Research Report

FEBRUARY 2002

The Security of Applications: Not All Are Created Equal

By Andrew Jaquith
ajaquith@atstake.com

Andrew Jaquith is a Program Director with @stake, Inc. He directs the Hoover Project, an initiative within the firm to improve clients' abilities to quantify the risks and returns of security investment. Mr. Jaquith actively works with leading clients in the financial services, supply chain and software sectors.

Companies increasingly require ways of prioritizing security initiatives. We have found that the best-designed e-business applications have one-quarter as many security defects as the worst. By making the right investments in application security, companies can out-perform their peers — and reduce risk by eighty percent. Here's how.

Corporate perimeters are rapidly dissolving, and Internet-facing applications are increasing in both number and reach. Line-of-business application initiatives — not IT infrastructure projects — are the engines driving technology investments. As applications have become more complex, so too have the types of risk.

Five years ago the favored munitions of cyber-attack — denials of service and website defacements — were crude Molotov cocktails aimed at firms' network infrastructures. Today, companies must defend against increasingly sophisticated threats that target vulnerabilities of specific applications [1].

@stake analyzed forty-five e-business applications to profile the state of security as it is practiced today. We focused on the applications themselves — rather than the firewalls and related network infrastructure — for two reasons. First, application-level attacks can traverse most firewalls with ease. Second, as Willie Sutton famously remarked, it's where the money is.

This article is the second in a series about Return on Security Investment (ROSI). Early findings appeared in the Q4 2001 issue of *Secure Business Quarterly*, which discussed how secure software engineering techniques can provide tangible return on investment by reducing developer re-work [2].

In this article, we provide empirical detail on nine classes of common security flaws that cause applications to become insecure. We discuss the impact of design choices on application security, and identify the most common application security mistakes. We also compare and contrast the *outliers* in our study — that is, the top and bottom performers as measured by business risk — as a way of benchmarking application security best practices. Finally, we recommend a course of action. Our analysis will enable corporate risk managers to better identify and manage sources of risk in firms' e-business portfolios. In addition, our data will help e-business executives, IT planners and quality assurance teams justify strategic investments to improve the security of applications.

The need for security risk analytics

In the last five years, the Internet has become a key part of most companies' business strategies. Even in the current economic climate, firms continue to fund new e-business initiatives at a rate that far outpaces the growth in IT spending as a whole [3]. But is the Internet a *safe* place to do business? In the face of mounting evidence that suggests that it is not [4], senior management, audit committees, outside investors and insurance firms increasingly require companies to formally measure and manage their e-business portfolio risk [5]. Few companies are aware of the security exposures of their deployed e-business applications. Even fewer understand how their efforts compare with prevailing practices — whether the relative risks presented by their e-business applications lead — or lag — the state of the art.

Historically, there have been few methods available for managers to use for measuring application security risk. Security firms, insurance companies, market strategists and product vendors have used techniques such as weighted questionnaires, annual loss expectancy, penetrate-and-patch numbers, and plain old FUD (fear, uncertainty, and doubt) statistics. While these can be useful, we believed a comparative, data-driven approach would shed more insight on application security best practices.

Show me the data

We studied the security practices of applications encountered during @stake's client application assessments. Using a subset of @stake's engagement data over an eighteen-month period between February 2000 and July 2001, we created anonymized security profiles for forty-five (45) e-business applications and their potential risk to our clients' business [6].

The applications we analyzed were responsible for generating $3.5 billion in revenues for our clients [7]. Applications in the analysis included commercial packages from leading software companies, middleware platforms and end-user e-commerce applications. This information has helped us achieve a high level of rigor in classifying and quantifying the risk associated with modern e-business applications.

To gain an understanding of the security of a typical application, we examined each assessment in our sample in detail, classifying each security defect we found based on the type of vulnerability, degree of risk, and potential level of business impact (see *Exhibits A* and *C*). In the assessments we examined, we found nearly five hundred significant security defects related to authentication, input validation, buffer overflows, session management and other areas, with an average ten to eleven defects per assessment.

