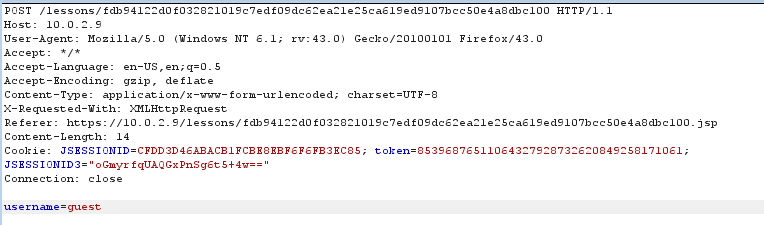
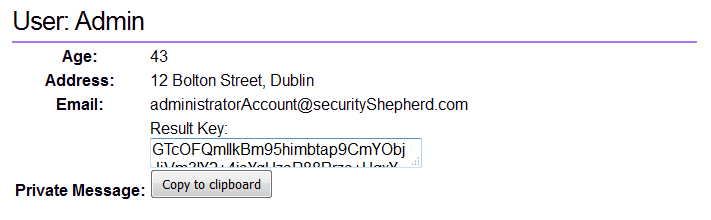
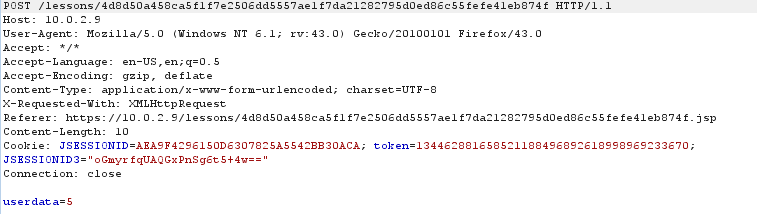
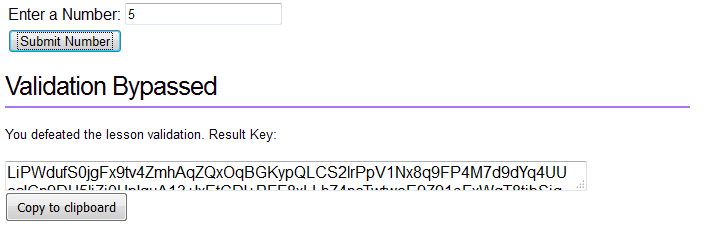
**Field Training**

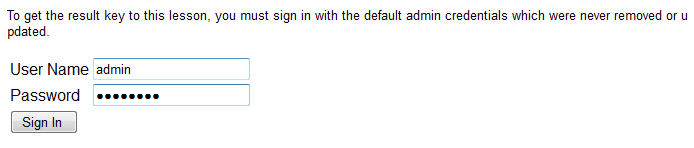
1. **Insecure Direct Object References**
   * Intercept the request
   * Replace the word “guest” with “admin”

