A Strategic Approach to Web Application Security

The importance of a secure software development lifecycle

Rachna Goel
Technical Lead – Enterprise Technology

SDG
[technology + passion] - risk

Web application security is clearly the "new frontier" of cyber security. Web applications are already prevalent and becoming more popular and pervasive every day. While Network and Operating System cyber security practitioners have had decades to perfect their techniques and methodologies, web applications have been around for less than 15 years, and are still vulnerable; and security is a moving target. Although web application security was clearly recognized as a big problem several years ago, many organizations have been too slow to act. This has given potential attackers a distinct advantage. We must carefully consider the most common and straightforward security issues first and work to perfect our approach to protecting applications against those attacks.

Security is similar to any other good engineering process: It requires education, skill and careful attention. As teams develop the skills and knowledge needed and continue to pay careful attention, they can achieve their goals of delivering secure software applications.

With technology continuing to evolve at a rapid pace, and with more users, all potentially malicious, accessing the Internet and its applications every day, the threats and challenges to web application security will only continue to grow. Implementing a comprehensive web application security program will be critical to the success of any online presence in the next decade.

The Problem: Web Applications are the new Security Perimeter

Today, almost every enterprise conducts business online. As the applications that run online businesses spread out over technologies and platforms, the security risks also increase. No company is immune to attack. In 2012 alone, there were more than 800 reported hacking incidents, and 70% of those were penetrated through web application flaws. According to the **INFORMATION SECURITY BREACHES SURVEY 2015,** there has been an increase in the number of both large and small organizations experiencing security breaches. Some statistical reports published by **ISBS 2015.**

90% of large organizations

74% of small businesses

had a security breach.

▲ Up from 81% a year ago.

▲ Up from 60% a year ago.

69% of large organizations

38% of small businesses

were attacked by an unauthorized outsider in the last year.

▲ Up from 55% a year ago.

▲ Up from 33% a year ago.

75% of large organizations

31% of small businesses

suffered staff related security breaches in the last year.

▲ Up from 58% a year ago.

▲ Up from 22% a year ago.

46% of large organizations

7% of small businesses

expect information security spend to increase in the next year.

▼ Down from 51% a year ago.

▼ Down from 42% a year ago.

Most organizations that develop custom software struggle with managing these new security risks. That's because application security is a continually evolving field - security standards and rules must evolve as fast as these technologies and the ever-changing array of attacks on them. And yet, solutions have focused primarily on reactive measures.

Many approaches for testing security after the application has been built have proven costly and caused critical deployment deadlines to be missed. Historically, there was no economical way to continuously test source code for security issues during the development process. ***Therefore, organizations were forced to consider security near the end of the development process, when the web application was nearly complete.*** Then, applications were typically tested either by expensive consultants, or by tools that required a high level of security expertise to use them effectively. Because no solution existed that could provide continuous security guidance to developers, security was forced to the end of the development cycle.

Due to the unprecedented increase in the scale and intensity of sophisticated web application attacks, organizations are scrambling to re-architect their development processes to make security a major component of each phase.

A more proactive approach is needed; one that incorporates a well designed secure development life cycle, and includes appropriate tools, processes and training. This will eliminate, mitigate or reduce the risk of harm from security vulnerabilities. Developers must analyze security from the moment the first line of code is written. This will assure that risk and development costs are reduced over time.

Question:

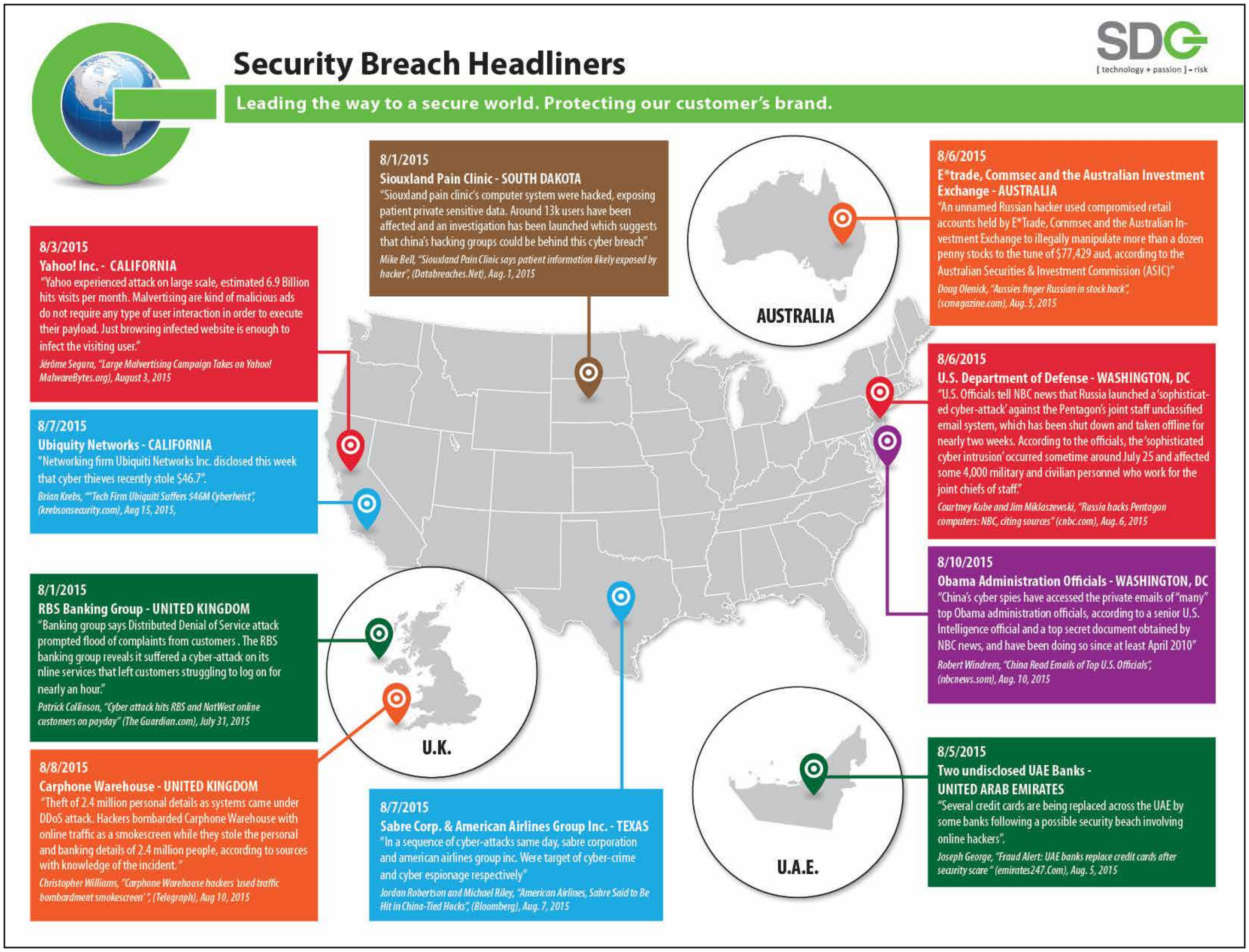
Is TESTING the best way to find security vulnerabilities in the development lifecycle?

Answer: ***It's impossible to incorporate security into an application through testing alone, just as it's impossible to incorporate quality into an application through testing alone.***

Security throughout the software development life cycle (SDLC)

Web application development remains a relatively new domain. As such, formal processes and best practices for developing web software are still being defined. Currently most enterprise organizations follow a set of standard steps, which define each phase of software creation. These phases are collectively referred to as the SDLC.

It has become apparent that lack of security was a serious issue and also the most vital missing piece in the development process. In fact, security assurance in the past was relegated to the QA phase of development when it was considered at all. Now, however, forward-thinking organizations are adding security activities to every phase of the SDLC in order to discover flaws earlier and to significantly increase the security of the applications that are in production.



Security Breach Headliners

Leading the way to a secure world. Protecting our customer's brand.

8/1/2015
Siouxland Pain Clinic - SOUTH DAKOTA
"Siouxland pain clinic's computer system were hacked, exposing patient private sensitive data. Around 13k users have been affected and an investigation has been launched which suggests that china's hacking groups could be behind this cyber breach"
Mike Bell, "Siouxland Pain Clinic says patient information likely exposed by hacker", (Databreaches.Net), Aug. 1, 2015

8/3/2015
Yahoo! Inc. - CALIFORNIA
"Yahoo experienced attack on large scale, estimated 6.9 Billion hits visits per month. Malvertising are kind of malicious ads do not require any type of user interaction in order to execute their payload. Just browsing infected website is enough to infect the visiting user."
Jérôme Segura, "Large Malvertising Campaign Takes on Yahoo! MalwareBytes.org), August 3, 2015

8/7/2015
Ubiquity Networks - CALIFORNIA
"Networking firm Ubiquiti Networks Inc. disclosed this week that cyber thieves recently stole $46.7".
Brian Krebs, ""Tech Firm Ubiquiti Suffers $46M Cyberheist", (krebsonsecurity.com), Aug 15, 2015,

8/1/2015
RBS Banking Group - UNITED KINGDOM
"Banking group says Distributed Denial of Service attack prompted flood of complaints from customers . The RBS banking group reveals it suffered a cyber-attack on its nline services that left customers struggling to log on for nearly an hour."
Patrick Collinson, "Cyber attack hits RBS and NatWest online customers on payday" (The Guardian.com), July 31, 2015

8/8/2015
Carphone Warehouse - UNITED KINGDOM
"Theft of 2.4 million personal details as systems came under DDoS attack. Hackers bombarded Carphone Warehouse with online traffic as a smokescreen while they stole the personal and banking details of 2.4 million people, according to sources with knowledge of the incident."
Christopher Williams, "Carphone Warehouse hackers 'used traffic bombardment smokescreen'", (Telegraph), Aug 10, 2015

8/7/2015
Sabre Corp. & American Airlines Group Inc. - TEXAS
"In a sequence of cyber-attacks same day, sabre corporation and american airlines group inc. Were target of cyber-crime and cyber espionage respectively"
Jordan Robertson and Michael Riley, "American Airlines, Sabre Said to Be Hit in China-Tied Hacks", (Bloomberg), Aug. 7, 2015

8/6/2015
E*trade, Commsec and the Australian Investment Exchange - AUSTRALIA
"An unnamed Russian hacker used compromised retail accounts held by E*Trade, Commsec and the Australian Investment Exchange to illegally manipulate more than a dozen penny stocks to the tune of $77,429 aud, according to the Australian Securities & Investment Commission (ASIC)"
Doug Olenick, "Aussies finger Russian in stock hack", (scmagazine.com), Aug. 5, 2015

8/6/2015
U.S. Department of Defense - WASHINGTON, DC
"U.S. Officials tell NBC news that Russia launched a 'sophisticated cyber-attack' against the Pentagon's joint staff unclassified email system, which has been shut down and taken offline for nearly two weeks. According to the officials, the 'sophisticated cyber intrusion' occurred sometime around July 25 and affected some 4,000 military and civilian personnel who work for the joint chiefs of staff."
Courtney Kube and Jim Miklaszewski, "Russia hacks Pentagon computers: NBC, citing sources" (cnbc.com), Aug. 6, 2015

8/10/2015
Obama Administration Officials - WASHINGTON, DC
"China's cyber spies have accessed the private emails of "many" top Obama administration officials, according to a senior U.S. Intelligence official and a top secret document obtained by NBC news, and have been doing so since at least April 2010"
Robert Windrem, "China Read Emails of Top U.S. Officials", (nbcnews.som), Aug. 10, 2015

8/5/2015
Two undisclosed UAE Banks - UNITED ARAB EMIRATES
"Several credit cards are being replaced across the UAE by some banks following a possible security beach involving online hackers".
Joseph George, "Fraud Alert: UAE banks replace credit cards after security scare" (emirates247.Com), Aug. 5, 2015

Typical security activities in each phase of the SDLC are as follows:

TRAINING

Everyone involved in web application development should be provided basic security training. Scalability and repeatability are critical aspects of effective security training programs.

REQUIREMENTS

As software requirements are defined, the corresponding security requirements should also be defined. For example, if sensitive customer data is to be collected and stored, requirements on how the data should be encrypted, both in transit and at rest should also be established as a requirement.

DESIGN

Once the application requirements are captured, architecture is designed to incorporate all the software requirements. At this stage of development, necessary security controls should also be identified and included as part of the application.

IMPLEMENTATION

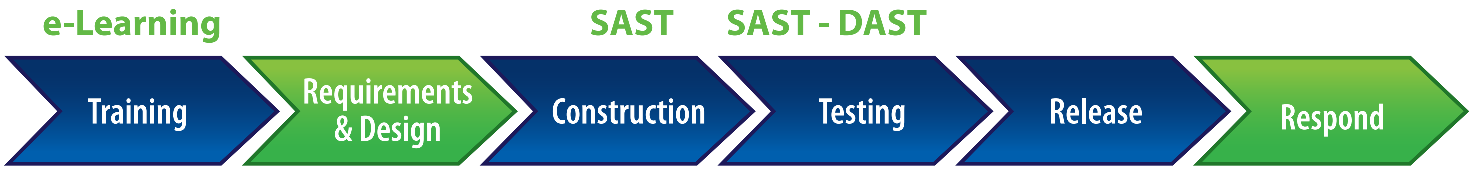
After requirements have been determined and an architectural design is in place, software development begins. Ideally, developers should receive security feedback while they are coding. This feedback should begin as early and as often as possible. Because this phase is often the most labor-intensive, continuously running automated security assessments should be performed, which will allow a developer to address issues in near-real time. This will allow organizations to develop applications that have been designed to be secure, rather than develop risky code; with security added as an afterthought.

QUALITY ASSURANCE

New application code needs to be tested before it goes into a production environment, to ensure that the code behaves as expected. While most organizations currently test applications to ensure that the functional requirements are being met; many are also beginning to test if the application is secure, based on the security requirements.

PRODUCTION

In the deployment phase, continual testing is vital to maintain security assurance and to protect against common application vulnerabilities. In addition, updates to applications that are already in production can introduce new flaws. Therefore, all code updates should be subjected to source, QA and production testing.



The above figure depicts the example of a Secure Waterfall SDLC.

Question: *Why can't developers code securely?*

Answer: ***The fact is that web application security is an extremely complicated problem. With millions of applications and lines of code already in existence online, web security is now more than ever an arcane knowledge-set that is continually evolving and adding new parameters. Furthermore, application code may contain a large number of obscure and hard-to-see issues that are further obscured by the ever-increasing complexity of modern web applications. This makes it nearly impossible to identify certain security issues in a timely way.***

Is it better to analyze source code or binary code?

Analyzing source code is absolutely the best approach because:

- Developers create the source code; vulnerabilities can be identified, and fixed by these developers.
- It allows analysis of the identified 'vulnerable method' calls being made, so that the remediation advice is both pertinent and actionable.

Features of binary code analysis:

- It performs assessments on compiled code that may not be decipherable by developers.
- It may be difficult to distinguish the code written by internal developers and the code inherent to the platform. This shortcoming results in non-actionable and confusing reports for the developers.
- A suitable option when source code is unavailable.

Advantages of Static and Dynamic Security Testing

The mitigation of application security risks is not a one-time exercise; rather it is a strategic initiative that requires paying close attention to the changing landscape of emerging threats and deploying new security measures to mitigate these new threats. To do this, an organization must create a security process that tracks an application throughout the SDLC using Static Application Security Testing (SAST) as well as Dynamic Application Security Testing (DAST).

The following lists advantages of SAST and DAST:

Static Application Security Testing (SAST)

- Performing static application security testing on source code greatly assists developers ensure that they code securely throughout the development process.
- Vulnerabilities such as ***SQL Injection, Command Injection, Improper and/or Custom Cryptography,*** and a host of other issues that occur primarily on the server are easy to find and eliminate early in the process.

Dynamic Application Security Testing (DAST)

- Vulnerabilities that can only be seen once the code is live on the web such as ***cross-site request forgery, business logic flaws and some forms of cross-site scripting*** are easier to identify using DAST.
- Once identified, organizations can automatically mitigate vulnerabilities using a web application firewall, while waiting for developers to remediate the security risk.

Using SAST and DAST technologies at the appropriate stage during the SDLC, an organization can create a strategic enterprise application security program

According to Gartner Research, ***"...next-generation modern Web and mobile applications requires a combination of SAST and DAST techniques, and new interactive application security testing (IAST) approaches have emerged that combine static and dynamic techniques to improve testing..."*** Because IAST combines SAST and DAST techniques, the results are highly actionable, can be linked to the specific line of code, and can be recorded for replay later for developers. Multiple DAST solutions now provide IAST capabilities. Some of the vendors evolving their offerings in this direction and offering IAST include Acunetix, HP, IBM, NTO, Parasoft and Quotium. However, most IAST solutions also require that an agent be deployed on the application platform, which relegates the technique largely to QA and also requires that the vendor explicitly support the platform or language being instrumented (such as PHP, Java or .NET/ASP).

Secure Development Lifecycle Protects Software from Vulnerabilities

The challenge for many teams is in designing a secure software development lifecycle. Here, are some tips to help teams implement more secure code. The key to preventing, detecting and resolving security vulnerabilities during the development lifecycle include ***good security architecture*** and should include the following concepts:

Centralize as much as possible: Experienced security engineers have a number of "secrets" for building secure code. Centralize input validation, use a common encoding library, develop or implement a centralized authentication and authorization mechanism, etc. Centralizing this functionality imposes a burden of consistency on developers - consistency breeds repeatability. It also forces the team to incorporate concepts such as authorization schemes and encryption libraries into the architecture and design.

Educate product development and QA teams: Educating the product team (including the product owner, developer and tester) incorporates security into every step of the process. Product team members should be trained on encryption as well as on how to design a good user password reset tool. Developers should be trained to encode output in the Enterprise Security API (ESAPI) tools. QA engineers should be trained on how to proxy a Web application and to perform basic penetration tests. As the team develops familiarity with security concepts, that familiarity will result in better security architecture and implementation.

Early detection through security reviews, static code analysis, threat modeling and "inline" penetration testing: By engaging, testing and thinking about security through the lifecycle, poor implementation is detected early. Static code analysis is a quick and effective way to scan an entire code-base, and to detect implementation issues. Threat models allow the team to consider the application as a complete system, and to identify where threats exist. Inline penetration tests are small, short-cycle pen tests aimed at validating security decisions made during feature security reviews. Pen tests also ensure that feature implementation was performed properly.

Make security activities a required process within the development lifecycle: In many organizations there is an emphasis on releasing features at almost any cost. Quality and security are sometimes sacrificed to benefit early release dates and additional features. In such organizations, it's helpful to have senior leadership 'buy-in', for them to understand how minimal the security processes are. This helps enforce best practices, even under the stress of release dates and client commitments. A team that approaches the implementation of security as a core value can build reliable and secure applications with very little overhead. It takes education, planning, a good set of tools and commitment to building secure applications. While few teams inherently possess these skills, there are numerous opportunities for external training (organizations like SANS and OWASP are a great resource for rigorous training).

Conclusion

We stand on the brink of a complete revolution in secure web development. By giving developers continuous feedback during the entire development process and by tracking vulnerabilities when they are introduced and when they are remediated, we can make major headway towards developing secure applications. This in turn results in helping to eradicate the current global epidemic of website hacking, by taking action at the beginning of the development process.

About SDG

SDG is a leading provider of technology, consulting and risk management solutions to strengthen enterprise businesses while managing IT risk. We focus on six practices: Risk and Security; Identity and Access Governance; Digital Collaboration; Quality Assurance; Mobility and Cloud. In addition we offer a GRC solution, called TruOps, to manage enterprise IT risk and compliance.

"SDG helps enterprises realize their dreams by helping them develop, manage and deploy solutions with acceptable risk."

For over two decades, SDG has enabled enterprises to realize their dreams by helping them develop, manage and deploy solutions with acceptable risk. We combine technology, thought leadership and a relentless passion for customer success. SDG partners with enterprise brands, but we specifically focus on mitigating client IT risk. Our ultimate goal is to help enterprises realize the opportunity of technology, increase innovation, improve speed-to-market and maximize returns on investment.