Brandon Young

CSCI – C437

Project

Summary:

The exploit that I will be reviewing is a vulnerability in the app mutillidae on the higher security sql injection pages.

Issue:

This issue resides in the fact that inputs can be manipulated after they have left the web app, but before they have hit the server side. In order to avoid sql injection developers often implement character filtering that does not allow users to input dangerous characters. In this application at security level 1, this app filters inputs from the user. However, hackers can avoid this entirely by simply editing the requests after they have left the webapp and before they hit the server. This could be done by anyone with the right tools.

Impact:

This vulnerability allows users to sign into accounts without authenticating and can also lead do stealing credit card numbers along with lots of other important information. The risk here is very high and there would be many people ready to take advantage of this if it had real data in it. This leads to a significant threat to the application.

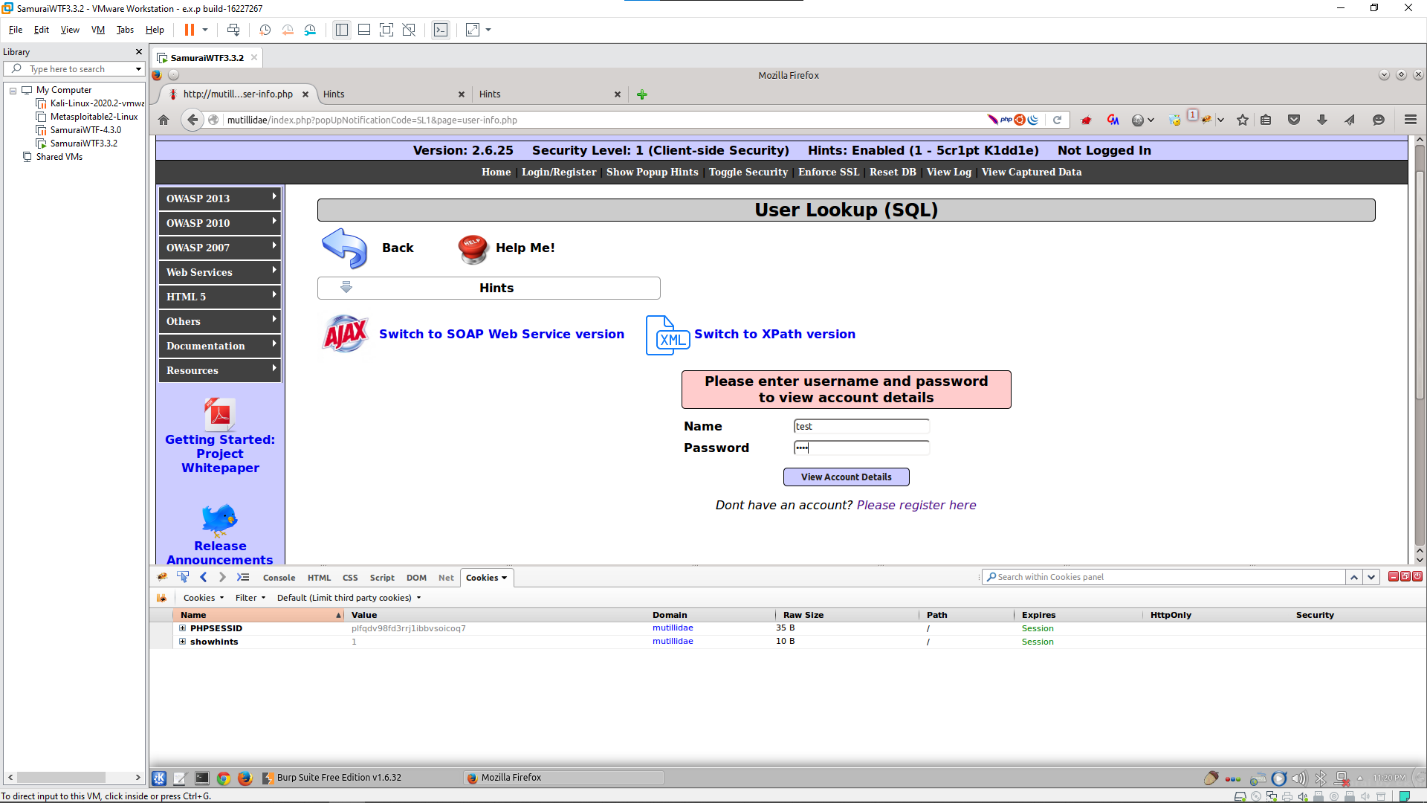
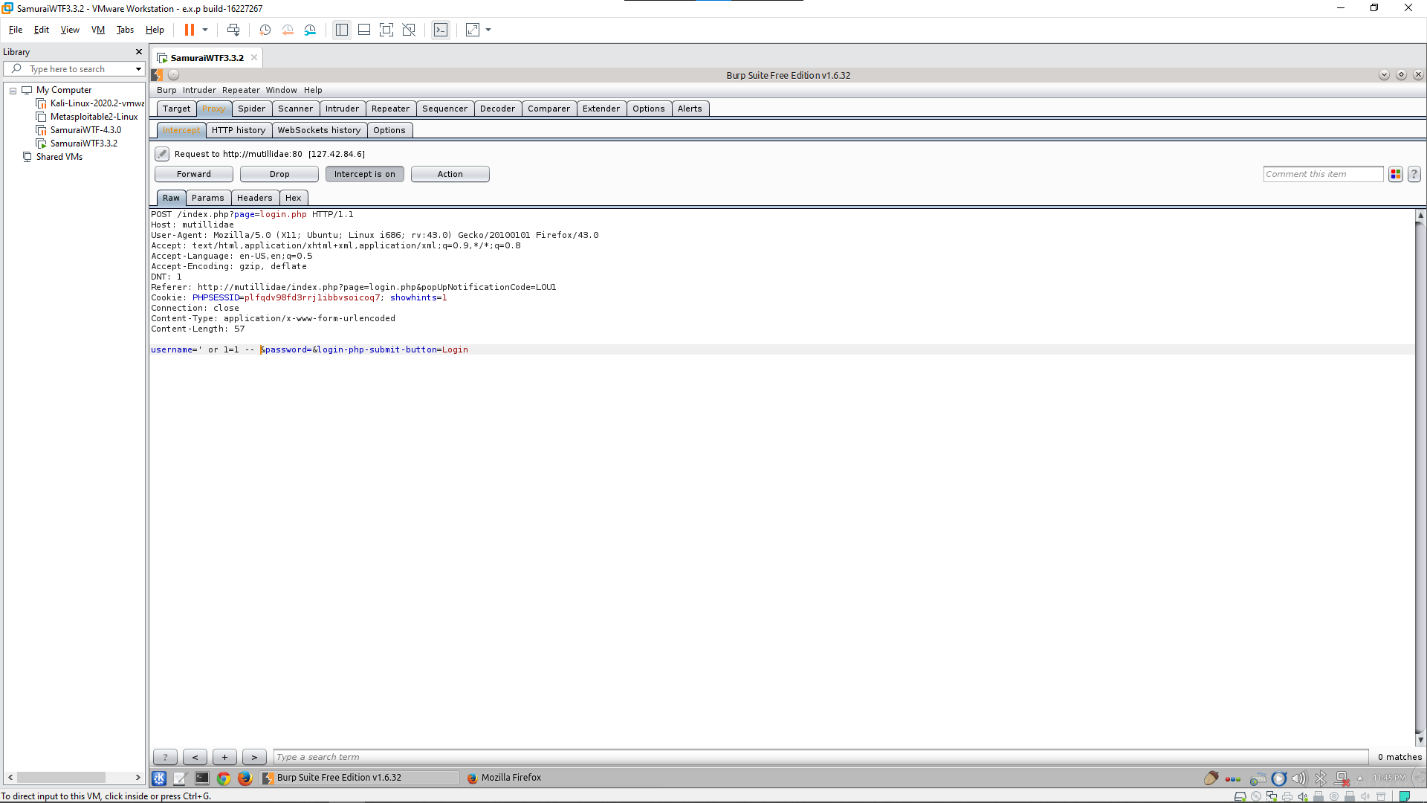
Reproduction:

Install mutillidae and burpsuite, these are the only tools required, and there may be other ways of doing it that don’t require burpsuite, but I am not aware of any. You then must navigate to the sql injection page “Login.” You then want to set your browser to use a proxy server. Set this proxy to the default burpsuite IP address and port. Then open burpsuite and set the proxy to intercept. Then return to mutillidae and submit a request using standard alpha-numeric values. It does not matter what the values are, just simply that a request is filled out correctly and sent to the proxy server. Then return to your proxy and look at the request you sent. You can then modify this request in burpsuite to do a SQL injection. For example, submit the value ‘ or 1=1 -- . Once you do this you will be logged in as admin.

Remediation:

The remediation for this is not to change the filtering, you simply need to move the filtering to the server the app runs off. When you do this, you remove all possibilities of the user editing requests and bypassing the filter. This would need to be implemented for all user input across all pages, as none can be treated as trusted.

Screenshots:

1, enter an alpha-numeric requst2, intercept and modiffy3, authentication bypassed