SecureSphere and OWASP 2010 Top Ten Most Critical Web Application Security Risks

Introduction

The Open Web Application Security Project (OWASP) Top Ten represents the most critical Web application security risks identified by a variety of Web application security experts from around the world. The OWASP Top Ten has become a de facto standard for application security. The U.S. Federal Trade Commission strongly recommends that all companies adopt the OWASP Top Ten as an internal benchmark. In addition, it is a key requirement in the Payment Card Industry Data Security Standard (PCI DSS), which states in section 6.5 that all web applications should be developed based on secure coding best practices such as the OWASP guidelines.

This Technical Brief discusses how the SecureSphere Web Application Firewall addresses the OWASP Top Ten 2010. It is not intended to replace reading the OWASP Top Ten document, which can be found here: http://www.owasp.org/index.php/OWASP_Top_Ten_Project. Instead, the document assumes the reader is familiar with the OWASP Top Ten and summarizes how Imperva SecureSphere can help to mitigate each Top Ten threat.

1. Injection

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
EASY	COMMON	AVERAGE	SEVERE

Definition:
Injection flaws such as SQL, OS, and LDAP injection occurs 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. Injection flaws, are very prevalent, often found in SQL queries, LDAP queries, XPath queries, OS Shell commands and program arguments. The impact is usually very severe as the entire database can be read or modified and allow access to the database schema, account or even OS level access.

SecureSphere Mitigation:
SecureSphere accurately detects and blocks injection attacks using a variety of techniques. SecureSphere protects against SQL injection by using a unique defense algorithm designed by the Imperva Application Defense Center (ADC). The Imperva ADC is a premier research organization that provides security analysis, vulnerability discovery, and compliance expertise. The SQL injection defense algorithm developed by the ADC combines information from the Web application profile (positive security model) and matches this information with attack signatures (negative security model) using SecureSphere's Correlated Attack Validation engine. Additional injection attacks such as LDAP, XPath, and OS injection are blocked by SecureSphere using pre-defined signatures.

Examples of anomalies detected by the Web application profile security rules include:

» If an attacker attempts to change the values of parameters that were fixed by the Web application and that should not be changed, SecureSphere will alert and block the request.
» If a parameter length exceeds the expected maximum length, SecureSphere will alert and block such an evasion attempt.
» If a parameter includes unexpected characters, such as quotation marks, angle brackets, and asterisks, that do not fit the application profile, SecureSphere will alert and block the request.

2. Cross Site Scripting (XSS)

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
AVERAGE	WIDESPREAD	EASY	MODERATE

Definition:
Cross Site Scripting allows attackers to execute script in the victim's browser which can hijack user sessions, insert hostile content, hijack the user's browser using malware, or redirect the user to phishing or malware malicious sites. XSS flaws occur when an application includes user-supplied data in a page sent to the web browser without properly validating or escaping that content. For example, an application using untrusted data in the construction of an HTML snippet without validation or escaping is vulnerable to attacks that can modify a browser parameter and causes the user's session ID to be sent to the attacker's website and hijack the user's current session.

SecureSphere Mitigation:
SecureSphere accurately detects and mitigates XSS attacks using both positive ("white list") and negative ("black list") models. The negative security model explicitly declines known attack signatures and includes XSS signatures, XSS keywords, suspicious patterns and other XSS indicators to correctly detect XSS attacks. SecureSphere accurately detects XSS attacks by combining custom signatures and special correlation rules that match signatures with the response code received from the Web application. SecureSphere also analyzes where the XSS pattern appears (for example, in a web page hyperlink) and whether the XSS pattern is new or a standard part of the Web application. An additional layer of defense against XSS is achieved by applying Dynamic Profiling to create a positive security model of the application structure and use the dynamic Web application profile to allow only legitimate user input. The profile serves as the baseline governing detailed application-layer behavior. Valid application changes are automatically recognized and incorporated into the profile over time.

Furthermore, SecureSphere utilizes a combination of algorithms that validate user input and application behavior to accurately detect and mitigate evasion techniques. Since XSS vectors are rendered and run at the client, there are a wide range of evasion techniques based on specific browser behavior. Although these techniques might not be malicious as is, they can be used to obfuscate malicious code. SecureSphere is the only WAF that can decode all browser encodings such as HTML, hex, and Unicode encoding and validate input against the application profile.

3. Broken Authentication and Session Management

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
AVERAGE	COMMON	AVERAGE	SEVERE

Definition:
Applications are frequently built using custom authentication and session management schemes that have flaws in areas such as logout, password management, timeouts, remember me, secret question, and account update. Such flaws may allow accounts to be attacked and compromise passwords, keys, session tokens, or exploit implementation flaws to assume other users' identities. One example is an airline reservation application that supports URL rewriting and inserts the session IDs in the URL. In this example, an authenticated user of the site informs his friends of the sale and emails the URL without knowing he is giving away his session ID. When his friends use the link they will also use his session and credit card information.