Exhibit A: Classifying Application Defects: the OWASP Framework

The Open Web Application Security Project (OWASP) is an industry group whose mission is to define common frameworks for secure web applications. The group is staffed and advised entirely by volunteers from leading companies, including Charles Schwab, @stake, IBM and Security Focus.

In our analysis, we classified our application security defects into nine (9) high-level and fifty-six (56) lower-level categories based in part on OWASP's Application Security Attack Components taxonomy. For more information, see the OWASP website (www.owasp.org) for complete definitions, details, and tools. *The Exhibit F: Application Defect Taxonomy* at the end of this article lists the complete set of categories we used.

Design issues dominate

@stake observed that many companies treat security as a “penetrate and patch” activity typically done after an application is deployed, rather than employing secure software engineering practices that would have produced a safer application from the start. This is akin to an automotive manufacturer ensuring quality by banging out the dings with rubber mallets at the end of the assembly line — or worse, after receiving customer complaints — rather than designing a better manufacturing process from the beginning.

Our anecdotal observations are confirmed by the empirical findings from our analysis. We found security design flaws in 70 percent of the defects we analyzed (see *Exhibit B*). After we excluded flaws that were of low business impact or were not easily exploitable, nearly half (47 percent) of the remaining serious defects could have been caught — and fixed inexpensively — during the design stage.

Application security design: common pitfalls

Application security flaws are generally introduced early in the design cycle. Given the severe scarcity of developers trained to write secure code, we were not surprised to see the same classes of issues repeated again and again. Our review of the defect data uncovered several recurring patterns.

First, most firms do not adequately provide secure authentication and access control features within applications. Nearly two-thirds (62 percent) of applications we assessed suffered from poor design and implementation choices that allowed access controls to be bypassed. Over one-quarter of the applications permitted user passwords to travel over the network unencrypted, where they could easily be stolen. Twenty-seven (27) percent of applications lacked password policies or controls that would have helped lock out would-be intruders trying to brute-force the login process. And despite the widespread popularity of cryptography for use with Secure Sockets Layer (SSL), one-third of companies stored sensitive information such as user passwords, confidential data and encryption keys insecurely.

Second, e-business applications typically trust user input implicitly or rely on client-side validation, rather than having the server check for inappropriate data. For example, a common trick with attackers is to submit web forms that contain embedded HTML, JavaScript, or overly long strings that do not conform to what the developer intended. Under the right conditions, this can cause the web server to fail, inadvertently disclose confidential information or redirect unsuspecting users to a server of the attacker’s choosing (this is referred to as a “cross-site scripting” vulnerability). Input validation errors plagued over two-thirds — 71 percent — of the applications in our sample.

Third, user session security remains the Achilles heel of most e-business applications. Most web application servers assign a unique, random number — called a session identifier — for users when they log in, which is used over the duration of the session to identify them. The application typically associates the session identifier with the user’s state — that is, which pages have been visited, the contents of a shopping cart, and where the user is in the purchase process. The session identifier is normally

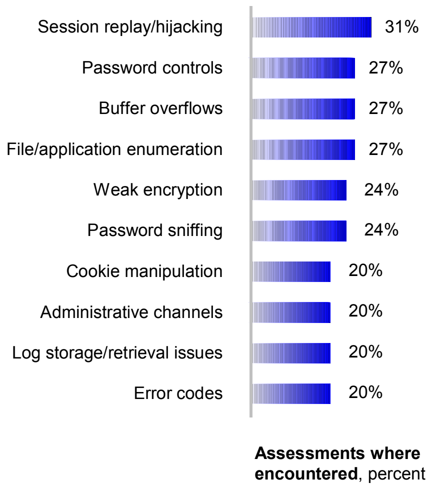
Exhibit B: Application Security Practices

Security Defects by Category

Category	Engagements where observed	Design related	Serious design flaws*
Administrative interfaces	31%	57%	**36%**
Authentication/access control	62%	89%	**64%**
Configuration management	42%	41%	**16%**
Cryptographic algorithms	33%	93%	**61%**
Information gathering	47%	51%	**20%**
Input validation	71%	50%	**32%**
Parameter manipulation	33%	81%	**73%**
Sensitive data handling	33%	70%	**41%**
Session management	40%	94%	**79%**
Total	**45**	**70%**	**47%**

*Scores of 3 or higher for exploit risk *and* business impact

Top 10 Application Security Defects



stored in a cookie in the user's browser or in an encoded URL. However, when the session is being conducted in the clear — unencrypted, without using SSL — a malicious attacker need only steal the session identifier to be able to masquerade as that user; obtaining the password is unnecessary. Session hijacking, therefore, is a serious risk. Thirty-one (31) percent of the applications we examined contained security defects that left them vulnerable to this form of attack.

