New issue

Jump to bottom

Bludit V3.12.0 -- Admin File Upload vulnerability #1218

Closed whiskey-jj opened this issue on Jun 27, 2020 · 2 comments

whiskey-jj commented on Jun 27, 2020

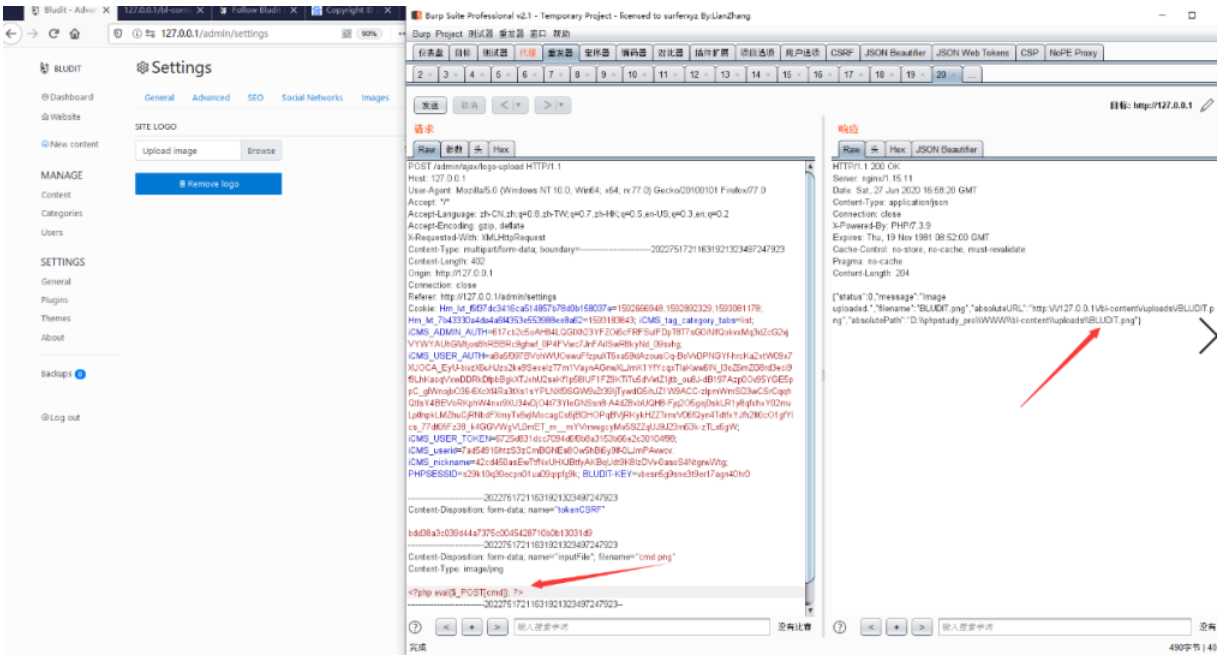
Describe your problem

A file upload vulnerability was discovered in Bludit V3.12.0
Hackers need administrator rights.
Hacker can use a backup file to control the server.

