

Cross side scripting/ XSS: code injection problem at server side/ client side, hence vulnerable to security issues. When url is hit, it is retrieving username from url and showing a form. But instead of a url may have be injected by a script with code that executes a program on client side. Hence security issue as we can then send new request to server with user cookies or could steal user cookies or any data from the form, can install new programs, delete the app

Reflected Xss: Client sends request to server to retrieve data and data is sent back to user. If data is a script, it would be executed on the client request. Due to bug in server-side code.

Persistent XSS: we send the data which is stored on the server and then if another client makes a request to that data, code will be executed on their machine. Due to bug in server side code.

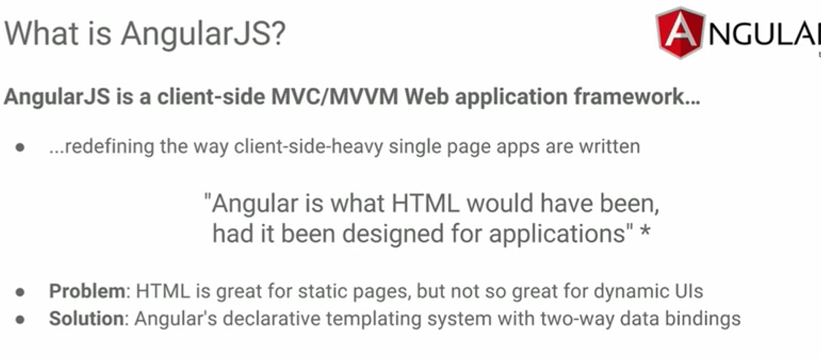
DOM based XSS: Due to bug in js code. Could be document write or inner html that writes user input to DOM and executes js

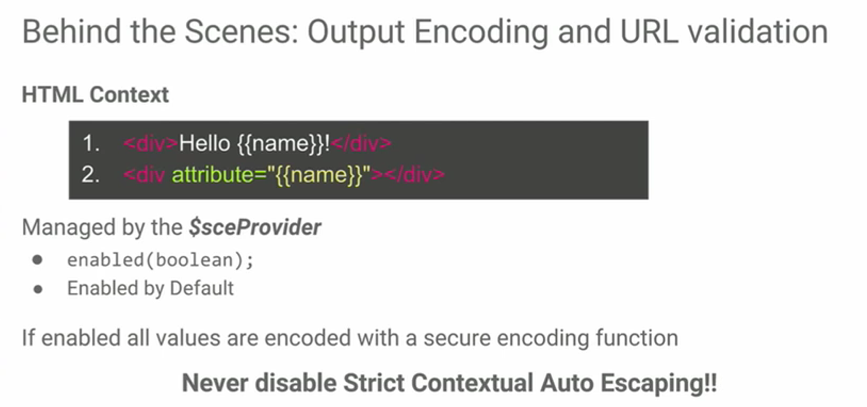
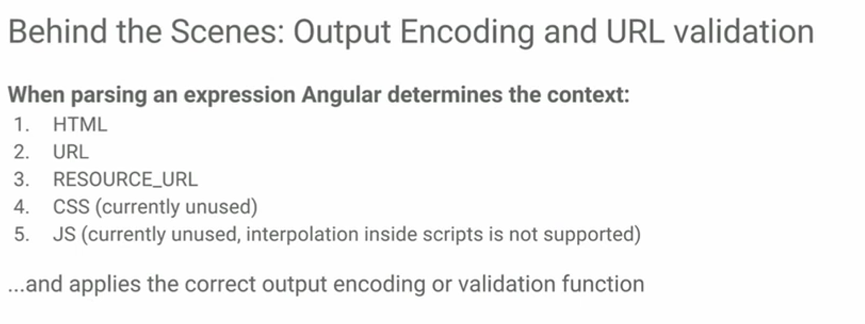
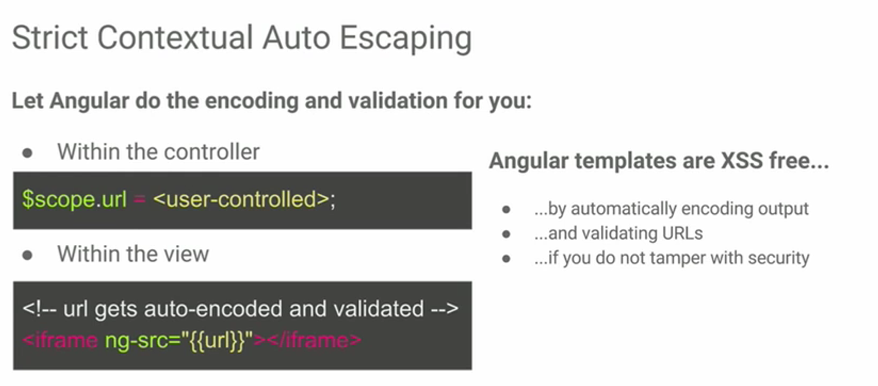
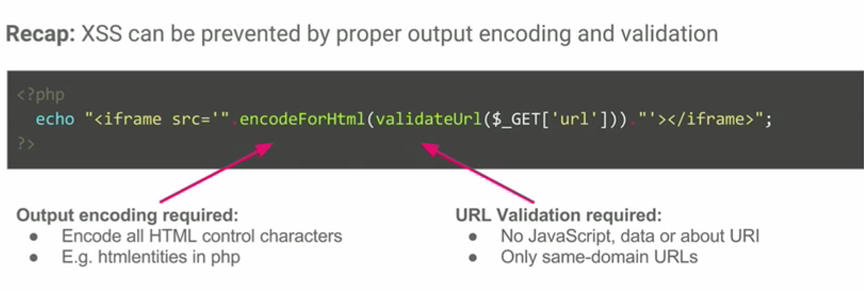
To prevent these, we may:

