They argue that their DNS resolvers already support DNSCurve and use it whenever possible. But, they then go on with the caveat that, "Of course, authoritative servers need to be upgraded to support DNSCurve as well..." which is something that has not happened. But the debate continues to hold back deployment.

There are also vexing problems on the consumer or client side of the DNSSEC issue. Some ISPs have indicated that they will not be providing DNSSEC validation because they have seen no demand for it.[22] This is hardly surprising, since the attacks have been few and DNSSEC is designed to be largely transparent to the end consumer. There doesn't seem to be a pain point on the consumer side to help push adoption. This should be a simple thing to provide. It's just a matter of enabling an option in the caching name servers, which most installations should now support.

For zones that are not signed, this situation results in no increase in server load and has no impact at all. If a domain is signed, those signatures can be checked and the failing records dropped. The end consumer may not even see that something has happened to help protect him. He might not even know if it is not checked and he does get trapped by some attack. This transparency problem has the associated problem of creating a lack of demand for a feature that really should be just there. Lacking some mandate from the registry authorities to the ISPs, this may be difficult to overcome.

This applies equally, if not more so, to change-adverse IT departments in corporations that are unwilling to make changes even if it is "doing the right thing." The risk, no matter how minuscule, of causing something to break due to a change which isn't going to provide them with some overt, observable, benefit can create reluctance in a corporate environment.

What's ahead now

Now that the root zone has been signed and more and more of the TLD zones are being signed, we can expect to see more progress in the coming years and some new and fresh ideas for taking advantage of DNSSEC. Already, proposals have been put forth to add e-commerce certificate hashes in DNS to firmly tie a certificate to the domain holder through the use of DNSSEC to sign those records.[23] While this is certainly no substitute for a Certifying Authority, it helps build trust in the authenticity and reliability that sites are what they say they are. There have also been proposals and discussions for IPsec keys and information to be overloaded into the DNS and authenticated by DNSSEC to facilitate opportunistic encryption and VPNs. These are desirable features that make DNSSEC more valuable but must wait for DNSSEC-aware applications.

It's important to keep in mind that DNSSEC was designed to deal with one particular threat, that of DNS spoofing and falsified DNS data. It is not the be all and end all of DNS security. But it has a valuable role to play. And, as we can learn from some of the proposed uses for DNSSEC, there may yet be other uses to which DNS with DNSSEC may be applied.

Conclusion

2010 was a watershed year for DNSSEC with many important milestones passed. Some would argue that DNSSEC has finally reached a critical mass and momentum is building behind it. But, in spite of all the good press, DNSSEC still has a long way to go in achieving true end-to-end validation of the Domain Naming System.

22 DNSSEC Deployment Among ISPs – The Why, How, and What – http://www.circleid.com/posts/20100629_dnssec_deployment_among_isps_the_why_how_and_what/
23 Dan Kaminsky's "The DNSSEC Diaries" – http://dankaminsky.com/2010/12/13/dnssec-ch1/

Section III—Developing Secure Software

In Developing Secure Software, we present data surrounding proven processes and techniques for developing secure software. We discuss how enterprises can find existing vulnerabilities and help prevent new ones from being introduced. If you use networked or web applications to collect or exchange sensitive data, your job as a security professional is harder now than ever before. We take a look at both the static and dynamic security testing done by the Rational® AppScan® group in all stages of application development and share insights on what was discovered.

Further analysis on web application trends

IBM Rational Security and Compliance provides further analysis on web Application Security trends in this year's report in two different ways. Continuing from its 2009 research, IBM® Rational® AppScan® onDemand Premium Service derives trends on web application vulnerabilities from 2010 assessment data. Additionally for this year's report, new automated technologies in the IBM® Rational® AppScan® portfolio are able to provide visibility to an organizational blind spot regarding web Application vulnerabilities.

Conclusions from real-world web application assessments

Methodology

IBM has collated real-world vulnerability data from hundreds of security tests conducted in 2010 from the IBM® Rational® AppScan® OnDemand Premium Service. This service combines application security assessment results obtained from IBM® Rational® AppScan® with manual security testing and verification. In all cases, identified false positives were removed from the results and the remaining vulnerabilities were categorized into the following key security categories:

- Cross-site request forgery
- Cross-site scripting
- Error message information leak
- Improper access control
- Improper application deployment
- Improper use of SSL
- Inadequate or poor input control
- Information disclosure
- Insufficient web server configuration
- Non-standard encryption
- SQL injection

For each of these categories, two core metrics were calculated:

1. The percent chance of finding at least one vulnerability in that category.
2. The average number of vulnerabilities that are likely to be found in that category.

Having collated similar data since 2007, it was also possible to trend this data over the past four years. In 2010 additional metrics were also captured for each test data point to gain deeper analysis of the data. This included the following areas.

Business Segment where test data was attributed to belong to one of the following:

- Financials
- Industrials
- Information technology
- Logistics
- Retail
- Other

Application Security Test Cycle depicting the type of test the application was involved in:

- One-time assessment—Applications tested for the first time
- Quarterly assessment—Applications tested in a regular, ongoing basis
- Retest—Follow-up test to confirm the findings (typically from the one-time assessment)

Application Technology depicting the main technology used to develop the application:

- ASP.NET application
- Java-based applications (including JSP)
- PHP-based applications

2007—2010 Application vulnerability trends

Several conclusions can be derived from our application assessment data, many of which indicate trends in the susceptibility of websites to these vulnerabilities. Since we started recording application security statistics in 2007 we have seen a steady decline in the instances of cross-site scripting (XSS) while, at the same time, cross-site request forgery (CSRF) has increased. In 2010, for the first time, we now find that CSRF is more likely to be found in our testing than XSS.

This change is attributed to better detection techniques for CSRF and also a greater awareness of the risk. We find that some organizations tolerate having some outstanding issues with CSRF if the risk of exploitation is minimized. This is generally not the case with XSS and these issues are often quickly resolved.

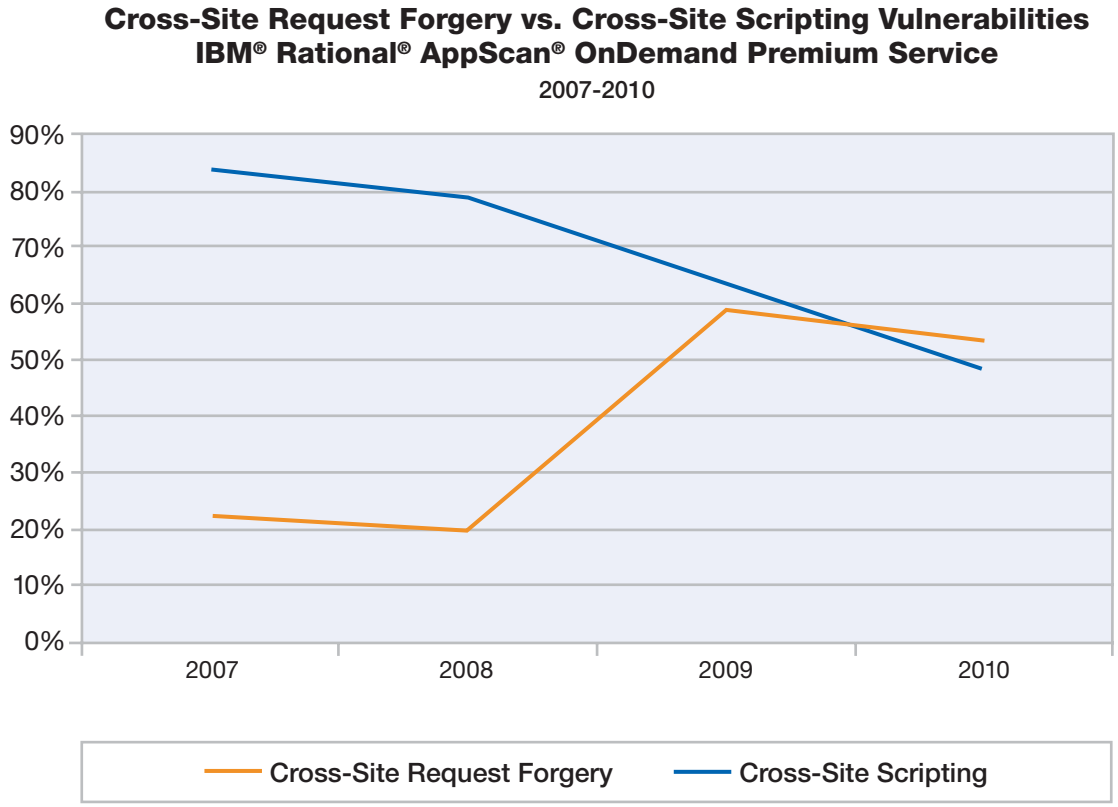

Figure 67: Cross-Site Request Forgery vs. Cross-Site Scripting Vulnerabilities IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

The true risk from CSRF is dependent on the specific application transaction that is vulnerable. It can be a simple search page or a more volatile money transfer transaction. As a consequence, we find that each instance of CSRF should be fully investigated. In the cases where it is a search page, the business may choose to accept this or put it on a slower track for mitigation.

XSS and SQL injection are both attributed directly to a lack of input control in code. Although we are seeing that instances relating to input control are on the decline, it is not as steady as XSS. We still find it present in our testing in excess of 60 percent of the time. SQL injection instances increased slightly in 2010, but are still down considerably from the numbers we had in 2007. Our data suggests that better database controls and methods appear to be the main reason for the decline, rather than any specific improvement in the lack of input control.

