

Spirent Avalanche™

Application Testing and Enhanced HTTP

Layer 4-7 Testing

Applications

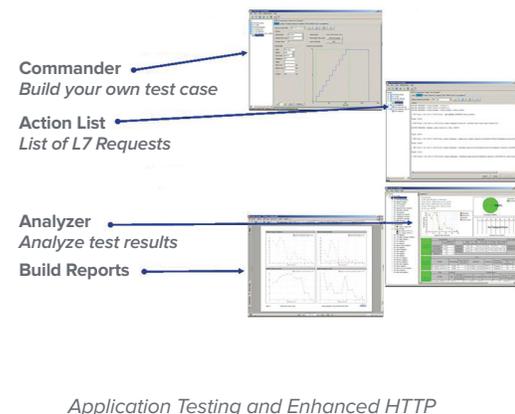
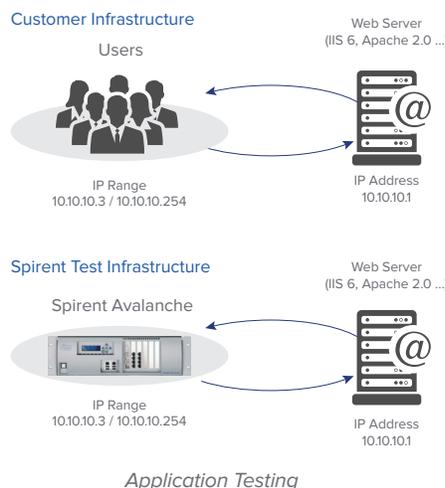
- Create real-world dynamic user behavior with advanced browser-to-application-interaction capabilities
 - Testing Web applications for scalability and response-time behavior under load
 - The Web applications handle increased load without dropping users
 - Test cloud-based web applications to ensure performance and resiliency of distributed applications and IaaS/PaaS/SaaS offerings
 - The Web applications continue to handle users without degrading the response time under load
 - Testing Web applications for high performance, user realism, and network realism
 - The Application Testing option provides support for testing of forms databases, dynamic variables (dynamic URLs, session IDs, etc.), and Web services
 - Test virtualized cloud applications with Avalanche Virtual testing software
- The Enhanced HTTP option for Avalanche provides support for testing of cookies, redirects, POSTS and content validation.

The Application Testing and Enhanced HTTP bundle are two features of Spirent Avalanche that are targeted specifically at Cloud and Web application testing—whether the end goal is to test a Web application, a Web service, a cloud application or a network device.

Features and benefits

Features of application testing

- **Forms databases:** This feature is very useful when requests are dynamically constructed based on external data utilizing a database of information. For example, a retailer may have a lot of inventory that can potentially be part of the test. The inventory list is maintained in a forms database and is picked up by the action list (which is parameterized). The forms databases in Spirent Avalanche are very flexible and allow for both global and user-level control.
- **Dynamic variables:** This feature allows users to test the web applications that are involved with the dynamic requests and contents. Dynamic variables are analogous to variables in a programming language.
- **Web services:** Spirent Avalanche supports testing with real HTTP content. This is especially useful for testing Web services (SOAP over HTTP) where the SOAP message is stored in an XML file. The data in this file can be parameterized as well.



Spirent services

Spirent Global Services provides a variety of professional services, support services and education services—all focused on helping customers meet their complex testing and service assurance requirements. For more information, visit the Global Services website at www.spirent.com or contact your Spirent sales representative.

spirent.com

AMERICAS 1-800-SPIRENT
+1-818-676-2683 | sales@spirent.com

EUROPE AND THE MIDDLE EAST
+44 (0) 1293 767979 | emeainfo@spirent.com

ASIA AND THE PACIFIC
+86-10-8518-2539 | salesasia@spirent.com

Features of application testing (continued)

- **User-based load profiles:** Large Websites have customers from different locations accessing the site independently but simultaneously. For example, a large retailer may have a set of users on the East Coast over DSL lines and another set of users from the West Coast using faster cable modems. Each of these users has separate characteristics and can be modeled separately using user-based load profiles. In user-based load profiles, the user characteristics and the loads of individual sets of users can be modeled separately. Spirent Avalanche aggregates the user descriptions before execution. This feature gives a tester a great deal of flexibility in modeling a test with as much realism as possible to the real customer base.
- **Nested directories:** This is a server-side feature of Spirent Avalanche. It supports a lot of protocols—including HTTP and HTTPS. On an HTTP (or HTTPS) server on Avalanche, a user can upload real static web content. With nested directories, Spirent Avalanche simulates a real web server.

Features of Enhanced HTTP

- **POSTs, cookies, and redirects:** At a minimum, modern Web applications use HTTP POST requests, cookies and redirects. Spirent Avalanche completely supports cookies, redirects—both in the response header and in the response body as a META-REFRESH tag—as well as HTTP POST statements.
- **Timers:** To adequately gauge the performance of a site, it is ideal to separate out and individually measure different parts of the Website. For example, a retailer may need to determine the time spent by users to browse the inventory as well as the performance of the checkout section of the website. To do this, the workflow is set up to group and separately measure the two sets of actions. This grouping of actions is done in Spirent Avalanche with Timers.
- **Loops:** A set of actions may need to be repeated several times. For example, if a workflow models the purchase of three books on a retailer's Website, then action can be set up once and repeated three times in a loop. Avalanche can execute loops either by count or for a specified amount of time.
- **Content validation:** In certain workflow situations, it is necessary to search for a string as part of the return content and take an action based on whether or not the search was successful. For example, if the retail Website has a login page, then the validity of the test is dependent on correct credentials being presented to the login page before the rest of the steps in the workflow can be executed. If correct credentials are not presented, then the workflow should not continue for that user. With content validation in Avalanche, the test can be set up to look for a pattern in the response (for example, "Sign-In Error") to determine whether the credentials presented have failed. At this point, the test or the current user can be stopped.
- **Search criteria:** This feature is useful when the pattern is known, but not the exact string or the position of the string in the response. With search criteria, when the prefix and the postfix are known, it is possible to grab a string from the response header or response body and reuse it in a subsequent request. This is useful when a Web application uses VIEWSTATE IDs or session IDs for state management. If the session ID is dynamic and has to be used across requests, then the session ID needs to be captured and reused. Another use case is to grab the URL from a response page and use that as the subsequent request. For example, this can be used when the first link from the response from a search engine should be set as the next request in the workflow.