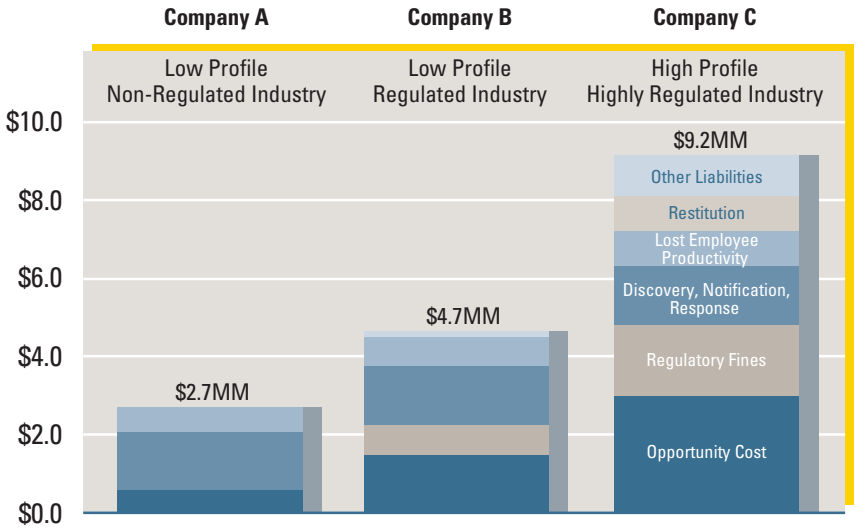
Source: Utimaco Safeware [10]

As more companies fall victim to data breaches, an increasingly accurate understanding of the true costs of a breach is being developed. The next section outlines these costs.

Part III: The Costs of a Data Breach

Various organizations have assessed the cost of a data breach. The estimated cost per compromised record currently ranges from a few dollars to upwards of $400. The breadth of this range can be attributed to variations in the size of the organization that experiences the breach, the number of customers affected, the nature of the organization's industry and the percentage of its revenue derived from online transactions.

The graph below outlines the results of a recent study by Forrester. It shows the average cost of a data breach that impacts between 20,000 and 30,000 records (an extremely conservative number given the size of breaches outlined in Part II). The study assessed the cost of a breach in three representative classes of businesses: a low profile company in a non-regulated industry, a low profile company in a regulated industry and a high profile company in a highly regulated industry.



Source: Forrester [11]

The Forrester data readily demonstrates that even a small breach can be extremely costly. To better illustrate the breakdown of these costs, Forrester offers the following examples of each.

Category	Description
Opportunity Cost	■ Customer churn ■ Difficulty in getting new customers
Regulatory Fines	■ FTC ■ PCI ■ SOX
Discovery, Notification, and Response	■ Outside legal counsel ■ Mail notification ■ Calls and call center ■ Discounted product offers
Lost Employee Productivity	■ Employees diverted from other tasks
Restitution	■ Civil courts may ask to put this aside
Other Liabilities	■ The security and audit requirements levied as a result of a breach ■ Credit card replacement costs ■ Civil penalties, if specific fraud can be traced to a breach

Source: Forrester [12]

The recent data breach at TJX is projected to cost the company a significant amount. The company announced that its second quarter profit fell 57% as it recorded a $118 million charge due to the breach - $11 million of costs incurred during the second quarter and $107 million for a reserve to cover costs, such as litigation and investigative expenses. In addition, TJX expects to record a non-cash charge of $21 million in the future and is currently under investigation by 37 states considering litigation to recoup costs incurred as a result of the breach.

Beyond the immediate costs outlined by Forrester and seen in the TJX breach, a public data breach can often have long-lasting repercussions down the road. One of the most extreme examples of these intangible costs is Cardsystems, which went from being the largest processor of credit-card transactions in the world to being sold in a fire sale after going bankrupt when a data breach in one of its applications compromised 40 million user records. An article in *The Wall Street Journal* entitled "Companies Pay a Price For Security Breaches" reads, "A look at stock moves following a string of losses of customer data by ChoicePoint Inc., LexisNexis and Bank of America Corp., among others, usually shows at least a moderate drop in a company's stock is likely after such an incident. On average, the data breaches shaved about 1% off the stocks right away, causing them to underperform the broader market, and there's some indication of a more serious, long-term effect."