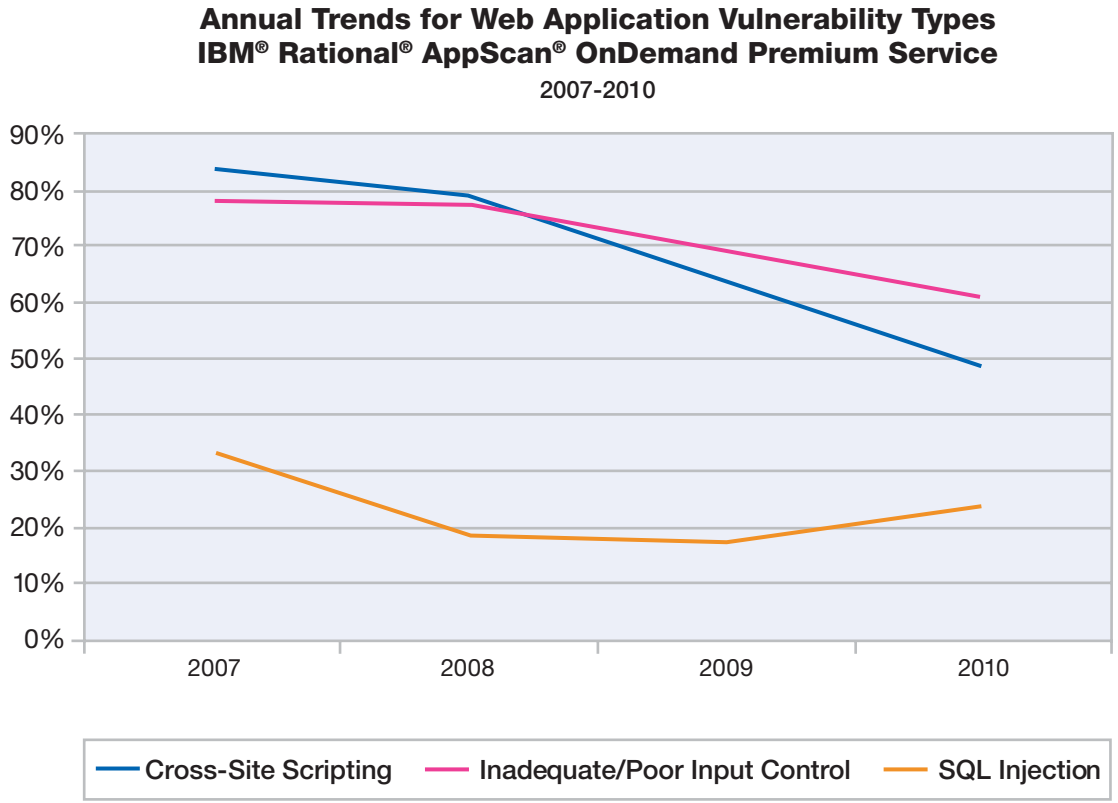

Figure 68: Annual Trends for Web Application Vulnerability Types IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

ANNUAL TRENDS								
Vulnerability Type	**2007**		**2008**		**2009**		**2010**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	1.9	22%	1.8	20%	7.9	59%	3.8	53%
Cross-Site Scripting	12.7	83%	17.9	79%	40.8	64%	5.8	49%
Error Message Information Leak	46.9	83%	22.6	74%	23.5	68%	15.3	56%
Improper Access Control	3.9	56%	2.4	67%	0.8	30%	0.9	31%
Improper Application Deployment	2.6	50%	3.2	54%	3.0	51%	1.9	33%
Improper Use of SSL	28.9	50%	23.8	74%	38.8	51%	26.4	60%
Inadequate / Poor Input Control	14.4	78%	28.1	77%	44.4	69%	10.5	61%
Information Disclosure	6.6	61%	8.7	63%	12.9	64%	16.6	84%
Insufficient Web Server Configuration	16.5	72%	5.4	46%	1.4	31%	4.4	44%
Non Standard Encryption	7.3	28%	2.4	17%	2.5	35%	1.6	22%
SQL injection	1.3	33%	5.3	19%	1.7	18%	2.3	23%

Table 14: Annual trends for Web application vulnerability types, 2007 – 2010, IBM® Rational® AppScan® OnDemand Premium Service

Business segments

As in 2009 we were able to split out our 2010 statistics by business segments. Where the number of data points would allow, we were able to split out data for five business segments.

In 2010, financial applications were again the best performing segment. Financial applications were found to not only have lower percentages attributed to the likelihood of finding each of the vulnerabilities covered, but they also have very low numbers for the instances of each finding found for a given test. So while XSS and SQL injection might be found in some financial applications, it would typically be an isolated occurrence and not a flaw seen throughout the application. The same is not true for applications for industrial and IT organizations.

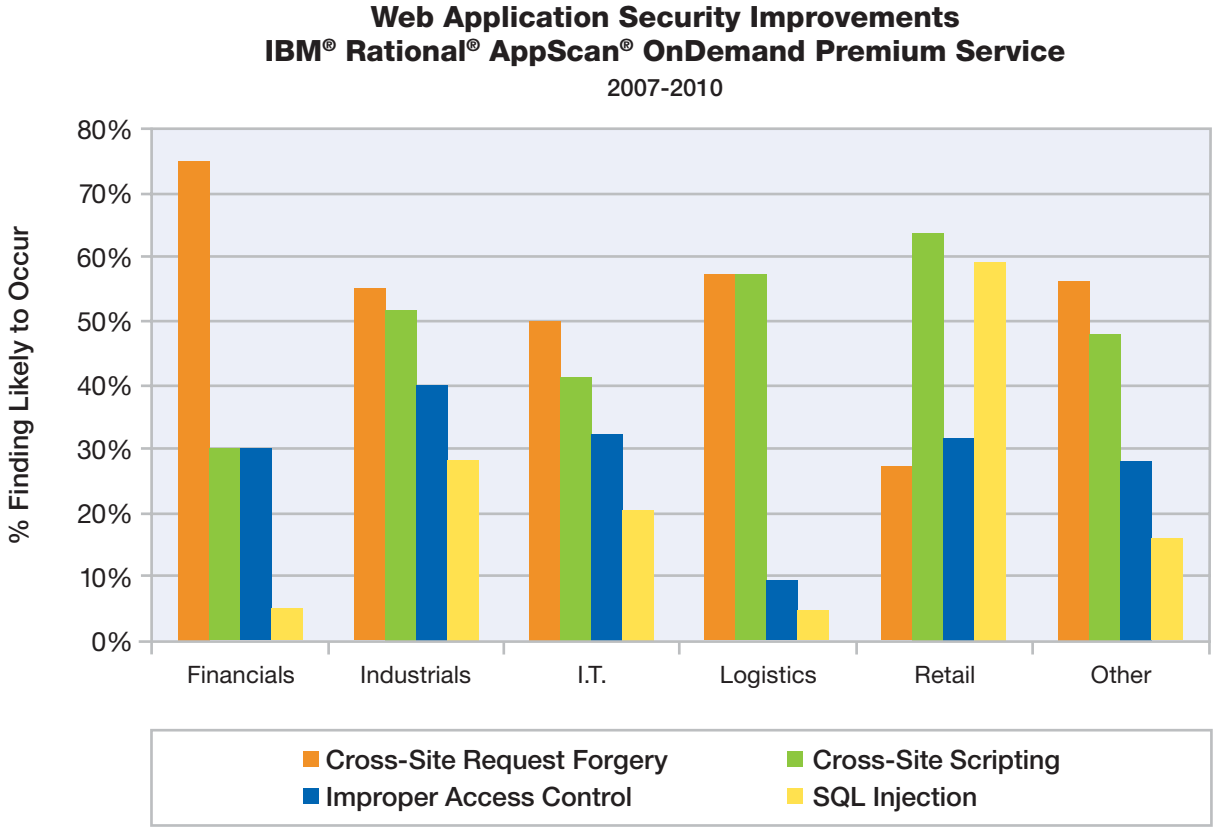

Figure 69: Web Application Security Improvements IBM® Rational® AppScan® OnDemand Premium Service – 2007-2010

	BUSINESS SEGMENT											
Vulnerability Type	**Financials**		**Industrials**		**Information Tech.**		**Logistics**		**Retail**		**Other**	
	Avg. vuln per test	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**	**Avg. vuln per test**	**% one vuln likely to occur**
Cross-Site Request Forgery	6.3	75%	2.6	55%	4.2	50%	9.2	57%	0.5	27%	2.4	56%
Cross-Site Scripting	0.4	30%	7.6	52%	6.4	41%	1.7	57%	2.6	64%	11.0	48%
Error Message Information Leak	10.9	80%	23.2	58%	14.6	47%	0.7	33%	17.8	59%	11.2	60%
Improper Access Control	0.4	30%	1.2	40%	0.7	32%	0.1	10%	2.2	32%	0.4	28%
Improper Application Deployment	2.4	55%	1.3	32%	2.1	24%	1.6	24%	0.4	27%	3.9	44%
Improper Use of SSL	32.1	90%	15.6	33%	19.4	50%	45.7	81%	20.0	73%	46.6	88%
Inadequate / Poor Input Control	3.5	40%	11.8	63%	13.8	65%	1.6	48%	11.3	82%	15.1	60%
Information Disclosure	10.9	75%	22.5	92%	17.4	82%	13.4	90%	7.8	82%	16.2	76%
Insufficient Web Server Configuration	1.0	50%	8.1	58%	4.4	44%	0.5	24%	2.3	27%	3.4	36%
Non Standard Encryption	2.1	10%	1.7	23%	0.7	24%	0.3	10%	4.6	41%	0.4	20%
SQL injection	0.1	5%	1.4	28%	5.0	21%	0.0	5%	7.0	59%	0.3	16%

Table 15: Most Prevalent Web Application Vulnerabilities by Industry, IBM® Rational® AppScan® OnDemand Premium Service

Application security test cycle

For the first time we collated data relating to the actual test cycle that was being conducted. This allowed us to see the correlation between the initial test of an application and the follow-up retest. In a pleasing way, the trend between these two statistics is that there is a significant decline in the likelihood of finding vulnerabilities in the retest. In many cases this reduction is more than half that of the original. This demonstrates the importance not only of testing applications, but also that follow up and mitigation are equally important.

