Mitigating the OWASP Top 10 2013 with Imperva SecureSphere

The Open Web Application Security Project (OWASP) Top 10 represents the most critical Web application security risks identified by broad consensus of application security experts from around the world. The OWASP Top 10 has become a de facto standard for application security.

Many regulatory standards and organizations, including the U.S. Federal Trade Commission, DISA, and MITRE, use the OWASP Top 10 as a benchmark for measuring security risks. In addition, the OWASP Top 10 is a key requirement in the Payment Card Industry Data Security Standard (PCI DSS), which states in section 6.5 that Web applications should be developed based on industry best practices such as OWASP guidelines.

This technical brief discusses how the SecureSphere Web Application Firewall addresses the OWASP Top 10 2013. It is not intended to replace reading the OWASP Top 10 report. Instead, the document assumes the reader is familiar with the OWASP Top 10 and summarizes how Imperva SecureSphere helps to mitigate each Top 10 threat.

A1—Injection

Threat Agents	Application Specific
Attack Vectors	Exploitability EASY
Security Weakness	Prevalence COMMON
	Detectability AVERAGE
Technical Impacts	Impact SEVERE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Injection flaws, such as SQL, OS, and LDAP injection, occur when an application sends untrusted data 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 data without proper authorization.

Injection flaws are very prevalent, particularly in legacy code. The impact is usually very severe as the entire database can be read or modified.

SecureSphere Mitigation

SecureSphere accurately detects and blocks injection attacks using a variety of techniques. SecureSphere protects against SQL injection by using a unique SQL injection correlation engine 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. SecureSphere using pre-defined signatures blocks additional injection attacks such as LDAP, XPath, and OS injection.

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 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

A2—Broken Authentication and Session Management

Threat Agents	Application Specific
Attack Vectors	Exploitability AVERAGE
Security Weakness	Prevalence WIDESPREAD
	Detectability AVERAGE
Technical Impacts	Impact SEVERE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.

Developers frequently build custom authentication and session management schemes, but building these correctly is hard. Authentication flaws may allow some or even all accounts to be attacked. Once successful, the attacker can do anything the victim could do. Privileged accounts are frequently targeted.

SecureSphere Mitigation

The SecureSphere Web Application Firewall prevents attackers from exploiting broken authentication and session management. SecureSphere stops session attacks like session hijacking, session fixation, and session tampering. When deployed as a transparent bridge, 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 often introduce by either breaking the application or requiring re-coding. When deployed as a reverse proxy, SecureSphere can also sign and encrypt cookies.

SecureSphere profiles application activity to determine which cookies are read-only and which cookies the client can modify. Based on this information, SecureSphere can correctly identify attacks like cookie injection or cookie poisoning. 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.

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 detects successful or failed logins and can prevent brute force attacks.

A3—Cross-Site Scripting (XSS)

Threat Agents	Application Specific
Attack Vectors	Exploitability AVERAGE
Security Weakness	Prevalence VERY WIDESPREAD
	Detectability EASY
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Cross-site Scripting (XSS) is the most prevalent Web application security flaw. XSS flaws occur whenever an application takes untrusted data and sends it to a Web browser without proper validation or 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.

Detection of most XSS flaws is fairly easy via testing or code analysis.

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 on the client, there are a wide range of evasion techniques based on specific browser behavior. Although these techniques might not be outright attacks, they can be used to obfuscate malicious code. SecureSphere is the only Web application firewall that decodes all browser encodings such as HTML, hex, and Unicode encoding and validates input against the application profile.

A4—Insecure Direct Object References

Threat Agents	Application Specific
Attack Vectors	Exploitability EASY
Security Weakness	Prevalence COMMON
	Detectability EASY
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

An insecure 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.

Applications frequently use the actual name or key of an object when generating Web pages. Applications don't always verify the user is authorized for the target object. This results in an insecure direct object reference flaw. Testers can easily manipulate parameter values to detect such flaws.

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 to allow for each form field. Dynamic Profiling enables SecureSphere to learn "normal" user activity and allows SecureSphere to detect attempts to manipulate input values. Using Dynamic Profiling, SecureSphere mitigates 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.

