Bug - Security Vulnerabilities (not serious)

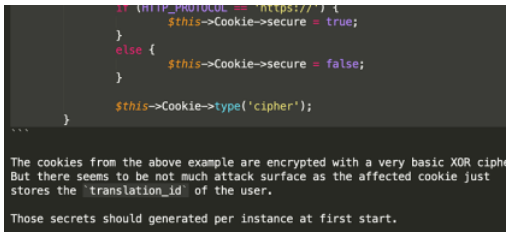
eramba **Aug '20**

We got reported today in the morning (European time) three bugs that work on community and enterprise. They are not serious (they can not mingle your data) but anyway we **fixed them already for enterprise** . The fix took 5hs from reporting (can anyone remind me which vendor provides a patch in 5 hours please?).

The details (from extracts on our email conversation with the guy that reported them):

```
The vulnerabilities are:

1- Hard coded secrets (eramba_static_secrets.md)
```



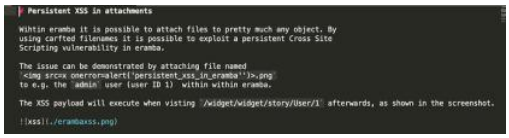
```
            $this->Cookie->secure = true;
        }
        else {
            $this->Cookie->secure = false;
        }

        $this->Cookie->type('cipher');
    }
```

```
The cookies from the above example are encrypted with a very basic XOR ciphe
But there seems to be not much attack surface as the affected cookie just
stores the `translation_id` of the user.

Those secrets should generated per instance at first start.

2- Persistent cross site scripting (eramba_persistent_xss.md & erambaxss.png)
```



```
Persistent XSS in attachments

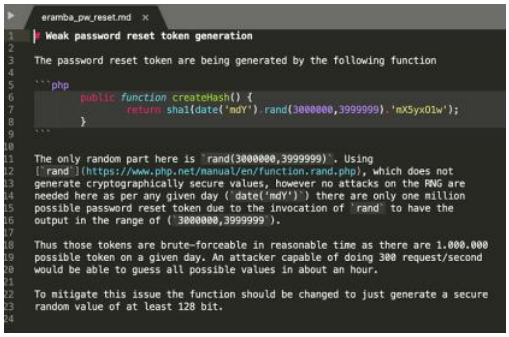
Wihtin eramba it is possible to attach files to pretty much any object. By
using carfted filenames it is possible to exploit a persistent Cross Site
Scripting vulnerability in eramba.

The issue can be demonstrated by attaching file named
`<img srcnx onerror=alert('persistent_xss_in_eramba')>.png`
to e.g. the `admin` user (user ID 1) within within eramba.

The XSS payload will execute when visting `/widget/widget/story/User/1` afterwards, as shown in the screenshot.

![xss](./erambaxss.png)

3- Weak password reset token generation (eramba_pw_reset.md)
```



```
eramba_pw_reset.md

# Weak password reset token generation

The password reset token are being generated by the following function

```php
        public function createHash() {
            return sha1(date('mdY').rand(3000000,3999999).'mX5yxO1w');
        }
```

The only random part here is `rand(3000000,3999999)`. Using
[`rand`](https://www.php.net/manual/en/function.rand.php), which does not
generate cryptographically secure values, however no attacks on the RNG are
needed here as per any given day (`date('mdY')`) there are only one million
possible password reset token due to the invocation of `rand` to have the
output in the range of (`3000000,3999999`).

Thus those tokens are brute-forceable in reasonable time as there are 1.000.000
possible token on a given day. An attacker capable of doing 300 request/second
would be able to guess all possible values in about an hour.

To mitigate this issue the function should be changed to just generate a secure
random value of at least 128 bit.
```

A summary on them:

1- is **not really relevant** so nothing was or will be done with this for a long time

2- this means that if a person HAS an account in eramba and CAN login (this means you know who the person is and you trust otherwise they would not have an account) , that person can upload as attachment a file with XSS embedded and that will load when the same (or another user) clicks on the attachment. This of course is a bit unlikely as whoever upload a file will need an account and it would be evident who the perpetrator was (the trails are on the modal). This is fixed by sanitising output on attachments.

3- When you “lost a password” (ONLY if you are using local accounts, most people uses LDAP and therefore this is not applicable) you can “recover” them if you forgot them, eramba will send an email with a token (valid for 24hs). In theory (this has not been tested as is unlikely to work) you could try to brute-force the token (if you know the url) by trying 300 (or more) keys per second non stop for one hour. **Frankly speaking no eramba deployment can stand 300 requests per second without going down so we see this again very unlikely**. Anyway is fixed by reducing the window to one hour and making the token more complicated (before 6 random characters now 50)

We want to thank ***“Joern Schneeweisz from the GitLab Security Research Team”*** because instead of sending a dull tool report which is almost never useful he took time and debug this and showed us in detail how to reproduce these issues. It took 5 hours issue a fix but im sure it took another couple of hours to find them !

Community will get these and many many other bugs fixed on the anual upgrade which will take place around November. Enterprise has a **fix on e2.19.3** .

Release - 2.19.3

eramba **Sep '20**

Skip to main content

we submitted the issue to mitre with the bugs and got the CVE’s issued:

- CVE-2020-25104
- CVE-2020-25105

Eramba Mail - Re_ [scr953893] eramba - c2.8.1.pdf (78.3 KB)