Intercepting API Calls

Intercepting API calls in Android applications allows you to uncover how an app nteracts with remote servers and external resources. By mastering this technique, you will gain deep insights into the inner workings of applications, including how they send and receive data and, more importantly, how they handle sensitive information. Intercepting API calls not only helps us understand the communication between an app and its server but also a gateway to discovering potential vulnerabilities that could be exploited. These vulnerabilities—covered extensively in our API Attacks module—can include insecure data transmission, exposure of sensitive information, and flaws in authentication.

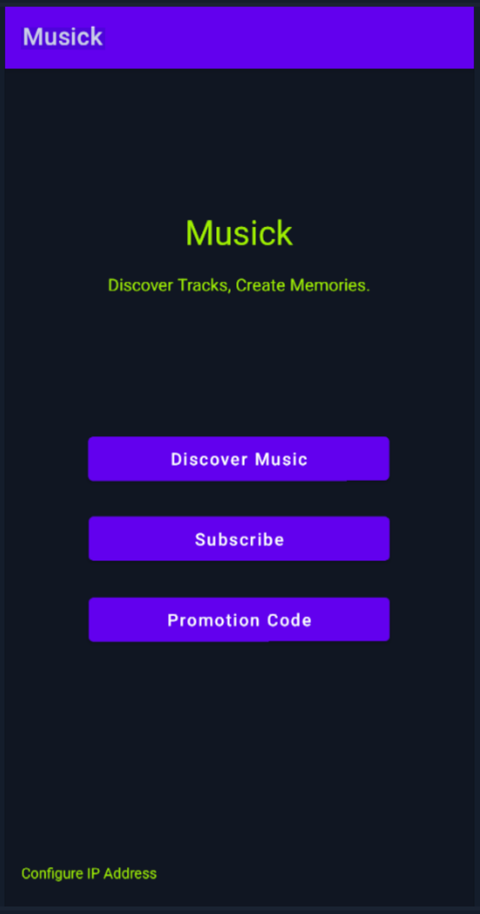
In the following example, we will use Burp Suite to intercept the network traffic of an Android application that communicates with a remote server. Burp is a tool that can be used for testing any web-based application communicating with a server, including Android apps. One of its features is including a proxy server, which we will utilize during this demonstration. In this example, we will primarily use an Android Virtual Device (AVD), though the process is compatible with any other Android device, physical or emulated. Let's connect to the device via ADB and install the application.

Intercepting API Calls

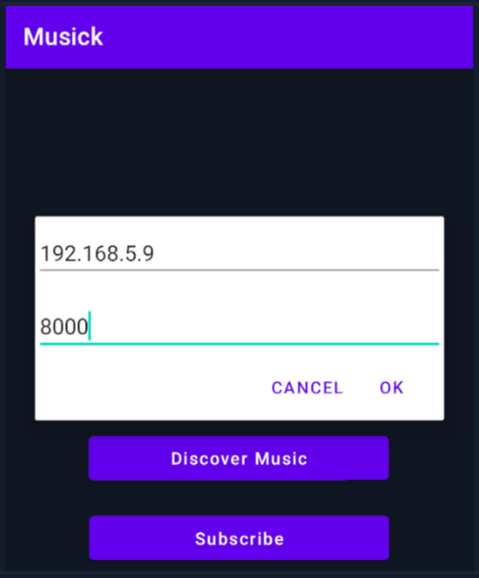
```
rl1k@htb[/htb]$ adb connect
rl1k@htb[/htb]$ adb install myapp.apk

Performing Streamed Install
Success
```

Running the application, we can see that it provides the user with three options: `Discover Music`, `Subscribe`, and `Promotion Code`.



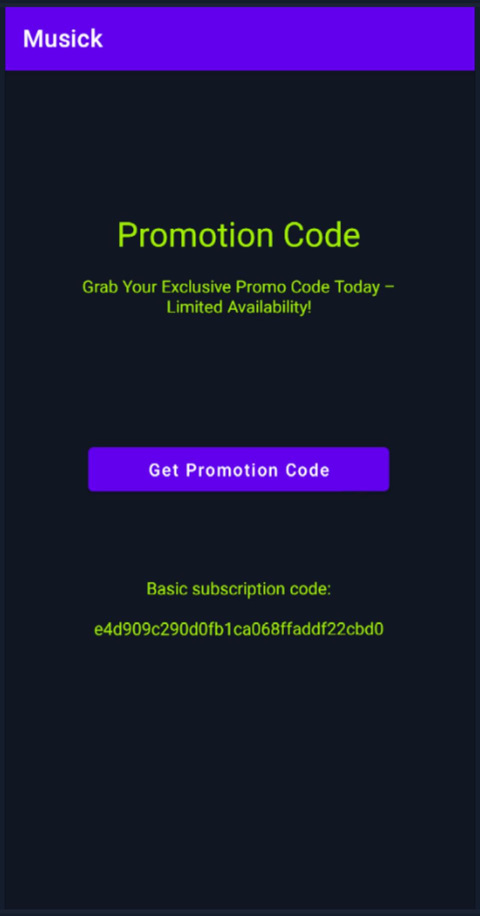
The first option displays a list of songs the user can navigate. The second option allows the user to make a new subscription, while the third allows the user to claim a promotion code. Before claiming a code, let's configure the IP address by tapping the `Configure IP Address` link on the bottom left.