Source: The Wall Street Journal [13]

Part IV: Preventing a Breach

The comprehensive approach to avoiding a data breach entails instilling security into the software development lifecycle. In order to do this, an organization must change certain processes, hire the right people and leverage available technology. In addition, it requires a constant level of threat intelligence, risk assessment, and sharing of best practices. A large body of literature exists today that is designed to help organizations create the right processes, policies and frameworks for instilling security into the software development lifecycle. There are also a number of case studies, which outline the steps that leading companies have followed in order to make their applications more secure.

Top technology companies will admit that technology alone can't solve the problem, but often offer services to help customers develop complete security solutions. In this paper we will focus on the available technologies that can help companies secure their data. We will provide an overview of these technologies as well as an assessment of the pros and cons of each.

Most of these technologies have been developed in recent years by university researchers and industry experts in order to counter the increased threats directed at applications. The fruits of this labor are demonstrated in three key technology areas:

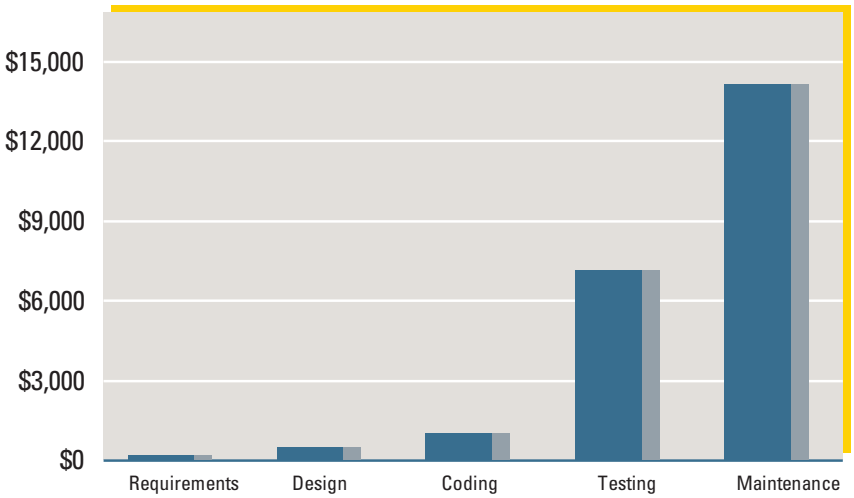
Source Code Analysis: Assessing the security of source code without executing it.

Security Testing: Dynamically testing a running application by simulating the behavior of a hacker.

Run-time Application Defense: Defending a deployed application by identifying attempted hacks and fraudulent behavior.

The "right" solution varies from one organization to another. In most situations, however, true security calls for a comprehensive solution involving a combination of all three approaches. Using the best that each approach has to offer, organizations should strive to find and fix vulnerabilities as early as possible in the software development lifecycle. The benefits of fixing bugs early in the development lifecycle are demonstrated in the following figure.

Cost of Fixing One Vulnerability
Based on the Stage It Was Identified



Source IEEE Computer, IEEE Computer Society [14]

The remainder of this paper is dedicated to an overview of these three approaches to application security, including information about when they provide the most benefit and how they can be used most successfully.