1. Encode user input.

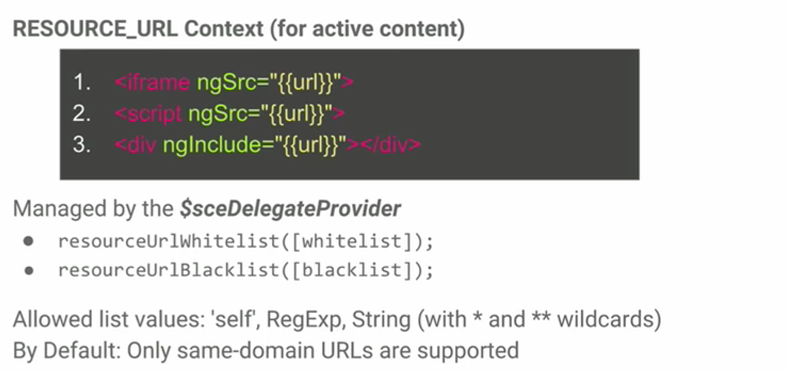
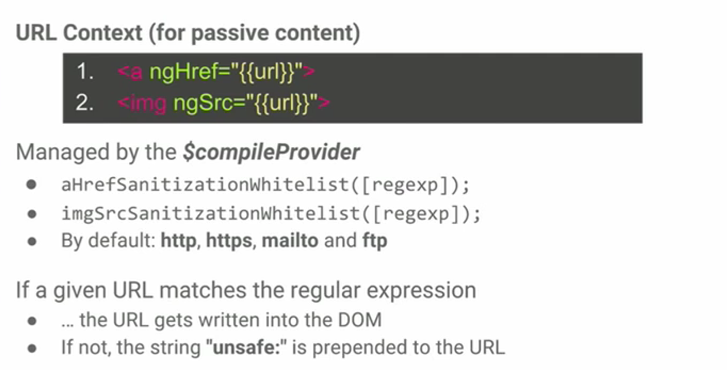