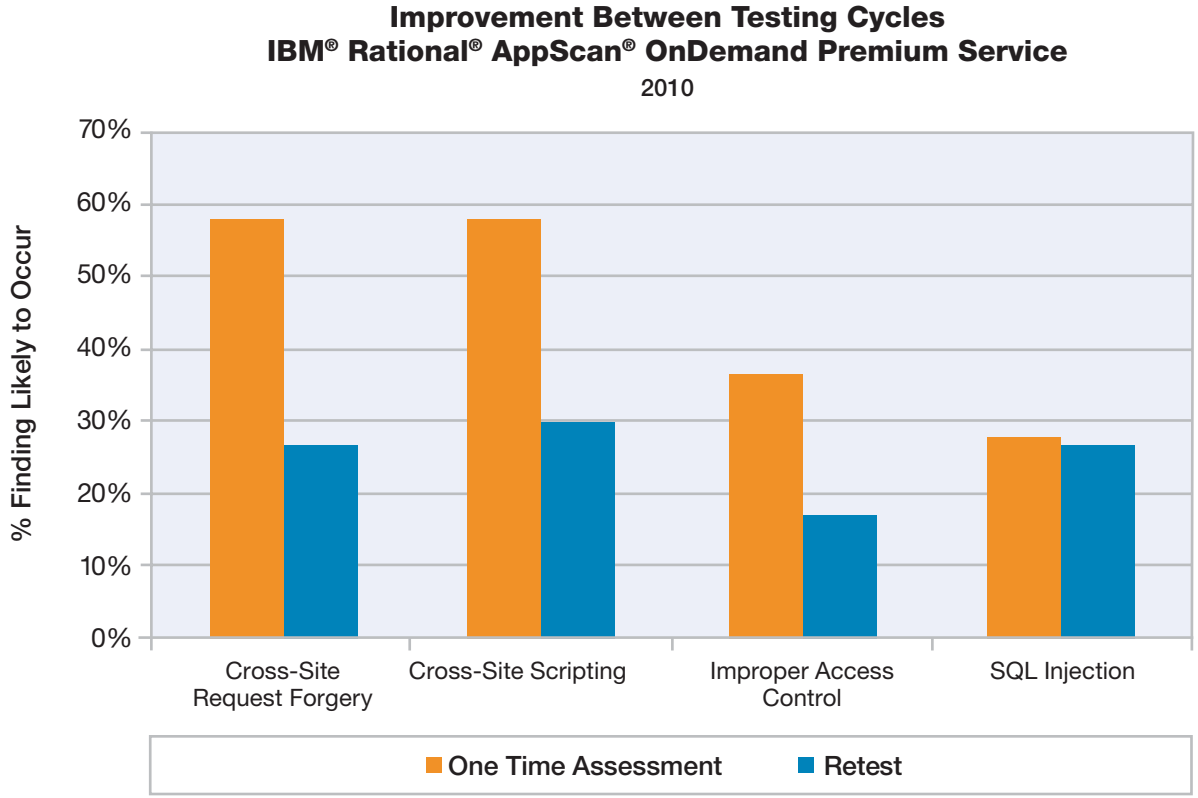

Figure 70: Improvement Between Testing Cycles IBM® Rational® AppScan® OnDemand Premium Service – 2010

SECURITY TEST CYCLE						
Vulnerability Type	**One Time Assessment**		**Quarterly Assessment**		**Retest**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	3.2	58%	7.8	58%	0.6	27%
Cross-Site Scripting	8.8	58%	1.0	35%	0.8	30%
Error Message Information Leak	22.5	63%	3.4	43%	4.5	43%
Improper Access Control	1.2	37%	0.4	25%	0.3	17%
Improper Application Deployment	2.4	35%	1.5	28%	0.4	30%
Improper Use of SSL	27.2	54%	35.5	83%	11.3	53%
Inadequate / Poor Input Control	15.8	74%	1.6	43%	2.3	33%
Information Disclosure	21.3	86%	10.3	78%	7.1	83%
Insufficient Web Server Configuration	6.0	48%	1.3	33%	2.7	40%
Non Standard Encryption	1.5	25%	1.3	13%	2.2	23%
SQL injection	3.3	28%	0.1	8%	1.4	27%

Table 16: Security test cycles by vulnerability type, IBM® Rational® AppScan® OnDemand Premium Service 2010

Application technology

Another new statistic for us in 2010 was taken from looking at the technology of the application. We were only able to split this across three types but this still showed some interesting results. ASP.NET applications were clearly more susceptible to SQL injection than Java or PHP. The likely reason is that ASP.NET applications would typically use SQL Server as a backend database. SQL injection is better documented and easier to detect in this technology.

PHP overall performed best of the three technologies. However, it is worth highlighting that our data is taken entirely from commercial applications.

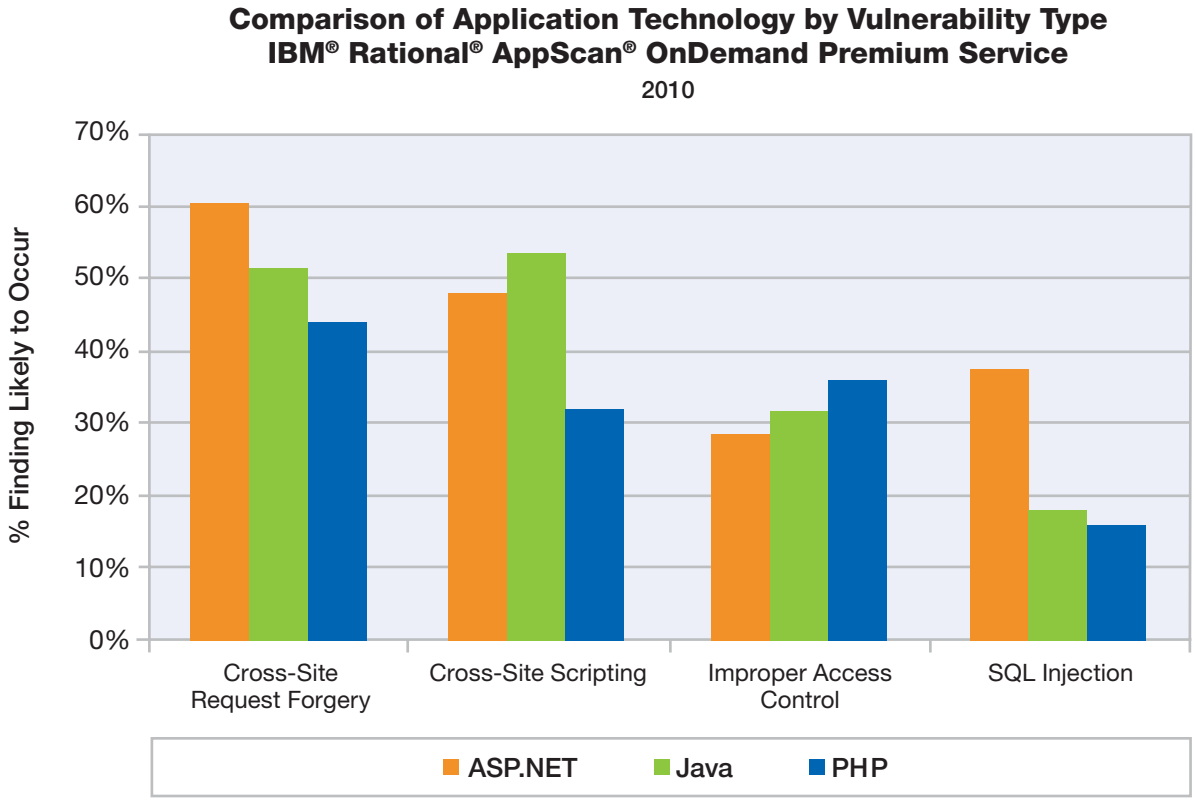

Figure 71: Comparison of Application Technology by Vulnerability Type IBM® Rational® AppScan® OnDemand Premium Service – 2010

APPLICATION TECHNOLOGY						
Vulnerability Type	**ASP.NET**		**Java**		**PHP**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	2.8	61%	4.4	51%	3.4	44%
Cross-Site Scripting	4.9	48%	7.2	53%	1.9	32%
Error Message Information Leak	23.6	71%	13.7	51%	3.4	40%
Improper Access Control	1.1	29%	0.8	32%	0.8	36%
Improper Application Deployment	2.5	48%	1.5	26%	2.0	28%
Improper Use of SSL	28.2	64%	28.8	55%	12.4	72%
Inadequate / Poor Input Control	10.6	66%	12.1	59%	3.7	56%
Information Disclosure	24.7	84%	14.5	88%	6.6	72%
Insufficient Web Server Configuration	3.0	50%	5.7	41%	2.3	44%
Non Standard Encryption	2.7	30%	1.1	17%	0.7	24%
SQL injection	3.3	38%	2.3	18%	0.2	16%

Table 17: Comparison of application technology by vulnerability type, IBM® Rational® AppScan® OnDemand Premium Service 2010

Hybrid analysis sheds light on vulnerability blind spot

Background and methodology

In the past ten years, many whitepapers, research articles, and Blog posts have been published on the subject of server-side web application vulnerabilities such as SQL injection, cross-site scripting, and HTTP response splitting. In addition, several projects such as the WASC web hacking incident database or the WASC statistics projects have tried to estimate the incidence of such issues in the real world.

On the other hand, there is a dearth of information and statistics on the incidence of client-side JavaScript™ vulnerabilities in web applications, even though these vulnerabilities can be just as severe as their server-side counterparts. We suspect that the main reason for this lack of information is that client-side vulnerabilities may be harder to locate, and require deep knowledge of JavaScript and the ability to perform code review of HTML pages and JavaScript files.