Source Code Analysis

Source code analysis is the most comprehensive solution to the application security problem. It leverages the ability to cover 100 percent of the application to analyze every feasible path that execution and data can follow to identify and help remediate hundreds of categories of vulnerabilities—more than any other application security solution. Source code analysis began its climb to popularity in 2005 and is used today by:

- The top five **commercial banks** and seven of the world's eight largest banks
- Five of the top seven **computer software** companies
- Three of the top five **aerospace and defense** industry leaders
- The three largest **armed services** for the U.S.
- Three of the top five **telecommunications** companies
- Three of the top six **securities** industry firms
- Three of the leading four **accounting** firms
- Two of the world's **most-visited Internet companies**
- Two of top three **insurance** companies
- The top two **wireless voice and data** carriers

- The #1 computer **peripherals** company
- The #1 **health-care services** company
- The world's largest dedicated **semiconductor** foundry

Increasingly, medium and small businesses have seen the value of this approach and are adopting source code analysis solutions at a quick rate.

In addition to being very effective at identifying and remediating security vulnerabilities, source code analysis is also a highly cost-effective approach. Research conducted by Cigital shows an average cost savings of over $2.1 million (on a code base of 2 million LOC) when vulnerabilities are identified during development, where source code analysis is most often leveraged.

Cost of Fixing Vulnerabilities Early			
Stage	**Critical Bugs Identified**	**Cost of Fixing 1 Bug**	**Cost of Fixing All Bugs**
Requirements		$139	
Design		$455	
Coding	200	$977	$195,400
Testing		$7,136	
Maintenance		$14,102	
Total	**200**		**$195,400**

Cost of Fixing Vulnerabilities Later			
Stage	**Critical Bugs Identified**	**Cost of Fixing 1 Bug**	**Cost of Fixing All Bugs**
Requirements		$139	
Design		$455	
Coding		$977	
Testing	50	$7,136	$356,800
Maintenance	150	$14,102	$2,115,300
Total	**200**		**$2,472,100**

Source: Cigital, "Case Study: Finding Defects Earlier Yields Enormous Savings" [15]

Some companies face a decision between automated and manual code review. Although manual code review has proven highly effective at finding bugs more efficiently than traditional security testing, automated source code analysis is equally as effective and can increase efficiency, cover more of the application and save substantial amounts of time and energy when used to assist in code review. Traditionally, manual audits have been expensive and are often limited to small portions of the code base under review, leaving potentially vulnerable code untested.

If a company decides to license a source code analysis tool, they often deploy it:

- On Developers' desktops
- With a security team, which will act like a gate
- With an individual auditor who will focus exclusively on performing analyses

Each of these deployment scenarios has advantages and disadvantages, but the following pros and cons are consistently associated with source code analysis.

Pros:

- 100 percent code coverage
- Identifies vulnerabilities when they are least expensive to fix
- Very comprehensive list of vulnerabilities

Cons:

- Uncovers a large number of potential vulnerabilities, which require human review

Security Testing

Traditional security testing can be grouped into two categories:

- Ethical Hacking: hiring a professional (ex-)hacker to break an application
- Running automated tools to simulate a hacker's actions

Both techniques attack the application at run-time, which allows them to identify vulnerabilities that only manifest themselves when the application is deployed, such as vulnerabilities that depend on the application's configuration and environment. Application security testing is uniquely suited for identifying these specific issues.

The downside of security testing is that, like other run-time testing, it is difficult to fully cover the application and occurs late in the software development lifecycle (because it requires that the application being tested be built and deployed).

Companies that leverage automated security testing often purchase a few licenses for a small group of users on a security team, who are then responsible for testing as many high-risk applications as possible.

In many organizations there is an effort to empower the QA group to help conduct security tests, but current solutions are designed for security professionals and not QA teams. The next wave of security testing technology will leverage existing QA tests and require little security expertise.

Pros:

- Quick and easy to get started
- Identifies issues only evidenced in the deployed application
- Accurate at finding vulnerabilities visible over the Web

Cons:

- Not comprehensive—difficult to exercise the entire application
- Unable to identify vulnerabilities that are not visible over the Web
- Lacks code-level details, making remediation difficult for developers
- Not tightly integrated with existing run-time testing capabilities in QA

Run-Time Application Defense

This category includes intrusion detection and prevention systems, application firewalls, and most recently, application security platforms. This group of solutions evolved from standard network firewalls by continually improving their ability to understand the applications and communications protocols they defend. This evolution has brought run-time application defense ever closer to the software it's designed to protect to the point that, today, some solutions reside inside the application itself.