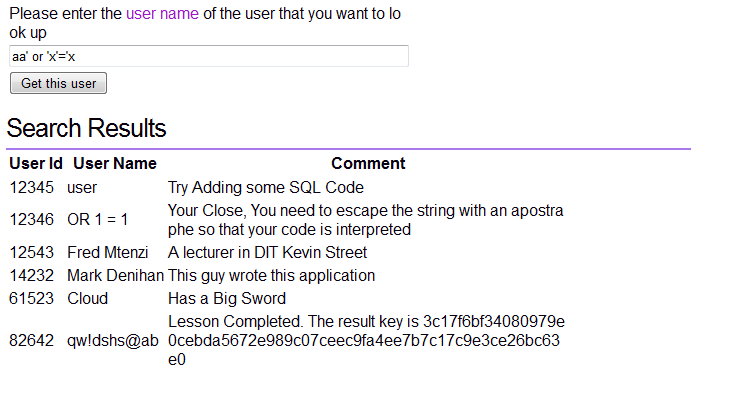
****

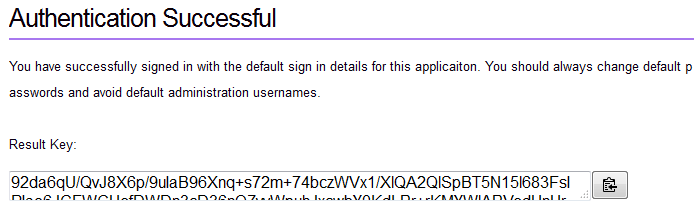
1. **Poor Data Validation**

* Intercept the request
* ****Change the userData to a negative value

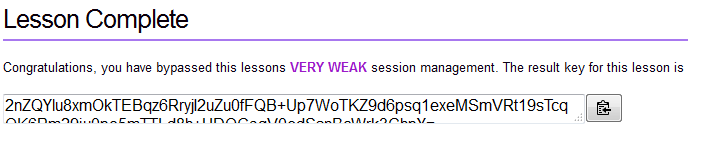
****

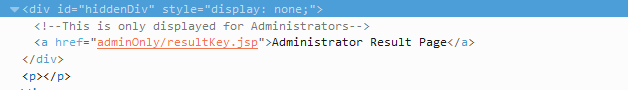
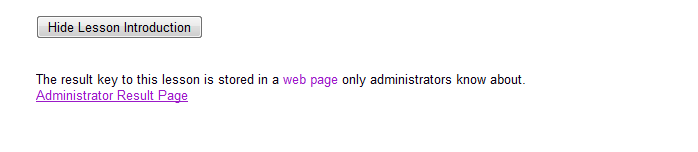
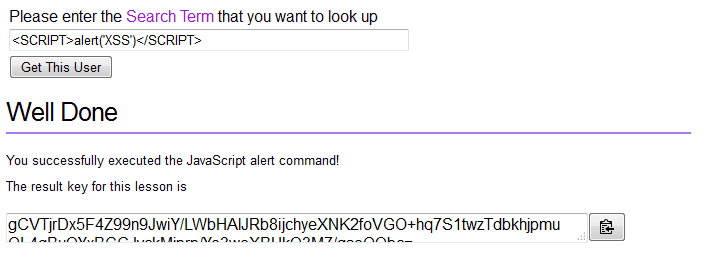
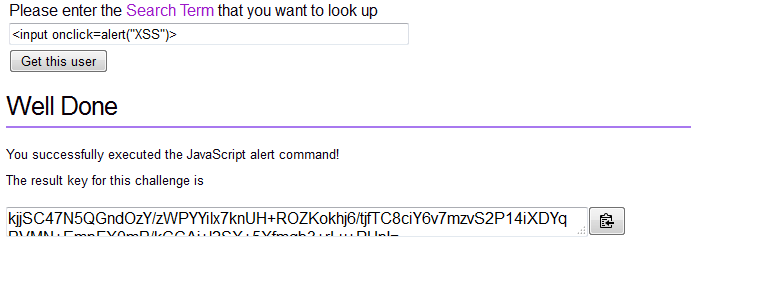
1. **Security Misconfiguration**

* Use default Username and Password (“Admin”, ”Password”).