As Web 2.0, AJAX applications, and Rich Internet Applications (RIAs) become more common, client-side JavaScript vulnerabilities may become more relevant, with a potential rise in the amount of such issues being exploited by malicious hackers.

This summary presents the results of research performed by the IBM Rational application security group into the prevalence of client-side JavaScript vulnerabilities, using a new IBM technology called JavaScript Security Analyzer (JSA). JSA performs hybrid analysis by applying static taint analysis on JavaScript code collected from web pages and extracted by an automated deep web-crawl process. From our perspective, this kind of analysis is superior to—and more accurate than—regular static taint analysis of JavaScript code because it includes the entire JavaScript codebase in its natural environment: fully rendered HTML pages and the browser's Document Object Model (DOM).

The research used a sample group of approximately 675 websites, consisting of all the Fortune 500 companies and another 175 handpicked websites, including IT, web application security vendors, and social networking sites. In order to avoid damage to the sites or interference with their regular behavior, we used a non-intrusive web crawler, similar to that of a web search engine, which retrieved approximately 200 web pages and JavaScript files per site into a repository. We then used the JavaScript Security Analyzer to analyze these pages offline for client-side JavaScript vulnerabilities. We concentrated on two main types of issues: DOM-based cross-site scripting, and open redirects.

JavaScript analyzer results

The results of our research were quite disturbing: about 98 sites (14 percent) of the 675 sites suffer from many severe client-side JavaScript issues, which could allow malicious hackers to perform attacks such as:

- Infecting users of these sites with malware and viruses.
- Hijacking users' web sessions and performing actions on their behalf.
- Performing phishing attacks on users of these sites.
- Spoofing web contents.

The troubling fact about these statistics is that most organizations have no efficient process or automated solution to assist them with the task of locating these types of issues.

Our research also showed that 38 percent of the vulnerable sites suffered from these vulnerabilities as a result of using third party JavaScript code such as:

- Marketing campaign JavaScript snippets.
- Flash embedding JavaScript snippets.
- Deep linking JavaScript libraries for Adobe® Flash and AJAX applications.
- Social networking JavaScript snippets.

Percentage of Sites Vulnerable to Client-Side JavaScript Issues

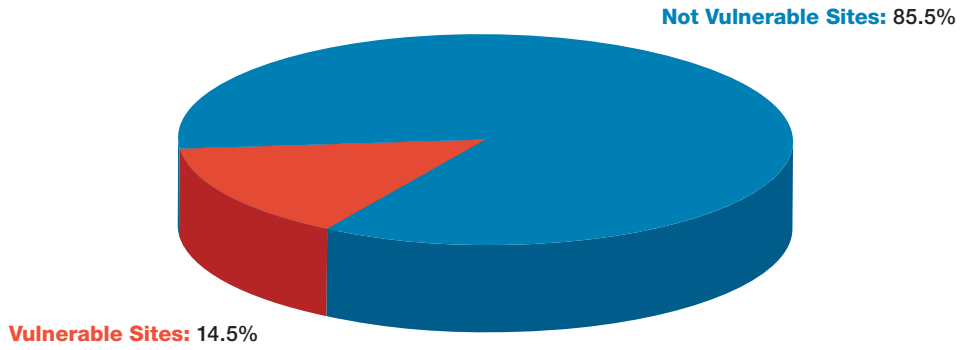

Figure 72: Percentage of Sites Vulnerable to Client-Side JavaScript Issues

Vulnerable Third-Party JavaScript Code Versus In-House Written Code

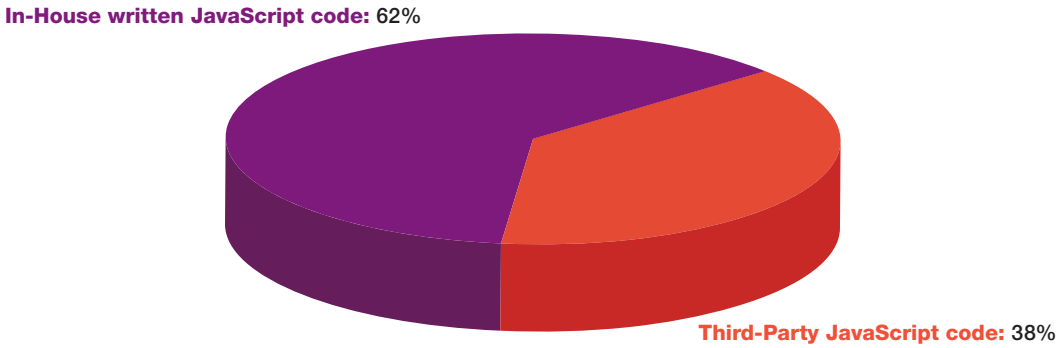

Figure 73: Vulnerable Third-Party JavaScript Code Versus In-House Written Code

Of the 98 vulnerable sites, 92 sites (94 percent) suffered from DOM-based cross-site scripting issues, whereas only 11 sites (11 percent) suffered from open redirects. The total amount of DOM-based cross-site scripting issues found was 2370, while only 221 open redirects were found.

Based on the dataset that we analyzed, we may extrapolate that the likelihood that a random page on the Internet contains a client-side JavaScript vulnerability[24] is approximately one in 55.

To summarize, from the information uncovered by this research we conclude that client-side vulnerabilities are quite common in modern web applications, especially those that rely on JavaScript for performing client-side logic—i.e. Web 2.0, AJAX, and Rich Internet Applications. In addition, a substantial number of the existing JavaScript client-side vulnerabilities on the Internet are introduced from 3rd party code that is not developed in-house, and usually is not reviewed for security issues.

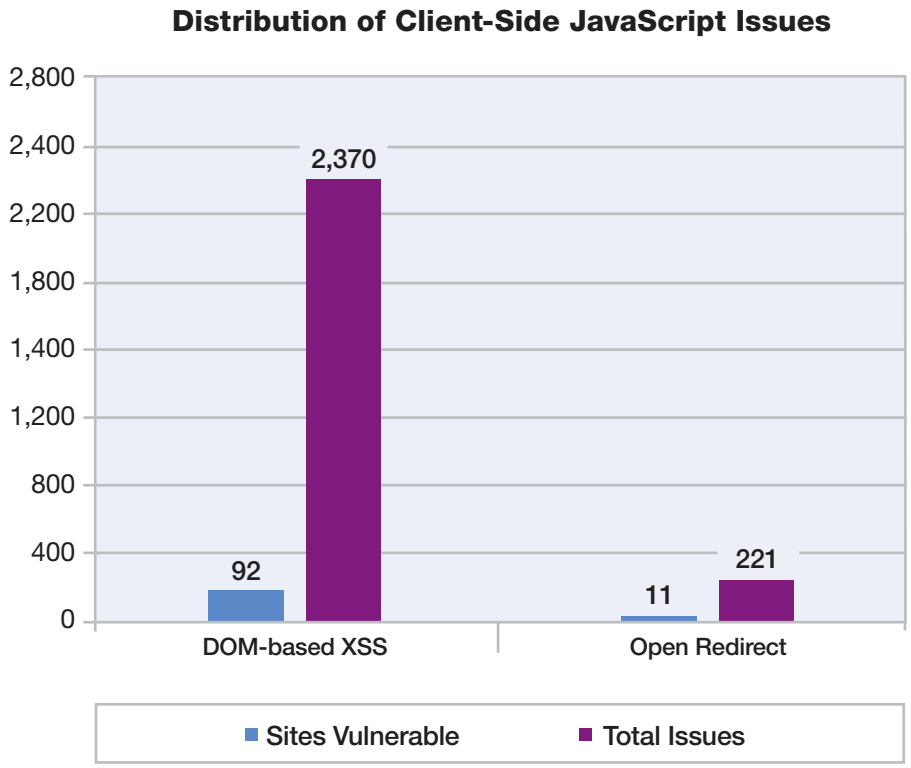

Figure 74: Distribution of Client-Side JavaScript Issues

24 Information about the prevalence of client-side JavaScript vulnerabilities was included from a Rational research paper titled "Close Encounters of the Third Kind" (http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=SA&subtype=WH&appname=SWGE_RA_RA_USEN&htmlfid=RAW14252USEN&attachment=RAW14252USEN.PDF).

Web application hack-ability and efficient defense

IBM Security provides both scanning products and services. The value of this combination is that, in aggregate, IBM can show how effective companies are in securing their web applications. While these numbers do not have direct bearing on your business, which has its own risk picture, they do provide a comparative view which is useful.

The following Web Application Vulnerability scanning is from IBM Professional services, and these vulnerability numbers represent vulnerabilities found by both Rational® AppScan® as well as manual site analysis by a professional penetration tester.

Figure 75 to the right shows the likelihood that each vulnerability will occur within a web application. One thing to understand is that some of these scans are repeat scans, so some of the decline shown is due to fixed vulnerabilities over time.