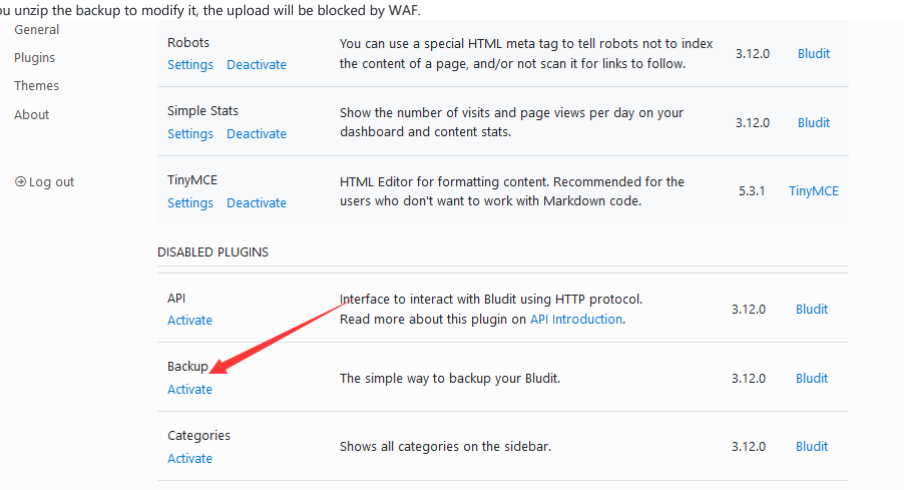
Steps to reproduce the problem

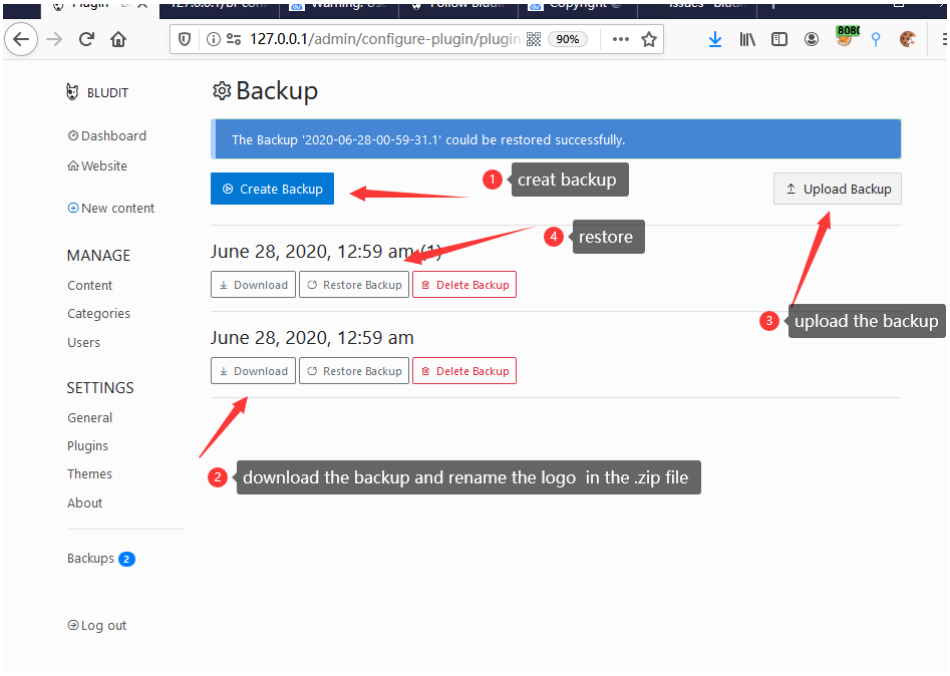
1. Download the latest version of bludit from GitHub.

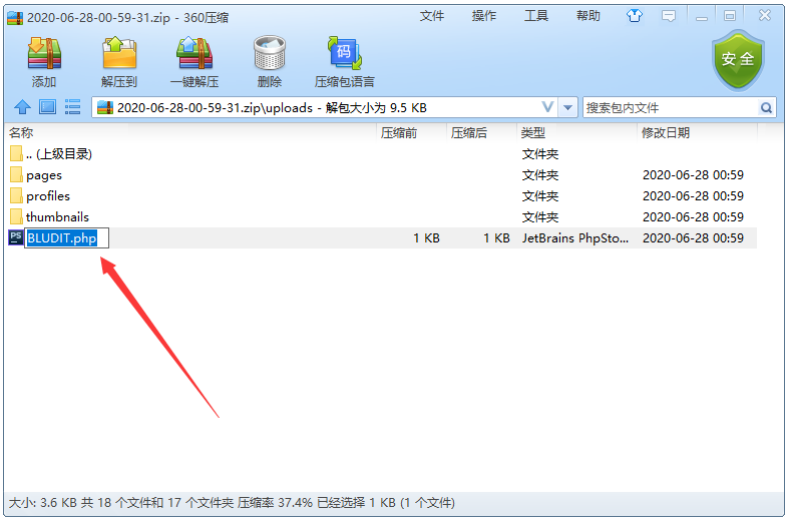
2. Using burpsuite when uploading logo in the background.
   Change picture content to PHP code
   `<?php eval($_POST[cmd]); ?>`



