Challenge: Pinned

Challenge Description:

This app has stored my credentials and I can only login automatically. I tried to intercept the login request and restore my password, but this seems to be a secure connection. Can you help bypass this security restriction and intercept the password in plaintext?

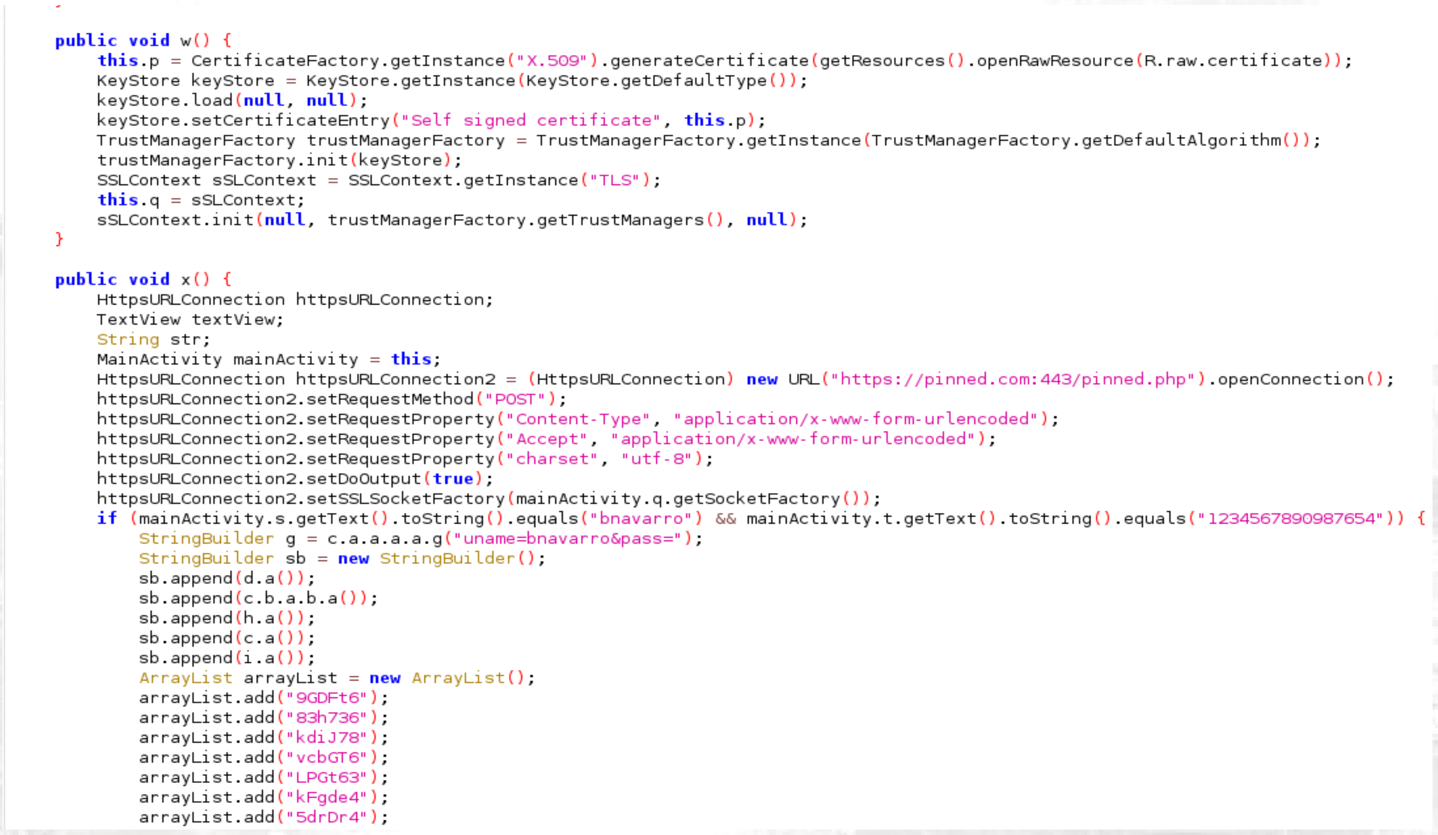
Context:

- You can either use jadx-gui for the unpacking of the apk file
- Or you can use MobSF, to run ,scan and monitor the apk file.

Challenge:

- For the use of jadx-gui after selecting the Pinned.apk file after extraction.

