Anísio Santos

CVE: 2021-3311 October CMS Token Reactivation

23 Feb 2021

Don't get me wrong, but I believe that 'luck' many times is really a thing in the cybersecurity field. I'm not talking just about the normal luck, that makes you hit a jackpot on a slot machine, I'm talking about another kind of luck, an 'experience luck', that comes just with knowledge. Many bounties and many pieces of researches emerge just by looking at the right amount of code at the right path number and them BOOM! Been able o have that feeling and timing is something that I believe comes with experience, time and familiarity.

I think I have much more to learn about knowledge and experience to get the 'experience luck' feeling, but sometimes I come across many normal luck moments, and the bug that I will talk about today is one of those moments. That's not a deep technical knowledge bug, but it gave me this lucky feeling of being at the right place at the right time.

The Bug

The bug here is simple. I was lost on a rabbit hole, trying to explore an upload field on an **October CMS** based site on a pentest, besides some teammates here at **Conviso**, and I ended up logging out now and then. I noticed that my Burp repeater tabs with the upload post request stopped work when I log out, but since I log in again the same old session token will be reactivated.

So the flux is basically:

1. Victim logs in
2. Attacker captures victim's session cookie
3. Victim logs out
4. Session cookie no longer works
5. Victim logs in a second time
6. ORIGINAL session cookie works (Attacker is also signed in)

That's really strange behavior, the victim receives a different session token but the old disabled one becomes available again. It's important to say that, to do this attack, the attacker has to find some way to get the session token of the victim, and the token last use must be on the time gap of the October CMS framework (the default is 2 hours), so the scenario is really strict, but the simple fact that this can happen makes it a vulnerability.

Why This Happens?

Since October is an open-source platform, my teammates and I started to look for some explication about this behavior on the source code. Let's take a look at **October CMS Logout function**:

```
1  public function logout()
2  {
3      // Initialize the current auth session before trying to remove it
4      if (is_null($this->user) && !$this->check()) {
5          return;
6      }
7
8      if ($this->isImpersonator()) {
9          $this->user = $this->getImpersonator();
```

```
            $this->stopImpersonate();
            return;
        }

        if ($this->user) {
            $this->user->setRememberToken(null);
            $this->user->forceSave();
        }

        $this->user = null;

        Session::flush();
        Cookie::queue(Cookie::forget($this->sessionKey));
    }
```

The part that caught our attention was that *Session::flush();* part. October is based on Laravel, and if you look **how Laravel implements this flush function** you can see why just flush the session can cause bad behavior.

```
public function flush()
{
    $this->attributes = [];
}
```

This function call basically just cleans the attributes of the session, without invalidating it.

The Correction

My suggestion for the October CMS team was to remove the flush and call *Session:invalidate*, as we can see on the **same Laravel file**, the invalidate method calls a method named migrate passing true as an argument.

```
 public function migrate($destroy = false)
 {
     if ($destroy) {
         $this->handler->destroy($this->getId());
     }

     $this->setExists(false);

     $this->setId($this->generateSessionId());

     return true;
 }
```

Since the invalidate function passes true, the old token will be destroyed and this bug will no longer work. That's exactly what was changed on this **October commit**.

That's basically it, A huge thanks to the October team and my Conviso teammates, hope that sharing this story with you brings any kind of luck to your journey! Thanks for the reading!

References

- **CVE 2021-3311** - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-3311
- **Conviso Application Security** - https://www.convisoappsec.com/
- **October correction commit** - https://github.com/octobercms/library/commit/642f597489e6f644d4bd9a0c267e864cabead024
- **Owasp Broken Authentication** - https://owasp.org/www-project-top-ten/2017/A2_2017-Broken_Authentication