validateUrl to validate not a javascript code or any other protocol evil handler

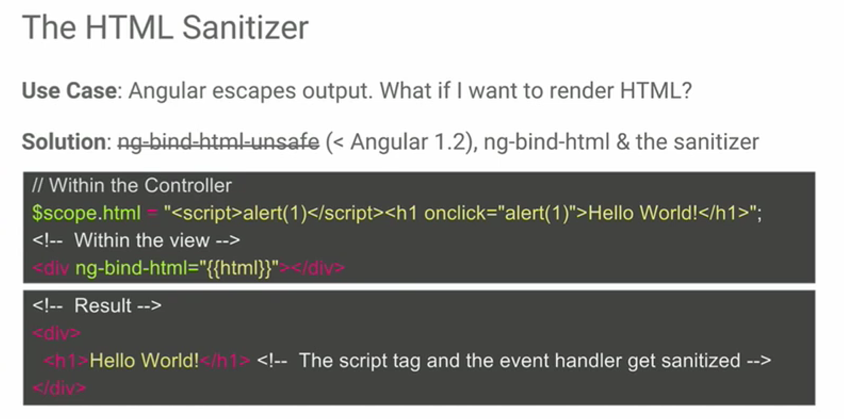




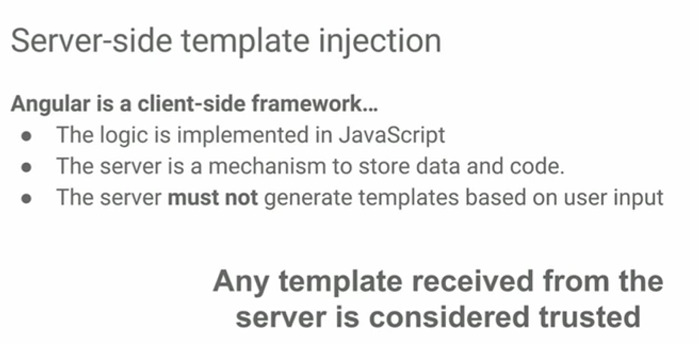
Passive content: links, images

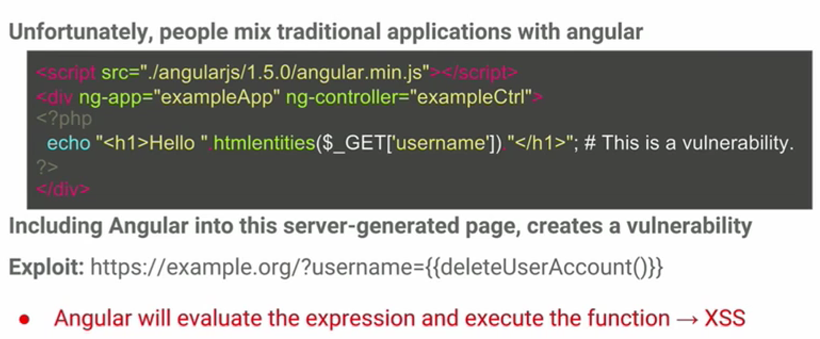


Self: same domain

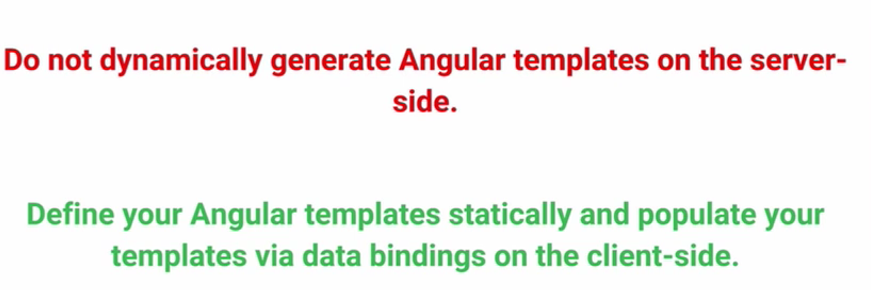


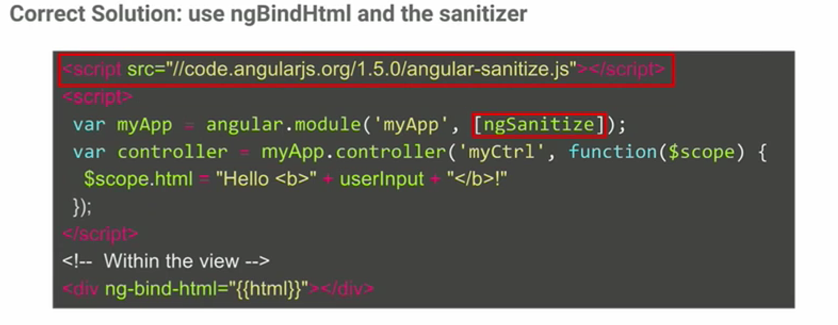
