Unauthenticated Stored Cross-Site Scripting (CVE 2015-3440).

Vulnerability found in 4.2 |  Fixed in: 4.2.1

An unauthenticated attacker can inject JavaScript in WordPress comments. The script is triggered when the comment is viewed.

If triggered by a logged-in administrator, under default settings the attacker can leverage the vulnerability to execute arbitrary code on the server via the plugin and theme editors.

Summary : Part of Testing we created a post with an admin account and we used a Subcriber account to comment the post. the comment text should be long enough, it will be truncated when inserted in the database. The MySQL TEXT type size limit is 64 kilobytes so the comment has to be quite long. By entering this comment

<a title='x onmouseover=alert(unescape(/hello%20world/.source)) style=position:absolute;left:0;top:0;width:5000px;height:5000px AAAAAAAAAAAA...[64 kb]..AAA'></a> the Subscriber will create a back door to access to the database and could change the administrator's password, create new administrator accounts, or do whatever else the currently logged-in administrator can do on the target system.

Source:

https://klikki.fi/wordpress-4-2-core-stored-xss/

---

User Enumeration (CVE-2020-7918)

Summary: for the testing purpose we created 3 new users in WordPress. we used this command

wpscan --url http://192.168.33.10 --random-user-agent -e u vp to enumerate user.

We used rockyou.txt to brute force the passwords

wpscan --url http://192.168.33.10 --random-user-agent --usernames username.txt --passwords rockyou.txt

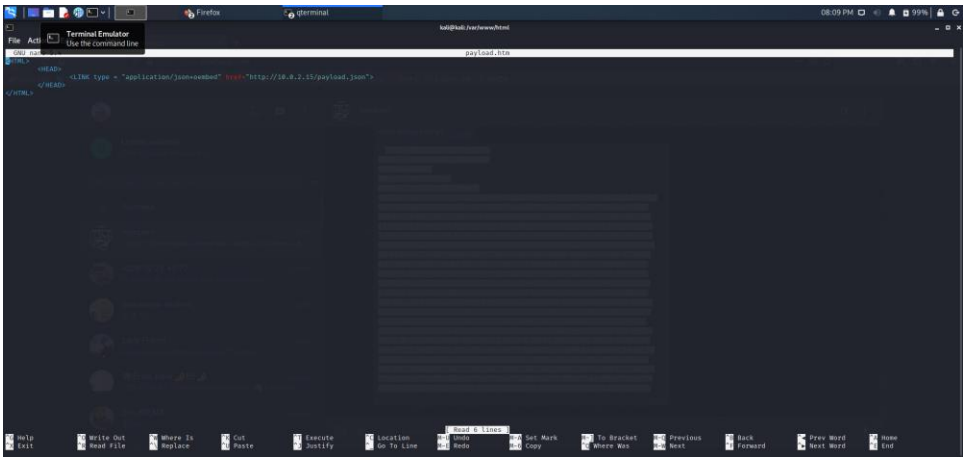
After knowing usernames and passwords you can easily login in the control panel of WordPress.

---

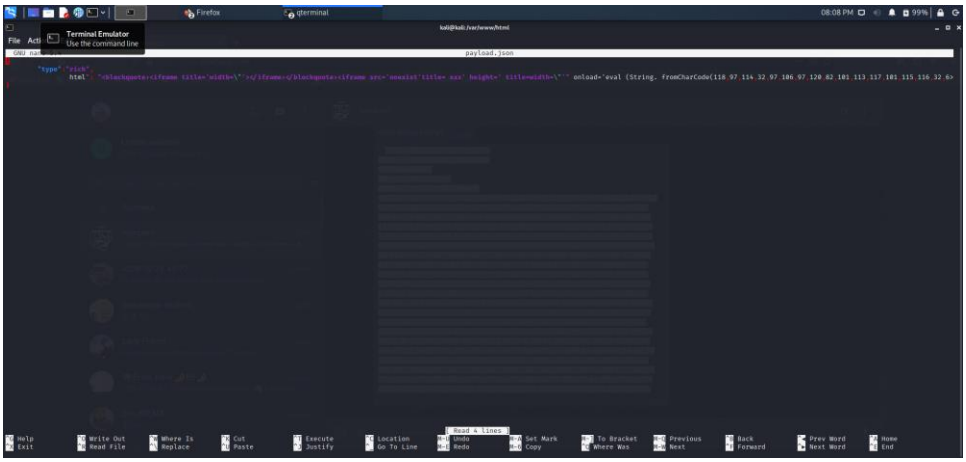
**Improper Neutralization of Script-Related HTML Tags in a Web Page (CVE-2020-4046)**