The first wave of pure application firewalls performed very poorly and met with a commensurate level of adoption and market growth. These solutions demanded unreasonable levels of effort to set up and administer, and suffered from highly inaccurate behavior that often broke application functionality.

However, recent solutions – namely in the form of application security platforms – have positioned themselves inside the application and leverage the improved context this offers them to provide more accurate, more automated and more complete defenses.

For example, an application security platform that resides inside the application might correlate a user session with requests to a credit-card processing API and prompt any user who submits more than five failed attempts to re-authenticate or contact customer support to complete their order. Such a defense would make brute-force attacks against credit-card security codes (CVV2) almost impossible. This is just one example of how being inside the application also allows for enhanced fraud protection and data mining prevention over network-based solutions.

Pros:

- Potentially quick and easy to deploy
- Fast solution to secure apps against common attacks
- Can be used to protect against fraud and data mining

Cons:

- Some solutions still require extensive effort to set up and manage
- Inaccurate tuning may result in turning away legitimate users

Summary

As our networks become more secure, the application is becoming the new frontier for cyber warfare. The results show that hackers are increasingly attacking vulnerabilities in applications to steal private records, often at great cost to the application's owner. Technology, including the advances commonly referred to as Web 2.0, has enabled applications to provide new levels of functionality, which brings with it added complexity and a related increase in the chance the software will contain security vulnerabilities.

Today, attacks against applications impact organizations across nearly every industry, including finance, healthcare, retail, telecommunication, education, ISVs, and government agencies.

Government regulations, such as SOX and HIPAA, mandate protections for certain industry sectors, and the PCI (Payment Card Industry) Data Security Standards require organizations that process credit-card transactions to implement specific application security activities. Organizations such as OWASP, ISACA, ISSA, SANS and others promote awareness and education regarding application vulnerabilities and attacks.

Every indication is that hackers, organized crime cartels and foreign entities are increasing their efforts and directing attacks against applications. The tools to protect you are here. What are you waiting for?

References

[1] Computer Emergency Response Team, "Vulnerabilities Reported" http://www.cert.org

[2] Web Application Security Consortium, "Web Application Security Statistics Project" http://www.webappsec.org/projects/statistics/

[3] Joseph Feiman, "Application Developers Should Assume Responsibility for Application Security" November 16, 2006, Gartner

[4] Mark Curphey, "Software Security Testing: Let's Get Back to Basics" October, 2004, SoftwareMAG.com

[5] Theresa Lanowitz, "Now Is the Time for Security at the Application Level" December 1, 2005, Gartner

[6] Bruce Jenkins, Major, USAF (Ret.)

[7] Computer Emergency Response Team, "Incidents Reported" http://www.cert.org/

[8] Sharon Gaudin, "Number of Hackers Attacking Banks Jumps 81%" August 2, 2007, *Information Week*

[9] Privacy Rights, Clearinghouse, "A Chronology of Data Breaches" http://www.privacyrights.org/ar/ChronDataBreaches.htm#2006

[10] Utimaco Safeware, "2005 Data Security Breach Statistics" http://americas.utimaco.com

[11] Khalid Kark, "Calculating the Cost of a Security Breach" April 10, 2007, Forrester Research

[12] Khalid Kark, "Calculating the Cost of a Security Breach" April 10, 2007, Forrester Research

[13] Michael Rapoport, "Companies Pay a Price For Security Breaches" June 15, 2005, *The Wall Street Journal*

[14] B. Boehm and V. Basili, "Software Defect Reduction Top 10 List," IEEE Computer, IEEE Computer Society

[15] Cigital, "Case Study: Finding Defects Earlier Yields Enormous Savings" http://www.cigital.com/solutions/roi-cs2.php