Using the SecureSphere Database Firewall and correlating it with the SecureSphere Web Application Firewall provides an additional layer of protection. For instance, if an attacker attempts to manipulate an object that is expected to be used for a database operation, then the SecureSphere Database Firewall will 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.

A5—Security Misconfiguration

Threat Agents	Application Specific
Attack Vectors	Exploitability EASY
Security Weakness	Prevalence COMMON
	Detectability EASY
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

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.

If an application lacks proper security controls, an attacker can potentially access default accounts, unused pages, or unprotected files or exploit unpatched flaws to gain unauthorized access to or knowledge of the system.

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 as 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 provide instant mitigation for imported misconfiguration by virtually patching vulnerabilities.

A6—Sensitive Data Exposure

Threat Agents	Application Specific
Attack Vectors	Exploitability DIFFICULT
Security Weakness	Prevalence UNCOMMON
	Detectability AVERAGE
Technical Impacts	Impact SEVERE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Many Web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers 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.

Even if sensitive data is encrypted, poor key generation and key management and weak encryption algorithms, particularly weak password hashing techniques, can expose sensitive data to compromise.

SecureSphere Mitigation

The SecureSphere Web Application Firewall mitigates sensitive data exposure by inspecting outbound traffic for sensitive data such as cardholder data and social security numbers. If SecureSphere detects sensitive data leaks, it blocks Web server responses before they reach the end user.

The SecureSphere Web Application Firewall also prevents many of the attacks that lead to sensitive data exposure—attacks like SQL injection and OS command injection. SecureSphere also stops malicious users from exploiting Web server and application vulnerabilities to gain control of a Web server and access sensitive data. By preventing the attacks that lead to a data breach, organizations can drastically reduce the opportunity for attackers to access encrypted data.

Organizations must also protect sensitive data where it lives—in databases and file servers. Otherwise, malicious insiders can bypass application controls to access sensitive data. Even when strong encryption algorithms and FIPS-certified key management are implemented, if the database dynamically decrypts sensitive data, then malicious users can potentially access and steal this data. SecureSphere Database Security solutions audit all access to sensitive data in the database. They can detect excessive user rights and dormant accounts to prevent unauthorized users from accessing sensitive records.

Together, SecureSphere Database Security solutions combined with the SecureSphere Web Application Firewall, provide end-to-end protection for sensitive data in the datacenter.

A7—Missing Function Level Access Control

Threat Agents	Application Specific
Attack Vectors	Exploitability EASY
Security Weakness	Prevalence COMMON
	Detectability AVERAGE
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Most Web applications verify function level access rights before making that functionality visible in the user interface. 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.

Weak or missing access controls can allow attackers to access unauthorized functionality. Administrative functions are key targets for this type of attack.

SecureSphere Mitigation

SecureSphere applies several levels of control to prevent unauthorized access to application functions. At the application level, SecureSphere uses dynamic application profiling to learn which URLs and functions require a valid session identifier like a cookie. If a cookie is not present or if the end user has manipulated it, then SecureSphere can block the session, the user, or the IP address.

SecureSphere also prevents forceful browsing—when attackers attempt to access hidden or protected URLs by enumerating different file and directory names. SecureSphere analyzes the Web page referrer and ensures that files and functions are accessed in the correct order. By preventing forceful browsing, SecureSphere thwarts attempts to circumvent function-level access controls.

Custom policies can be created to restrict access to certain files by IP address or user name. SecureSphere's Dynamic Profiling capability, coupled with forceful browsing rules and granular custom policies, together enforce function-level access controls for sensitive application functions.

A8—Cross-Site Request Forgery (CSRF)

Threat Agents	Application Specific
Attack Vectors	Exploitability AVERAGE
Security Weakness	Prevalence COMMON
	Detectability EASY
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

A Cross-Site Request Forgery (CSRF) attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any potentially other authentication information, to a vulnerable Web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

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 Dynamic Profiling to identify and block CSRF attacks; hence, SecureSphere blocks all CSRF attacks that go through a Web server. SecureSphere inspects requests from external sources to the Web application and applies security controls based on default and custom CSRF policies.

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 from generating requests, thus preventing CSRF.

A9—Using Components with Known Vulnerabilities

Threat Agents	Application Specific
Attack Vectors	Exploitability AVERAGE
Security Weakness	Prevalence WIDESPREAD
	Detectability DIFFICULT
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.