In affected versions of WordPress, users with low privileges (like contributors and authors) can use the embed block in a certain way to inject unfiltered HTML in the block editor. When affected posts are viewed by a higher privileged user, this could lead to script execution in the editor/wp-admin. This has been patched in version 5.4.2,

SUMMARY: for the testing we created a file payload.htm that we linked our script on it

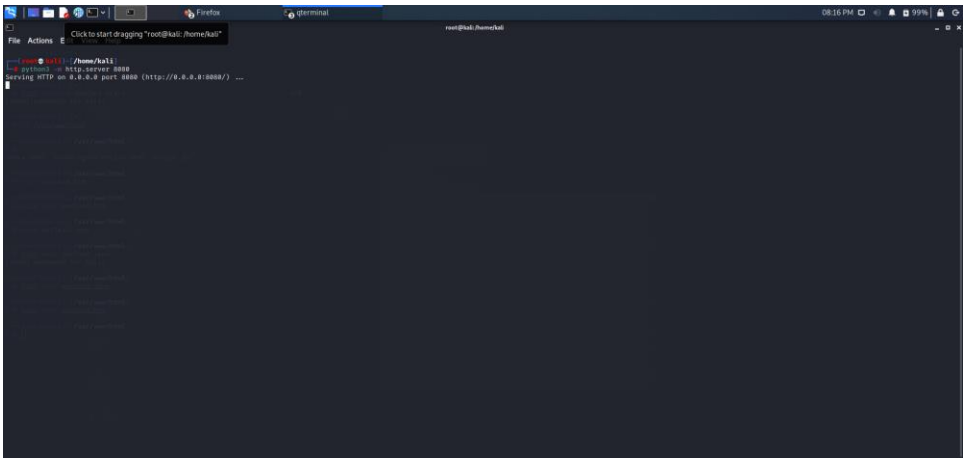


and another file payload.json

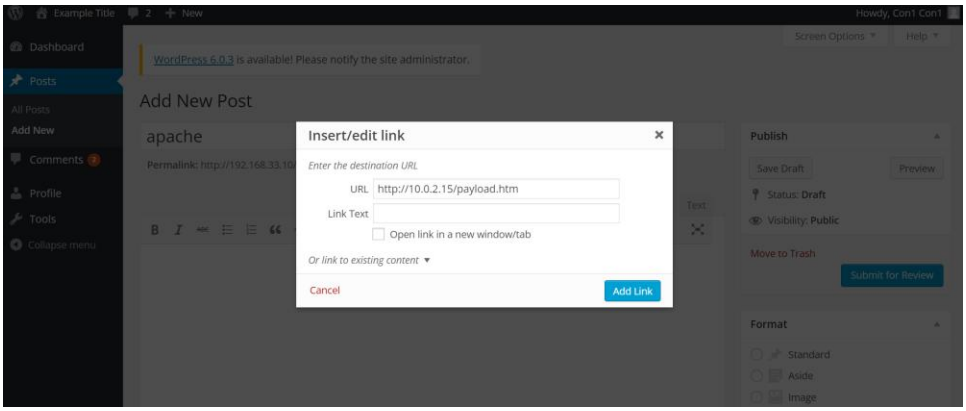


We started the local server using this command

sudo  python3  -m http.server 80



We created a post with Contributor account and link a to the payload.htm




After the administator clicked on the post the contributor could excecute some linux command on the browser using this url

http://192.168.33.10/wp-admin/wp-content/plugins/hello.php?0