

ATTACK

DEFENSE

by PentesterAcademy

Name	Arachini: Automated Vulnerability Scanning
URL	https://www.attackdefense.com/challengedetails?cid=2056
Type	DevSecOps: Dynamic Code Analysis

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

[Arachni](#) is an open-source framework for performing dynamic analysis on web applications.

The Arachni instance is running in the lab. The credentials provided below can be used to log in to the Arachni Framework.

Username	Password
admin@admin.admin	administrator

Three examples of vulnerable web portals are also provided. The details of these portals:

Web Portal	Web Portal URL
School Homework Web Portal	school-homework
Article Web Portal	article-site

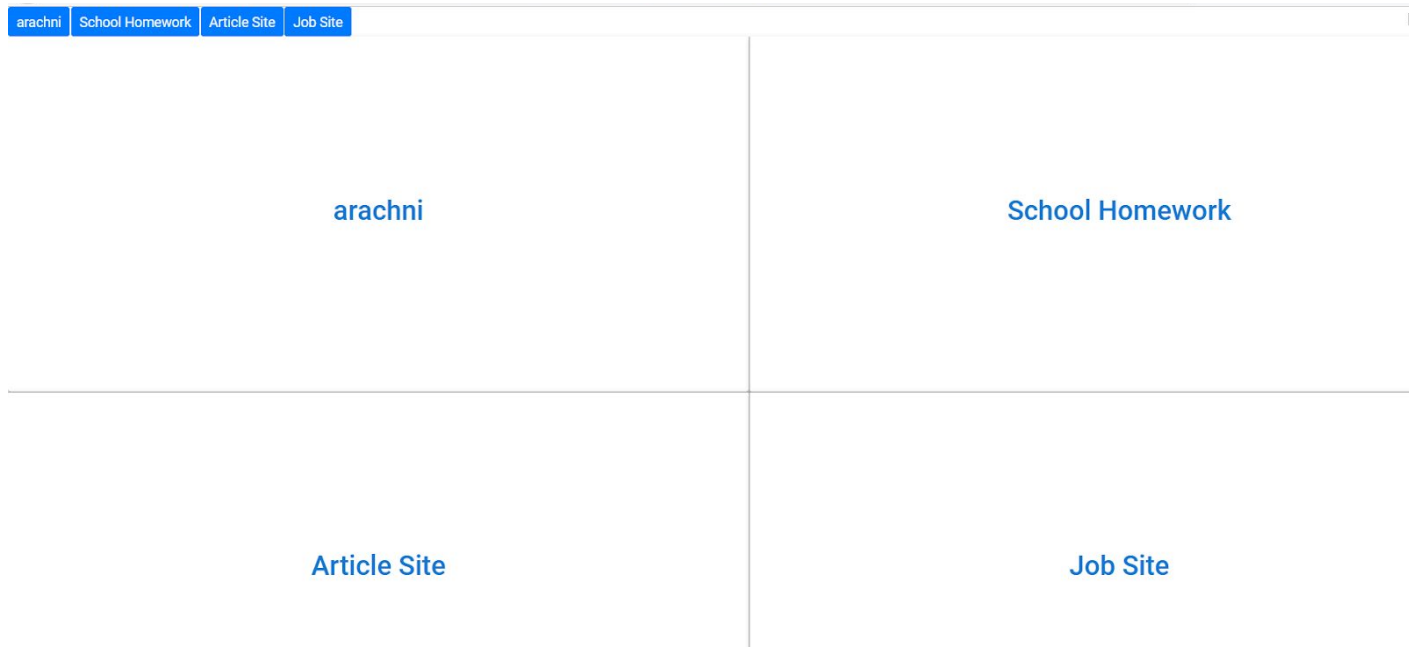
Job Advertisement Web Portal

job-site

Objective: Identify the vulnerabilities in web applications using Archini.

Lab Setup

On starting the lab, the following interface will be accessible to the user.



On choosing (clicking the text in the center) top left panel, a web UI of **Arachni** will open in a new tab

Sign in

Please consult the [Wiki](#) for default credentials.

Email

Password

Remember me

☐

Sign in

Similarly on selecting the top right panel, a web UI of **School Homework UI** will open in a new tab.

MSHW Page

Welcome to the HWPPage System!

