



IT Architecture | Delivery | Support | Management

WEB APPLICATION PENETRATION: WHERE ARE WE HEADED?

WHITEPAPER

ABSTRACT

The security community has experienced many high value corporate penetrations in recent months. Learn about some of the recent exploitation and vectors for attack in this resource. Explore the many possible measures that could have prevented such attacks. Specifically, this looks web security testing and ROI of a web application security investment.

VERSION 1.2: 10 AUGUST 2011

AUTHORED BY: CHRIS SERAFIN
RKON TECHNOLOGIES

HIGH VALUE CORPORATE PENETRATION – WHERE ARE WE HEADED?

Over the last couple months, the security community has witnessed a number of successful compromises to enterprise companies. From the cornerstone of two factor authentication, popular scripting, and database entities, to trusted certificate authorities, a popular blogging company, corporate email clearinghouse, and even a company that sells the same web application firewalls that should have prevented their own compromise. Could these attacks have been prevented and what do organizations need to be adding to their security strategy to prevent future attacks?

PREVENTATIVE MEASURES

While most organizations have adopted firewalls, anti-virus, SPAM filtering, IDS/IPS, and risk auditing, these preventative measures are no longer enough. Malicious users are finding new vectors for attack that cannot be patched, blocked, or filtered with out-of-the-box technology like firewalls and IPS signatures. Further, the growth of in-house web applications often without secure coding have contributed to increases in web application penetration. Web application attacks are even becoming automated and incorporated into botnets. Once a resolution is found for a current security threat, the attackers will migrate to or add additional emerging vectors for attack to their toolbox. All of these factors exponentially increase the risk associated with hosting a web site—making tools such as Data Leak Protection (DLP), Web Application Firewalls (WAF), and Proactive Web Security Testing more essential to your security strategy than ever. DLP and WAF must become viewed as not only integral to the well being of the data center, but also to achieving governance, risk, and compliance.

DLP, WAF, AND PROACTIVE WEB SECURITY TESTING

The business case for DLP, WAF, and Proactive Web App Security is growing due to the advanced methods of hackers, which we have witnessed in these recent attacks. However, there are many common misunderstandings as to why people don't think they need to invest in WAF, such as:

- **Companies think they are safe from attack because they have their servers locked down, patch them every week, have IDS, IPS, signatures are being blocked, SIM.** However, this is not enough. You can write signatures all day long but 1) if they are for web application vulnerabilities they are so broad that they are either going to get labeled as false positives and completely get tested out or 2) it is not going to be caught because it is not specific to your custom, in-house web application.
- **Companies don't realize that the increase in custom, in-house developed web applications poses a great threat because they don't do input validation. When users input data into forms, these apps don't do boundary checking or input validation—which is a vector of attack.** For example, is your web application checking to see that zip code field is only accepting six-digit numeric, and not alphanumeric or different types of scripts? This is the vector of attack in which all of these web app penetration incidents are occurring. They are not doing input validation—so hackers are able to enter tick marks or sql statements and leave that open in a form field such as “zip code.” It will dump the entire database of information—credit card numbers, passwords, usernames, and phone numbers—whatever else is stored in database.
- **Web application databases take in data (such as email addresses on a subscription list) but are not set up to verify that all the data is correct, actual email addresses and not scripting code or java code.** The application needs to put extra code in—called consistent cross side scripting—to prevent that risk (for medium to high

severity). Not only could this potentially put something into the database that is wrong, it could be read back. What are you going to be injecting? It depends on your level of boundary checking. Sometimes there is a good level of boundary checking, sometime there is very little.

- **Companies do not see WAF as an integral part of their security strategy. WAF spiders your entire web application, harvests your entire program, every page, sees all the perimeters, and URLs.** You customize the perimeters and what data is allowed. The WAF will permit any bad data from coming through. The WAF will prevent against any penetration.

ROI OF YOUR WEB APPLICATION SECURITY INVESTMENT

The ROI of scanning every week or month with web application security checks will save you exponentially in the end. This is known as your mean time for execution—once you get further along in the development of your web application without these checks, it so difficult to fix problems. You will essentially have to re-architect your entire application if you do not start your security approach from the beginning and on a reoccurring basis. The threat landscape is changing hourly. A check done last month will not stand up to a new threat that is being developed today. If you have built your foundation with web application security checks on a reoccurring basis, it will withstand future attacks. The attacks summarized below, which occurred recently to enterprise companies, could have been prevented with a web application security investment.