The combination of these three patterns — among others — means the typical e-business application is at serious risk of compromise. As noted, of the ten to eleven average defects we found in our survey, 70 percent were design-related. This contrasts sharply with network security defects, which are predominantly configuration-related in nature [8]. Moreover, nearly half of application security defects — 47 percent — should be regarded as significant design flaws: they are both readily exploitable *and* could cause significant loss of reputation or customer revenue.

What is most surprising about the defects we found was not their severity, but the degree to which they were entirely preventable. Armed with the right skills and tools, we believe the companies in our survey could have readily detected and fixed the defects we found during the design cycle. As one CEO who endured a humiliating public hack put it: "we were penny-wise but pound-foolish." [9]

Separating the best from the worst

Although the state of application security practice is grim, not all applications are created equally. To understand what differentiates one application from another, we used an analytical technique called *outlier analysis* on twenty-three of the assessments in our survey. For each engagement, we calculated an overall *business risk index*, based on the sum of the individual BAR scores (see *Exhibit C*). The engagements were ranked

Exhibit C: Analyzing Business-Adjusted Risk

The Hoover Project is an initiative within @stake to provide security consultants and analysts with a vocabulary and framework for quantifying risks and returns of security. A significant portion of the project focuses on mining historical client engagement data for patterns and trends.

The Hoover repository contains thousands of qualitative and quantitative findings from customer engagements. For this paper we selected a subset (forty-five) of the application-related engagements completed to date. We filtered the data to include only security defects that were application-related (database, application server software and packages, middleware and custom code). We excluded network infrastructure-related findings (firewalls, servers, network hardware).

For each security defect we found, we calculated a *business-adjusted risk (BAR)* score as follows:

	Business impact	1-5 score
x	Risk of exploit	1-5 score, depending on business context
=	Business adjusted risk	1-25 score

Risk of exploit indicates how easily an attacker can exploit a given defect. A score of 5 denotes high-risk issues that are well known and can be exploited with off-the-shelf tools or canned attack scripts. A score of 3 indicates that the defect requires intermediate skills and knowledge to exploit, such as the ability to write simple scripts. Finally, certain classes of defect can only be exploited by a professional-caliber malicious attacker; these were given a score of 1.

Business impact indicates the damage that would be sustained if the defect were exploited. An impact score of 5 represents a flaw that could cause significant financial impact, negative media exposure and damage to a firm's reputation. A score of 3 indicates that a successful exploit could lead to limited or quantifiable financial impact, and possible negative media exposure. Those defects that would not have significant impact (monetary or otherwise) received a score of 1.

Business-adjusted risk is a simple tool for measuring risk: the higher the number, the higher the risk. Because BAR includes relative ratings for both likelihood of occurrence and business impact, it behaves in a manner similar to Annual Loss Expectancy (ALE) calculations used by insurers. A BAR score of 20, for example, denotes an order of magnitude more risk than 2.