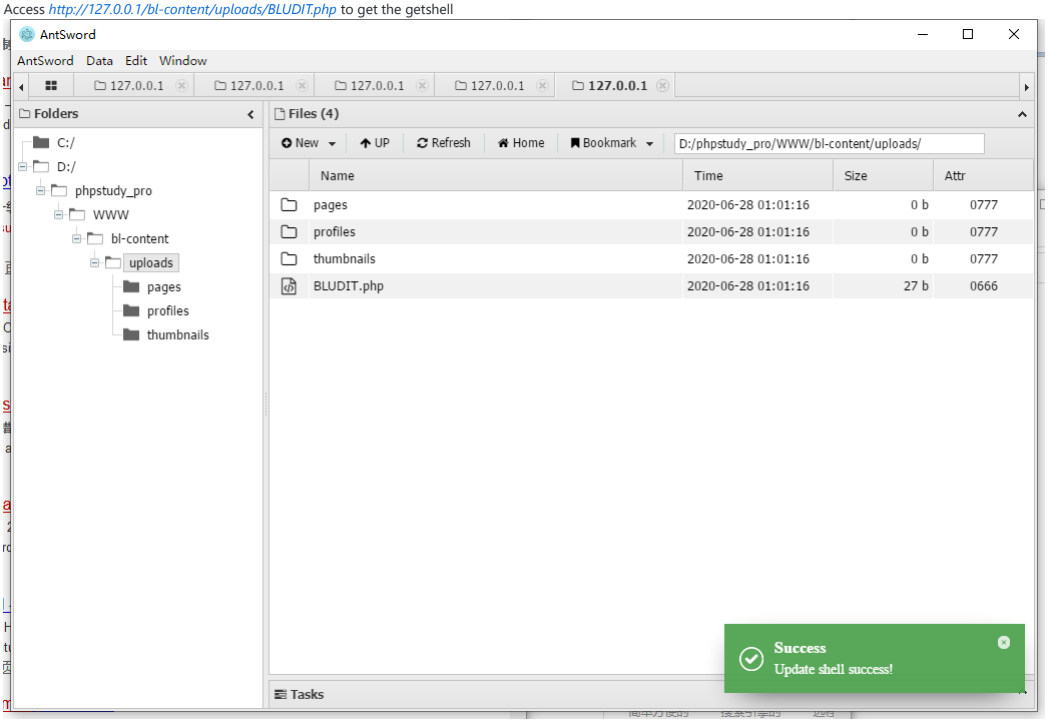
3. Enable backup plugin.Generate and download a backup file, modify the extension of logo file directly in the zip file.
   If you unzip the backup to modify it, the upload will be blocked by WAF.







4. Access *http://127.0.0.1/bl-content/uploads/BLUDIT.php* to get the getshell



Vulnerability in /bl-plugins/backup/plugin.php

```
        public function uploadBackup($backup)
        {
                global $L;

                // Check File Type
                if ($backup["type"] !== "application/zip" && $backup["type"] !== "application/x-zip-compressed") {
                        return $this->response(415, $L->get("The passed file is not a valid ZIP Archive."));
                }

                // Check File Extension
                if (stripos($backup["name"], ".zip") !== (strlen($backup["name"]) - 4)) {
                        return $this->response(415, $L->get("The passed file does not end with .zip."));
                }

                // Check ZIP extension
                if(!$this->zip) {
                        return $this->response(400, $L->get("The passed file could not be validated."));
                }

                // Validate ZIP File
                $zip = new ZipArchive();
                $zip->open($backup["tmp_name"]);
                if($zip->addEmptyDir("databases") || $zip->addEmptyDir("pages") || $zip->addEmptyDir("uploads")) {
                        $zip->close();
                        return $this->response(415, $L->get("The passed file is not a valid backup archive."));
                }
                $zip->close();

                // File Name
                $name = $backup["name"];
                $count = 0;
                while (file_exists($this->workspace() . $name)) {
                        $name = substr($backup["name"], 0, -4) . "." . ++$count . ".zip";
                }

                // Move File to Backup Directory
                Filesystem::mv($backup["tmp_name"], $this->workspace() . $name);
                return $this->response(200, $L->get("The backup file could be uploaded successfully."));
        }
}
```

We can check the image content uploaded by users.
Or just delete the backup module

Bludit version

V3.12.0

PHP version

PHP7.3.9nts

This was referenced on Jun 28, 2020

MIME Type Check for Issue #1218 and #1212 #1219

Merged

Validate Backup Files #1221

Merged

ghost commented on Jun 28, 2020

Hellow,

my solution, as submitted as 2 pull requests, is

- validating the MIME Type, next to the file extension
- adding a MD5 checksum file to the backup archive

This should fix this vulnerability.

