**Open Redirect Development Mitigation SOP**

Open redirect vulnerabilities occur when unvalidated input is allowed to control the URL used in a redirect, which can aid phishing attacks. Redirects allow web applications to direct users to different pages within the same application or to external sites. Redirects can trick users into going to a malicious site. Encoding the URL can allow attacks to trick users into believing the site they have gone to is not malicious. These attacks are often used as part of phishing scams to harvest sensitive end-user data.

**Classic Example**

<%

…

String strDest = request.getParameter(“dest”);

pageContext.forward(strDest);

…

%>

In the example above, the JSP code above instructs the user’s browser to open a URL parsed from the *dest* request parameter when a user clicks the link. If a victim received an email to follow the link, “http://trusted.example.com/ecommerce/redirect.asp?dest=www.wileyhacker.com”, they could be redirected to “http://www.wileyhacker.com” without knowing the site is untrusted.

**Example**

//needs to be added

**Defense Against *Specific Vulnerability***

Do not allow unvalidated user input to control the destination URL in a redirect. Alternatively, a level of indirection could be used instead by creating a list of URLs that users are allowed to specify and only allow users to select from that list.

**Defense Example *(fixing Classic example above)***

The code below has the user select a URL from a valid list instead:

<%

…

try {

int strDest = Integer.parseInt(request.getParameter(“dest”));

if((strDest >= 0) && strDest <= strURLArray.length -1)){

strFinalURL = strURLArray[strDest];

pageContext.forward(strFinalURL);

}

} catch(NumberFormatException nfe) {

//handle exception

}

%>

**Specific Explanation**

*Explanation of why and how this works within goes here*/Lorem ipsum dolor sit amet, mel et esse audiam. Te primis comprehensam sit. Vix ex decore debitis complectitur, cu mutat melius sed. Altera erroribus est in, ei eos deserunt partiendo posidonium. Eos tempor albucius ex. Id sea sale commodo molestiae, nonumy pertinacia ad quo. Lorem ipsum dolor sit amet, mel et esse audiam. Te primis comprehensam sit. Vix ex decore debitis complectitur, cu mutat melius sed. Altera erroribus est in, ei eos deserunt partiendo posidonium. Eos tempor albucius ex. Id sea sale commodo molestiae, nonumy pertinacia ad quo.

**Recommendation *(highlight important changes in Bold, fixing example above)***

//needs to be added

**References**

1. [HP Enterprise Security – Open Redirect](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/javascript/open_redirect.html)