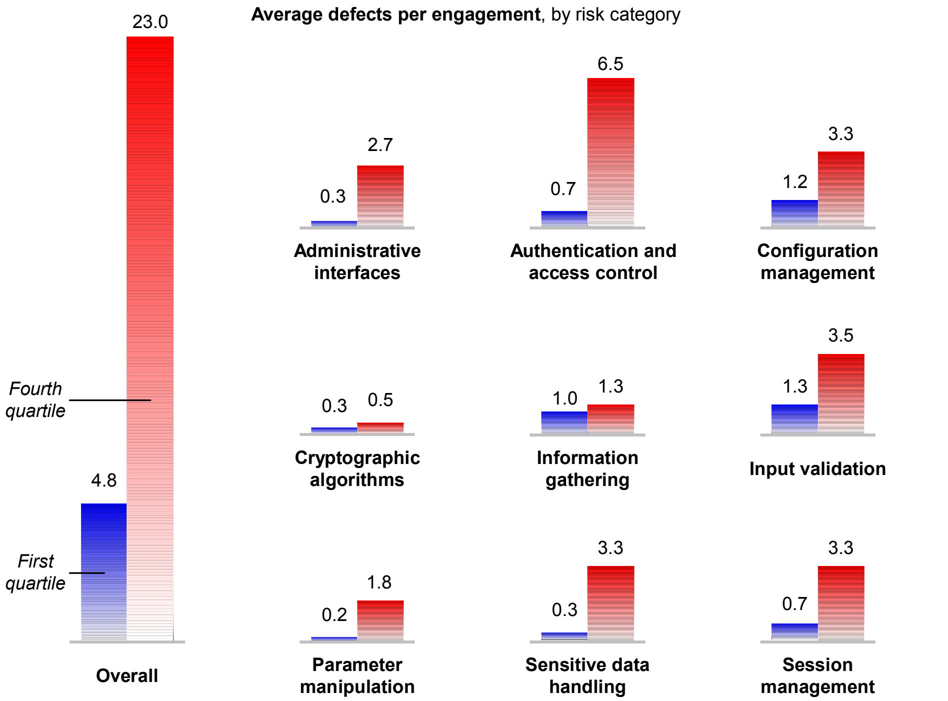
by their index score, from best to worst, and divided into quartiles. Engagements with the lowest business risk index formed the first quartile; those with the highest formed the fourth.

Of the applications in our analysis, we found that the most secure contained, on average, about one-quarter of the number of defects compared to the least secure (see *Exhibit D*). The top performers' reduced defect rates translated into much lower risk scores as well. The least secure applications carried, on the basis of business-adjusted risk, nearly *six times* more risk than the most secure (*Exhibit E*).

Security secrets of the top performers

Most of the components that make up modern e-business applications are largely interchangeable. Development languages, web servers, application servers, middleware and databases are well established in the marketplace, and all can be used in both secure and insecure manners. We conclude, therefore, that the difference between the top- and bottom-quartile performers is due to superior *practices* in designing, coding and deploying secure applications, rather than any inherent advantage bestowed by the components themselves.

Exhibit D: Security Comparison of First and Fourth Quartiles



Our analysis reveals six patterns that differentiated the top-quartile assessments from the bottom:

1. **Early design focus on user authentication and authorization**. Leading practitioners design and implement secure methods for managing user passwords and privileges. Some firms opt to integrate their application's login processes with third-party authentication and access control products such as RSA's SecurID or Netegrity SiteMinder. Others prefer a customized approach using LDAP in combination with JSSE or RSA cryptography [10]. The most secure applications ensure passwords are sent encrypted rather than in plaintext, and do not rely on the web server's "Basic Authentication" feature. As a result, applications in the top quartile of our analysis had, on average, 89 percent fewer authentication and access control-related defects as the bottom quartile, which represented an 88 percent reduction in risk (10.3 average business risk score versus 85.2).