Version 1.3 Beta 1

Select a page:

[Student Index](#)
[Classes](#)
[Subjects](#)
[Teacher Interface](#)

[View Classes](#)
[View Subjects](#)

Select your Class:

Classes:

[6106A](#)
[6206B](#)
[7107A](#)
[7207B](#)

On selecting the bottom left panel, a web UI of **Article Site UI** will open in a new tab.

[Pentester Academy](#)

[Login](#) | [Submit Articles](#) | [Register](#)

- [Home](#)

[Cheap hotels](#)

Find Hotels By Price,
Star Rating Or Location
Cheap hotels
www.ResortGateway.com

Ads by Google

All Categories

- [Arts & Entertainment](#)
- [Business](#)

And on selecting the bottom right panel, a web UI of **Job Site UI** will open in a new tab.

Welcome Back! this is your **NaN** visit.

JobSite Logo

Home | Jobseekers | Search Jobs | Post a CV | Register

Employers

Search Jobs

Put Job Titles, Location, Company Name, Skills, Industry, etc.

Job Listings

Post Date	Job Title
11/13/2008	tsd
11/13/2008	tsd
11/13/2008	Web developer

Lastest News

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque nisl. Integer dapibus nulla


Login


Username:


Password:


Select:
Employer


Featured Employers



AL FARA'A GROUP


Al Khail Group


SAMSUNG


SAMSUNG ENGINEERING


SYMBIOSIS


Ez Abash Holding Co

Solution

Step 1: Open the Arachni Web UI from the provided Interface.

Arachni v1.5.1 - WebUI v0.5.12

Sign in

Please consult the [Wiki](#) for default credentials.

Email

Password

Login using the credentials provided in the challenge description

Credentials:

- **Username:** admin@admin.admin
- **Password:** administrator

Arachni v1.5.1 - WebUI v0.5.12 Scans ▾ Profiles ▾ Dispatchers ▾ Users ▾

Signed in successfully.

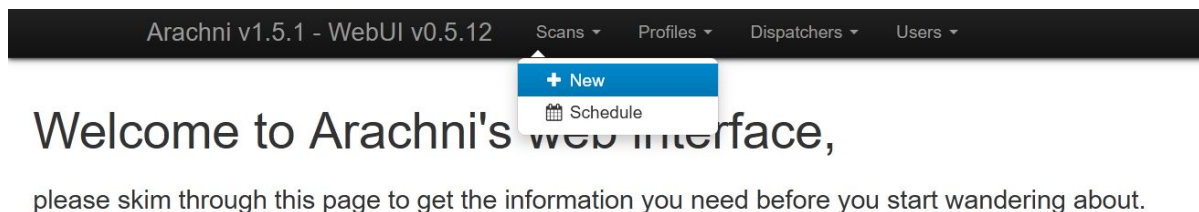
Welcome to Arachni's web interface,

please skim through this page to get the information you need before you start wandering about.

Choosing the right database

This is the web interface of the Arachni framework from where a user can start scans on the target websites.

Step 2: Click on the “New” button under Scans drop-down menu.



Choosing the right database

Example 1: School Homework

Step A: Enter the target URL in the attack field.

URL: http://school-homework

Arachni v1.5.1 - WebUI v0.5.12 Scans ▾ Profiles ▾ Dispatchers ▾ Users ▾

Start a scan

The only thing you need to do is provide some basic information and make a simple choice about

Full URL of the targeted web application (must include the appropriate protocol, http or https).
http://school-homework

Configuration profile to use.
Default (Global)

Description

Share with:
Regular User

Click on the Go button.

The screenshot shows the Arachni web interface. The top navigation bar includes 'Arachni v1.5.1 - WebUI v0.5.12', 'Scans', 'Profiles', 'Dispatchers', and 'Users'. The breadcrumb trail is 'Scans / http://school-homework'. On the left, there's a sidebar with 'TOGGLE VISIBILITY OF' (Comments) and 'ACTIONS' (Share, Edit schedule, Full edit). The main area displays 'http://school-homework/' with an 'Edit description' button. A light blue status bar at the bottom says 'The scan is initializing, please wait...'.

The tool will start the automated attack on the target website.

