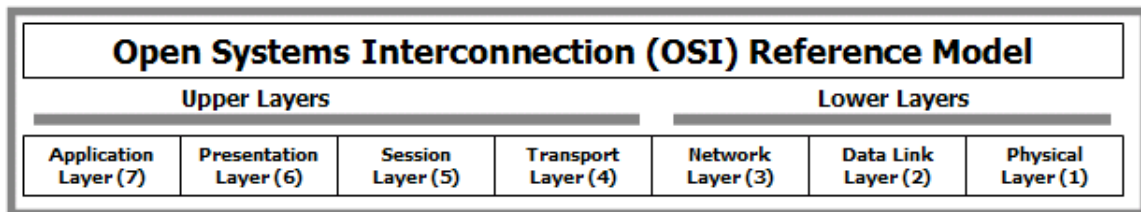


# Web Application Vulnerabilities

Where Attacks Occur and How AppScan Can Protect You

Web applications and web application users are exchanging data on the Internet every minute of the day. Sometimes this exchange is done through human or machine data collection for personal research or for site content indexing. More often than not, however, the data exchange is a result of businesses moving their customer interactions to the web to leverage economies of scale and meet the ever-increasing demand to connect customers to data.

From a technical perspective, web transactions and data exchange are commonly described in terms of the Open Systems Interconnection (OSI) standard reference model that illustrates communication between two end users in a network. The OSI model has seven layers, as shown below, and it's the application layer where over 75 percent of successful attacks exploit application data vulnerabilities, according to data from Gartner Group., a leading research firm for the global IT industry.



While some web applications are simple and limited, requiring and providing minimal services to the user, others are more complex and allow the user to personalize them for a specific need. Web applications that are more sophisticated, such as banking applications, medical record databases and systems that access third-party information, are those most inviting for illegal tampering. When asked why he robbed banks in the 1900's, Willie Sutton simply replied, "Because that's where the money is." The same is true for your web applications.

Knowing where your vulnerabilities lie, and resolving them, is not a new requirement, but one that continues to broaden as technologies evolve and applications are enhanced. Watchfire® AppScan™ knows how to protect you from the "old tricks" and is constantly evolving to meet an ever-evolving threat. In its fifth generation, AppScan knows these threats well and can alert you to all of these vulnerabilities.

<b>Application Threat</b>	<b>Negative Impact</b>	<b>Example Business Impact</b>
Buffer overflow	Denial of Service (DoS)	Site Unavailable to Customers
Cookie poisoning	Session Hijacking	Cash out someone else's account
Hidden fields	Alter Site, Illegal Transactions	Change hidden fields ie. Account Balance
Debug options	Admin Access	Access to all accounts and information
Cross Site scripting	Run Malicious Code	Identity Theft
Stealth Commanding	Access O/S and Application	Get list of customer accounts
Parameter Tampering/ SQL Injection	Fraud, Data Theft, Data Access	Alter distributions and transfer accounts, direct database access
HTTP Response Splitting	Web Cache Poisoning	e-Graffiti, Identity Theft
Forceful Browsing	Unauthorized Site Access	IP and Confidential data exposure, application misuse and abuse
3rd Party Misconfiguration	Admin Access	Create new unauthorized database of customers
Published Vulnerabilities	Admin Access, DoS	Create new unauthorized account

Older generation testing tools cannot protect you from the newer generation of these vulnerabilities. Without properly testing a web application for ALL possible vulnerabilities in these staid categories, the security test itself is incomplete, thus the results are inaccurate. Here are a few complex examples of basic vulnerabilities where you can count on AppScan for complete and accurate testing results.

### **Complex Cookie Poisoning**

Most reputable scanning tools will clearly state that they handle cookie poisoning attacks. In reality, however, such attacks are increasingly rare. AppScan takes generic cookie poisoning attacks to a higher level and performs several key attacks on cookies such as SQL Injection in cookies, buffer overflows in cookies, numeric cookie prediction attacks and more. This is where tampering activity is found -- not just in basic manipulation of cookies passed from server to browser.

### **Complex Parameter Tampering**

Most scanning tools will send a bulk of pre-defined tests for each parameter to test for parameter tampering, but they stop short of determining the original value and won't use it in order to modify the tests accordingly. When performing eShoplifting tests (manipulating e-shopping cart price values in hidden parameters), AppScan will keep the original currency format attached to new values (i.e. \$ [price] USD) when sending new values. This ability is the most relevant, yet exists only in the AppScan product line – in fact, it's one of AppScan's core features -- and complements its ability to increment or decrement numeric values out of observed ranges.

### **SQL Injection / Port Listener Attacks**

SQL Injection attacks allow a hacker to modify/manipulate the original SQL query that is executed by the web application, and change it to perform unauthorized actions on the back-end SQL server. In some cases, the output of the SQL Injection attack will not appear in the returning web page. AppScan is the only scanner to include a port listener mechanism, which accepts connections from the attacked database server (out-of-bound requests), allowing AppScan to validate the existence of SQL Injection in 100 percent of such cases.

### **Try AppScan Now**

To be sure your web applications are secure; we encourage you to scan them with an IP-specific trial license, with a Watchfire engineer at your side. Let us help you make sure you know how to digest and filter AppScan's test results, and help you understand what vulnerabilities are the most critical for your organization.

If this is a process that you would like to walk through with us, or if you have any questions at all, please feel free to contact Watchfire directly at [info@watchfire.com](mailto:info@watchfire.com). All of the site-specific tests are offered at no expense or obligation to you or your company.