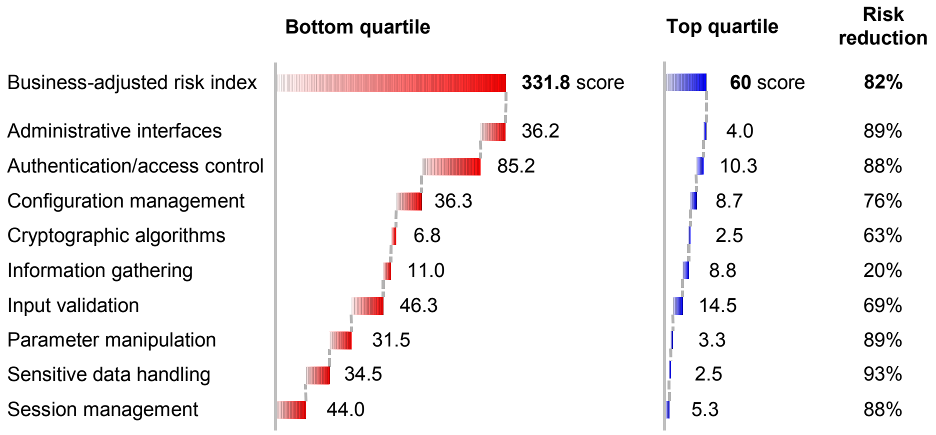
2. **Mistrust of user input**. Secure applications essentially treat end-users as hostile agents. All user-submitted data is checked on the server side — in addition to the client — to make sure inputs conform to business rules prior to processing by the application. Strings that could contain HTML or JavaScript are stripped of suspicious tags, and are checked against length constraints to prevent buffer overflows. In our analysis, assessments in the top quartile contained one-and-a-half issues related to input validation or parameter manipulation. In contrast, those in the bottom quartile contained

between five and six issues, nearly four times as many. This represents a fourfold increase in business risk, with an average risk index of 77.8 versus 17.8. Top-quartile companies, therefore, reduced their business risk by 77 percent relative to the bottom.

3. **End-to-end session encryption**. Because of the all-important role session identifiers play in tracking logged-in users on a website, it is critical to protect this vital piece of information. Since most applications create and manage session identifiers insecurely, the easiest way to prevent an eavesdropper from hijacking a user's identifier is to encrypt the entire session from login to logout, not just the sensitive portions such as checkout or payment. In addition, the web application server should enforce short session timeout periods — for example, after a short period of inactivity users should be required to log in again, thus generating a new session identifier. Therefore, even if a rogue attacker obtains a user's session identifier, the window of attack in which it can be "replayed" will be narrow. In our survey, top-quartile applications had nearly 90 percent less business risk associated with session management issues compared to the bottom quartile (44 average business risk score versus 5.3).

4. **Safe data handling**. While session encryption is critical to safeguarding information in motion, top-quartile companies also take steps to ensure that it stays secure *while at rest*. That means encrypting user passwords within back-end databases, removing hard-coded passwords and backdoors from application code, and segregating stored information so that customers cannot access each other's data, even in the event of a compromise. When it comes to cryptography, leading companies stick with tried-and-true algorithms such as RSA, AES, 3DES and Diffie-Hellman [11], and make sure that random numbers used in the calculations really *are* random. Our analysis showed that bottom-quartile companies had ten times the number of issues

Exhibit E: Business Risk Averages — The Good, the Bad and the Ugly



Average business-adjusted risk (BAR) index per engagement, with breakdown by risk category with respect to sensitive data handling and cryptography, which represented an eightfold increase in risk (41.3 combined business risk score versus 5.0)

5. **Elimination of administrator backdoors, mis-configurations and default settings**. Most commercial software and web application servers ship with example code and demonstration applications to help developers learn the package. In addition, many packages install developer utilities or administrative features onto the server by default. Leading companies eliminate or lock down these features because they often contain significant security flaws. As a result, for companies in the top quartile, business risk associated with configuration management and administrator interface issues was 82 percent less than with companies in the bottom quartile (12.7 average business risk score versus 72.5).

6. **Security quality assurance**. While not reflected in our data, we noted anecdotally that many of the firms whose applications fared best in our assessments were also the ones who treated security as a core discipline. Some companies have set up dedicated Security Quality Assurance (SQA) teams to ensure that potential security defects are identified, tracked and eliminated during the course of development. In addition, tools [12], methodologies and training provide valuable support for developers, eliminating expensive re-work costs and minimizing reputation damage.

A plan for action

Our analysis shows that application security defects are common but unevenly distributed. As a result, the least secure applications carry, on average, six times as much business risk as the most secure. Many companies seek guidance on how to manage their e-business portfolio risk. @stake recommends firms focus on six areas:

- **Stop depending on the firewall**. With the emergence of targeted application attacks, firewalls are now no more than a speed bump into (and out of) your company. Recognize that your security team will need to adjust their skill sets and infrastructure to better protect applications. Identify and fund initiatives to protect sensitive data and implement stronger session, authentication and access controls. Work with your IT group and developers to prioritize efforts, based on the risk areas identified in this paper.