192.168.5.9

8000

Once this is set, tap the `Promotion Code` button and the `Get Promotion Code`.


Musick

Promotion Code

Grab Your Exclusive Promo Code Today – Limited Availability!

Get Promotion Code

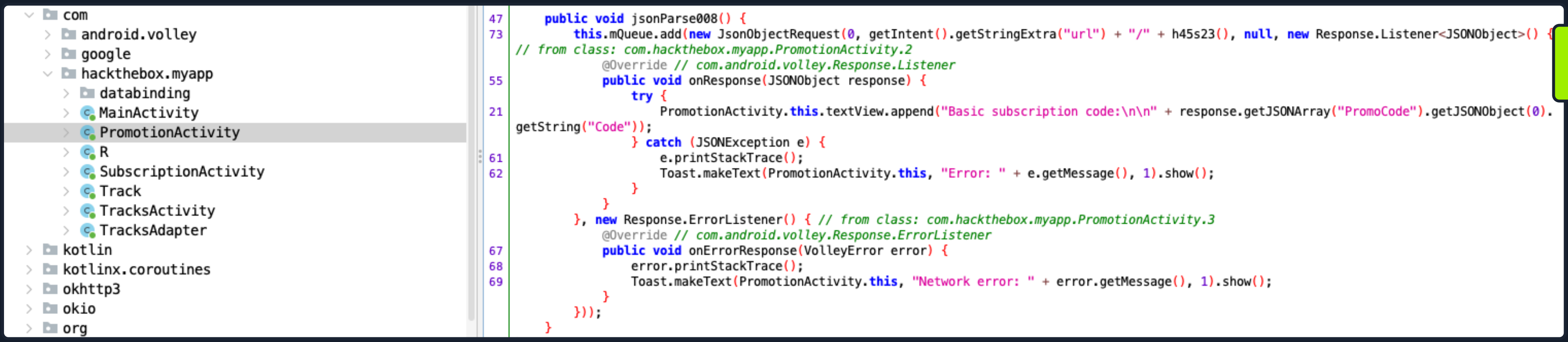
Basic subscription code:

e4d909c290d0fb1ca068ffaddf22cbd0

The code `e4d909c290d0fb1ca068ffaddf22cbd0` is displayed on the screen, indicating it's for the `Basic` subscription. This code can be used on the `Subscribe` screen to activate the `Basic` tier. Let's examine the app's source code using JADX to see whether other subscription levels are also accessible (or properly secured).

Intercepting API Calls

```
rl1k@htb[/htb]$ jadx-gui myapp.apk
```


```
com
  android.volley
  google
  hackthebox.myapp
    databinding
    MainActivity
    PromotionActivity
    R
    SubscriptionActivity
    Track
    TracksActivity
    TracksAdapter
kotlin
kotlinx.coroutines
okhttp3
okio
org
```