  With jadx-gui i look around the unpacked files and i find the username and password needed to login into the actual apk,

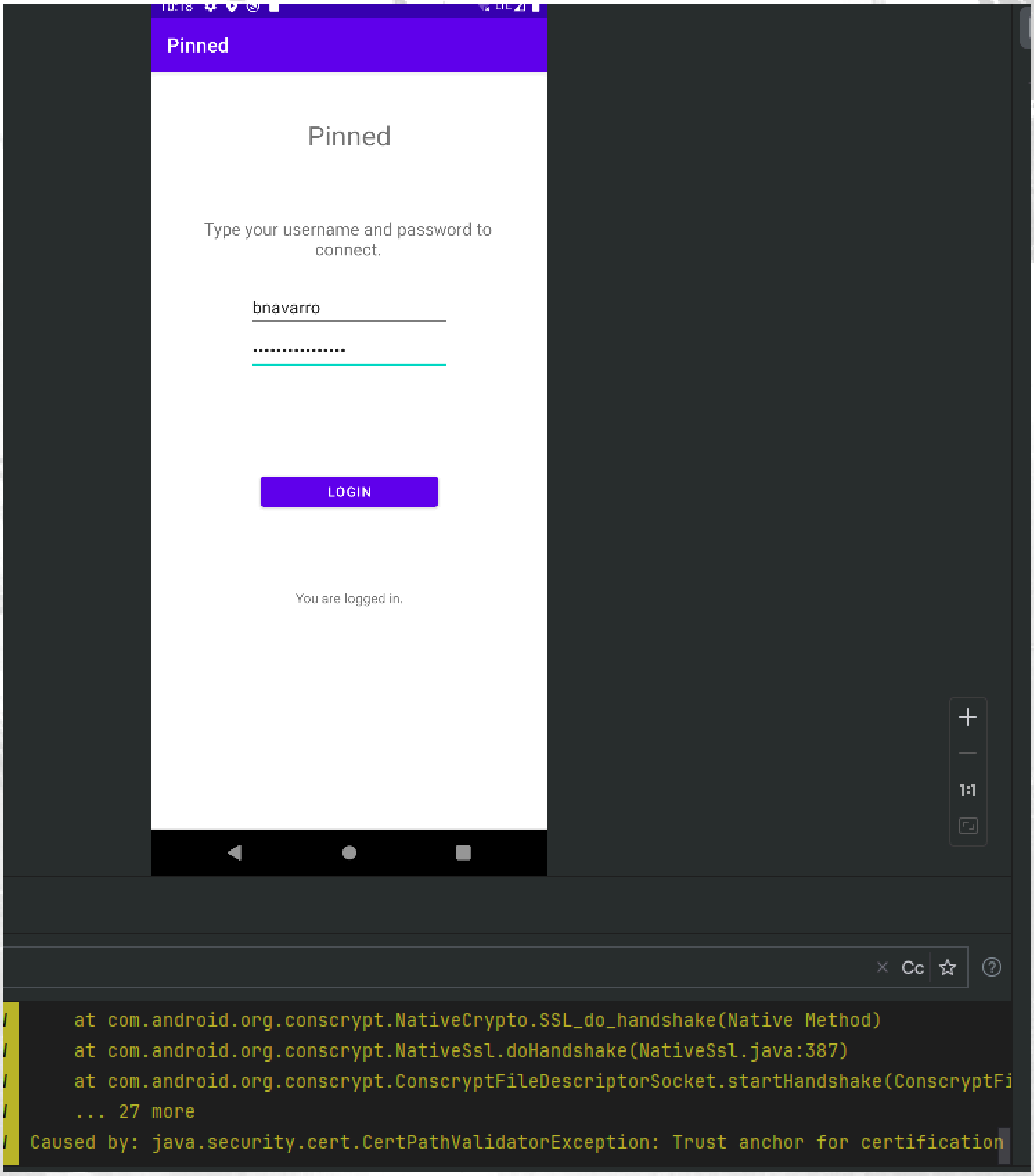


```java
public void w() {
    this.p = CertificateFactory.getInstance("X.509").generateCertificate(getResources().openRawResource(R.raw.certificate));
    KeyStore keyStore = KeyStore.getInstance(KeyStore.getDefaultType());
    keyStore.load(null, null);
    keyStore.setCertificateEntry("Self signed certificate", this.p);
    TrustManagerFactory trustManagerFactory = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
    trustManagerFactory.init(keyStore);
    SSLContext sSLContext = SSLContext.getInstance("TLS");
    this.q = sSLContext;
    sSLContext.init(null, trustManagerFactory.getTrustManagers(), null);
}

public void x() {
    HttpsURLConnection httpsURLConnection;
    TextView textView;
    String str;
    MainActivity mainActivity = this;
    HttpsURLConnection httpsURLConnection2 = (HttpsURLConnection) new URL("https://pinned.com:443/pinned.php").openConnection();
    httpsURLConnection2.setRequestMethod("POST");
    httpsURLConnection2.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
    httpsURLConnection2.setRequestProperty("Accept", "application/x-www-form-urlencoded");
    httpsURLConnection2.setRequestProperty("charset", "utf-8");
    httpsURLConnection2.setDoOutput(true);
    httpsURLConnection2.setSSLSocketFactory(mainActivity.q.getSocketFactory());
    if (mainActivity.s.getText().toString().equals("bnavarro") && mainActivity.t.getText().toString().equals("1234567890987654")) {
        StringBuilder g = c.a.a.a.a.g("uname=bnavarro&pass=");
        StringBuilder sb = new StringBuilder();
        sb.append(d.a());
        sb.append(c.b.a.b.a());
        sb.append(h.a());
        sb.append(c.a());
        sb.append(i.a());
        ArrayList arrayList = new ArrayList();
        arrayList.add("9GDFt6");
        arrayList.add("83h736");
        arrayList.add("kdiJ78");
        arrayList.add("vcbGT6");
        arrayList.add("LPGt63");
        arrayList.add("kFgde4");
        arrayList.add("5drDr4");
```