The screenshot shows the Arachni web interface during a scan. The top navigation bar includes 'Arachni v1.5.1 - WebUI v0.5.12', 'Scans 1', 'Profiles', 'Dispatchers', 'Users', and an 'Administrator' button. The breadcrumb trail is 'Scans / http://school-homework'. On the left, the sidebar shows 'TOGGLE VISIBILITY OF' (Comments, Statistics) and 'ACTIONS' (Share, Edit schedule, Full edit). The main area displays 'http://school-homework/' with an 'Edit description' button. A 'Scanning' button is visible. A light blue status bar says 'Currently auditing: http://school-homework/'. Below this, a table shows scan statistics:

Pages discovered	2	Requests performed	881	Requests per second	93.44	Request concurrency	20
Running for	00:00:14	Responses received	879	Timed out requests	1	Response times	0.069 s

Below the table, it says 'Issues [12]'.

Step B: Scroll down and check the issues found by the tool.

Arachni v1.5.1 - WebUI v0.5.12 Scans 1 Profiles Dispatchers Users Administrator

/ Scans / http://school-homework

All [26] Fixed [0] Verified [0] Pending verification [7] False positives [0] Awaiting review [0]

Listing all logged issues.

TOGGLE BY SEVERITY

Reset Show all Hide all

Severity	Count
High	5
Medium	13
Low	4
Informational	4

NAVIGATE TO

Cross-Site Scripting (XSS) in HTML tag	1
Cross-Site Scripting (XSS)	1
SQL Injection	1
Blind SQL Injection (timing attack)	1
Code injection (timing attack)	1
Common directory	12
Unencrypted password form	1

URL	Input	Element
Cross-Site Scripting (XSS) in HTML tag 1		
<p>Client-side scripts are used extensively by modern web applications. They perform from simple functions (such as the formatting of text) up to full manipulation of client-side data and Operating System interaction.</p> <p>Cross Site Scripting (XSS) allows clients to inject scripts into a request and have the server return the script to the client in the response. This occurs because the application is taking untrusted data (in this example, from the client) and reusing it without performing any validation or sanitisation.</p> <p>If the injected script is returned immediately this is known as reflected XSS. If the injected script is stored by the server and returned to any client visiting the affected page, then this is known as persistent XSS (also stored XSS).</p> <p>Arachni has discovered that it is possible to insert content directly into an HTML tag. For example <code><INJECTION_HERE href=.....etc></code> where <code>INJECTION_HERE</code> represents the location where the Arachni payload was detected.</p> <p>(CWE)</p>		

Issues Detected

- XSS vulnerability
- Blind SQL Injection
- Code Injection

Step C: The detailed report for a detected vulnerability can be viewed by clicking the question mark.

Arachni v1.5.1 - WebUI v0.5.12 Scans 1 Profiles Dispatchers Users Administrator

Scans / http://school-homework (Default profile) / Cross-Site Scripting (XSS) in HTML tag in Link Input 'class'

References

- Secunia
- WASC
- OWASP

Overview Identification data Vector data HTTP data Request Response

Request

```
GET /view.php?class=610%27%20arachni_xss_in_tag%30%27ef5ae55ac3d5a4b0a9066eeb6e7dd27d%27%20blah%30%27 HTTP/1.1
Host: school-homework
Accept-Encoding: gzip, deflate
User-Agent: Arachni/v1.5.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,he;q=0.6
X-Arachni-Scan-Seed: ef5ae55ac3d5a4b0a9066eeb6e7dd27d
```

Response

Proof is highlighted in red and scroll-centered.

```
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.26
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 582
Content-Type: text/html

<head>
<link rel="stylesheet" type="text/css" href="stu.css" /></head><IMG SRC="logo.gif" alt="MSHW Page Logo" width="576" height="100"><div id=menu>
<A HREF='http://school-homework/index.php'>Student Index</A>
<br><A HREF='http://school-homework/class.php'>Classes</A><br>
<A HREF='http://school-homework/subject.php'>Subjects</A><br>
<A HREF='http://school-homework/post/get_hw.php'>Teacher Interface</A>
</div><H1>Welcome to the MHWPage System!</H1>
<div id=content><TITLE>MSHWPage -- View Homework</TITLE>
<h2>Homework for as of Sep 19 6:19:16 UTC 2020:</h2>
<a href='view.php?print=1&&class=610' arachni_xss_in_tag='ef5ae55ac3d5a4b0a9066eeb6e7dd27d' blah='><IMG SRC='printButton.png'> Printable Versions</a>
<h3>Homework due today or later:</h3>
Query failed : You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'' arachni_xss_in_tag='ef5ae55ac3d5a4b0a9066eeb6e7dd27d' blah=' and assignment.su' at line 2
```

Here, the details of XSS vulnerability are provided. The payload and response of the request are highlighted in red.