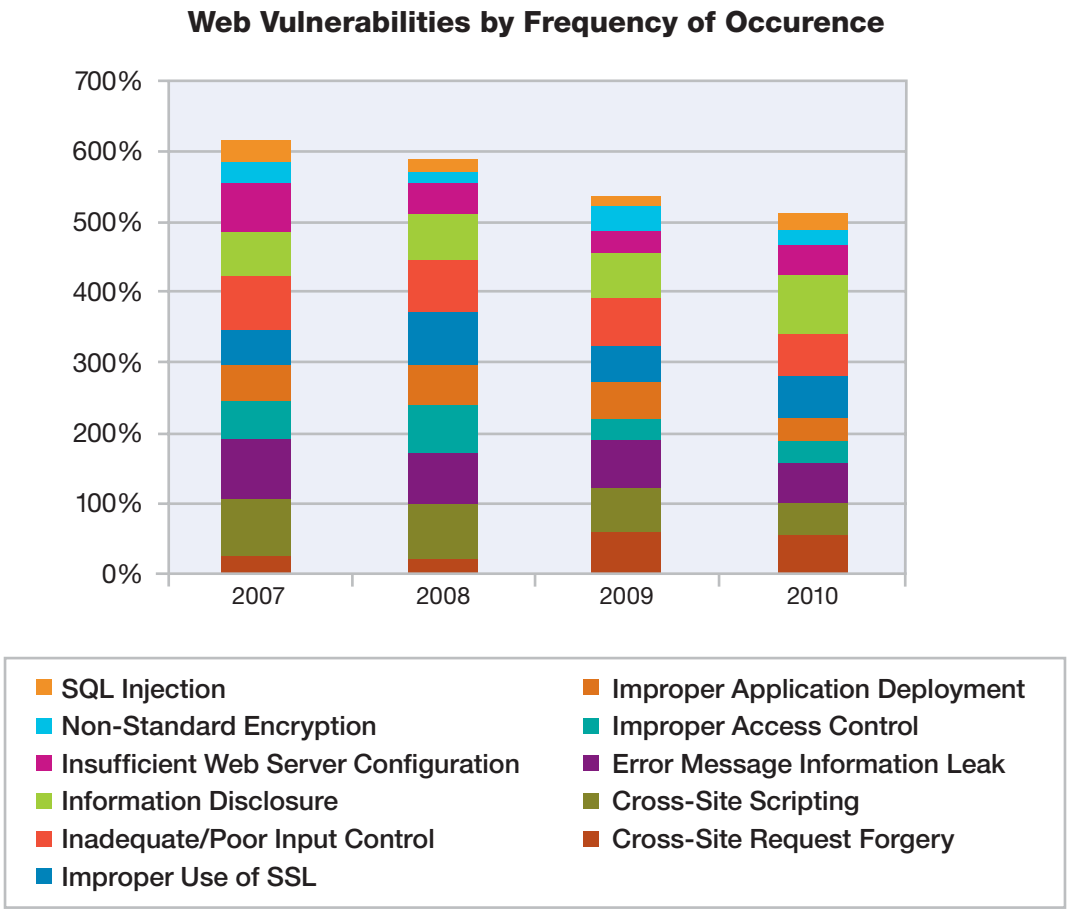

Figure 75: Web Vulnerabilities by Frequency of Occurence

If you want to compare your web vulnerability levels with other companies in your business segment, the chart below shows the average number of instances of a given vulnerability type across industries.

BUSINESS SEGMENT						
Vulnerability Type	**Financials**		**Industrials**		**Information Tech.**	
	Avg. vulnerability per test	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**	**Avg. vulnerability per test**	**% one vulnerability likely to occur**
Cross-Site Request Forgery	6.3	75%	2.6	55%	4.2	50%
Cross-Site Scripting	0.4	30%	7.6	52%	6.4	41%
Error Message Information Leak	10.9	80%	23.2	58%	14.6	47%
Improper Access Control	0.4	30%	1.2	40%	0.7	32%
Improper Application Deployment	2.4	55%	1.3	32%	2.1	24%
Improper Use of SSL	32.1	90%	15.6	33%	19.4	50%
Inadequate / Poor Input Control	3.5	40%	11.8	63%	13.8	65%
Information Disclosure	10.9	75%	22.5	92%	17.4	82%
Insufficient Web Server Configuration	1.0	50%	8.1	58%	4.4	44%
Non Standard Encryption	2.1	10%	1.7	23%	0.7	24%
SQL injection	0.1	5%	1.4	28%	5.0	21%

Table 18: Vulnerability type for Financials, Industrials, Information Technology, IBM® Rational® AppScan® OnDemand Premium Service 2010

Vulnerability Type	BUSINESS SEGMENT					
	Logistics		Retail		Other	
	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur
Cross-Site Request Forgery	9.2	57%	0.5	27%	2.4	56%
Cross-Site Scripting	1.7	57%	2.6	64%	11.0	48%
Error Message Information Leak	0.7	33%	17.8	59%	11.2	60%
Improper Access Control	0.1	10%	2.2	32%	0.4	28%
Improper Application Deployment	1.6	24%	0.4	27%	3.9	44%
Improper Use of SSL	45.7	81%	20.0	73%	46.6	88%
Inadequate / Poor Input Control	1.6	48%	11.3	82%	15.1	60%
Information Disclosure	13.4	90%	7.8	82%	16.2	76%
Insufficient Web Server Configuration	0.5	24%	2.3	27%	3.4	36%
Non Standard Encryption	0.3	10%	4.6	41%	0.4	20%
SQL injection	0.0	5%	7.0	59%	0.3	16%

Table 19: Vulnerability type for Logistics, Retail, Other, IBM® Rational® AppScan® OnDemand Premium Service 2010

If you are considering what technology to use for your next web application, these numbers may help you focus your research.

Vulnerability Type	ASP.NET		Java		PHP	
	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur	Avg. vulnerability per test	% one vulnerability likely to occur
Cross-Site Request Forgery	2.8	61%	4.4	51%	3.4	44%
Cross-Site Scripting	4.9	48%	7.2	53%	1.9	32%
Error Message Information Leak	23.6	71%	13.7	51%	3.4	40%
Improper Access Control	1.1	29%	0.8	32%	0.8	36%
Improper Application Deployment	2.5	48%	1.5	26%	2.0	28%
Improper Use of SSL	28.2	64%	28.8	55%	12.4	72%
Inadequate / Poor Input Control	10.6	66%	12.1	59%	3.7	56%
Information Disclosure	24.7	84%	14.5	88%	6.6	72%
Insufficient Web Server Configuration	3.0	50%	5.7	41%	2.3	44%
Non Standard Encryption	2.7	30%	1.1	17%	0.7	24%
SQL injection	3.3	38%	2.3	18%	0.2	16%

Table 20: Vulnerability type by web application (ASP.NET, Java, PHP), IBM® Rational® AppScan® OnDemand Premium Service 2010

At first review, these numbers may seem to be of limited use. While every security fanatic holds the theoretical concept of zero vulnerability as a laudable but perhaps impossible goal, there is value in looking at a comparative vulnerability. Understanding this stems from understanding the nature of your adversary, the attacker. Some may claim attackers as being lazy, but it is demonstrated that attackers range from disciplined ascetics focused only on the acquisition of hacking skills to the lazy buyers of hacking products. One thing is certain, attackers pursue efficiency as the following examples illustrate.

- Attackers use scanning tools and automated propagation tools which are designed to use any and all vectors to fulfill one simple intent: Give control of as many computers as possible to a master.
- They use cached pages on search sites to assess your vulnerability so that they can "probe the ghost of your defenses" without probing you directly. Your cached page can tell them what to attack without directly examining your live web pages.
- Attack business sites rank targets, building search engines for hacking targets. So the most vulnerable targets are attacked the most. This is where the comparative view starts to make sense, in that a less vulnerable website will be ranked lower and therefore hacked less.
- There is a self-sustaining cycle where vulnerable websites allow the propagation of bots, which then generate more fake sites with malware, etc. This cycle is self-reinforcing.

Understanding that hacking is mainly about efficiency, we can prioritize and strategize our web application defense to be as efficient as possible. Our unobtainable goal of zero vulnerability (unplugged, powered off, and placed inside a Faraday cage) can shift to becoming a relatively inefficient target so that it takes more effort to compromise your company rather than another. You can use numbers in the previous chart to have an idea of how attractive your business servers may be to attackers.

Avoid the Net cast by automation

Automated systems sweep the net for easily exploited websites. Typically, these automated attacks are mitigated most effectively by a separate web access control system and Intrusion Prevention System (IPS) with web application protection capabilities.

For web applications, a good choice is to separate your authentication solution from your web application. This can provide you with vulnerability mitigation for several types of web application vulnerabilities at once. Separate authentication also makes access control itself more efficient for administrators to manage than from within the web application code.

When it comes to intrusion prevention, efficiency should be measured in actions taken over vulnerabilities blocked. The perfect Intrusion Detection System has an efficiency ratio approaching 0, where turning it on results in perfect protection. This of course is driven by the accuracy of detection. Threat prevention accuracy is driven in turn by security research, so accuracy should be viewed as a "historical trend" of pre-emptiveness. The most efficient threat-mitigation systems block whole classes of threats with a few detection algorithms. Part of the value of assessing accuracy as a historical track record is taking into account the background and motivations of the researchers.

These tools rapidly can close down vulnerabilities, giving you more time to fix your vulnerabilities efficiently.

Fix vulnerabilities efficiently

Vulnerability prioritization is a balance between the difficulty of the fix versus the ease of the attack. This is where professional penetration testing and vulnerability assessment services provide additional value because they identify relationships that help you prioritize. Vulnerabilities in web applications are often related, one hard-to-fix vulnerability may be mitigated by fixing several easy-to-fix vulnerabilities. For example, request forgery is often difficult to fix, but to be more effective, it is often combined with link injection as a vehicle for delivering malicious content. In addition to identifying complex relationships, the professional penetration tester can find vulnerabilities that are recognized only by intelligent human probing.

Clearly, those vulnerabilities which are blocked by Intrusion Prevention and access control are less important to fix, especially if the fix is difficult, but it is always a good idea to fix broken applications, if for no other reason than to help your application developers avoid the same mistakes in the future.

The best defense against the elite

If you avoid the net cast by automation, you should fix the vulnerabilities you can, and make your remaining vulnerabilities difficult to access; and you hopefully will be left exposed only to the hacker elite. From here you can continuously work toward the unobtainable "Zero Vulnerability" posture with relative safety.