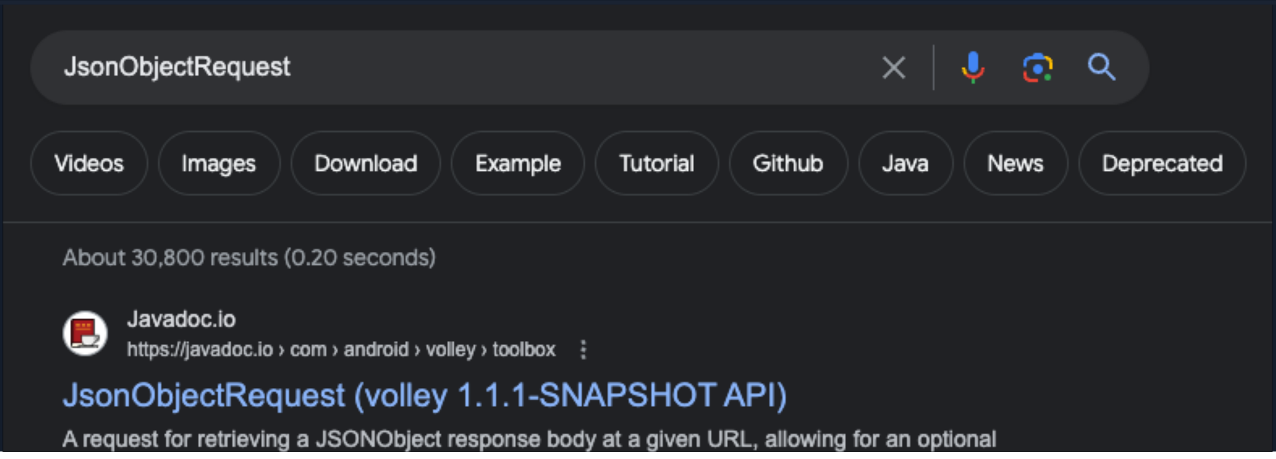
```
    public void jsonParse008() {
        this.mQueue.add(new JsonObjectRequest(0, getIntent().getStringExtra("url") + "/" + h45s23(), null, new Response.Listener<JSONObject>() {
// from class: com.hackthebox.myapp.PromotionActivity.2
            @Override // com.android.volley.Response.Listener
            public void onResponse(JSONObject response) {
                try {
                    PromotionActivity.this.textView.append("Basic subscription code:\n\n" + response.getJSONArray("PromoCode").getJSONObject(0).
getString("Code"));
                } catch (JSONException e) {
                    e.printStackTrace();
                    Toast.makeText(PromotionActivity.this, "Error: " + e.getMessage(), 1).show();
                }
            }
        }, new Response.ErrorListener() { // from class: com.hackthebox.myapp.PromotionActivity.3
            @Override // com.android.volley.Response.ErrorListener
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
                Toast.makeText(PromotionActivity.this, "Network error: " + error.getMessage(), 1).show();
            }
        }));
    }
```

Inside the `onCreate()` method of the `PromotionActivity` class, we find that the method `jsonParse008()` is invoked. This function initiates a network request to retrieve promotion codes from a JSON file hosted on a remote server, which are then displayed in the app's UI. Within this method, there's a notable line:

Code: java

```java
JsonObjectRequest(0, getIntent().getStringExtra("url") + "/" + h45s23(), null, ....
```

Here, the app appends the return value of `h45s23()` to a base URL that was passed via an `Intent` extra. This most likely references a specific JSON file containing data about available subscription types. The obfuscated function name, rather than a plain path, suggests the developer intended to hide or obscure the endpoint. Searching for `JsonObjectRequest` confirms it's part of the Volley library, commonly used for HTTP networking.



About 30,800 results (0.20 seconds)

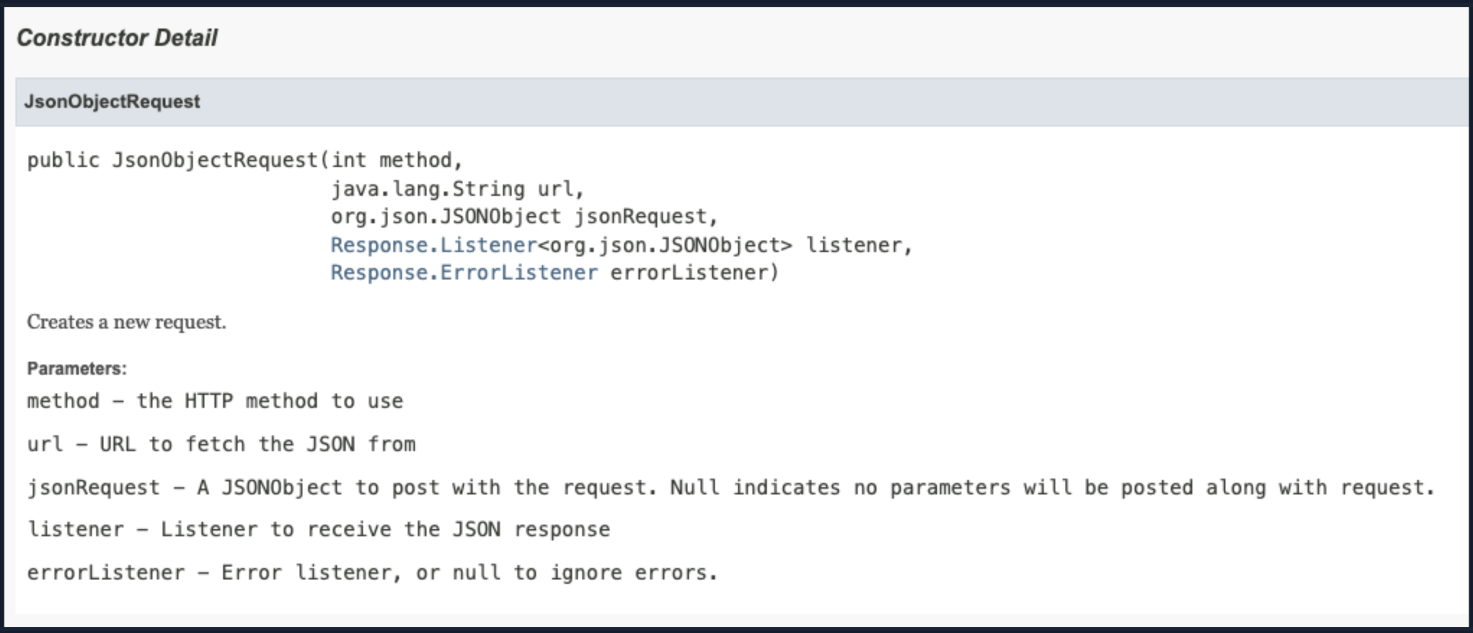
Javadoc.io
https://javadoc.io › com › android › volley › toolbox