RECENT EXPLOITATION AND VECTORS FOR ATTACK

RSA/EMC

Vector: Targeted Phishing via Zero Day Excel/Adobe Exploit.

Summary: On March 17, 2011, RSA (the leading provider of two-factor authentication and SEIM solutions) was compromised via a spear phishing attack using a zero day exploit in Microsoft Excel and Adobe Flash. Although they are stating this was an ‘Advanced Persistent Threat,’ the attack was no more advanced than having someone open a file, exploit their system, install a backdoor, and tunnel proprietary information out of the company. In other words, this was a very basic attack methodology.

Compromised: Possibly source code and customer seed files for the SecureID infrastructure.

Prevention Ideas: DLP, Host Based IPS (HIPS), and a working SEIM would have prevented or at least detected this malicious activity before the crown jewels left the building.

Source: <http://bits.blogs.nytimes.com/2011/04/02/the-rsa-hack-how-they-did-it/>

PHP.NET

Vector: Unnamed Web App Vulnerability w/ Linux Root Exploit.

Summary: PHP is an extremely popular scripting language for web applications and is extensively used by a number of entities. On March 19, 2011, Wiki.php.net was compromised using an unnamed vulnerability in the Wiki web application, and when combined with a local Linux root exploit, enabled the attacker to gain complete control over the Wiki server.

Compromised: Although initial reports stated that the PHP source code repositories were compromised, this proved to be untrue. The attacker placed his name in the PHP code credits, but no other changes to the source code were discovered after extensive research. The site's account credentials were, however, compromised.

Prevention Ideas: A properly configured WAF could have prevented against many if not all web app vulnerabilities by filtering the input parameters allowed in the web app. Even if they did get past the WAF, a properly configured HIPS would have prevented the local root account escalation. This coupled with local file integrity checking and alerting would have caught the source code changes immediately.

Source: <http://www.php.net/archive/2011.php#id2011-03-19-2>

COMODO ROOT CA COMPROMISE

Vector: Partner Vendor's Credentials Compromised.

Summary: According to Phillip Hallam-Baker on the IT Security Comodo blog, "On March 15, 2011, a Comodo affiliate RA was compromised resulting in the fraudulent issue of 9 SSL certificates to sites in 7 domains. Although the compromise was detected within hours and the certificates revoked immediately, the attack and the suspected motivation require urgent attention of the entire security field" (<http://blogs.comodo.com/it-security/data-security/the-recent-ra-compromise/>).

Compromised: Issuance of fraudulent SSL certificates for major companies, including Google, AOL, Yahoo, and Skype, among others.

Prevention Ideas: A few months ago, the general consensus in the security community would be: use RSA or another vendor for two factor authentication. While that may still be the case, additional security checkpoints must be put in place to address future breaches leveraging compromised two factor authentication. DLP could have potentially prevented this if they were filtering for major companies and brand names in SSL certificates or CSRs leaving the organization, as well as a more rigorous validation process for new certificates.

Source: <http://blogs.comodo.com/it-security/data-security/the-recent-ra-compromise/>

MYSQL.COM AND SUN.COM

Vector: Cross Side Scripting (XSS) and Blind SQL Injection (Blind SQLi) in the web applications.

Summary: On March 28, 2011, Mysql.com and Sun.com were compromised via two popular web application vulnerabilities to gain access to the underlying databases. Given the size and popularity of MySQL and SUN, this is a major black eye for the organizations—both of which are now owned by Oracle.

Compromised: The databases were compromised and user credentials were posted on pastebin.com (<http://pastebin.com/raw.php?i=BayvYdcP>)

Prevention Ideas: These vectors for attack and the disclosure of the databases could have been prevented via input validation and HTTP/S response scrubbing (DLP for a web app) with many popular web application firewalls (WAF).

Sources: <http://www.net-security.org/secworld.php?id=10807>

http://seclists.org/fulldisclosure/2011/Mar/309?utm_source=twitterfeed&utm_medium=twitter

EPSILON

Vector: Unstated Email Server Breach.

