

Pentest Report – Pagekit 01-2017

<https://github.com/pagekit/>

SecureLayer7

Index

- **Intro**
- **Scope**
- **Identified Vulnerabilities**
- **Conclusion**

Intro

“Pagekit is a modular and lightweight content management system. It gives the tools to create websites.” - As per Pagekit website

This penetration test was carried out by Securelayer7 team. It was carried out for 4 days. The test identified one critical vulnerability which leads to vertical/horizontal authentication bypass. It was also found that Pagekit users swiftmailer version 5.4.1 for sending mails which has been recently hit with remote code execution vulnerability. In addition to that, 4 minor and 1 medium vulnerabilities were discovered that require to be addressed.

Scope

- Source code available on Github (<https://github.com/pagekit/>)
- Locally hosted web application

Testing environment

- Latest Kali Linux 64 bit 2016.2
- LAMP

Identified Vulnerabilities

The following sections list vulnerabilities. The findings are listed in chronological order and not by their degree of severity. The severity is given in brackets following the title heading. Each bug is given a unique identifier for the purpose of future reference and follow-up.

SL7_PGKT_01: Vertical/Horizontal Authentication Bypass (Critical)

Description: A vulnerable end point reveals the password reset links clicked by all the users of Pagekit. This means whoever clicks on those links will have their accounts compromised. No privilege level is required to see that end point. It is public. So, an unauthenticated attacker can perform complete account takeover. Note that the password reset link also has the username mentioned in it. So, an account takeover can be performed because the attacker knows both the password and username (password was changed by him and the username is displayed in the reset link).

The vulnerable end point is: `/pagekit/index.php/_debugbar/<the secret code here>`

The ‘secret code’ is nothing but the value of debugbar in `<script>var $debugbar = {"current":"60b4b0f6a3fcdf3bff05668401c2ec12c75ee152"};</script>` js in the HTML source of the login page .

SecureLayer7

Time and Again, Securing you



```
9 <meta property="twitter:card" content="summary_large_image">
10 <meta property="og:site_name" content="PageKit Test">
11 <meta property="og:url" content="http://127.0.0.1/pagekit/index.php/user/login">
12 <link href="/pagekit/app/system/modules/theme/favicon.ico" rel="shortcut icon" type="image/x-icon">
13 <link href="/pagekit/app/system/modules/theme/apple_touch_icon.png" rel="apple-touch-icon-precomposed">
14 <title>Login | PageKit Test</title>
15 <link href="/pagekit/packages/pagekit/theme-one/css/theme.css?v=3218" rel="stylesheet">
16 <link href="/pagekit/app/modules/debug/assets/css/debugbar.css?v=3218" rel="stylesheet">
17 <script>var $debugbar = {"current":"60b4b0f6a3fcd33bfff05668401c2ec12c75ee152"};</script>
18 <script>var $pagekit = {"url":"\pagekit\index.php","csrf":"6a7d47c6d525d5615b1002794609f96f5ffa9050"};</script>
19 <script src="/pagekit/app/assets/jquery/dist/jquery.min.js?v=3218"></script>
20 <script src="/pagekit/app/assets/uikit/js/uikit.min.js?v=3218"></script>
```

PoC:

1. Along with this report, an exploit developed in Ruby is attached.
2. Execute it using command prompt.
3. Login as any user and reset your password and click on your password reset link.
4. Go to the command prompt and observe that the executing script captures the password reset link.

Mitigation: When login page is loaded, the long cryptographic code is revealed in client side JavaScript. Do not disclose it there. If it is not displayed, then no one can use it to find the reset links. It is very long so brute forcing is not possible, so this will make it safe.