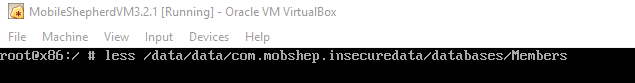
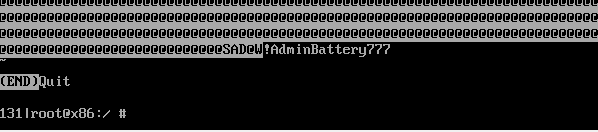
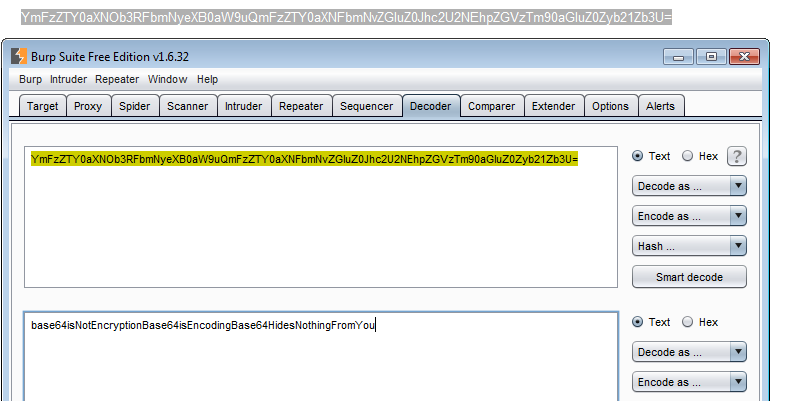
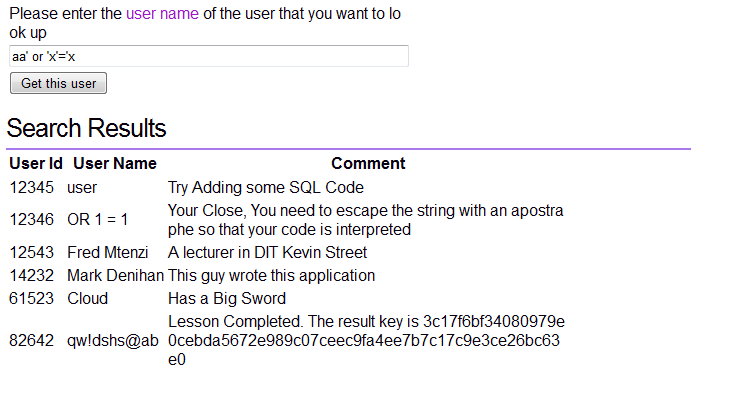
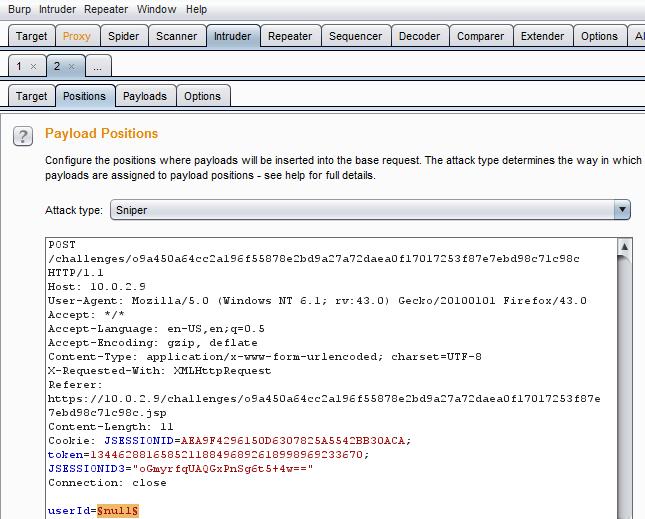