Summary: On April 4, 2011, unknown attackers compromised one of Epsilon's email servers and accessed names and email accounts of their 2,500 high profile clients. Clients include Target, Walgreens, Citigroup, Tivi, Best Buy, JP Morgan Chase, Kroger, US Bank, and Marriot, among others. This email breach is stated to be the largest of its kind.

Compromised: The names and email accounts of their 2,500 high profile clients were compromised. This data will likely be used in high value targeted spear phishing scams within the next year.

Prevention Ideas: Since the vector of the compromise is unclear at this moment, a preventative measure cannot be provided at this time.

Source:

http://www.computerworld.com/s/article/9215488/About_50_clients_hit_by_Epsilon_e_mail_marketing_breach

BARRACUDA

Vector: SQL Injection while their WAF was offline.

Summary: In response to this compromise, Barracuda's blog stated that the WAF in front of their corporate web site was unintentionally placed in a passive mode and then completely offline for a planned maintenance window. During this maintenance window, automated scripts began combing the site for input parameters that were not properly validated. Once a valid SQL injection flaw was discovered in a seemingly innocent customer case study, the shared database was compromised.

Compromised: This attack compromised lead, employee, and partner contact information.

Prevention Ideas: A well-configured web application firewall could have prevented this attack. This should serve as a reminder to the enterprise that altering one's security model, even if only for a downtime window, can pose a significant threat.

Source: <http://www.barracudalabs.com/wordpress/index.php/2011/04/11/learning-the-importance-of-waf-technology-the-hard-way/>

LISAMOON MASS INFECTIONS

Vector: Mass Automated SQL Injection Botnet.

Summary: On March 29, 2011, security researchers noticed that an automated botnet was exploiting SQL injection attacks on various web applications such as CRM and blogs. It injected code into the HTML of the web page to redirect users to an infected third party site, which exploited the client's system via malicious JavaScript.

Compromised: Over a million web sites were infected, which resulted in malware to clients that visited their site and further spreading fake anti-virus malware.

Source: <http://twitteling.com/2011/04/liza-moon-infect-millions-of-sites/>

WORDPRESS

Vector: Database Credentials in PHP File and Improper File Permissions.

Summary: On April 18, 2011, Wordpress—one of the largest and most popular web blogging sites—underwent a mass compromise of personal blog sites. The attackers infected the blogging sites with malware intended to further compromise systems.

Compromised: The credentials of each affected database and the integrity of the web content of the personal blogs was compromised.

Prevention Ideas: Considering that your average blogger does not know nor understand web app security or Linux file permissions, most experts are stating that this is a massive design flaw in the WordPress application itself; a low level security review of the code would have discovered this threat. A web app firewall may have prevented this in the tuning of the WAF, but a manual security assessment would have been needed.

Source: <http://news.softpedia.com/news/WordPress-Design-Flaw-Blamed-for-Recent-Mass-Blog-Compromise-139623.shtml>

SONY PLAYSTATION

Vector: Unknown, possibly SQL Injection

Summary: In late April 2011, The Sony PlayStation network was compromised from an 'external threat.' The global PSN and Sony networks were shut down while forensic experts examined the vector for attack. While Sony has been suspiciously silent on the details, many experts believe the method of attack was an SQL injection in one of their web applications.

Compromised: The breach had led to the theft of the data of the 77 million users, including credit card data.

Prevention Ideas: A properly configured DLP and SIEM solution should have caught the data trying to leave the organization. The information should have been encrypted in the databases from the beginning to prevent this data leakage. If this was a web application attack, a WAF solution would have caught and been able to mitigate these series of threats.

Source: <https://www.infosecisland.com/blogview/13337-Sony-PlayStation-Hack-70-Million-Users-Details-Stolen.html> and <http://www.techopsguys.com/2011/05/05/sony-compromised-by-apache-bug/>

ABOUT RKON

We specialize in the architecture, delivery, support, and management of Virtualization, Security Infrastructure, and Network Infrastructure. We care about providing our clients with world-class IT services, support, and long-term relationships. We strive to provide superior client satisfaction while offering solutions that secure, scalable, and highly available—on site or in the cloud.

Looking to move your data center to a private cloud? We have enabled many IT departments to refocus on adding value to the business while we supply the on-demand computing platform. Please visit www.rkon.com to learn more about our capabilities.