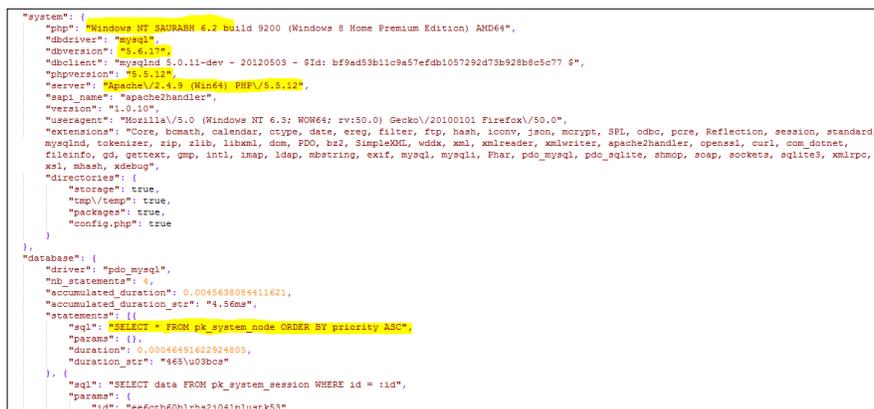
SL7_PGKT_02: Server side information disclosure (Medium)

Description: The application has an endpoint that reveals following information:

- Some database queries
- Server information including computer name, PHP version, SQL server name and version
- Full path of the web application

The end point is the same as the one mentioned in SL7_PGKT_05.

PoC:



```
"system": {
  "php": "Windows NT SARABH 4.2 Build 9200 (Windows 8 Home Premium Edition) AMD64",
  "dbdriver": "mysqli",
  "dbversion": "5.6.17",
  "dbclient": "mysqli 5.0.11-dev - 20120503 - ID: bf9ad59b1c9a57efdb1057292d73b928b8c5e7? $",
  "phpversion": "5.5.12",
  "server": "SARABH/4.2.9 (Win64) PHP/5.5.12",
  "sapi_name": "apache2handler",
  "version": "1.0.10",
  "useragent": "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0",
  "extensions": "Core, bcmath, calendar, ctype, date, exec, filter, ftp, hash, iconv, json, mcrypt, SPL, openssl, session, standard, mysqlnd, tokenizer, zip, zlib, libxml, dom, PDO, bz2, SimpleXML, wddx, xml, xmlreader, xmlwriter, apache2handler, openssl, curl, com_dotnet, fileinfo, gd, gettext, gmp, intl, imap, ldap, mbstring, exif, mysqli, mysqli, Phar, pdo_mysql, pdo_sqlite, shmop, soap, sockets, sqlite3, xmlrpc, xsl, mbasm, xdebug",
  "directories": {
    "storage": true,
    "tmp/temp": true,
    "packages": true,
    "config.php": true
  }
},
"database": {
  "driver": "pdo_mysql",
  "nb_statements": 4,
  "accumulated_duration": "0.0046630094611621",
  "accumulated_duration_str": "4.66ms",
  "statements": [
    {
      "sql": "SELECT * FROM pk_system_node ORDER BY priority ASC",
      "params": {},
      "duration": "0.00046491622924805",
      "duration_str": "465u03bc"
    }
  ],
  "sql": "SELECT data FROM pk_system_session WHERE id = :id",
  "params": {
    "id": "ee6ctb601zhs2j041plustck53"
  }
}
```

A file named informationdisclosure.json is attached with this report. Please view it.

Mitigation: The mitigation is same as that of SL7_PGKT_01.

SL7_PGKT_03: Misconfiguration .htaccess (Low)

Description: During the penetration testing we have used the LAMP to test the application and we found that the .htaccess does not restrict users to access the file phpunit.xml.dst. It has sensitive data such as temporary usernames and passwords of DB, SMTP and FTP. Although, the credentials are not used by the application, but they can be used for testing purpose or in future if there is any enhancement in the application. This is the reason, this vulnerability is reported with low severity. It can also be treated as informational.

PoC: Request the file in browser and observe that it is publicly available

<http://yourpagekitsite.com/phpunit.xml.dst>

Mitigation:

- 1) For apache server. Please add '|phpunit.xml.dist' in line 2 of .htaccess file. After this is done, any user who requests it will get a forbidden error.
- 2) For Nginx server – the phpunit.xml.dist used for the internal purposes, so you can provide IP based access to the user.