SecureSphere Mitigation:
SecureSphere protects against session attacks such as session hijacking, session fixation, and session tampering without changing the application's session handling mechanism. SecureSphere will track and enforce session variables, but it will not rewrite or intrude on an application's session management facility. This approach mitigates the issues that intrusive mechanisms introduce often either breaking the application or requiring re-coding. SecureSphere also blocks session attacks by profiling cookies for session hijacking, session replay, cookie injection and cookie manipulation. Additionally, SecureSphere's strong correlation capabilities address complex attacks and allow users to define complex rules in seconds. For instance, users can define advanced session hijacking detection rules such as the simultaneous user authentication from two different IPs.

In addition, SecureSphere's Application User Tracking automatically captures Web application user names and associates all subsequent session activity with that specific username. As a result, SecureSphere can detect successful or failed logins and act accordingly. Lastly, the aforementioned protection mechanisms combined with SecureSphere's LDAP integration enables strong access control validation mechanism that is both user and role aware.

4. Insecure Direct Object References

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
EASY	COMMON	EASY	MODERATE

Definition:
A direct object reference occurs when a reference to an internal implementation object, such as a file, directory, or database key is exposed. Applications do not always verify the user is authorized for the target object, resulting in an insecure direct object reference flaw. Such flaws can compromise all the data that can be referenced by the parameter. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

SecureSphere Mitigation:
SecureSphere utilizes dynamic Web application profiling to identify and understand what the expected behavior is for elements of an application, for example knowing what type of input should be within each form field. Dynamic Profiling gives SecureSphere the ability to learn what is normal and accepted input and allows SecureSphere to detect attempts to manipulate input values. Using Dynamic Profiling, SecureSphere can mitigate insecure direct object references by blocking parameter tampering. Moreover, SecureSphere's parameter read-only capability ensures users only follow the links provided by the application without the ability to tamper with them.

An additional layer of protection can be achieved by using the SecureSphere database security layer and correlating it with the Web security layer. For instance, if an attacker attempts to manipulate an object that is expected to be used for a database operation, then the SecureSphere Database Security gateway can detect and block this attack. This serves as an example of the value that SecureSphere's integrated database capability can uniquely provide for comprehensive application data security.

5. Cross Site Request Forgery (CSRF)

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
AVERAGE	WIDESPREAD	EASY	MODERATE

Definition:
A CSRF attack creates forged HTTP requests and tricks a victim into submitting them via image tags, XSS, or other techniques. If the user is authenticated, the attacker succeeds. CSRF takes advantage of Web applications that allow attackers to predict all the details of the transaction. Since browsers send credentials like session cookies automatically, attackers can create malicious web pages which generate forged requests that are indistinguishable from legitimate ones.

SecureSphere Mitigation:
SecureSphere provides defense against Cross Site Request Forgery (CSRF) attacks by defining a rule within the correlation security engine. The CSRF rule leverages the dynamic profiling to identify and block CSRF attacks; hence SecureSphere can block all CSRF attacks that go through a Web server. SecureSphere has the ability to detect requests from external sources to the web application and applies security controls based on the rules defined.

In addition, new application security hazards produced by Web 2.0 technologies have elevated the risk of CSRF attacks. SecureSphere automatically detects CSRF injections using its profiling mechanism to learn which external web applications are allowed to generate requests to the protected domain. By simultaneously supporting both white list and black list security models, SecureSphere's flexible architecture offers granular control of security policies to determine which external applications are allowed and prevent any other external source to generate requests thus preventing CSRF.

6. Security Misconfiguration

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
EASY	COMMON	EASY	MODERATE

Definition:
Security misconfiguration can happen at any level of an application attack, including the platform, web server, application server, framework, and custom code. Such flaws can give attackers access to default accounts, unused pages, unpatched flaws, unprotected files, and directories to gain unauthorized access to system data.

SecureSphere Mitigation:
SecureSphere protects against security misconfiguration through Dynamic Profiling by understanding normal input. For instance, if a developer or network admin misconfigured anything within the environment such the OS, Web or Application Server, or the application code; SecureSphere detects this deviation from normal usage and blocks it. In addition, SecureSphere integrates with the leading scanners that scan web servers and web applications for misconfigurations and vulnerabilities. Through the scanner integration, the scan results are imported into SecureSphere and provides instant mitigation for imported misconfiguration and vulnerabilities using a "virtual patch."