Section IV—Emerging Trends in Security

The Emerging Trends in Security section takes a look at fast developing technology that presses upon enterprises considering whether or not it is time to make investments in these future areas. We explain where threats and exploits are being utilized in these early technology adoptions and how enterprises can stay focused.

Mobile security trends

As enterprises approach the huge potential in efficiency that mobile computing has to offer, the two primary hurdles they will likely face are complexity (due to proliferation of platforms) and security. This section explores the approaches, strategy, and suggested controls as a perspective on the external threat landscape in this area.

In approaching the mobile security topic, there are two fundamental observations to consider. First, most of what is considered best practice around securing mobile devices is still not nearly as well defined as it is in the corresponding personal computing space. Second, the underlying platforms themselves are substantially untested and likely contain years of vulnerability discovery ahead of them.

2010 saw significant increases in the number of vulnerabilities disclosed for mobile devices as well as the number of public exploits released for those vulnerabilities, but it's important to keep these increases in perspective. Many of the vulnerabilities impacted shared software components that are used by both mobile and desktop software. The vulnerability research that is driving these disclosures is not necessarily mobile-centric.

Likewise, many of the public exploits that have been released for these vulnerabilities are not actually designed to function properly on mobile platforms, although they could be retooled to do so by an interested party.

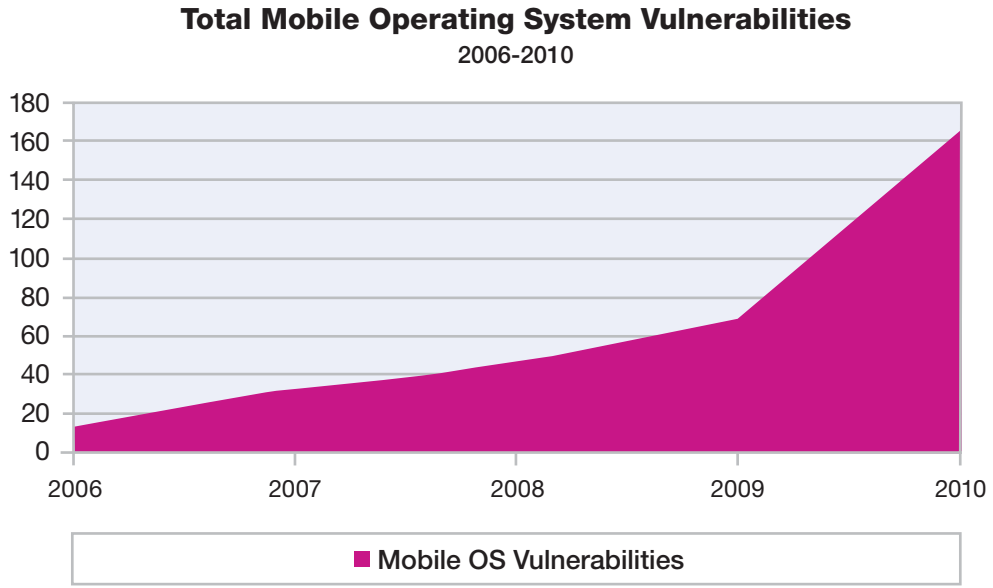

Figure 76: Total Mobile Operating System Vulnerabilities – 2006-2010

Nevertheless, there have been exploits released this year that are designed to function on various popular mobile platforms. One of the motivations of these exploit writers is a desire by mobile device users to "jailbreak" or "root" their devices to enable various kinds of functionality not intended by the manufacturers. This motivation drives the creation of mature, reliable exploit code that is widely disseminated and can be readily repurposed for malicious use. For example, early in 2011 malicious applications were distributed in the Android app market that used widely disseminated exploit code to obtain root access to devices and steal information. The vulnerabilities exploited by these malicious applications had been publicly disclosed for months at the time of the attacks. While attacks like this are not yet common place, they may happen more frequently in the future. It's also worth pointing out that the use of mobile devices in an enterprise environment brings other software systems into play, such as enterprise management servers and desktop sync software, which have also been subject to vulnerability disclosures and exploit releases.

We aren't seeing a lot of widespread attack activity targeting these vulnerabilities today, because mobile devices likely do not represent the same kind of financial opportunity that desktop machines do for the sort of individuals who create large Internet botnets. As e-commerce involving mobile phones increases in the future, it may bring with it a greater financial motivation to target phones, and an associated increase in malware attacks. However, mobile devices do represent opportunities for sophisticated, targeted attackers today. There are a number of vulnerabilities to target, and there is exploit information available. Malicious software on the devices can be used to spy on users, access sensitive information on the phones, and reach back into corporate networks. Therefore, enterprises should take the risk of targeted malware on phones seriously.

Because of these risks, enterprises may be apprehensive to move forward with significant enablement of multiple mobile device platforms. However, in addition to the potential efficiency benefits of enablement, it may be more useful to implement effective management technologies rather than provide technical controls needed to prevent the forward movement that will be attempted without their support anyway. It will likely become more expensive to implement technical controls to help ensure enterprise data is not finding its way to employee smartphones in an ad hoc fashion. Investing that same funding into properly securing some level of additional platforms to enable this trend and its subsequent efficiency gains may make the most amount of sense for many environments.

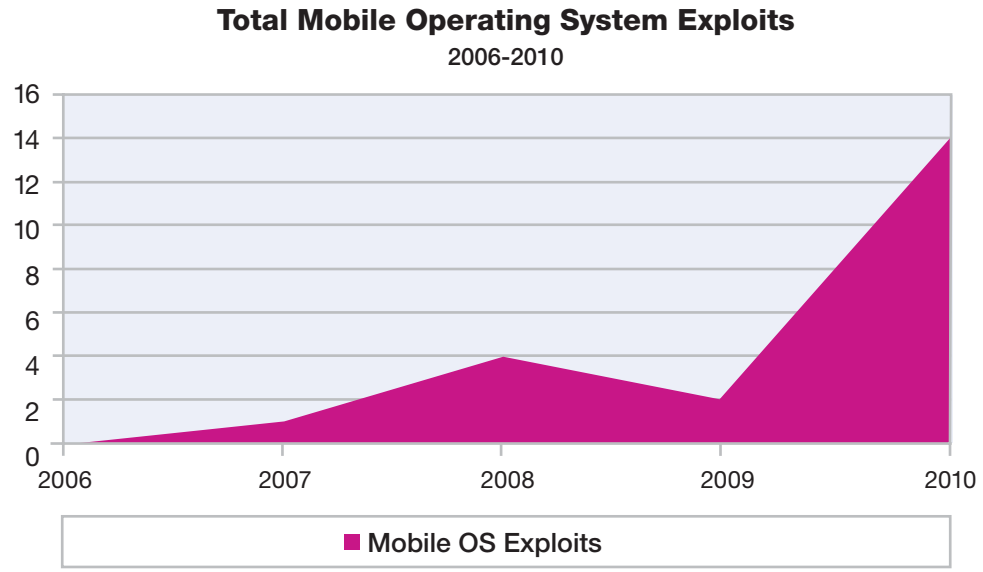

Figure 77: Total Mobile Operating System Exploits – 2006-2010

Effective controls to manage mobile devices

Existing enterprise security standards serve to help protect the integrity of our data and its corresponding IT infrastructure. Therefore, it should be relatively straightforward for the enterprise to identify the required controls for a given data classification. The data-focused approach should provide the foundation of the appropriate security standards to adequately protect this same data on mobile devices while taking into account the unique aspects of mobile technology. This enterprise data is of no less value because it now resides on the latest, shiny new smartphone rather than on existing personal computers or servers.

As enterprises arrive at the specific controls they need to enforce, it is vital to establish correct assumptions on the various classifications of data that will end up residing on their devices. This can be approached in different ways including identifying classification based on employee roles or services that are expected for device support. Regardless of the approach, it is paramount that this classification is established in order to clearly define the resulting controls that are required.

This statement ideally results in a fairly small set of controls required to host, transmit, and process this data so the controls can be clearly defined in employee security standards as well as implemented and enforced via technology.

For example, here is a typical set of controls.

- A device password of adequate strength to protect the data classifications expected to reside on the device.
- A timeout and lockout feature controlled by the device password and set for a period of minimal time. This is typically anywhere from five to 30 minutes; the shorter the better from a security perspective.
- Device configuration such that any data stored on the device is removed after “X” failed login attempts, or the device is managed by a remote service with this ability. If both controls are possible, they should both be used. This data removal should include data stored on memory media (i.e., flash memory) used by the device if possible.
- Password prompt on the device should pause for an incremental time after each unsuccessful login attempt to protect against brute-force login attempts if possible.
- Install and run an anti-malware program on any device that has access to the enterprise infrastructure or has access to enterprise data.
- Install and run a firewall program on the device if possible. Limiting access into the enterprise is an effective means of decreasing risk.
- Remote access for synchronization of data or access to enterprise infrastructure should always go through an approved Remote Access Service (RAS) gateway using adequate access credentials. It is a sound security practice to minimize or discourage the practice of making internal services available externally. Doing so simply increases attack surface area.
- Configure Bluetooth so that it is not discoverable and it will connect only with paired devices on all handheld devices supporting these features.

If devices cannot meet these minimum requirements, they should not be suitable for enterprise use. Ideally, technology should be implemented to properly configure devices for employees as part of the boarding process. This establishes a trusted relationship at the completion of the boarding process.

As you review these defined controls, notice that there are some controls unique to smartphones. The requirement to either remotely or locally remove information is a compensatory control to address the unique nature of smartphones. Because of their size and common use cases, enterprises should expect that loss and theft will be higher than they’ve typically witnessed in laptop programs. The reality is that even the most conscientious employee can use their smartphone in an airport, cab, hotel, or anywhere they go because that is the nature and benefit of the technology.