JsonObjectRequest (volley 1.1.1-SNAPSHOT API)

A request for retrieving a JSONObject response body at a given URL, allowing for an optional

JSONObject to be passed in as part of the request body.

According to the documentation, the request uses the HTTP protocol.



Constructor Detail

JsonObjectRequest

```
public JsonObjectRequest(int method,
                         java.lang.String url,
                         org.json.JSONObject jsonRequest,
                         Response.Listener<org.json.JSONObject> listener,
                         Response.ErrorListener errorListener)
```

Creates a new request.

Parameters:

`method` - the HTTP method to use

`url` - URL to fetch the JSON from

`jsonRequest` - A JSONObject to post with the request. Null indicates no parameters will be posted along with request.

`listener` - Listener to receive the JSON response

`errorListener` - Error listener, or null to ignore errors.

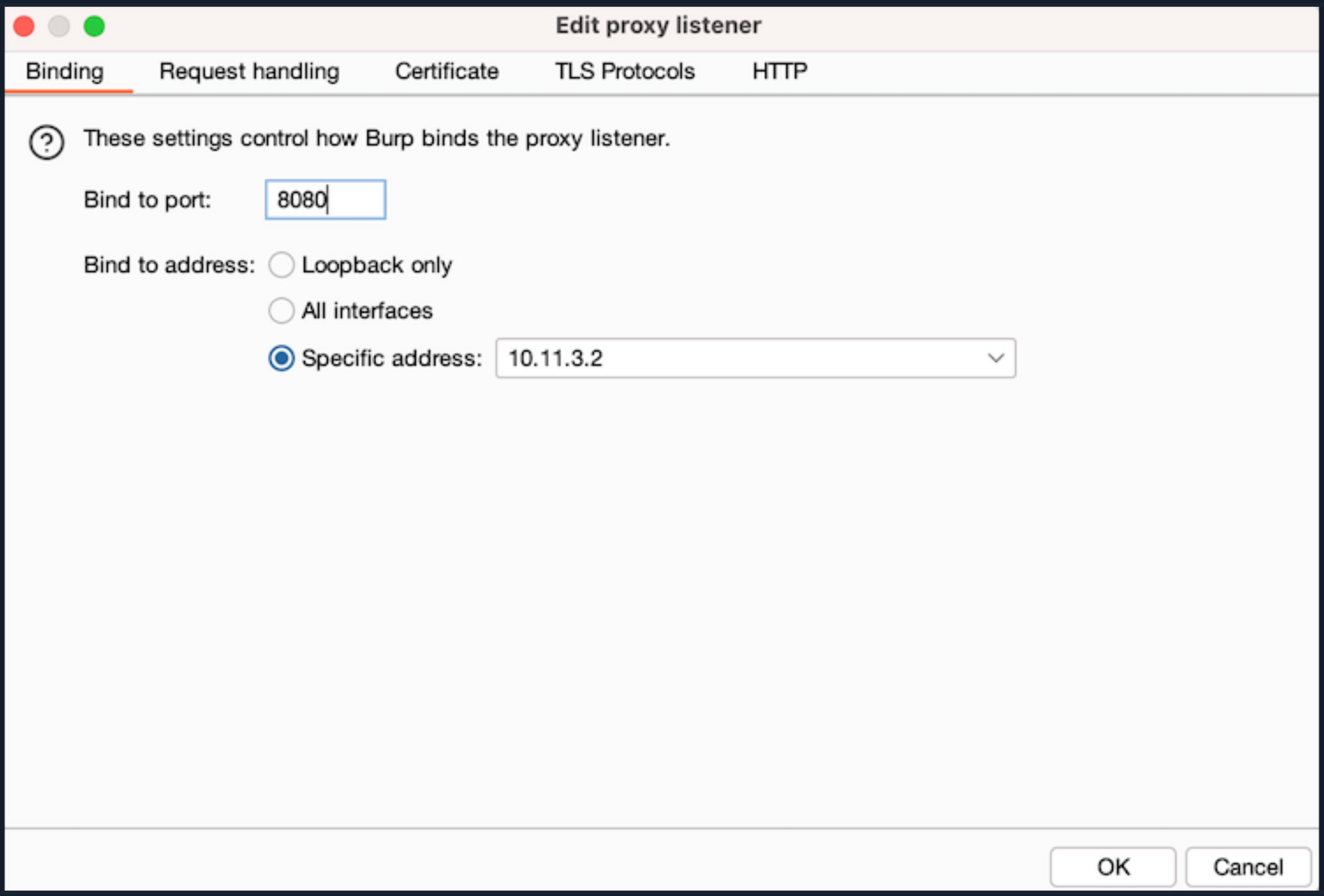
This means we can intercept and inspect the request using a tool like Burp Suite. However, to use Burp Suite with our Android Virtual Device, we first need to set up the environment. Start by identifying the IP address of your host machine, as this is where the proxy server will run.