7. Insecure Cryptographic Storage

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
DIFFICULT	UNCOMMON	DIFFICULT	SEVERE

Definition:
Many web applications do not properly protect sensitive data such as credit cards, Social Security Numbers (SSNs), and authentication credentials with appropriate encryption or hashing. Attackers may use this weakly protected data to conduct identity theft, credit card fraud, or other crimes.

SecureSphere Mitigation:
SecureSphere can mitigate insecure cryptographic storage by inspecting outbound traffic to identify potential leakage of sensitive data such as cardholder data and SSNs through the application. Furthermore, SecureSphere reports on where sensitive data is used in the application and can prevent this information from leaving the organization. In addition, when the storage device is a database (as it commonly is), SecureSphere's Database Gateways can be deployed to monitor and identify sensitive information sent to the database in an un-encrypted format using similar capabilities that the SecureSphere WAF uses to prevent data leakage from the application. This serves as another example of why the integration of Web and database security in SecureSphere provides unique value for comprehensive application data security.

8. Failure to Restrict URL Access

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
EASY	UNCOMMON	AVERAGE	MODERATE

Definition:
Many web applications do not always properly protect page requests. Applications need to check URL access rights or perform access control checks when these pages are accessed, otherwise, failure to restrict URL access will enable attackers to forge URLs to access these hidden pages. For example, if the URLs are supposed to require both authentication and admin rights for access to the "admin_getappinfo" page, and the attacker is not authenticated and access to the pages are granted, then unauthorized access was allowed. If an authenticated, non-admin user is allowed to access the "admin_getappinfo" page, this is a flaw and may lead the attacker to more improperly protected admin pages.

SecureSphere Mitigation:
SecureSphere protects against unauthorized URL access by invoking control at various levels; both at the application level using dynamic Web application profiling and at the user level using SecureSphere's correlation security engine. In the event that unauthorized URL access violations occur, SecureSphere generates an alert and returns a custom, detailed error page to the end user.

9. Insufficient Transport Layer Protection

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
DIFFICULT	COMMON	EASY	MODERATE

Definition:
Applications frequently fail to encrypt network traffic when it is necessary to protect sensitive communications. When they do, they sometimes support weak algorithms, use expired or invalid certificates, or do not use them correctly.

SecureSphere Mitigation:
SecureSphere can actively terminate SSL connections when configured as a transparent or reverse proxy and thus has the ability to enforce SSL to specific resources. Therefore if the application is HTTP only, SecureSphere can

add the SSL layer on top of it. SecureSphere has the ability to configure SSL access to configured URLs using URL rewriting rules that redirect cleartext requests to use HTTPS. Additionally, SecureSphere is FIPs compliant and has the ability to prevent any SSL connection fthat uses non-FIPs ciphers.

10. Unvalidated Redirects and Forwards

Attack Vector Exploitability	Security Weakness Prevalence	Security Weakness Detectability	Impact
AVERAGE	UNCOMMON	EASY	MODERATE

Definition:
Unvalidated redirects and forwards occur when attackers redirect users to other pages and websites such as phishing or malware sites or use forwards to access unauthorized pages. Web applications frequently redirect and forward users to other pages in a similar manner. Sometimes the target page is specified in an unvalidated parameter, allowing attackers to choose the destination page. Such redirects may attempt to install malware or trick victims into disclosing passwords or other sensitive information and unsafe forwards may allow access control bypass. Without proper validation, attackers can use unvalidated forwards to bypass authentication or authorization checks.

SecureSphere Mitigation:
SecureSphere addresses unvalidated redirects and forwards in multiple ways. By basing security decisions upon multiple events, SecureSphere's advanced correlation rules incorporate all security elements to detect complex, multi-stage attacks. For example, SecureSphere's advanced correlation rules checks for user input in the form of URLs and is then followed by a profile violation such as a parameter tampering that is not supposed to be used to reference to an external domain. By correlating these different suspicious aspects of the same request, SecureSphere can conclude that it is an unvalidated redirect and forward and will block this attack. Furthermore, unvalidated redirects will frequently operate as a XSS attack via a link in an email or forum. SecureSphere can effectively protect against this attack by profiling requests from external web applications and determining which external sources are legitimate and which are not. In addition, ThreatRadar-powered SecureSphere WAF provides an extra layer of protection against malicious sources, anonymous proxies, and Phishing URLs. By integrating credible, timely information on known attack sources into SecureSphere WAF, ThreatRadar can accurately block traffic from malicious sources before an attack is launched.

Headquarters
3400 Bridge Parkway
Suite 101
Redwood Shores, CA 94065
Tel: +1-650-345-9000
Fax: +1-650-345-9004