In reviewing the myriad of platforms that have become available in the last couple of years, the primary observation from a security perspective is that platform vendors have designed their products to appeal to consumers with the enterprise being a secondary concern. Most smartphone platforms did not lend themselves to immediate enterprise use in their initial versions. Nor did they support the typical controls that an enterprise would expect. In fairness, nearly all vendors have recognized this and have begun to embrace that their customers desire to use their devices across both their work and personal lives. As a result, typically as platforms hit version two or three, they include most or all of the minimum enterprise requirements. It is particularly important that enterprises consider patch management of these devices as a part of their overall strategy for managing them. As discussed above, the desire to "jailbreak" or "root" the devices has been one of the drivers for the public dissemination of reliable exploit code for mobile devices, and this sort of exploit code has been used in malicious attacks.

Although it is the responsibility of the ecosystem of mobile device makers and telecom companies to make sure that updates are available that fix these vulnerabilities, those updates may have to be manually installed by end users. Experience shows that manual end user update processes are inconsistently complied with and users who aren't keeping up with updates may have devices that are exposed to attack. One way to combat this problem is to develop a mechanism for regularly reminding corporate mobile device end users that installing updates is an important part of keeping their device and their corporate data secure.

It is very likely that at least initially, as software updates become more important and more frequent to fix exploited vulnerabilities that enterprises may only be able to rely on their MDM (Mobile Device Management) solutions to simply limit synchronization to updated versions. Currently, the platform vendor/hardware vendor/carrier ecosystem has not embraced the notion of frequent updates that can be distributed by third parties, like enterprises, in order for them to more closely manage vulnerabilities on their enterprise devices. Obviously, as this moves forward, it may vary from platform to platform, adding an additional challenge of inconsistency for the enterprise.

Encryption

While encryption of data at rest is not required for some types of information in some industries, it should be used for a subset of specific types of data in nearly every enterprise. This is driven by legislation as well as by customer expectation so we'll continue to see this apply to at least a portion of employees for every enterprise. Whether enterprises leverage native encryption capabilities that may exist in some platforms or seek some of the third party encryption solutions that exist, it is crucial to thoroughly understand the implementation you've selected to help ensure that it meets the specific encryption requirements defined in your security policy.

Note that nearly all data encryption approaches for smartphones have been software-based and do not provide an ideal architecture for the typical smartphone. It is hard to determine if this is simply a point in time in the development of mobile devices and more will eventually include hardware-based encryption capabilities. This concern may also be mitigated as processor capacity continues to increase in smartphones and we see both faster and multiple processors in these devices.

Until the summer of 2010, some felt smartphone malware was an urban legend but as a result of multiple security research disclosures that summer, there is now more recognition that this is both possible and likely common moving forward. Enterprises should not discount this threat because it is not as pervasive as the existing personal computer threat landscape.

It is valuable to maintain an information-based objective approach as we look at the current threat landscape in this arena. While the threat of mobile malware has existed as long as the devices have been available, it remains far less prevalent than malware attacks against many other devices. In fact,

most of the activity that drives the most malware today remains focused on targeting Windows XP computing devices. This should not be a surprise—they exist in the hundreds of millions and are typically manned by a wide range of user expertise. We should expect this target to exist as it does today until the prevalence of XP devices begins to decrease as the 2014 Windows XP end-of-life support date arrives. Until then, XP will remain a primary target, especially with common malware development kits available.

One of the reasons Windows XP grew to the primary attack target is simple pervasiveness. Windows XP market share drove this attractiveness. The discovery of numerous vulnerabilities allowed it to grow and the existence of malware development kits allowed it to flourish. If you apply this same logic to the current smartphone landscape, you would note that at present, there is no single dominant platform. As there become clear winners in this space, we should expect them to be targeted.

In discussing smartphone malware, we may see a slightly different attack approach than we've seen in the personal computing space. Specifically, we may see malware introduced voluntarily by the device owner by using "vetted" application hosting in one of the many platform-specific application stores. This approach is already evidenced in existing malware and should be expected to increase as the number of available applications skyrockets. This will also challenge the end user because of the nature of smartphone application stores.

Unlike personal computers where this approach isn't prevalent, users likely will perceive the application store as a trusted source of software for their device. This couldn't be farther from the truth, with no existing application store providing secure code reviews. In fact, most do not provide any code review whatsoever, simply providing a place for developers who complete the registration process (which may include a minimal fee) to sell or give away their work. While it is undoubtedly possible to remotely compromise a smartphone device by socially engineering a user into clicking a link or visiting a URL, these attacks require remote code execution vulnerabilities, unlike the application store approach. It is likely that malicious behaviors in what appear to be trustworthy applications may provide an easy vector.

We should also expect that many of the same malware components we see in desktop malware will exist in their mobile counterparts. Components like keystroke loggers and proxies that redirect traffic and steal information have already been observed in smartphone malware. Multiple types of Premium SMS toll fraud malware exist; these are unique to smartphones and represent an easy way of generating quick revenue for the attacker.

Remote Access Service

Since smartphones in their essence exist as mobile devices and are typically outside of both the enterprise infrastructure and premise, a secure remote access service is a fundamental enabler of enterprise mobile computing.

In an ideal circumstance, a Remote Access Service (RAS) would only allow access to those devices it could demonstrate as trustworthy, rejecting all others. In addition, given the specific defined use cases for mobile devices, risk can be lowered by limiting this access to those destinations and services needed by the device and restricting those that are not required. RAS is another area where the desire for platform diversity becomes a challenge. Ideally, the enterprise would desire adoption of common, industry-standard secure access solutions that are commonly supported in many or most platforms.

Enterprise selection of RAS service should also focus on the technology selection that is best suited to smartphone devices. Typically, most personal computer RAS services use IPsec (Internet Protocol Security) as a means to establish an authenticated, secure tunnel across the Internet between the personal computer and the enterprise gateway. This approach has supported the needed, secure algorithms to help provide confidence that data in transit was well protected between the two

points. The obvious approach would be to transport that same approach to smartphone devices. Many smartphone platforms include IPsec VPN clients natively and work with most industry standard gateways. The benefit of this approach is that existing infrastructure can be leveraged, using the same level of security required.

The downside to this approach, when used with smartphones, is a real issue with device battery life and usability. Maintaining a constant tunnel between device and gateway, which is needed to synchronize data, quickly saps battery life. The alternative approach is to manage the use of this tunnel, leaving it connected only long enough to synchronize or access data, and then turning it off. Unfortunately, this loses many of the benefits gained by mobile efficiency.

An alternate approach is the use of Secure Sockets Layer (SSL) as a tunneling protocol within the remote access solution. While SSL is able to support similar encryption algorithms as IPsec (in terms of bit strength), it exists natively in http (s). SSL can provide an on-demand secure connection into the enterprise that does not require the mobile device to maintain a constant secure tunnel; it is only needed for actual data exchange. The primary concern with the use of SSL in a remote access service is in terms of the gateway. In most cases, this function is simply a reverse-proxy SSL-based exchange that is easily compromised and provides little security isolation. That said, there are SSL-based gateways available that do provide security functions which allow for the discovery of a trusted device (hence preferable from a security perspective) while still maintaining the battery-friendly, user-friendly approach observed with an on-demand secure access service.

Future security vision

Nirvana, as it applies to future of smartphone security and enterprise use, is likely the ability for smartphone devices and associated platforms to support dual personas on a single device. Since much of the smartphone growth within the enterprise likely will be comprised of employee-owned devices, the ability for enterprise data and controls to peacefully co-exist on a personal smartphone is the most desired state. In today's environment, enterprises should ensure control of their data regardless of where it is and this includes employee-owned smartphones. As a result, enterprise requirements should be applied to all smartphones enabled to access or store this data, regardless of owner liability. The ideal future state would allow the enterprise to properly secure access to its data and infrastructure to the degree required while allowing the individual to decide the security controls for their data and access to personally-subscribed services. For the enterprise, this would mean that all enterprise data, applications, and network access to and from the enterprise would be secured in their prescribed manner but would be enforced to only that "container" where those applications, associated data, and connectivity existed. Outside of the container, the user would be free to decide what kind of controls the device itself should contain and what applications they were comfortable with without regard to any impact on enterprise data, applications, and access.

The need for this approach and separation is necessary as we look at the future enterprise use of smartphones. Certainly, while starting with the need for malware prevention, we shouldn't expect that enterprise protection ends there. It is only a matter of time before things like intrusion prevention and data leakage prevention are requirements on smartphones as they've become on personal computers. Given the likely relatively limited nature of computing resources present on smartphones, the most viable approach to these needs is to push the execution of them into the enterprise remote access connection in a way that helps ensure that all connection to and from the device is forced through a common service that performs this inspection before allowing the traffic to its ultimate destination. To some degree, this inspection can be exacerbated by finite segregation between enterprise data and applications, access, and personal use but ultimately, even if only the enterprise portion needs this level of inspection, the most favorable approach will be to push a lot of this into the cloud or, in more likely terms, into the remote access enterprise connection.

