Project 4 Write Up

Noam Lerner
nlerner3
903047143

Target 1 Epilogue

Target 1 was vulnerable because it did not remember which challenge it had sent to the user, and because the response was based only on the new values and not the old.
If the values were based on the old values, I would not have been able to hard code the response in.
If the server remembered which challenge it had sent to the client, I could not hard code a challenge I had already received and to which I knew the answer to (I would have to do it dynamically).
To fix, this, remember which challenge was sent to the client in the session (don't use the challenge sent from the post request) and base it on values being changed, not just the values they are being changed to.

Target 2 Epilogue

Target 2 was vulnerable because it directly took user input and rendered it back onto the page (in this case, an incorrect username). This allowed me to craft a username that would inject a script and change the page in almost any way I would want to. The way to get around this vulnerability would be to not render user input on the page, or to properly sanitize it before doing so.

Target 3 Epilogue

All though auth.php sanitized some of the data, the way it did so was flawed. I was able to exploit the fact that that function first removed "/*" and then removed spaces. The string "/ *" would not match "/*" until after the space was removed, which was later in the function. The function also did not check for any single quotes which was crucial in injecting code.
The fix for target 3 would be to remove any single quotes in the sanitation process, and to remove strings that are lower in length before removing strings that are higher in length.