Credentials are: bnavarro:1234567890987654

- I Ran the apk with Android Studio, it works but when i try to login i keep getting a certificate error.

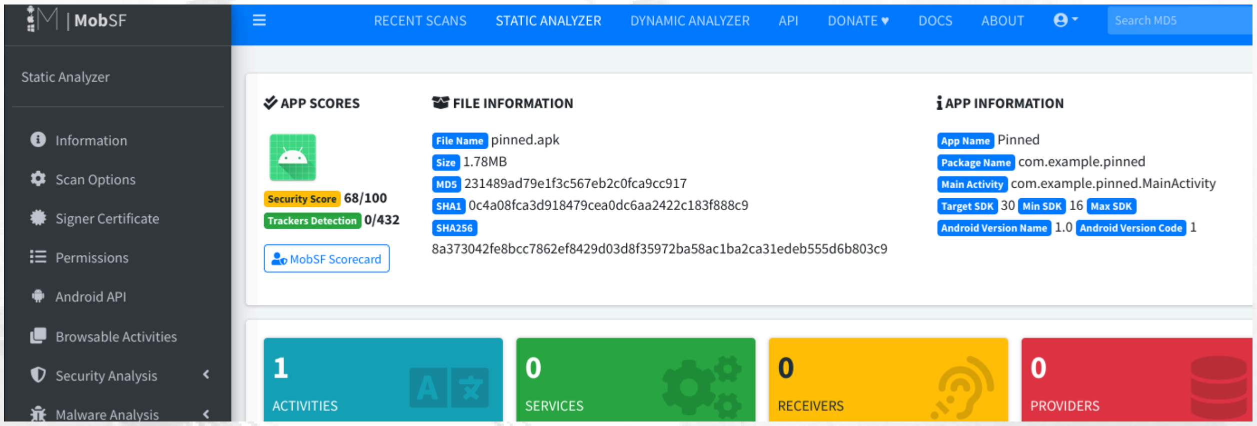


```
    at com.android.org.conscrypt.NativeCrypto.SSL_do_handshake(Native Method)
    at com.android.org.conscrypt.NativeSsl.doHandshake(NativeSsl.java:387)
    at com.android.org.conscrypt.ConscryptFileDescriptorSocket.startHandshake(ConscryptFi
    ... 27 more
Caused by: java.security.cert.CertPathValidatorException: Trust anchor for certification
```

- One way to bypass this would be to remove the certificate' [s] that are pinned on the apk.

Flag:

- Either use Frida to run the application with a Certificate unpinned script 'https://github.com/httptoolkit/frida-interception-and-unpinning'
  Or
  'https://codeshare.frida.re/@sowdust/universal-android-ssl-pinning-bypass-2/'
- I Attempted this and it was successful after the correct configuration, you should also get a flag either via burp-suite or a MITM / proxy service you might have set up.
- To use The MobSF, first install and run the setup then, or just Use docker to run it knowing it will work, then place the apk into the upload section.



