INTRODUCTION

The Open Web Application Security project (OWASP) Top Ten provides a powerful awareness document for web application security. The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are as identified by a variety of security experts from around the world who have shared their expertise to produce this list.

The U.S. Federal Trade Commission strongly recommends that all companies use the OWASP Top Ten and ensure that their partners do the same. In addition, the U.S. Defense Information Systems Agency has listed the OWASP Top Ten as key for best practices.

In the commercial market, the Payment Card Industry (PCI) standard has adopted the OWASP Top Ten and lists it as a key requirement as part of section 6 – “Develop and maintain secure systems and applications” mandating that all web applications be developed according to security guidelines to protect against the OWASP Top 10.

OWASP TOP 10

The OWASP team updates the Top 10 list periodically as the threat landscape changes, new technologies released, attack vectors are changed and utilized differently, and new weaknesses found.

Broken Authentication and Session Management moved up in prevalence based on our data set. This area is being looked at harder, not because issues are actually more prevalent. This caused Risks A2 and A3 to switch places.

Cross-Site Request Forgery (CSRF) moved down in prevalence based on our data set from 2010-A5 to 2013-A8. We believe this is because CSRF has been in the OWASP Top 10 for 6 years, and organizations and framework developers have focused on it enough to significantly reduce the number of CSRF vulnerabilities in real world applications.

We broadened Failure to Restrict URL Access from the 2010 OWASP Top 10 to be more inclusive. 2010-A8: Failure to Restrict URL Access is now 2013-A7: Missing Function Level Access Control – to cover all of function level access control. There are many ways to specify which function is being accessed, not just the URL.

We merged and broadened 2010-A7 & 2010-A9 to CREATE 2013-A6, Sensitive Data Exposure. This new category was created by merging 2010-A7 – Insecure Cryptographic Storage & 2010-A9 - Insufficient Transport Layer Protection, plus adding browser side sensitive data risks as well. This new category covers sensitive data protection (other than access control which is covered by 2013-A4 and 2013-A7) from the moment sensitive data is provided by the user, sent to and stored within the application, and then sent back to the browser again.

We added: 2013-A9: Using Components with Known Vulnerabilities. This issue was mentioned as part of 2010-A6 – Security Misconfiguration, but now has a category of its own as the growth and depth of component based development has significantly increased the risk of using components with known vulnerabilities.

WEB APPLICATION SECURITY CHALLENGES

Web applications are attractive targets to hackers as often they are public facing applications that are required to be open to the Internet as they provide major e-commerce and business driving tools for organizations. Connected to backend databases web applications are perfect for hackers as these databases are the primary repository for cardholder data, company data and other sensitive information.

According to the SANS, attacks against web applications constitute more than 60% of the total attack attempts observed on the Internet. Web application vulnerabilities such as SQL injection and Cross-Site Scripting flaws in custom-built applications account for more than 80% of the vulnerabilities being discovered.

The difficulty in protecting web applications is their architecture and dynamics. While network security is simple–define security policies to allow/block traffic to/out of several applications consist of hundreds and sometimes thousands of different elements (URLs, parameters and cookies).

Manually creating different policies for each of these items is almost impossible and obviously does not scale. On top of this, web applications change frequently with new URLs and parameters being added making it difficult for security administrators to update their security policies.
FORTIWEB WEB APPLICATION FIREWALL

Combining both a Web Application Firewall, Web vulnerability scanner and a layer 7 load balancer in a single platform the FortiWeb solution allows enterprises to protect against application level attacks, help identifying application vulnerabilities in web applications while ensuring availability and accelerating application access with capabilities such as hardware SSL offload, compression and caching. Using advanced techniques to protect against SQL injection, Cross site scripting and a range of other attacks FortiWeb helps protect sensitive data and prevent identity theft, financial fraud and brand erosion which can result in significant damage while ensuring application availability.

