Writeup

Yen-Yun Wu / Baijun Chen

SQL Injection Recommendation

I recommend you use one of these safe tools instead of constructing SQL queries by yourself, no matter how professional and knowledgeable you are: - Parameterized (AKA Prepared) SQL: Parameterized SQL allows you to pass in query separately from arguments and let your server handle escaping data automatically. - ORMs (Object Relational Mappers): Object Relational Mappers (ORM) provide an interface between native objects and relational databases.

XSS Scripting Recommendation

I recommend to leverage a "whitelist" approach instead of a "blacklist" approach to specify what kind of inputs your system would accept. While filtering is a possible defense, it is difficult to generate a perfect filter. Hence, I recommend to utilize existing filters proved to be powerful at a certain level unless you are a genious expert who could create a much more effective filter. "Content Security Policy" is another method eliminating XSS attacks, which in this case could help browser to ignore inline scripts passed in as an user input.

CSRF Recommendation

I recommend you to examine Fetch-Metadata headers in every incoming HTTP requests, because such headers tell the server who they are talking to and how they got there. With that being said, it has a limitation of not being supported on old browsers. Another method is to let the server set the "SameSite" attribute on your website cookies. The "Lax" or "Strict" mode can prevent browser from sending session cookie along with a cross-site request, which could eliminate CSRF attacks.

Did you like the PA? Anything that could be improved? Give your suggestions here

Yes. Maybe you can provide some back-end code when we solve extra credit.