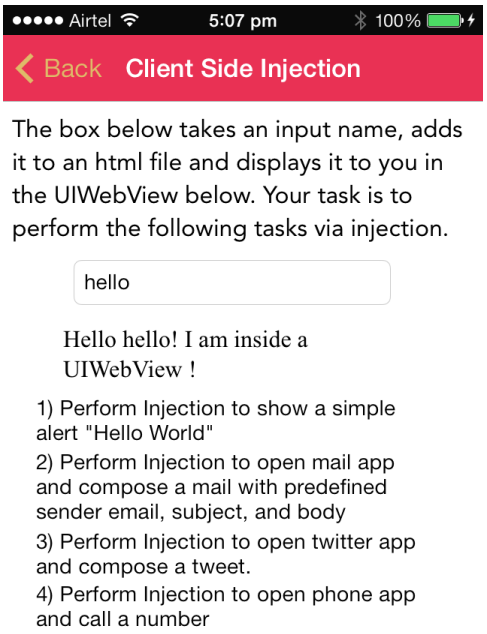
Damn Vulnerable IOS Application Solutions
http://damnvulnerableiosapp.com/

Client Side Injection

Let's enter some dummy name and tap on return. We get the text *Hello $text !* inside a UIWebView. From this, it is clear that some data is being added to the UIWebView at runtime and shown to the user. If the data is not validated properly then we can just add some javascript in the UIWebView and get it executed.

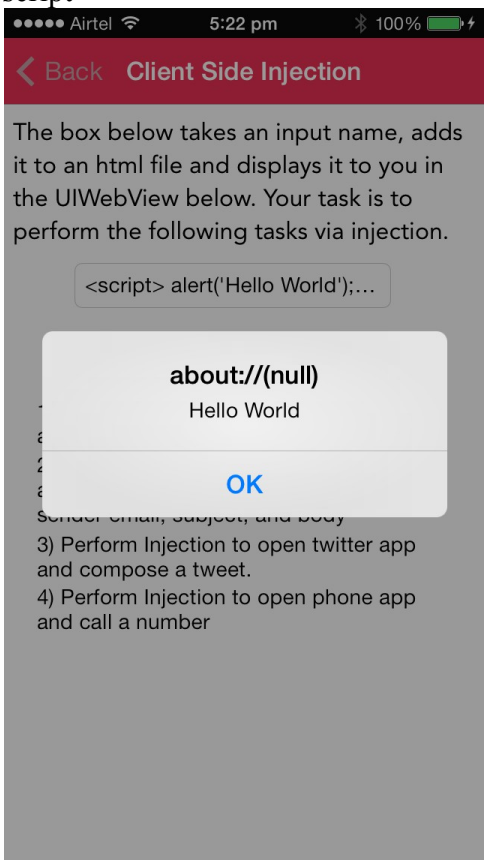


Now the important thing is how to write javascript code that can interact with the device ? The answer is URL schemes. URL schemes are used for communicating between applications. Every application can register for a particular url scheme. For e.g, the phone application on IOS registers for the url scheme *tel:// .*This means any url starting with tel:// will be redirected to the phone application and the phone application is supposed to handle any incoming url's starting with that url scheme. For e.g, the url *tel://1-123-456-789* when opened in a browser from an IOS device will redirect the user to the phone application and will try to initiate a call to the number *1-123-456-789*. Well coded applications will always try to perform some validation before performing some action with url schemes.

A comprehensive list of applications with their URL schemes can be found at http://wiki.akosma.com/IPhone_URL_Schemes#TikiSurf

With that being said, here are the inputs you should provide to perform the tasks from 1) to 4)

a) <script>alert('Hello World');</script>



Similarly, the solutions for b), c) and d) are...

b) <script>document.location='mailto://xyz@example.com?cc=prateek@damnvulnerableiosapp.com&subject=Greetings%20from%20DVIA!&body=I%20performed%20client%20injection%20successfully!'</script>

c) <script>document.location='twitter://post?message=Hello%20World'</script>. In this case you must make sure that the twitter application is installed on your device.

d)<script>document.location='tel://1123456789'</script>