- **Act up**. Commercial applications contain flaws that you may not know about. If you enjoy good relations with your software vendors, ask them to provide you with documentation on how to use and deploy their applications securely. You should also ask for copies of the results of any security assessments they may have had.

- **Educate application developers**. Security mistakes cannot be avoided if your developers do not know about them. Several firms provide high-quality training in secure software engineering techniques. To help your development team reduce security defects and increase quality, send them to classes at least once a year.

- **Assess early and often**. Nobody likes surprises. Do not wait until your application goes live — integrate security into your application from the start. Assess security during design; before, during, and after development; and prior to testing and deployment. Security consulting firms can deliver focused expertise to quickly identify your security needs, and create a roadmap for implementing solutions.
- **Engage Finance and Audit**. E-business risk management is a boardroom issue. Allocating funds for application security due diligence becomes much easier when senior management understands the risk reduction potential that can be achieved with targeted investments in security.
- **Get outside help**. Security is a scarce skill set in most companies. In most cases, an outside firm can provide fresh insight into industry-wide practices, emerging attack techniques, and the latest methods for writing secure code. Outside assessors can also transfer knowledge to your team so that they can incorporate Security Quality Assurance into current and future e-business initiatives.

We hope this paper provides useful insight on current application security practices. The data presented in the preceding pages should give you insights on how to develop strategies for managing e-business portfolio risk. By making strategic investments in application security, we expect that most firms can significantly reduce their exposure, safeguard their reputations and gain competitive advantage.

While every precaution has been taken in the preparation of this document, @stake assumes no responsibility for errors, omissions or damages resulting from the use of the information herein. Products or corporate names may be trademarks or registered trademarks of other companies and are used only for the explanation and to the owner's benefit, without intent to infringe

Exhibit F: Application Defect Taxonomy

Risk management begins with risk classification. The Hoover Project uses a multi-level classification scheme to ensure that findings can be equitably compared, aggregated and contrasted. The list below shows the categories we used. The nine top-level categories correspond to those that appear in *Exhibits B, D,* and *E*.

Administrative interfaces

- Administrative channels
- Log storage and retrieval
- Public interfaces

Authentication & access control

- Brute force
- Email interception
- Implicit component trust
- No authentication
- Password controls
- Password sniffing
- Authentication - other

Configuration management

- Classpath mis-configuration
- Configuration file integrity
- Default accounts
- Default services
- File permissions
- License checking
- Privileged applications
- Sample code
- Untrusted service reliance
- Vendor patches
- Configuration management - other

Cryptographic algorithms

- Hardcoded credentials
- Random number generation
- Weak encryption
- Cryptographic - other

Information gathering

- Account enumeration
- Browser cache
- Browser history
- Client-side comments
- Debug commands
- Error codes
- File/application enumeration
- System & user information - other

Input validation

- Buffer overflows
- Case sensitivity
- Client-side validation
- Cross-site scripting
- Direct OS commands
- Direct SQL commands
- Meta characters
- Null characters
- Path traversal
- Unicode encoding
- URL encoding
- Input validation - other

Parameter manipulation

- Cookie manipulation
- Form field manipulation
- URL manipulation
- Parameter manipulation - other

Sensitive data handling

- Credential storage
- Data segregation
- Database mis-configuration
- Sensitive data handling - other

Session management

- Cleartext data
- Session replay/hijacking
- Session management - other

Notes and references

[1] In the last year, the security community has documented an increasing number of vulnerabilities targeting business applications such as Microsoft SQL Server, Oracle and Internet Information Server, in addition to those targeting personal productivity applications like Outlook. John Pescatore, an analyst with Gartner Group, notes that "the current generation of firewalls focuses on the network level, kind of like the walls of a fort stopping direct attack. However, close to 75 percent of today's attacks are tunneling through applications." Source: *Computer World*, "Airline Web Sites Seen As Riddled With Security Holes," February 04, 2002.