FortiWeb provides flexible and reliable protection to address the OWASP Top Ten by utilizing a range of in-depth security modules and technologies. Sophisticated attacks are blocked using a multi-layered security approach. Incorporating a positive and a negative security module based on bi-directional traffic analysis and an embedded behavioral based anomaly detection engine FortiWeb can protect against a broad range of threats without the need for network re-architecture and application changes.

FORTIWEB INCORPORATES TWO SECURITY MODELS IN ORDER TO PROTECT BOTH KNOWN AND UNKNOWN VULNERABILITIES:

POSITIVE SECURITY MODEL

FortiWeb uses Positive Security Model to protect against any known and unknown vulnerabilities. Once a new web application policy is defined, FortiWeb starts monitoring traffic flowing to the application and based on a behavioral analysis technology called Auto-Learn FortiWeb builds a dynamic baseline of allowed elements for the application. By analyzing normal user behavior FortiWeb understands how the web application should be accessed. Authorized URLs are created with the relevant parameters for each of the URLs. Parameter characteristics are profiled as well to provide an overall picture of the web application structure and what constitutes as normal user behavior.

The Auto-Learn profiling capability is completely transparent and does not require any changes to the application or network architecture. FortiWeb does not scan the application in order to build the profile, but rather analyzes the traffic as it monitors it flowing to the application.

By creating a comprehensive security model of the application FortiWeb can now protect against any known or unknown vulnerabilities, zero day attacks such as SQL Injection, Cross Site Scripting, and other application layer attacks. These types of attacks require manipulating URLs and parameters by inserting different characters that are not part of learned profiled and as such can be immediately blocked.

NEGATIVE SECURITY MODEL

FortiWeb includes a full application signature dictionary to protect against known application layer attacks and application logic attacks. A sophisticated engine scans both inbound and outbound traffic, matching elements with pre-defined known exploits. FortiWeb provides an enhanced flexible engine that also allows customers to write their own signatures using a regular expression engine which provides the ability to create new and customized signatures for every application and vulnerability.

FortiWeb’s signature dictionary is updated regularly and automatically via FortiGuard Labs, a Security Subscription Service which delivers continuous, automated updates and offers dynamic protection based on the work of the Fortinet® Global Security Research Team, which researches and develops protection against known and potential security threats.

FortiWeb extends the negative security layer with an HTTP RFC enforcement layer making sure any access to the protected application is done according to the HTTP standard. FortiWeb provides an enhanced policy configuration with multiple rules to protect against buffer overflows, encoding based attacks and any other attack that tries to manipulate the HTTP protocol.
ADDITIONAL FORTIWEB FUNCTIONALITY HELPS PROTECT AGAINST OWASP TOP 10

DATA LEAK PREVENTION AND INFORMATION DISCLOSURE

FortiWeb extends monitoring and protection to outgoing traffic such as credit card leakage and information disclosure. Providing multiple policies for information disclosure FortiWeb immediately alerts any web application abnormalities that may be caused by an attack or by poor application configuration. Rules can be extended to rewrite information disclosure such that users are not exposed to any sensitive application data. Additional credit card leakage rules make sure any outgoing traffic does not include credit card numbers.

FortiWeb also allows customers to create their own custom rules providing an entire solution for Data Leak Prevention.

WEB VULNERABILITY SCANNER

FortiWeb is the only vendor that provides a Vulnerability Scanner module within the web application firewall that automatically scans and analyzes the protected web applications and detects security weaknesses, known and unknown vulnerabilities. Together with the web application firewall FortiWeb completes a comprehensive solution for PCI DSS requirement 6.6 and 6.5 allowing organizations to scan their applications, identify vulnerabilities and protect them in real time from the same platform.
PAGE ACCESS AND START PAGE ENFORCEMENT

Enforcing access to the application according to the correct business logic is an important part of web application security. Applications that do not enforce this open themselves to Cross Site Request Forgery (CSRF) attacks. For example, e-commerce applications should not allow access to the shipping or payments stage directly, without prior access to the ordering pages.