The evolving state of security in the cloud

While security is still considered one of the major inhibitors to cloud adoption, organizations are increasingly adopting cloud-based technologies to address competitive market needs. This contradiction highlights the fact that many of the perceived challenges associated with cloud computing have been of less concern for a large subset of the market that have already adopted the cloud. We are seeing a shift in perception as cloud adoption evolves and knowledge increases. A recent study from Appirio focused on the state of the public cloud from the perspective of corporations that are using the public cloud for one or more service. Of the 155 medium-to-large companies responding to their survey, 28 percent agreed that security is the number one misconception about cloud computing and 39 percent said that cloud computing would be a pivotal enabler of an overall business transformation for their organization.[25]

Unlike other emerging technologies, the interest in security as it relates to cloud computing began close to its inception, and concerns about cloud security have received considerable attention in the marketplace. This has translated into hesitancy on the part of some organizations to aggressively adopt the cloud. In fact, many organizations are looking at a private cloud implementation as their initial foray into cloud computing in order to maintain control over data processing and security. Although perceptions about cloud computing in general may be changing, the fact remains that an organization's willingness to utilize the public cloud for mission-critical work usually depends on their understanding of potential risks and their assessment of whether their data can be adequately safeguarded. They also may rely on their experience with and knowledge of specific cloud-based solutions. In fact, we see greater adoption of cloud technologies with which the market has become more familiar. For example, although email is clearly a business critical application and can contain confidential data, many organizations have already leveraged some form of web-based email as part of a collaboration solution. The question for organizations is not whether the cloud as a whole is secure, but whether the organization is comfortable placing their workload on the cloud. As shown in Figure 78, the relevant component of the adoption curve is that most workloads can be suitable for cloud-based technologies. Whether that technology is adopted depends on the business benefits of the organization and their perception of the risks.

25 "State of the Public Cloud: The Cloud Adopters' Perspective" published October 2010, Appirio.

As with many outsourcing technologies, public cloud computing requires that the subscriber trust the provider to manipulate and handle their data with appropriate security measures. This trust relationship is paramount in cloud computing given that many providers are unable, or in some cases, unwilling to share their security controls or the details of the environment for the very purpose of maintaining security. Security best practices guidelines specific to cloud computing have begun to emerge from organizations such as the Cloud Security Alliance (CSA), which has a stated focus of providing security and privacy guidance for subscribers of cloud computing. This helps organizations evaluate their risk tolerance for using the cloud. With the increasing challenge of compliance that subscribers are facing, organizations should leverage industry recognized best practices while establishing their strategy around security and their use of cloud-based technology.

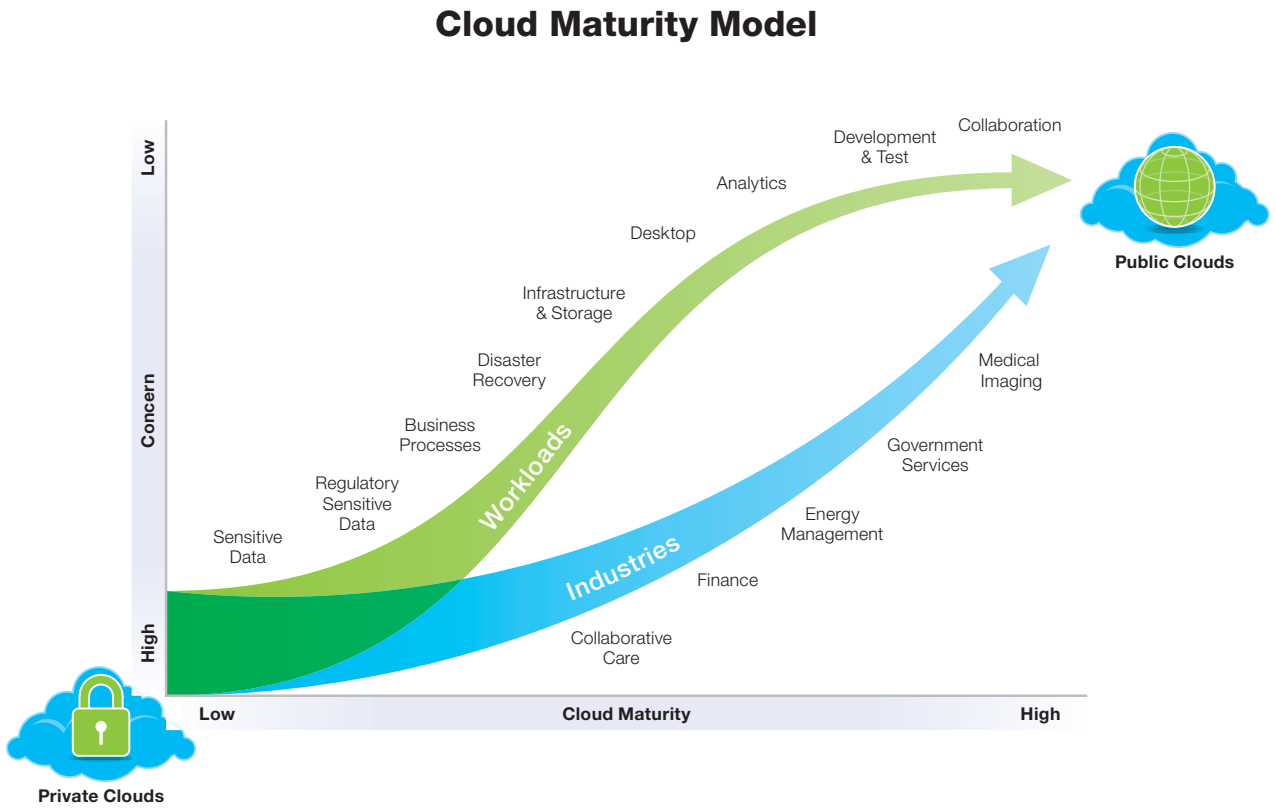

Figure 78: Cloud Maturity Model

Design elements for security in the cloud

Secure by design

Cloud computing is largely driven by financial and operational efficiency motivators. As a result, organizations should build security into the fabric of their cloud activities to achieve the expected returns. Retroactive attempts to apply security later in the cloud life cycle often result in diffusing the value of cloud computing. For example, if an organization wants to use the cloud, public or private, as a platform for delivering cloud-based application services, but has not ensured that the targeted application has been securely designed and implemented, then regardless of the controls their provider has put into place, the application vulnerabilities could leave the solution open to unintended data loss or compromise. Extending existing security policies and standards, leveraging sound physical security protections already in place, and assessing systems and applications for security weaknesses are examples of security design elements that should be included when establishing a secure cloud environment.

Purpose-built security

IBM believes that there is no "one-size-fits-all" approach to security within clouds. Rather there are common sets of foundational security controls which apply to all types of clouds. On top of these foundational controls, organizations should implement workload-specific controls that align with the work being done in that particular cloud. For example, in a cloud solution dedicated to workplace collaboration, anti-spam is certainly an appropriate and needed control. However, in a cloud designed for development, anti-spam probably is not a control necessary to reduce risk associated with the workload. This approach allows the cloud provider to address the specific security needs of each cloud solution and control costs, which should translate into cost savings for their subscribers. The delivery and deployment models (SaaS, IaaS, PaaS, etc.) can also determine the types of security controls that are appropriate based on differences in attributes such as data flow, integration points, and user access scenarios.

Improving security via the cloud

Although a vast amount of public attention has been given to the security risks of cloud computing, it is likely that for many organizations the cloud could be considered more secure than their traditional legacy environment. Cloud providers may contribute security capabilities and skills that subscribers do not or cannot support within their own organizations. Cloud adoption is typically aligned with specific initiatives, and as such, the security requirements are narrowly focused and thus can be more deliberate. As such, security can be applied more appropriately and effectively to that workload or task than was applied as part of the organization's enterprise-wide security program.

Cloud computing can also allow organizations to apply layers of security that they previously were not able to implement due to lack of skilled resources or budget by actually moving security as a workload into the cloud. Cloud-based security services not only can offer customers cost savings over performing that function in house, but may allow some organizations to take on new security controls that they otherwise would not have added to their security management program, such as ongoing vulnerability scanning.

> Security in the cloud is a product of ongoing due diligence rather than a point in time statement. Organizations should plan on engaging in security over the life cycle of their cloud activities with the same level of diligence they execute within their enterprise environments.

Cloud providers who understand security threats and are able to adapt as threats evolve, are best equipped to help subscribers strengthen their security posture via the cloud. Ongoing gap assessments against best practices for secure cloud computing and testing for weaknesses against external attack via penetration testing are ways that cloud providers can assess and maintain their security posture.

Organizations should understand the implications of their cloud initiative in terms of security and privacy. Organizations new to the cloud should look towards seasoned experts to help them consume cloud-based technologies and security vendors, like IBM, can help these organizations plot out their security requirements and help ensure that their security strategy for cloud computing is sound.

Cloud computing opportunities

As acceptance of cloud computing advances, we expect to see the cloud being used in new ways that can serve to advance security. For example, IBM is exploring the use of advanced analytics to help organizations identify threats to their environments and respond to those threats without impacting business value. These advanced analytics capabilities are being developed to allow the processing of millions of events per second to identify the key threats or the needle in a haystack which an organization should focus on from a security perspective. IBM is also leveraging social network concepts such as crowd sourcing to evaluate the impacts of collective group experiences and knowledge to identify and address vulnerabilities. Finally, IBM is evaluating emerging endpoints such as mobile technologies to provide protection from new avenues of attack against cloud subscribers.

Summary

As cloud adoption continues to grow, and cloud providers apply controls appropriate to the function and purpose of their cloud solution, acceptance of the cloud, even the public cloud, as a platform for handling increasingly sensitive and mission critical workloads is expected to grow. Building security into the foundation of each cloud initiative should be the joint responsibility of the cloud subscriber and their providers. This requires a deep understanding of the security requirements surrounding that initiative, and a commitment to meet those requirements without applying security controls that are unnecessary or ineffective. If vendors and subscribers are able to do this, then the efficiencies and cost savings that cloud computing affords can be better realized.

As security concerns quell regarding cloud computing, more organizations may take advantage of the security benefits that can be gained from cloud computing either as a beneficiary of the security controls the provider offers for their specific security initiative or as a consumer of cloud-based security services. In the meantime, organizations should continue to seek guidance from security vendors like IBM for help in evaluating and developing their cloud security strategy, assessing the controls around their cloud initiatives, and providing them with secure solutions for enabling cloud computing within their organization.

IBM Corporation
Software Group
Route 100
Somers, NY 10589 U.S.A.

Produced in the United States of America
March 2011

WGL03007-USEN-00