Intercepting API Calls

```
rl1k@htb[/htb]$ ifconfig

<SNIP>
en11: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
	options=400<CHANNEL_IO>
	ether b6:30:9f:19:84:2c
	inet 10.11.3.2 netmask 0xfffffc00 broadcast 10.11.3.255
	nd6 options=201<PERFORMNUD,DAD>
	media: autoselect
	status: active
```

In this example, the host IP is `10.11.3.2`, found on the `en11` interface—though yours may differ (e.g., `eth0`, `tun0`). Next, download Burp Suite Community Edition and configure a proxy listener. Open Burp, navigate to `Proxy` → `Proxy Settings`, then under `Proxy Listeners`, select the current entry (`127.0.0.1:8080`) and click `Edit`. In the pop-up window, set the bind address to `Specific address` and enter the IP `10.11.3.2`.



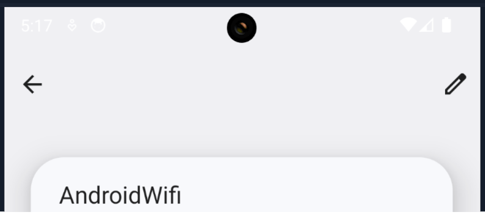
Edit proxy listener

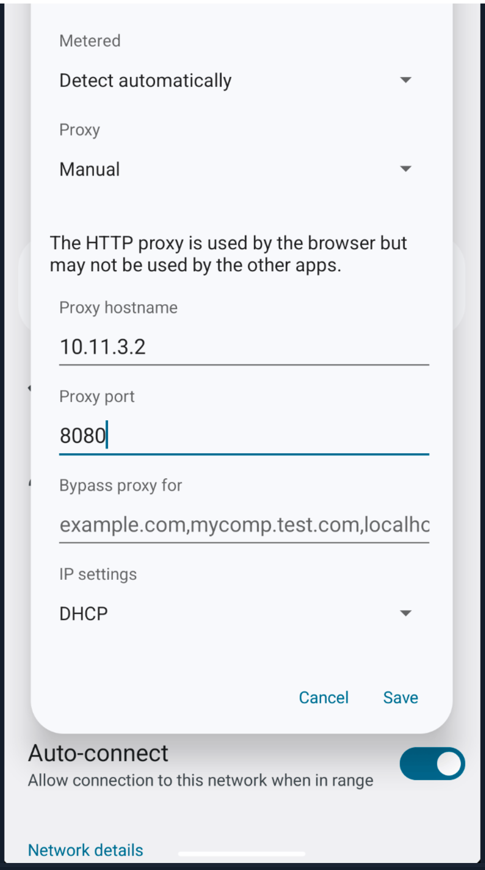
These settings control how Burp binds the proxy listener.

Bind to port: 8080

Bind to address: ( ) Loopback only
( ) All interfaces
[✓ Specific address:] 10.11.3.2

Now configure your Android Virtual Device. Go to `Settings` → `Internet` → `Android Wi-Fi`, tap the pencil icon, expand `Advanced options`, and under `Proxy`, choose `Manual`. Enter the host IP and port (e.g., `10.11.3.2:8080`).




Metered

Detect automatically

Proxy

Manual

The HTTP proxy is used by the browser but may not be used by the other apps.