Example 2: Article Site

Step A: Enter the target URL in the attack field

URL: http://article-site

Start a scan

The only thing you need to do is provide some basic information and make a simple choice about the type of scan you want to perform.

<input type="text" value="http://article-site"/>	<input type="text" value="Default (Global)"/>
Full URL of the targeted web application (must include the appropriate protocol, http or https).	Configuration profile to use.
<div>Description</div>	<div>Share with: Regular User</div>
You can use Markdown for text formatting.	

Click on the Go button to start the attack on the website.

Arachni v1.5.1 - WebUI v0.5.12 Scans Profiles Dispatchers Users

Scans / http://article-site

TOGGLE VISIBILITY OF

Comments

ACTIONS

Share

Edit schedule

Full edit

http://article-site/

Edit description

The scan is initializing, please wait...

The tool will start the automated attack on the target website.

Arachni v1.5.1 - WebUI v0.5.12 Scans 1 Profiles Dispatchers Users Administrator

Scans / http://article-site

TOGGLE VISIBILITY OF

- Comments
- Statistics

ACTIONS

- Share
- Edit schedule
- Full edit

http://article-site/

Edit description

Scanning

Currently auditing:

- http://article-site/category.php?id=2

Pages discovered	14	Requests performed	14231	Requests per second	318.58	Request concurrency	20
Running for	00:00:59	Responses received	14125	Timed out requests	21	Response times	0.037 s

Issues [19]

Issues may be missing some context while the scan is running.
You better wait until the scan is over to review them as the meta-analysis phase will flag probable false-positives and other untrusted issues accordingly.

Step B: Scroll down and check the issues found by the tool.

Arachni v1.5.1 - WebUI v0.5.12 Scans 1 Profiles Dispatchers Users Administrator

Scans / http://article-site

Issues [20]

Issues may be missing some context while the scan is running.
You better wait until the scan is over to review them as the meta-analysis phase will flag probable false-positives and other untrusted issues accordingly.

All [20] * Fixed [0] ✓ Verified [0] ! Pending verification [0] ✖ False positives [0] ! Awaiting review [0]

Listing all logged issues.

TOGGLE BY SEVERITY

Reset Show all Hide all

High 4

Medium 2

Low 8

Informational 9

NAVIGATE TO

Cross-Site Scripting (XSS) in HTML tag 1

Cross-Site Scripting (XSS) 1

SQL Injection 2

Common directory 2

URL	Input	Element
Cross-Site Scripting (XSS) in HTML tag 1		
Client-side scripts are used extensively by modern web applications. They perform from simple functions (such as the formatting of text) up to full manipulation of client-side data and Operating System interaction.		
Cross Site Scripting (XSS) allows clients to inject scripts into a request and have the server return the script to the client in the response. This occurs because the application is taking untrusted data (in this example, from the client) and reusing it without performing any validation or sanitisation.		
If the injected script is returned immediately this is known as reflected XSS. If the injected script is stored by the server and returned to any client visiting the affected page, then this is known as persistent XSS (also stored XSS).		
Arachni has discovered that it is possible to insert content directly into an HTML tag. For example <INJECTION_HERE href=.....etc> where INJECTION_HERE represents the location where the Arachni payload was detected.		

Issues Detected

- Cross-Site Scripting
- SQL Injection

Step C: The detailed report for a detected vulnerability can be viewed by clicking the question mark.

The screenshot displays the Arachni v1.5.1 - WebUI v0.5.12 interface. The top navigation bar includes links for Scans, Profiles, Dispatchers, and Users, along with a user profile for Administrator. The main content area is divided into two sections: References and Request/Response details.

