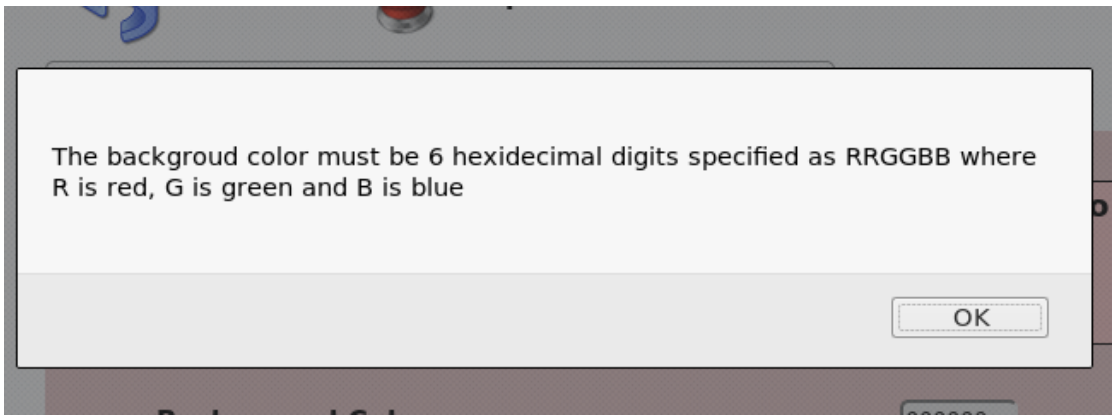
關於作業二的補充說明
Client-side Security

關於今日的作業二，我在此有一些說明。第一大題的 mutillidae II 的題目，主要是讓同學們了解對於 Client 防護對於攻擊者而言作用不大。

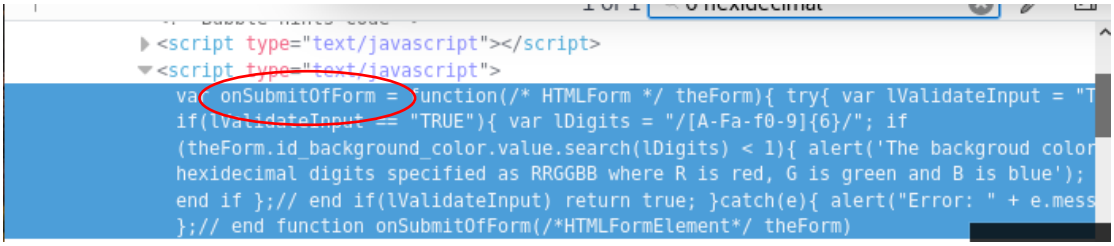
一般來說有兩種方式，第一種修改前端程式，如利用 Inspect Element 來修改對成前端輸入的檢查，包括輸入長度、字元內容等等。第二種方式是利用中間的 Local Proxy，如 Burp Suite 的方式，把輸入資料在送往 Server 前先擋住，可進行一些修改、分析，或進行大量的攻擊等。第二種方式可以做的事很多。若修改前端程式有某些困難或限制，或是想進行大規模測試，使用 Burp Suite 都是很好的選擇。

同學們若有測試第一大題的第三小題。若想利用 Burp Suite 的話，好像在一開始輸入顏色代碼會有困難，因為無論怎麼輸入都會被前端程式擋住，因此無法送給 Burp Suite 來處理。這應該是該程式設定不讓你通過檢查。若你仍想使用 Burp Suite。由於該前端程式使用 Javascript 程式撰寫檢查的函式，因此，方法大概就是，先關掉瀏覽器的 Javascript 功能，然後便能順利送出資料，然後使用 Burp Suite 擋住後，再回到瀏覽器開啟 Javascript 後，再將 Burp Suite 的資料放行送給 Server。便可以看到結果。

但其實要打 Client 的防禦，如果你對於前端程式了解的話，改前端程式就可以。不一定要用 Burp Suite。同學學習利用 Burp Suite 其實可以學到一些 Local Proxy 攻擊的概念。以下我說明一下第一大題第三小題，如何使用前端程式修改如何進行。類似長度限制的修改。首先你看到輸入之後的檢查，如下面畫面。



使用 Inspect Element 查詢一下“6 hexidecimal”這幾個字，找到他的防護是一段 Javascript，如下：



看到 onSubmitOfForm 因此找一下使用它的位置，在 form 的設定上。

然後將 onsubmit=”return onSubmitOfForm(this);” 刪除之。便去除了檢查。之後無論輸入什麼都可以了。包括 Script 的命令也可以。同學們可以試試。

其他許多前端的保護攻擊方式，可能也可使用類似方法。如前面的 SQL Injection 題目，也可以的。