Proxy hostname

10.11.3.2

Proxy port

8080

Bypass proxy for

example.com,mycomp.test.com,localhc

IP settings

DHCP

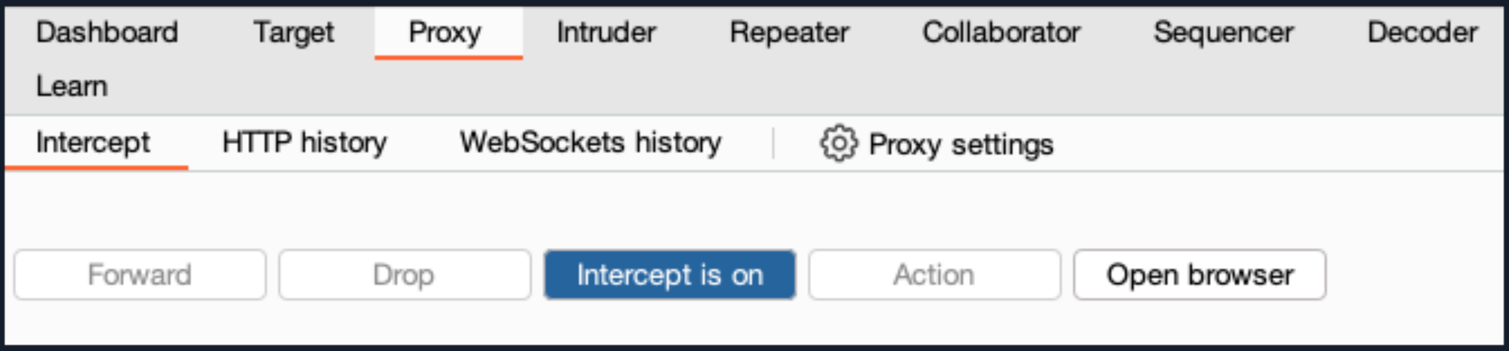
Cancel Save

Auto-connect

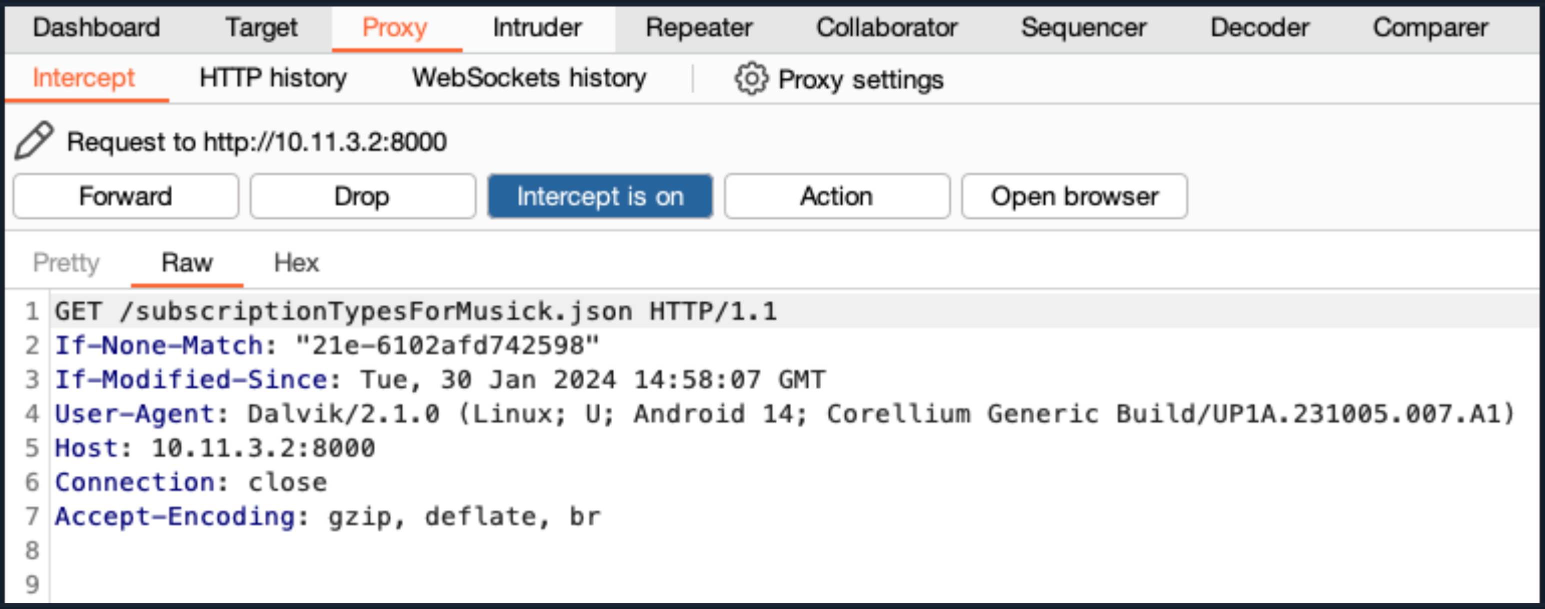
Allow connection to this network when in range

Network details

After configuring the proxy on both the host and the emulator, ensure that the `Intercept is on` button is toggled on inside Burp Suite.



