Directed Session

(Mini-Pattern)

Abstract

The *Directed Session* pattern ensures that users will not be able to skip around within a series of Web pages. The system will not expose multiple URLs but instead will maintain the current page on the server. By guaranteeing the order in which pages are visited, the developer can have confidence that users will not undermine or circumvent security checkpoints.

Problem

Web applications often have to collect a great deal of data from a user in order to complete a single transaction. E-commerce purchases, for example, require that a user select items to buy, provide contact information and a shipping address, select shipping options, and provide credit card information. Rather than simply present the user with a huge form with a bewildering array of options, most sites prefer to guide the user through the process, validating each piece of data as it is provided.

This approach can be vulnerable to attacks. An attacker can use known URLs to jump between the different pages, in attempt to bypass some of the data validation checks. For example, if the application developer is not extremely careful, it may be possible for the attacker to add items to the order after having paid.

Solution

The *Directed Session* pattern exposes a single URL to the end user. All pages on the server are accessed using that URL. Session data stored on the server is used to determine which page is served. When no session data is present, the user is given the initial home page. As the user navigates the system, the current selected page is maintained in the session data.

This approach ensures that users are not able to request specific URLs and thereby bypass data validation checks. When a transaction consists of several separate pages, the user cannot request the second page until the first page has been validated. The application designer has some lenience here. If the user goes back to cached pages and resubmits an earlier page, it may be acceptable to accept that page and notes that all subsequent pages must be resubmitted. Alternately, the application designer can enforce a strict sequence and require that the user navigate using “forward” and “back” commands on the page itself.

Note that many IIS .ASP pages use an approach like this to regulate user transactions. While not intended for security, it can have positive impact on security. It also has some shortcomings from a usability perspective, particularly where the back button on the browser is concerned.

Related Patterns

- *Authenticated Session* – a related pattern that can use this directed session mechanism to enforce a particular session interaction.

- *Client Input Filters* – a related pattern that can use this directed session mechanism to enforce that client input is accepted and validated in a particular order.

- *Validated Transaction* – a complementary pattern for ensuring that client input validation is performed on all user input.

References

None.