SQLi in ph_simpleblog CVE-2021-36748

This blog post details an SQLi I found in Blog for Prestashop (ph_simpleblog) by Prestahome, it is also my first CVE 😊

To begin with I had to identify that the module was installed, it is a blog plugin so this can generally be spotted by looking at the page source of the blog but you can also test if Prestashop modules are installed more directly by checking `https://example.com/modules/ph_simpleblog/config.xml`.

This Prestashop module uses the controller functionality so that's where I started to look as it's usually where the user input will go first.

The controllers of interest were `controllers/front/list.php` and `controllers/front/single.php`. In Prestashop modules the init() and initContent() functions are called whenever the endpoint is reached. In the init function we can see that the `sb_category` parameter isn't sanitised.

```
public function init(){
    parent::init();
    $sb_category = Tools::getValue('sb_category');
    // ...
    if($sb_category) $this->sb_category = $sb_category;
    //more code...
}
```

Later in the initContent() function sb_category gets passed to the getByRewrite() function in the SimpleBlogCategory class.

```
public function initContent(){
    // ...
    if($this->sb_category != ''){
        $this->context->smarty->assign('is_category', true);
        $SimpleBlogCategory =
        SimpleBlogCategory::getByRewrite($this->sb_category, $id_lang);
```

In the getByRewrite function we can see it gets used in an SQL query without any sanitisation.

```
public static function getByRewrite($rewrite = null, $id_lang = false){
    if(!$rewrite) return;
    $sql = new DbQuery();
    $sql->select('l.id_simpleblog_category');
    $sql->from('simpleblog_category_lang', 'l');
    if($id_lang)
        $sql->where('l.link_rewrite = \''.$rewrite.'\' AND l.id_lang =
    '.(int)$id_lang);
    else
        $sql->where('l.link_rewrite = \''.$rewrite.'\'');
    $category = new SimpleBlogCategory(Db::getInstance()->getValue($sql),
    $id_lang);
    return $category;
}
```

Okay, success, found the SQLi but where is this endpoint? In Prestashop the controllers for modules are located at `/module/modulename/filename` (not `/modules`!) so for this case: `https://example.com/module/ph_simpleblog/list?sb_category=*`. No exploit script or manual SQLi was required as sqlmap was able to detect it as boolean-based blind.

Fixing the issue

Prestashop provide a built in function for sanitising strings to be used in SQL queries called pSQL. This is the quick fix in situations like this but one must be sure to surround the parameter with quotes or the query will still be vulnerable to SQLi as I will show in the next blog post.

The most correct way to patch this would be to use PDO as desribed in Prestashop's Best Practices for the DB Class. PDO eliminates the risks of faulty parameter sanitisation and makes it hard to do things the wrong way.

I found the contact details for the developer of the module and they were quickly able to patch the issue.

Timeline

Date	Action
18/06/2021	Issue discovered during a pentest
13/07/2021	Reported issue to Prestahome
14/07/2021	Prestahome patched the issue in version 1.7.8
15/07/2021	Number CVE-2021-36748 assigned
18/08/2021	Blog post released
20/08/2021	pajoda released a Nuclei template for this CVE

2021-08-18

#ph_simpleblog #simpleblog #prestahome #CVE-2021-36748

SQLi in SmartBlog C...

Adminer - A Guide F...

Powered by Hugo, theme Anubis.

Dark theme