~ Sam.

PS.: Thanks for mention Burp Suite, didn't know about this program. It's really cool. :D

👍 3

dignajar added a commit that referenced this issue on Jun 29, 2020

`Merge pull request #1219 from SamBrishes/patch-010` ··· `4282a97`

ghost mentioned this issue on Jul 4, 2020

REMOTE CODE EXCEUTION BY FILE UPLOAD IN bludit version 3.12.0 #1224

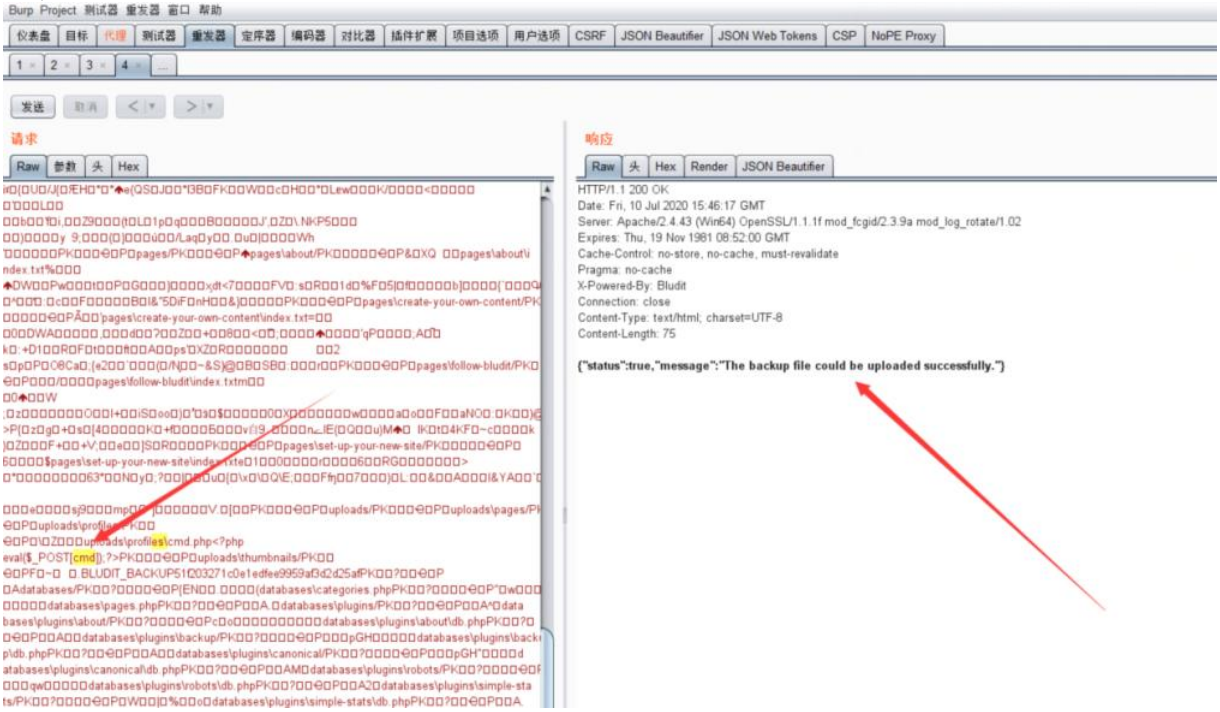
Closed

whiskey-jj commented on Jul 10, 2020 Author

> Hellow,
> my solution, as submitted as 2 pull requests, is
>
> validating the MIME Type, next to the file extension
> adding a MD5 checksum file to the backup archive

> This should fix this vulnerability.
> ~ Sam.
> PS.: Thanks for mention Burp Suite, didn't know about this program. It's really cool. :D

Hello,
I found a way to bypass the new version of the restrictions to execute files.
For MD5 detection, attackers only need to set up a local environment, add PHP files in the folder (for example,\bludit-master\bl-content\uploads\profiles), and then use the zip file generated by backup function to attack other servers.



whiskey-jj closed this as completed on Aug 25, 2020

Assignees

No one assigned

Labels

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant