IJCSMS International Journal of Computer Science & Management Studies, Vol. 12, Issue 02, April 2012
ISSN (Online): 2231 –5268
www.ijcsms.com

To Study and Explain the Different Methods to Built a Secure Web Application

Vipin Kumar
Research Scholar, CMJ University, Shillong, Meghalaya (India)

Abstract

The secure web application is the most important thing for any type of transaction or similar things. Information security should enable, to the extent possible, a business to take the risks it is prepared to take on, by designing and deploying countermeasures that allow for sensible business risk. Additionally, seemingly small exposures should be dealt with if there is a business case. The role of the security architecture is not to steer the business away from risk, but rather to educate their business partners about the risks they are taking and provide countermeasures that enable the business to take as much risk as suits their goals. This is very important, it is no longer acceptable for enterprise security to exclusively function as an arbiter; security in the enterprise needs architecture and design advocates, and backing at runtime. Security policy and standards are not end goals in themselves, they need to be backed by a governance model that ensures they are in use, and that it is practically possible to build, deploy, and operate systems based on their intent. In practice this means that the security architecture must define reusable security services that allow developers to not be security experts yet still build a secure system.

Keywords: *Information Security, Secure Web Application, Security Threats.*

Introduction

Web application security is a branch of Information Security that deals specifically with security of websites, web applications and web services.

At a high level, Web application security draws on the principles of application security but applies them specifically to Internet and Web systems. Typically web applications are developed using programming languages such as PHP, Java EE, Java, Python, Ruby, ASP.NET, C#, VB.NET or Classic ASP.

OWASP is the emerging standards body for Web application security. In particular they have published the OWASP Top 10 which describes in detail the major threats against web applications. The Web Application Security Consortium (WASC) has created the Web Hacking Incident Database and also produced open source best practice documents on Web application security.

Security Threats

With the emergence of Web 2.0, increased information sharing through social networking and increasing business adoption of the Web as a means of doing business and delivering service, websites are often attacked directly. Hackers either seek to compromise the corporate network or the end-users accessing the website by subjecting them to drive-by downloading.

As a result, industry is paying increased attention to the security of the web applications themselves in addition to the security of the underlying computer network and operating systems.

The majority of web application attacks occur through cross-site scripting (XSS) and SQL injection attacks which typically result from flawed coding, and failure to sanitize input to and output from the web application. These are ranked in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors. According the security vendor Cenzic, the top vulnerabilities in March 2012 include:

- Cross Site Scripting, 37%
- SQL Injection, 16%
- Path Disclosure, 5%
- Denial of Service, 5%
- Code Execution, 4%
- Memory Corruption, 4%
- Cross Site Request Forgery, 4%
- Information Disclosure, 3%
- Arbitrary File, 3%
- Local File Include, 2%
- Remote File Include, 1%
- Overflow 1%
- Other, 15%

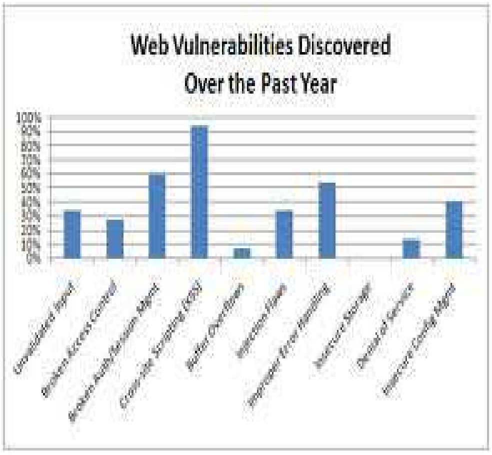

Figure1: Web Vulnerabilities

Methodology

While security is fundamentally based on people and processes, there are a number of technical solutions to consider when designing, building and testing secure web applications. At a high level, these solutions include:

- Black Box testing tools such as Web application security scanners, vulnerability scanners and penetration testing software
- White Box testing tools such as static source code analyzers
- Fuzzing Tools used for input testing
- Web application firewalls (WAF) used to provide firewall-type protection at the web application layer
- Password cracking tools for testing password strength and implementation

Application Service Architecture

Application service architecture (ASA) is an emerging discipline within IT that involves a top down approach to monitoring, controlling, securing, and optimizing applications in transit. This Application layer approach allows companies to manage the application service independently of the infrastructure to promote flexibility in the deployment, use, and provisioning of their application infrastructure. It also allows companies to better align IT with the business by bridging the gap between infrastructure and applications.

A properly designed ASA solution will enable companies to more effectively manage and secure applications across any kind of network medium. As such, ASA can be broken down into specific categories that represent the primary disciplines associated with managing applications as they transit network resources. Those disciplines include:

- Monitoring: Real-time information, as well as detailed reporting are the keys to supporting the other three principals, and ensure end-to-end visibility on the different components of the application service (application and network performance, backend infrastructure health, application security, trending, capacity planning, etc.)
- Controlling: Understanding the different aspects of a service, through monitoring, helps define specific behaviors in the interactions between the customers and the service provider. These behaviors can be mapped to specific use requirements, and further enforced.
- Securing: Protect against application layer attacks such as; malware, botnet, SQL injections, cross-site scripting, etc. In conjunction with traditional network layer security solutions; application aware security is quickly becoming an important

part of a comprehensive defense-in-depth strategy.

- Optimizing: In this new paradigm, the underlying networks used to deliver the business applications can change from one end to the other. Network optimization is there to maintain end-to-end consistency, and allow optimal performances when delivering the service to the customer.

This approach to managing applications provides a comprehensive strategy for delivering any application to any user. This focus is especially important for initiatives like virtualization, cloud computing, software or infrastructure as a service, and any external applications, like Web 2.0, that are used by the business.

This framework is focused on the application delivery process not the application server environment, which becomes an important distinction when contrasted against application performance management solutions.

Disciplines incorporated in ASA:

- Infrastructure: Application monitoring, Layer 4 -7, quality of service, application optimization, network behavior analysis, application delivery controllers
- Security: Secure web gateways, web application security, network behavior anomaly detection, content filtering, malware protection

APM or Monitoring

Application performance management (APM) or monitoring works alongside other IT monitoring systems including End-User Experience Monitoring, Synthetic Transaction Monitoring, Deep-Dive Monitoring and Business Activity Monitoring (BAM) solutions. According to Gartner, BTM and deep dive monitoring are "fundamentally distinct and their associated processes are typically carried out by different communities with different skill sets. As the technologies mature APM is now being viewed as a complete solution set. Maximum productivity can be achieved more efficiently through event correlation, system automation and predictive analysis which is now all part of APM. The buyer should still implement multiple products, even if it means greater architectural complexity and apparent functional overlap.

Applications

The solutions capture all of the transaction instances in the production environment and as such can be used for monitoring as well as for analysis and planning. Some applications include:

- Outage avoidance and problem isolation: Identification and isolation of tier-specific performance and availability issues.
- Service level management: Monitoring of SLAs and alerting of threshold breaches both at the end-user and infrastructure tier level.
- Infrastructure optimization: Modification of the configuration of data center infrastructure to maximize utilization and improve performance.
- Capacity planning: Analysis of usage and performance trends in order to estimate future capacity requirements.
- Change management: Analysis of the impact of change on transaction execution.
- Cloud management: Track the end-to-end transaction flow across both cloud (private, hybrid, public) and dedicated (on-premise, off-premise) infrastructure.

Web Application scanner

A web application security scanner facilitates the automated review of a web application with the expressed purpose of discovering security vulnerabilities and these are required to comply with various regulatory requirements. Web application scanners can look for a wide variety of vulnerabilities, including:

- Input /Output validation: (Cross-site scripting, SQL Injection, etc.)
- Specific application problems
- Server configuration mistakes/errors/version

In a copyrighted report published in March 2012 by security vendor Cenzic, the most common application vulnerabilities in recently tested applications include:

- Cross Site Scripting, 37%
- SQL Injection, 16%
- Path Disclosure, 5%
- Denial of Service, 5%
- Code Execution, 4%
- Memory Corruption, 4%
- Cross Site Request Forgery, 4%
- Information Disclosure, 3%
- Arbitrary File, 3%
- Local File Include, 2%
- Remote File Include, 1%
- Overflow 1%
- Other, 15%

Weaknesses and limitations

- Because the tool is implementing a dynamic testing method, it cannot cover 100% of the source code of the application and then, the application itself. The penetration tester should look at the coverage of the web application or of its attack surface to know if the tool was configured correctly or was able to understand the web application.
- It is really hard for a tool to find logical flaws such as the use of weak cryptographic functions, information leakage, etc
- Even for technical flaws, if the web application doesn't give enough clue, the tool cannot catch them
- The tool cannot implement all variants of attacks for a given vulnerability. So the tools generally have a predefined list of attacks and do not generate the attack payloads depending on the tested web application.
- The tools are usually limited in their understanding of the behavior of applications with dynamic content such as JavaScript, Flash, etc.

Strengths

- The tool can detect vulnerabilities of the finalized release candidate before shipping
- It simulates a malicious user by attacking and probing, and seeing what results are not part of the expected result set
- As a dynamic testing tool, it is not language dependent. A web application scanner is able to scan JAVA/JSP, PHP or any other engine driven web application.

Conclusion

Web applications have been highly popular since 2000 because they allow users to have an interactive experience on the Internet. Rather than just view static web pages, users are able to create personal accounts, add content, query databases and complete transactions. In the process of providing an interactive experience web applications frequently collect, store and use sensitive personal data to deliver their service. Customers benefit from the convenience of these applications, while tacitly taking on risk that private information stored in web applications will be compromised through hacker attacks, insider leaks etc.

According to the Privacy Rights Clearinghouse, more than 18 million customer records have been compromised in 2012 due to insufficient security controls on corporate data and web applications.

References

[1] "The Ghost in the Browser". Niels Provos et al. May 2007.
[2] "All Your iFrames Point to Us". Niels Provos et al. February 2008.
[3] "Improving Web Application Security: Threats and Countermeasures". Microsoft Corporation. June 2003.
[4] "Microsoft fortifies IE8 against new XSS exploits". Dan Goodin, the Register. February 2009.
[5] "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks". Fonseca, J.; Vieira, M.; Madeira, H., Dependable Computing, IEEE. Dec 2007.
[6] "CWE/SANS Top 25 Most Dangerous Programming Errors". CWE/SANS. May 2009.
[7] "2012 Trends Report: Application Security Risks". Cenzic, Inc.. 11 March 2012. Retrieved 9 July 2012.

[8] "The Web Hacking Incidents Database". WASC. January 2010.

[9] "Web Application Vulnerability Scanners". NIST.

[10] "Source Code Security Analyzers". NIST.

[11] "Fuzzing". OWASP.

[12] "Web application firewalls for security and regulatory compliance". Secure Computing Magazine. February 2008.