# SSD Advisory – PHP Melody Multiple Vulnerabilities

**blogs.securiteam.com**/index.php/archives/3464

SSD / Maor Schwartz October 9, 2017

**Vulnerabilities Summary**
The following advisory describes three (3) vulnerabilities found in PHP Melody version 2.7.3.

PHP Melody is a “self-hosted Video CMS which evolved over the last 9 years. SEO optimization, unbeaten security and speed are advantages you no longer have to compromise on.
A truly great CMS should help you save time and make your life easier not complicate it. Nobody enjoys spending time and money on inferior solutions. If you value your time, don’t settle for anything but the best video CMS with a proven track record, constant support and updates.”

The vulnerabilities found in PHP Melody are:

- Stored PreAuth XSS that leads to administrator account takeover
- SQL Injection (1)
- SQL Injection (2)

**Credit**
An independent security researcher, Paulos Yibelo, has reported this vulnerability to Beyond Security’s SecuriTeam Secure Disclosure program.

**Vendor response**
PHP Melody has released patches to address this vulnerability.

For more information: http://www.phpsugar.com/blog/2017/10/php-melody-v2-7-3-maintenance-release/

**Vulnerabilities details**

**Stored PreAuth XSS that leads to administrator account takeover**
User controlled input is not sufficiently sanitized, such that by sending a POST request to *page_manager.php* with the following parameters (vulnerable parameter – page_title)

```
1 page_manager.php?do=new&id=&author=&showinmenu=0&meta_keywords=555-555-0199@example.com&
status=0&submit=Publish&page_name=Peter+Winter&page_title=408b7<script>alert(1)<%2fscript>f2faf
```

An attacker can trigger the vulnerability and when administrator/moderator/editor or anyone with privileges visits Admin access */admin/pages.php?page=1* the payload is triggered and the alert is executed.

**SQL Injection (1)**
User controlled input is not sufficiently sanitized, by sending a POST request to */phpmelody/admin/edit_category.php* with the following parameters:

```
1 category=3&meta_keywords=555-555-0199@example.com&
tag=categoryone&
save=Save$name=Sample+Category+%231&image='&
meta_title=555-555-0199@example.com
```

The vulnerable parameter is the POST “image” parameter. We can send a single quote (‘) to verify and the server will

respond with an SQL error. We can inject SQL Queries here or extract data.

This attack requires an admin/modernator or editor to visit a malicious website that will submit the form with a malicious “image” parameter as an Injection

**SQL Injection (2)**

SQL Injection is on a cookie-value and can be exploited without any user interaction.

The cookie value “aa_pages_per_page” is the vulnerable parameter and we can use time based SQL Injection techniques to verify,

The payload we used ‘ AND benchmark(20000000%2csha1(1))—makes the server sleep for a long time (5-20 seconds).