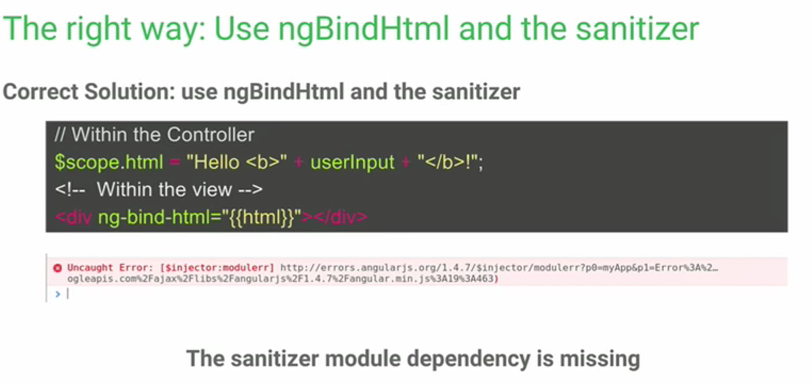
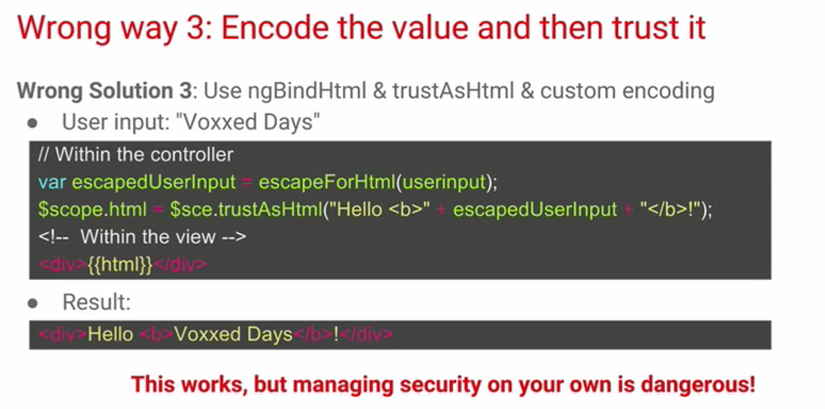
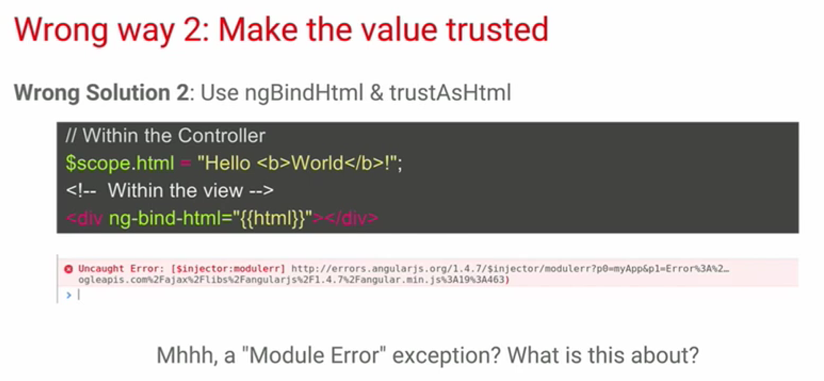
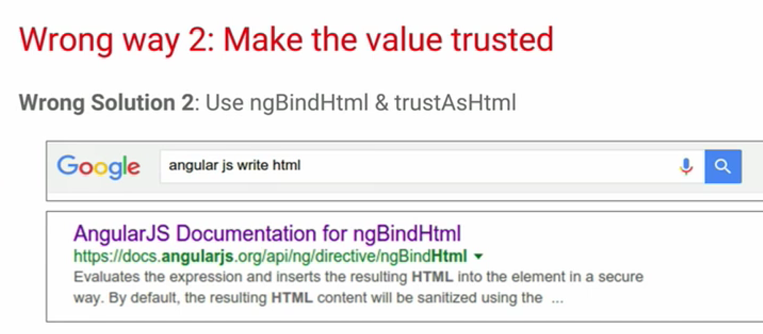
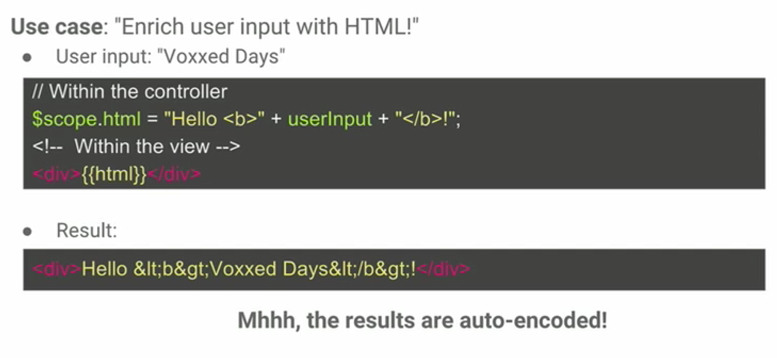
Security Bugs:

1. 



So templates should be static and not generated dynamically as the templates are thought to be trusted as coming from server



1. 
2. 