References:

- UnixWiz
- Wikipedia
- SecuriTeam
- OWASP
- WASC
- W3 Schools

Request:

```
POST /login.php HTTP/1.1
Host: article-site
Accept-Encoding: gzip, deflate
User-Agent: Arachni/v1.5.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,he;q=0.6
X-Arachni-Scan-Seed: 366cf4f8f2d866fec49ae3d8e4231e86
Cookie: PHPSESSID=o85nbru1m2n6pqgfofmd6p45d5
Content-Length: 96
Content-Type: application/x-www-form-urlencoded

userid=arachni_user%22%27%60--&password=5543%21%25arachni_secret&remember=1&submitstyle=log%20in
```

Response:

Proof is highlighted in red and scroll-centered.

```
HTTP/1.1 200 OK
Date: Sat, 19 Sep 2020 06:39:17 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.25
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 167
Content-Type: text/html

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''--' and password = md5('55431%arachni_secret')' at line 1
```

Here, The detailed summary of the SQL Injection is provided where the SQL error is highlighted in the **red** colour.

Example 3: Job Site

Step A: Enter the target URL in the attack field

URL: http://job-site

Start a scan

The only thing you need to do is provide some basic information and make a simple choice about the type of scan you want to perform.

Default (Global)

Full URL of the targeted web application (must include the appropriate protocol, http or https).

Configuration profile to use.

Description

Share with:
Regular User

You can use Markdown for text formatting.

[Advanced options](#)

Click on the Go button.

Arachni v1.5.1 - WebUI v0.5.12 Scans 1 Profiles Dispatchers Users

/ Scans / http://job-site

TOGGLE VISIBILITY OF
Comments

ACTIONS
Share
Edit schedule
Full edit

http://job-site/

Edit description

The scan is initializing, please wait...

Step B: Scroll down and check the issues found by the tool.

Arachni v1.5.1 - WebUI v0.5.12 Scans 2 Profiles Dispatchers Users 1 Administrator

Scans / http://job-site/

TOGGLE VISIBILITY OF

- Comments
- Statistics

ACTIONS

- Share
- Edit schedule
- Full edit

http://job-site/

Edit description

Scanning

Currently auditing:

- http://job-site/

Pages discovered	7	Requests performed	2152	Requests per second	207.88	Request concurrency	20
Running for	00:00:20	Responses received	2121	Timed out requests	20	Response times	0.087 s

Issues [14]

Issues may be missing some context while the scan is running.
You better wait until the scan is over to review them as the meta-analysis phase will flag probable false-positives and other untrusted issues accordingly.

All [14] * Fixed [0] ✓ Verified [0] Pending verification [0] ✗ False positives [0] Awaiting review [0]

Step C: Scroll down and check the issues found by the tool

All [16] * Fixed [0] ✓ Verified [0] Pending verification [0] ✗ False positives [0] Awaiting review [0]

Listing all logged issues.

TOGGLE BY SEVERITY

Reset Show all Hide all

- High 5
- Medium 6
- Low 2
- Informational 3

NAVIGATE TO

- Cross-Site Scripting (XSS) 3
- Blind SQL Injection (differential analysis) 2
- Unencrypted password form 1
- Common directory 5
- Missing "X-Frame-Options" header 1
- Password field with auto-complete 1
- Interesting response 2
- HttpOnly cookie 1

Cross-Site Scripting (XSS) 3

Client-side scripts are used extensively by modern web applications. They perform from simple functions (such as the formatting of text) up to full manipulation of client-side data and Operating System interaction.

Cross Site Scripting (XSS) allows clients to inject scripts into a request and have the server return the script to the client in the response. This occurs because the application is taking untrusted data (in this example, from the client) and reusing it without performing any validation or sanitisation.

If the injected script is returned immediately this is known as reflected XSS. If the injected script is stored by the server and returned to any client visiting the affected page, then this is known as persistent XSS (also stored XSS).

Arachni has discovered that it is possible to insert script content directly into HTML element content.

(CWE)

URL	Input	Element
http://job-site/job.php	username	Link
http://job-site/job.php	title	Link
http://job-site/search.php	keyword	Form

Issues Detected

- Cross-Site Scripting
- SQL Injection

Step D: The detailed report for a detected vulnerability can be viewed by clicking the question mark.