SL7_PGKT_04: Weak Password Policy (Low)

When a new user is added, the application does not enforce a password policy resulting in users created having weak passwords. This is not a good practice. Although it is not a major vulnerability, but letting users use weak passwords make the job of an attacker easier because the passwords are then very easy to guess and cracked with less efforts.

PoC:

1. Login as admin and go to create new user page (link: <http://127.0.0.1/pagekit/index.php/admin/user/edit>)
2. Enter valid details in all the fields
3. Enter password as test, 1234.
4. Click on SAVE
5. Observe that the application accepts it

Mitigation: Implement server side check that validate whether entered password is of min 6 characters or not. In the following function: validate() (line number 220-260) in

`pagekit/app/system/modules/user/src/models/user.php` . It is also recommended to user to change the password change password on the first login.

SL7_PGKT_05: Sensitive Information leakage via referrer header (Low)

Description: Password reset security token gets leaked to third party websites via referrer header. The homepage has a footer section which has links to third party websites such as github, twitter. If a user clicks on any of the links then the URL of the current page gets sent to them via referrer header and will be seen in logs of the web server of those websites. Although you can trust those sites, but it is not a good practice to include security tokens in referrer header while navigating to third party websites.

PoC:

```
GET /pagekit HTTP/1.1
Host: twitter.com
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer:
http://127.0.0.1/pagekit/index.php/user/resetpassword/confirm?user=saurabh&key=51mwy/ethCt2SqrTFwKarkk90MMzZkWC
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

Mitigation: In order to resolve this, whenever the password reset link gets loaded have the following in code between <head> tags of the html: <meta name="referrer" content="never" /> This will not send referrer headers to third party websites.

SL7_PGKT_06: Plain text storage of credentials (Low)

The password for SMTP connection is stored in plain text in the database. If some other vulnerability like SQL Injection causes the DB to be compromised, then attacker will obtain SMTP credentials.

PoC:

1. In order to reproduce this issue, set smtp settings using admin panel.
2. Go to DB and execute the following query: `SELECT * FROM `pk_system_config` where name like '%mail%'`

SecureLayer7

Time and Again, Securing you

```
id int(10) auto_increment
9
name varchar(255)
system/mail
value longtext
{"driver":"smtp","host":"smtp.gmail.com","port":"587",
"username":"saurabh.banawar@securelayer7.net","p
assword":"<the password here>","encryption":"tls","auth_mode":null,"from_nam
e":"SaurabhAdmin","from_address":"saurabh.banawa
...
 Save changes to original
 Insert as new row
Submit Cancel
```

Mitigation: Encrypt the password using a script and then store it in DB. At the time of sending email, decrypt it and then send to the SMTP server.

SL7_PGKT_07: SWIFTMAILER Remote Code Execution (Low)

Description: Pagekit uses swiftmailer version 5.4.1 for sending mails to users. All swiftmailer versions <= 5.4.5 are vulnerable to remote code execution (CVE-2016-10074). Note that in order to exploit this vulnerability, an attacker has to enter payload in the 'From' address of the mails. But the current Pagekit application does not have any functionality that lets users do that (for e.g. contact us). But it is recommended that swiftmailer library should be upgraded to latest patched version because there is a chance that a contact us page can be developed by Pagekit team in the future as an enhancement.

PoC: Due to no functionality that lets user modify 'From' field of an email, a PoC could not be made.

Mitigation: Upgrade swiftmailer to latest patched version as soon as the vendor patches it.

Conclusion

One authentication bypass vulnerability has been found which should be the point of concern. The fact that this can be remote exploited by an unauthenticated user makes this a severe impact issue. Another thing to be concern of is the swiftmailer library version which is vulnerable to remote code execution. However, no functionality that lets users change the 'From' mail address makes this very hard to be exploited. It should be noted that the problems are all implementation flaws and not design flaws. The findings are not very hard to fix.