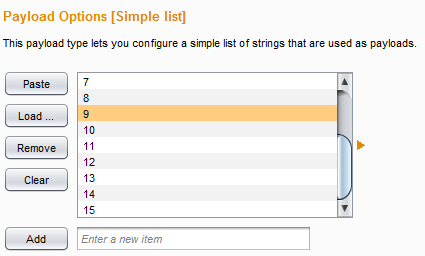
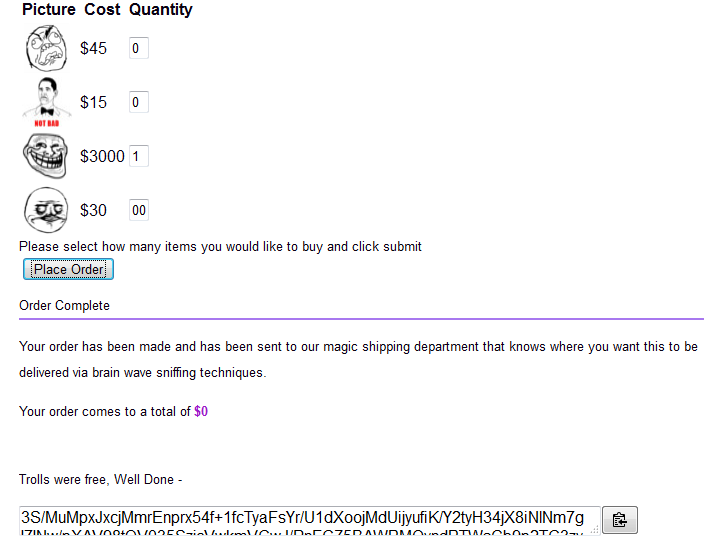
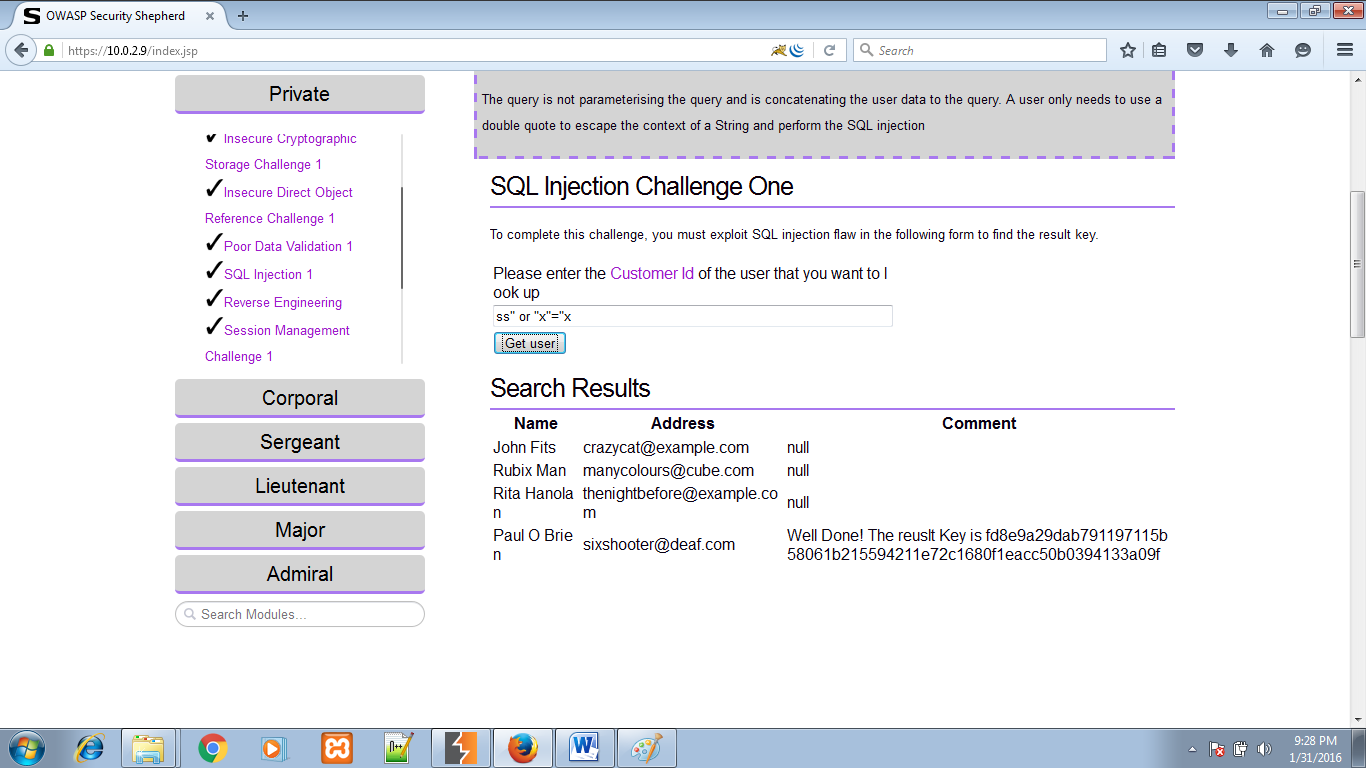
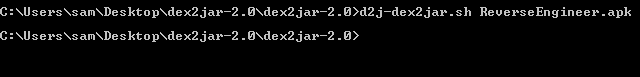