Malicious users first identify weak components through scanning or manual analysis. Then they customize the exploit as needed and execute the attack.

Virtually every application has these issues because most development teams don't focus on ensuring their components/libraries are up to date. In many cases, developers don't even know all the components they are using, not to mention their versions.

SecureSphere Mitigation

The main difference between the OWASP Top 10 2013 and previous Top 10 lists is the addition of "Using Components with Known Vulnerabilities." This new Top 10 item not only reinforces the popularity of third-party components in application development, but also underscores the risks introduced by these components. Because thousands or even hundreds of thousands of Websites may include the same vulnerable code, attackers are highly motivated to locate and exploit vulnerabilities in application components. This may be one reason why many of today's high-profile breaches are caused by vulnerable third-party components.

The SecureSphere Web Application Firewall protects Web applications that use vulnerable components. Through defenses such as patented Dynamic Profiling technology, SQL injection and XSS correlation engines, and detection of HTTP protocol violations, SecureSphere identifies zero-day attempts to exploit vulnerable components. In addition, once a new vulnerability is published, the Imperva Application Defense Center (ADC) quickly develops a signature or a set of policies to virtually patch the vulnerability. Through automatic security updates, all SecureSphere appliances receive the latest security content and are protected against newly published vulnerabilities.

Many organizations do not know what third-party components are used in their Web applications or track vulnerability announcements for these components. As a result, applications built with vulnerable components are often exposed to attack for long periods of time. The SecureSphere Web Application Firewall, with its multiple layers of defense, is the perfect solution to protect applications with third-party components.

A10—Unvalidated Redirects and Forwards

Threat Agents	Application Specific
Attack Vectors	Exploitability AVERAGE
Security Weakness	Prevalence UNCOMMON
	Detectability EASY
Technical Impacts	Impact MODERATE
Business Impacts	Application/ Business Specific

OWASP Top 10 Definition

Web applications frequently redirect and forward users to other pages and Websites, and use untrusted data to determine the destination pages. Sometimes the target page is specified in an unvalidated parameter, allowing attackers to change the destination page.

Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages.

SecureSphere Mitigation

SecureSphere detects and stops unvalidated redirects and forwards. 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 engines can detect HTTP protocol violations along with unusual characters in application form fields, such as colons, forward slashes and periods that are often indicative of an attack. By correlating these different suspicious aspects of the same request, SecureSphere correctly identifies and blocks unvalidated redirects and forwards.

Furthermore, unvalidated redirects frequently operate as a XSS attack via a link in an email or forum. SecureSphere effectively protects against this attack by profiling requests from external Web applications and determining which external sources are legitimate and which are not. In addition, ThreatRadar Reputation Services provides an extra layer of protection against malicious sources, anonymous proxies, and phishing URLs. By integrating near real-time information about known attack sources into SecureSphere, ThreatRadar accurately blocks traffic from malicious sources and bots.

Summary

The OWASP Top 10 is an excellent benchmark for organizations to measure the security of their Web applications. The Top 10 not only raises awareness about security threats, but it also empowers developers and security professionals to prioritize and tackle these threats. Recommended by countless organizations and standards bodies, including the Payment Card Industry (PCI) Security Standards Council, the OWASP Top 10 is a great first step towards identifying and mitigating application risks.

The SecureSphere Web Application Firewall, the most trusted and most widely deployed Web application firewall in the world, can protect Websites from a myriad of threats including the OWASP Top 10. Besides stopping exploits like SQL injection and XSS, SecureSphere also blocks attacks like Distributed Denial of Service (DDoS), site scraping, and online fraud—attacks that cannot easily be mitigated through application code fixes alone.

Because of its ironclad security defenses, its scalable centralized management, and its flexible, transparent deployment options, Imperva SecureSphere is the ideal choice for real-time Web application protection.

About Imperva

Imperva, pioneering the third pillar of enterprise security, fills the gaps in endpoint and network security by directly protecting high-value applications and data assets in physical and virtual data centers. With an integrated security platform built specifically for modern threats, Imperva data center security provides the visibility and control needed to neutralize attack, theft, and fraud from inside and outside the organization, mitigate risk, and streamline compliance.

IMPERVA®

www.imperva.com