Application

Charting Dependencies and Drawing Conclusions

--------------------------------------------------------------------------------

Software defects almost never exist in a vacuum. That is, the problem with ensuring application security is that there are usually additional defects or issues that have to be identified, understood, and resolved along with the actual defect in the code to which they are attached. These additional defects are considered dependencies.

In this assignment, you will examine dependencies associated with mitigating CSRF holes in websites and develop data-supported conclusions about the complexity of ensuring against CSRF holes.

Go to Bugzilla and enter the term "CSRF." Scroll to the bottom of the results page. There will be a way to display dependencies based on a graph and a tree.

Then, using the graph, the tree, and any explicit empirical measure that you feel is appropriate to characterize that defect, prepare a 2- to 3-page paper that presents the following:

•An estimate of how difficult this problem will be to resolve. Base that estimate on a quantitative measure that will support your assessment of the defect and the dependencies involved.

•An assessment of the relative complexity of the process needed to solve CSRF problems using quantitative measures and what you know about those dependencies.

•Your conclusions about how to resolve the problem based on the same quantitative measures.