[2] K. Soo Hoo, A. Sudbury, A. Jaquith. "Tangible ROI Through Secure Software Engineering," *Secure Business Quarterly*, Q4 2001.

[3] Companies continue to be bullish about e-business, according to a recent corporate technology spending survey by management consultancy A.T. Kearney. Eighty-three (83) percent of firms plan to initiate new e-business projects in 2002. Approximately one-half of all companies surveyed will increase e-business spending compared to last year, with an average increase of 10.6 percent compared to 4.4 percent for IT overall. Source: A.T. Kearney, *E-Business Outlook: 2002*.

[4] CERT/Carnegie-Mellon recently reported that it had recorded over 56,000 security incidents for calendar year 2001. Since 1999, the number of incidents reported has quintupled in absolute terms. This represents a rate of increase that far outpaces the growth of e-commerce as a whole. Source: CERT/Carnegie-Mellon.

[5] Expected to come into force in 2004, the Basel II Capital Accords require financial services firms to explicitly set aside cash reserves to cover operational and market risk. This is in addition to reserves required for credit risk. Digital security comprises a significant portion of operational risk. Firms that can demonstrate effective management of their e-business portfolio risk will be allowed to reduce the amount of operational set-aside. Therefore, better application security will — literally — be like money in the bank.

[6] Every statistical survey contains potential sources of bias, including this one. The e-business application assessments in this study, for example, are not truly random — they were essentially "self-selected" by the companies commissioning the analyses. A reviewer of this paper suggested that a reason we found so many mistakes could be because those companies who commissioned @stake *needed* the security assessment. One could argue, however, that the applications *not* represented in the survey might be even worse. We make no claim as to which is the more persuasive argument.

A second potential source of bias relates to how we collected our data. We identified security defects primarily through methods such as interviews, manual inspection and facilitated design reviews — rather than strictly through the use of tools. Application assessments do not always lend themselves to tool-based analysis; no tool can reverse-engineer a developer's intent. @stake's application assessment methodology ensures a high degree of analytical rigor through the use of consistent assessment yardsticks, strict documentation controls and a structured peer review process. We believe tools have their place — they complement but cannot substitute for good old-fashioned consulting.

[7] Revenue figures are annualized, based on total product sales or online revenues directly related to the applications we analyzed. We used a combination of public and private data sources to derive our estimate. Because we could not obtain reliable revenue information for many of the applications in our sample, the total revenue figure of $3.5 billion should be regarded as conservative.

[8] Based on @stake proprietary sources, chiefly the Hoover Project.

[9] L. McCauley, "Anatomy of a Break-In," *Secure Business Quarterly*, Q3 2001.

[10] Java Secure Socket Extension (JSSE) provides integrated session security for Java applications, and was introduced as part of the Java 2 specification. Lightweight Directory Access Protocol (LDAP) is a commonly used method for accessing user and resource information stored in corporate directories. RSA is the *de facto* standard for public-private key cryptography. The acronym refers variously to the inventors of the standard (Ron Rivest, Adi Shamir, and Leonard Adelman), the algorithms themselves, or the company that bears their name (RSA Security).

[11] Dan Geer, @stake's CTO, has stated that there are only twenty truly great cryptographers working in the world today. Given the shortage of this specialized skill set, we contend that do-it-yourself cryptography is nearly always fatally flawed. We have seen several clients make outlandish claims about the strength of their proprietary algorithms. In one case, a client boasting about the strength of their homegrown encryption code was rudely surprised when our consultants exhausted the number of possible encryption keys in fifteen minutes, *using pencil and paper*.

[12] In contrast to network security tools, there are few security tools geared towards software developers. The situation has improved markedly in the last year, however, with the emergence of "black-box" application security testing tools from vendors such as Cenzic (formerly ClickToSecure), Sanctum and Qualys. Open source alternatives (OWASP's Security Testing Framework, WebSleuth) are also emerging. We are not aware of any "white-box" tools that explicitly integrate into development environments, although these should be forthcoming. We expect that traditional software quality assurance heavyweights such as Mercury Interactive to become much more active in this area.