By defining a Page Access rule on FortiWeb CSRF attacks are immediately blocked helping with OWASP Top Ten A5 requirement.

Additionally, FortiWeb offers a Start Page policy as well which defines the entry point to the application helping with requests that try to circumvent authentication or any other environment that requires users to start browsing at a specific page.
## OWASP TOP TEN AND FORTIWEB MITIGATION TECHNIQUE

The table below lists the OWASP Top Ten and the corresponding FortiWeb mitigation techniques.

<table>
<thead>
<tr>
<th>OWASP Top 10</th>
<th>Explanation</th>
<th>FortiWeb Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1. Injection</td>
<td>Injection flaws, such as SQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker’s hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.</td>
<td>Auto-Learn profiling automatically builds an allowed baseline to provide comprehensive request validation capability to enforce strict URL and parameter control. Enhanced application signature detection engine updated regularly by the FortiGuard Labs team adds a secondary layer for abnormal characters and known injection strings.</td>
</tr>
<tr>
<td>A2. Broken Authentication and Session Management</td>
<td>Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users’ identities.</td>
<td>FortiWeb enforces session management with strict cookie control by tracking all sessions and cookies. Any attempt to compromise cookies is mitigated. Additionally various Authentication Offload capabilities (supporting Local, LDAP and NTLM) are included to offer additional security layer.</td>
</tr>
<tr>
<td>A3. Cross-Site Scripting (XSS)</td>
<td>XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation and escaping. XSS allows attackers to execute scripts in the victim’s browser, which can hijack user sessions, deface web sites, or redirect the user to malicious sites.</td>
<td>FortiWeb application signature layer engine includes various XSS signatures to protect against the most sophisticated XSS attacks. Additionally, FortiWeb’s Auto-Learn builds a comprehensive baseline of normal behavior such as URLs and parameters. Any attempt to inject illegal and unknown characters to arguments can be immediately blocked.</td>
</tr>
<tr>
<td>A4. Insecure Direct Object References</td>
<td>A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.</td>
<td>Auto-Learn profiling builds a comprehensive profile of allowed elements within the application. Any attempt to manipulate a parameter will trigger an alert and immediately be blocked. Hidden Fields Rules detect and block any attempt by the client to alter a hidden parameter value.</td>
</tr>
<tr>
<td>A5. Security Misconfiguration</td>
<td>Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.</td>
<td>1) Using Auto-Learn FortiWeb will block any attempt made by an attacker to exploit a misconfigured web application. 2) Through monitoring application responses, FortiWeb is able to identify any application failure. 3) Vulnerability Scanner module scans the protected applications, finds inherent misconfigurations and quickly turns them to security rules.</td>
</tr>
<tr>
<td>A6. Sensitive Data Exposure</td>
<td>Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attacker may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.</td>
<td>FortiWeb protects against attacks that lead to sensitive data exposure such as SQL Injection and other injection types. Additionally, FortiWeb inspects all web server outgoing traffic for sensitive data such as Social Security numbers, credit card number and other predefined or custom based sensitive data. When identified, FortiWeb can either mask the data or block it from reaching the client all together.</td>
</tr>
<tr>
<td>A7. Missing Function Level Access Control</td>
<td>Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access functionality without proper authorization.</td>
<td>FortiWeb validates sessions and any attempt to manipulate it to gain unauthorized or elevated application access control. FortiWeb can also perform authentication with local or RADIUS and LDAP servers to provide another security layer.</td>
</tr>
</tbody>
</table>
SUMMARY

The OWASP Top Ten provides a great starting point for customers to measure their application security and prioritize their risk. Mandated by the Payment Card Industry (PCI) standard as a key requirement as part of section 6 and widely adopted by many organizations the OWASP Top 10 is an important guideline that helps organizations focus on application security.

FortiWeb’s integrated Web Application Firewall and Vulnerability assessment scanner allow customers to protect against high-risk attacks such as defined in the OWASP Top Ten and many others.