Next, open the app, configure the IP and port using the in-app `Configure IP Address` option (bottom-left corner), then navigate to the `Promotion Code` screen and tap `Get Promotion Code` to send the request. Check the Burp proxy to view the captured request:

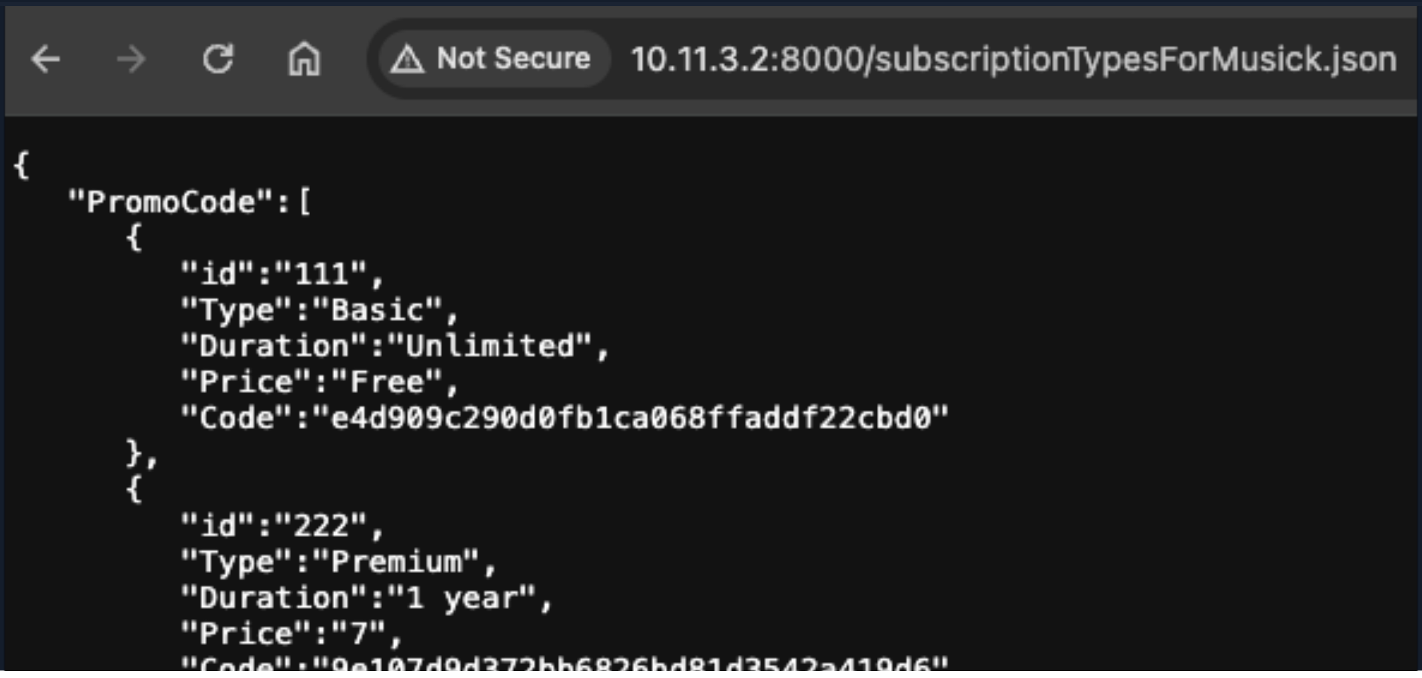


Request to http://10.11.3.2:8000

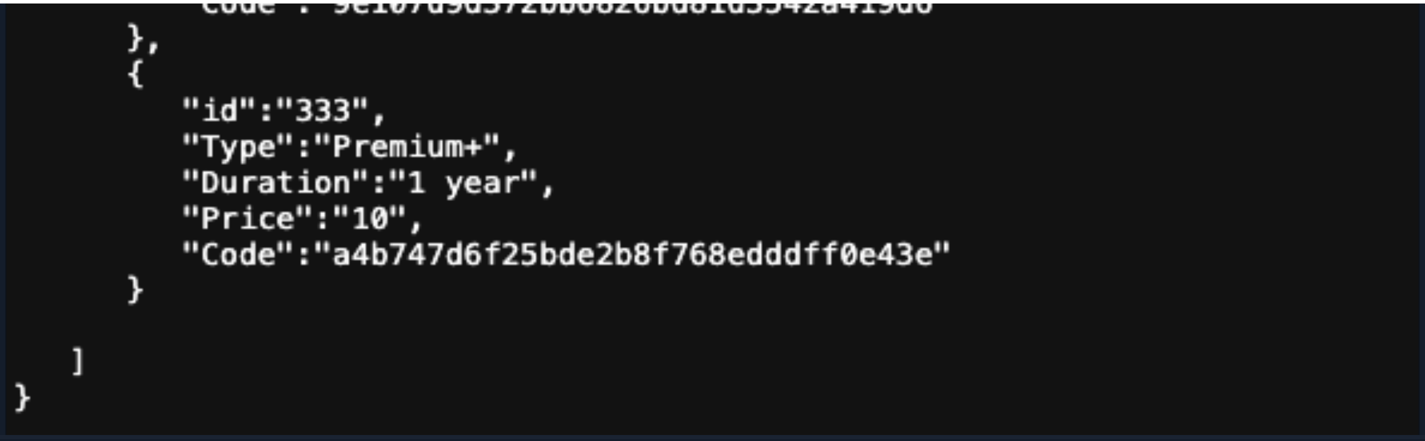
```
GET /subscriptionTypesForMusick.json HTTP/1.1
If-None-Match: "21e-6102afd742598"
If-Modified-Since: Tue, 30 Jan 2024 14:58:07 GMT
User-Agent: Dalvik/2.1.0 (Linux; U; Android 14; Corellium Generic Build/UP1A.231005.007.A1)
Host: 10.11.3.2:8000
Connection: close
Accept-Encoding: gzip, deflate, br


```

Looking at the intercepted request, we see the app trying to fetch a JSON file from a remote server. Open the URL `http://10.11.3.2:8000/subscriptionTypesForMusick.json` in your browser.



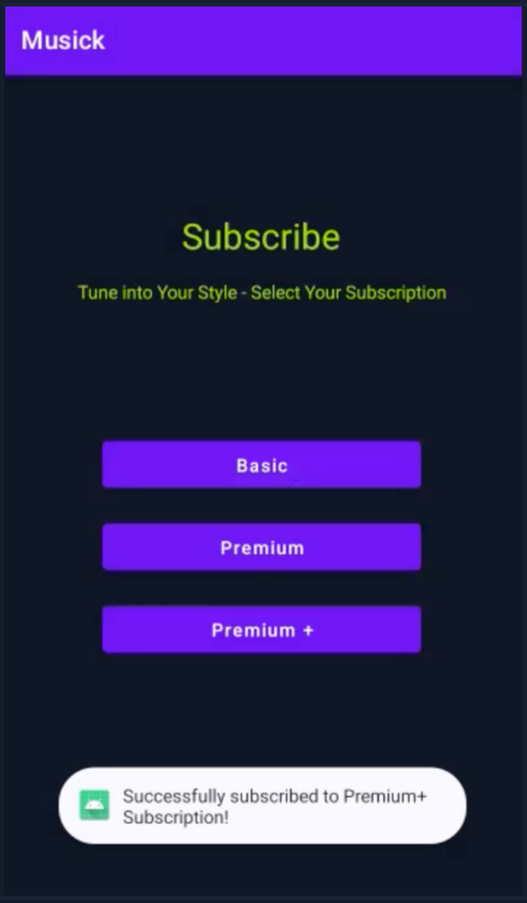
```
{
  "PromoCode":[
     {
        "id":"111",
        "Type":"Basic",
        "Duration":"Unlimited",
        "Price":"Free",
        "Code":"e4d909c290d0fb1ca068ffaddf22cbd0"
     },
     {
        "id":"222",
        "Type":"Premium",
        "Duration":"1 year",
        "Price":"7",
        "Code":"9e107d9d372bb6826bd81d3542a419d6"
```



```
            "Code":"9e107d9d372bb6826bd81d3542a419d6"
        },
        {
            "id":"333",
            "Type":"Premium+",
            "Duration":"1 year",
            "Price":"10",
            "Code":"a4b747d6f25bde2b8f768edddff0e43e"
        }

    ]
}
```

Before pasting the code into the app's `Subscribe` screen, ensure interception is disabled by clicking the `Intercept is on` button in Burp Suite so that it switches to `Intercept is off`. This prevents the proxy from blocking the request. Now, paste the code into the `Premium+` field and submit.



Successfully subscribed to Premium+ Subscription!

The process is successful, and the message `Successfully subscribed to Premium+ Subscription` is displayed in the app.

Connect to Pwnbox
Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK 27ms

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...

Enable step-by-step solutions for all questions

Questions

Cheat Sheet

Answer the question(s) below to complete this Section and earn cubes!

Target(s): Click here to spawn the target system!

+ 5 Sign up for the Premium+ subscription. Submit the flag printed to the screen as your answer.

Submit your answer here...

+10 Streak pts Submit API_calls.zip

Previous Next

Cheat Sheet

? Go to Questions

Table of Contents

Skills Assessment

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left