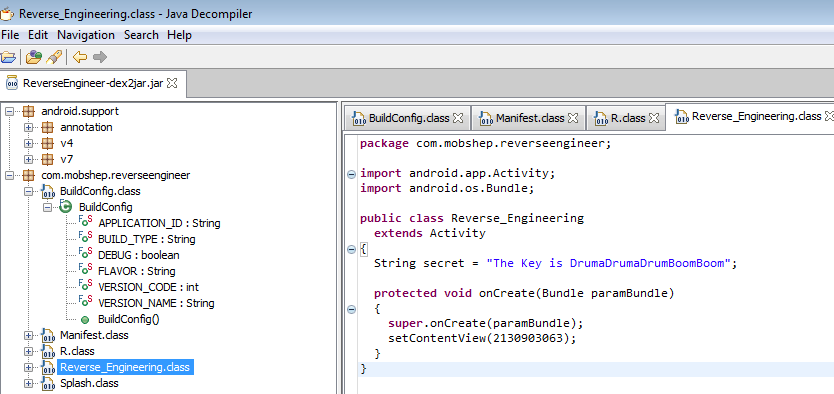
1. **Broken Session Management**

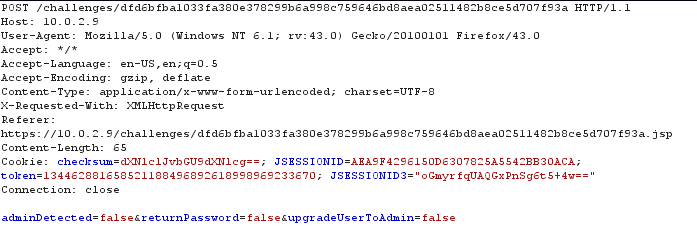
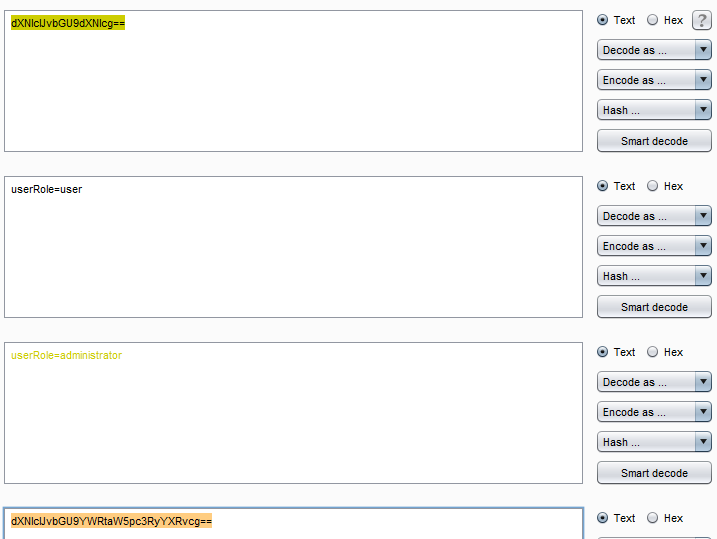
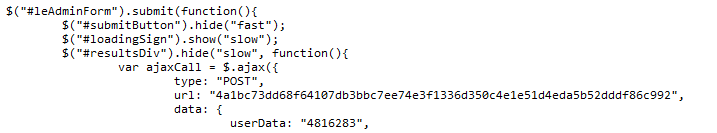
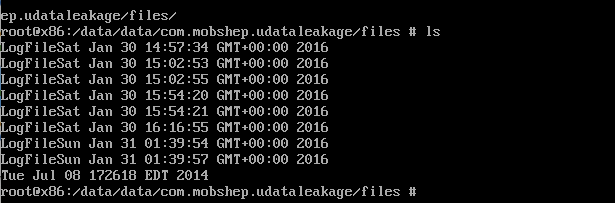
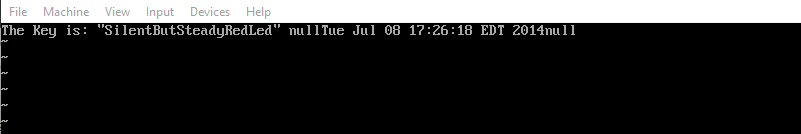
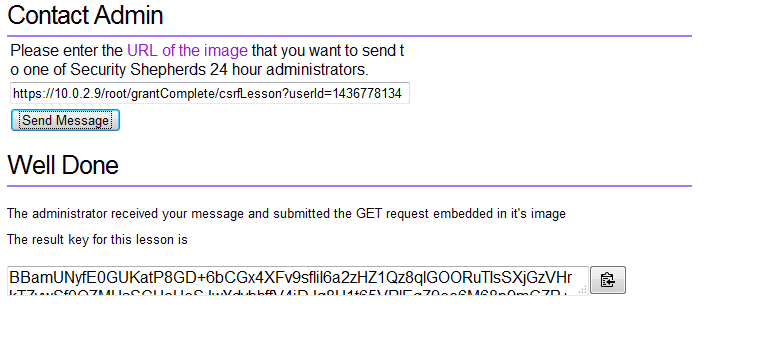
* Intercept the request
* Change the cookie value to “lessonComplete”

1. **Failure to restrict URL Access**
   * Inspect the HTML elements.
   * Remove the “Display: none” CSS property.
2. **Cross Site Scripting**
   * Insert Javascript in the input field
3. **Cross Site Scripting 1**
   * Insert Javascript using the attribute “onclick” of an input field

**Private**

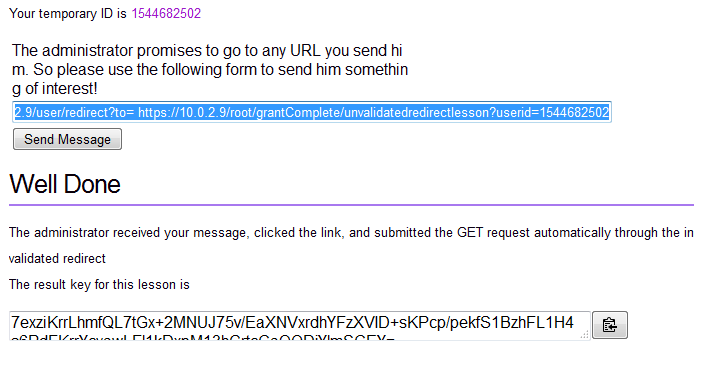
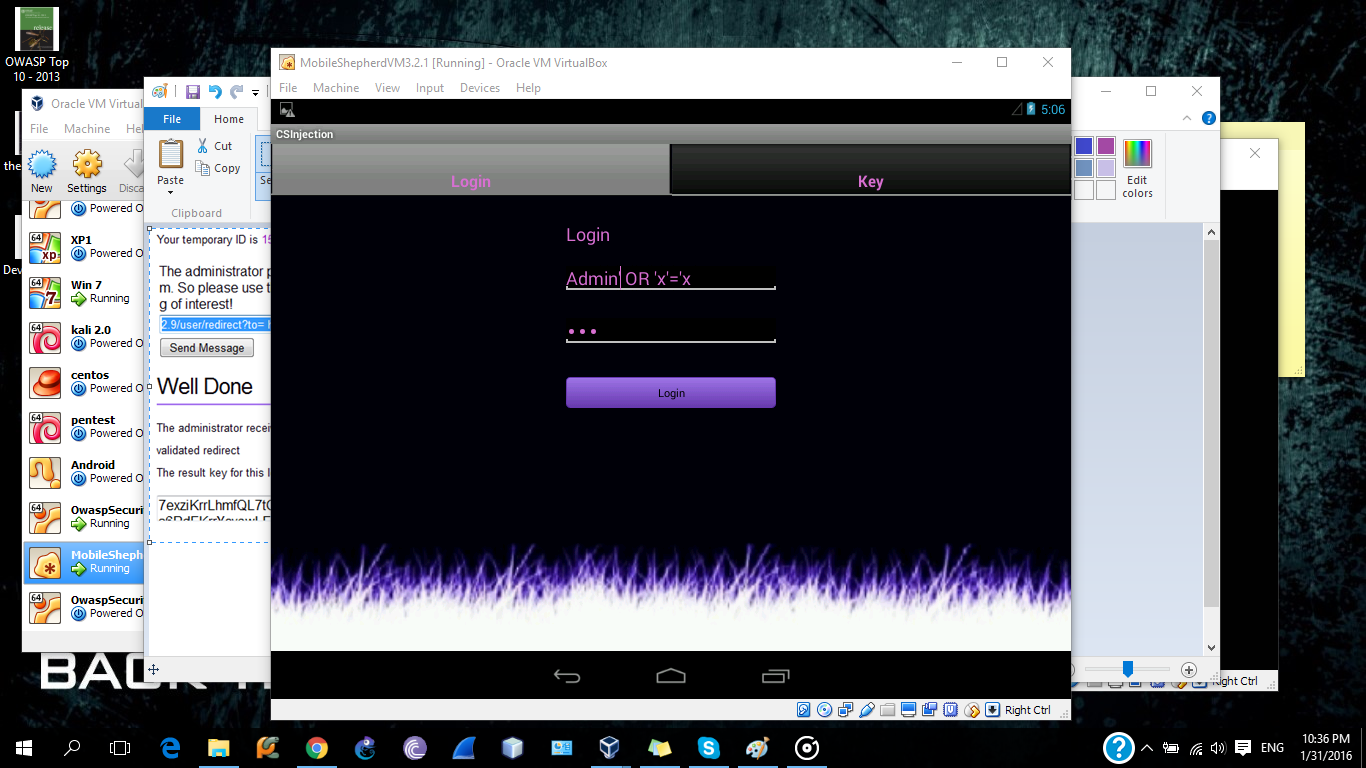
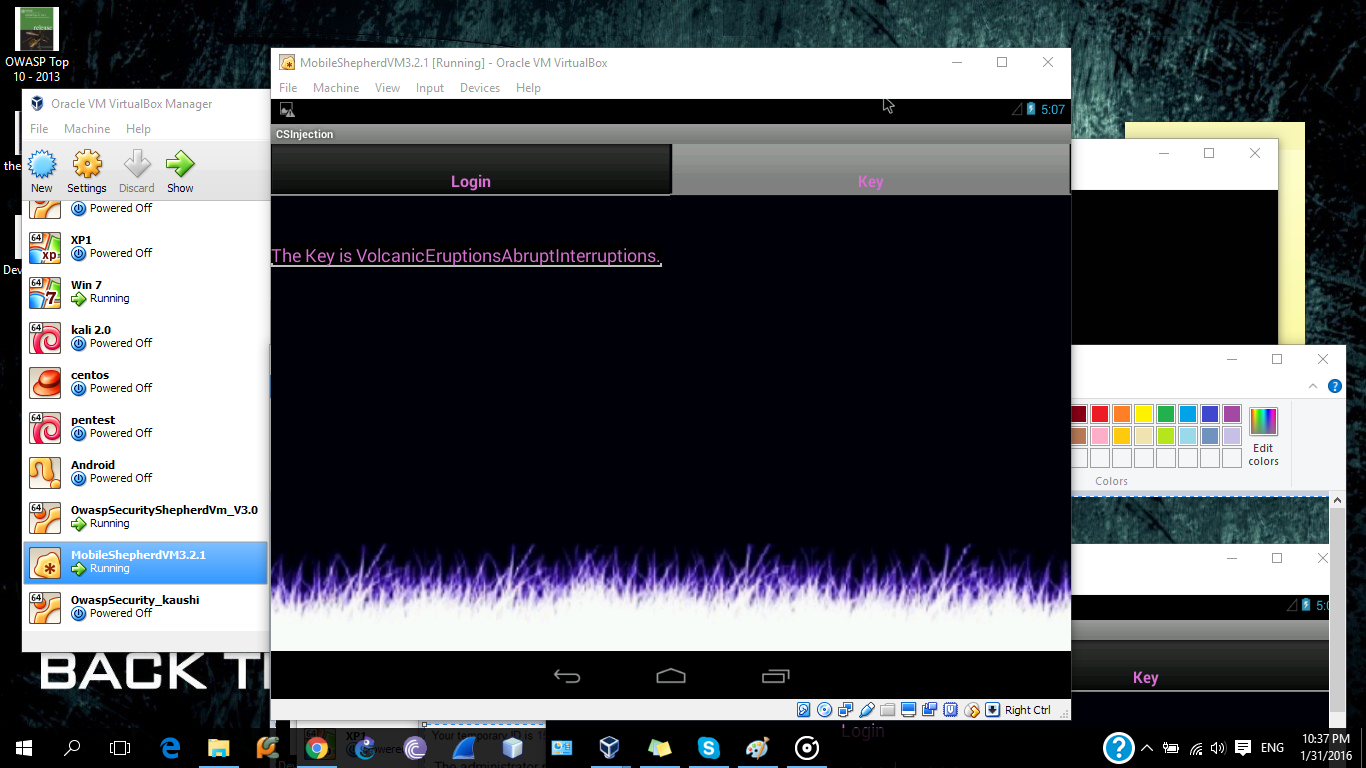
1. **Insecure Data Storage**
   * Using the less command the database can be viewed
2. **Insecure Cryptographic Storage**
   * Used a Base64 decoder to decode the message.
3. **SQL Injection**
   * ****Enter “ aa’ or ‘x’=’x ”
4. **Insecure Cryptographic Storage Challenge 1**
   * Use the 21 as the key for the decryption
   * The cipher text : Ymj wjxzqy pjd ktw ymnx qjxxts nx ymj ktqqtbnsl xywnsl; rdqtajqdmtwxjwzssnslymwtzlmymjknjqibmjwjfwjdtzltnslbnymdtzwgnlf
   * The message is : mylovelyhorserunningthroughthefieldwhereareyougoingwithyourbiga
5. **Insecure Direct Object Reference Challenge 1**
   * Intercept the request
   * Using the intruder in the Burpsuit designed a payload with various inputs as follows,
   * The User id is “11”
6. **Poor Data Validation 1**
   * Using a negative quantity the challenge can be completed.
7. **SQL Injection 1**
   * ****Same as the previous SQL injection
8. **Reverse Engineering**
   * First convert apk to jar using dex2jar
   * Using a Java decompiler the secret key is found which was hidden in the code



1. **Session Management Challenge 1**
   * Intercept the request
   * Decode the checksum.
   * Modify and encode the checksum again.
   * Forward the modified checksum
2. **Failure to Restrict URL 1**
   * The URL of the admin submission can be found in one of the scripts
3. **Unintended Data Leakage**
   * The log file which contains the key is kept in the following path,
   * The Key is
4. **Cross Site Request Forgery**
   * Used the modified URL to retrieve the key,
5. **Content Provider Leakage**
   * The key can be found in following path using Android Debug Bridge.

C:\Users\sam\Desktop\security shepherd\20.png

**Corporal**

1. **Reverse Engineering 1**
2. **Unvalidated Redirects and Forwards**
   * The key can be found using the following modified input
3. **Client Side injection**
   * Same as the previous SQL injections