APP SCORES

Security Score 68/100
Trackers Detection 0/432

FILE INFORMATION

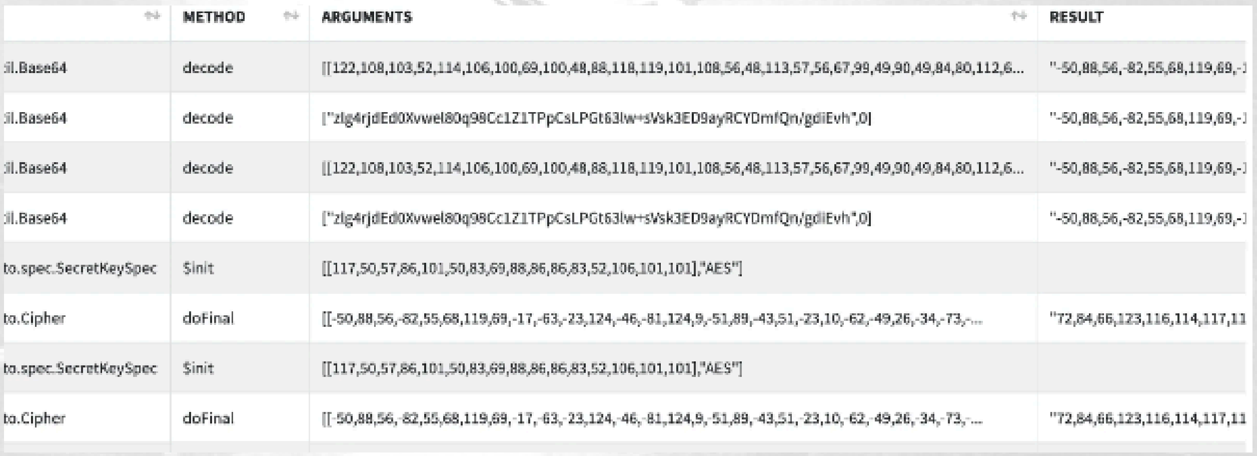
File Name pinned.apk
Size 1.78MB
MD5 231489ad79e1f3c567eb2c0fca9cc917
SHA1 0c4a08fca3d918479cea0dc6aa2422c183f888c9
SHA256 8a373042fe8bcc7862ef8429d03d8f35972ba58ac1ba2ca31edeb555d6b803c9

APP INFORMATION

App Name Pinned
Package Name com.example.pinned
Main Activity com.example.pinned.MainActivity
Target SDK 30 Min SDK 16 Max SDK
Android Version Name 1.0 Android Version Code 1

1 ACTIVITIES
0 SERVICES
0 RECEIVERS
0 PROVIDERS

- In the Second Half we first look at the static analysis results, nothing interesting in the report, we then start with the dynamic analysis and get something interesting in the Live API monitor section.

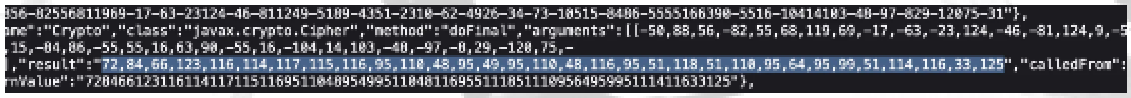


	METHOD	ARGUMENTS	RESULT
il.Base64	decode	[[122,108,103,52,114,106,100,69,100,48,88,118,119,101,108,56,48,113,57,56,67,99,49,90,49,84,80,112,6...	"-50,88,56,-82,55,68,119,69,-1
il.Base64	decode	["zlg4rjdEd0Xvwel80q98Cc1Z1TPpCsLPGt63lw+sVsk3ED9ayRCYDmfQn/gdiEvh",0]	"-50,88,56,-82,55,68,119,69,-1
il.Base64	decode	[[122,108,103,52,114,106,100,69,100,48,88,118,119,101,108,56,48,113,57,56,67,99,49,90,49,84,80,112,6...	"-50,88,56,-82,55,68,119,69,-1
il.Base64	decode	["zlg4rjdEd0Xvwel80q98Cc1Z1TPpCsLPGt63lw+sVsk3ED9ayRCYDmfQn/gdiEvh",0]	"-50,88,56,-82,55,68,119,69,-1
to.spec.SecretKeySpec	$init	[[117,50,57,86,101,50,83,69,88,86,86,83,52,106,101,101],"AES"]	
to.Cipher	doFinal	[[-50,88,56,-82,55,68,119,69,-17,-63,-23,124,-46,-81,124,9,-51,89,-43,51,-23,10,-62,-49,26,-34,-73,-...	"72,84,66,123,116,114,117,11
to.spec.SecretKeySpec	$init	[[117,50,57,86,101,50,83,69,88,86,86,83,52,106,101,101],"AES"]	
to.Cipher	doFinal	[[-50,88,56,-82,55,68,119,69,-17,-63,-23,124,-46,-81,124,9,-51,89,-43,51,-23,10,-62,-49,26,-34,-73,-...	"72,84,66,123,116,114,117,11

- After looking at the results I try to see if I can get any identification on what it might be. I finally came to the conclusion that it is Decimal (Bytes) that I need to convert to txt.

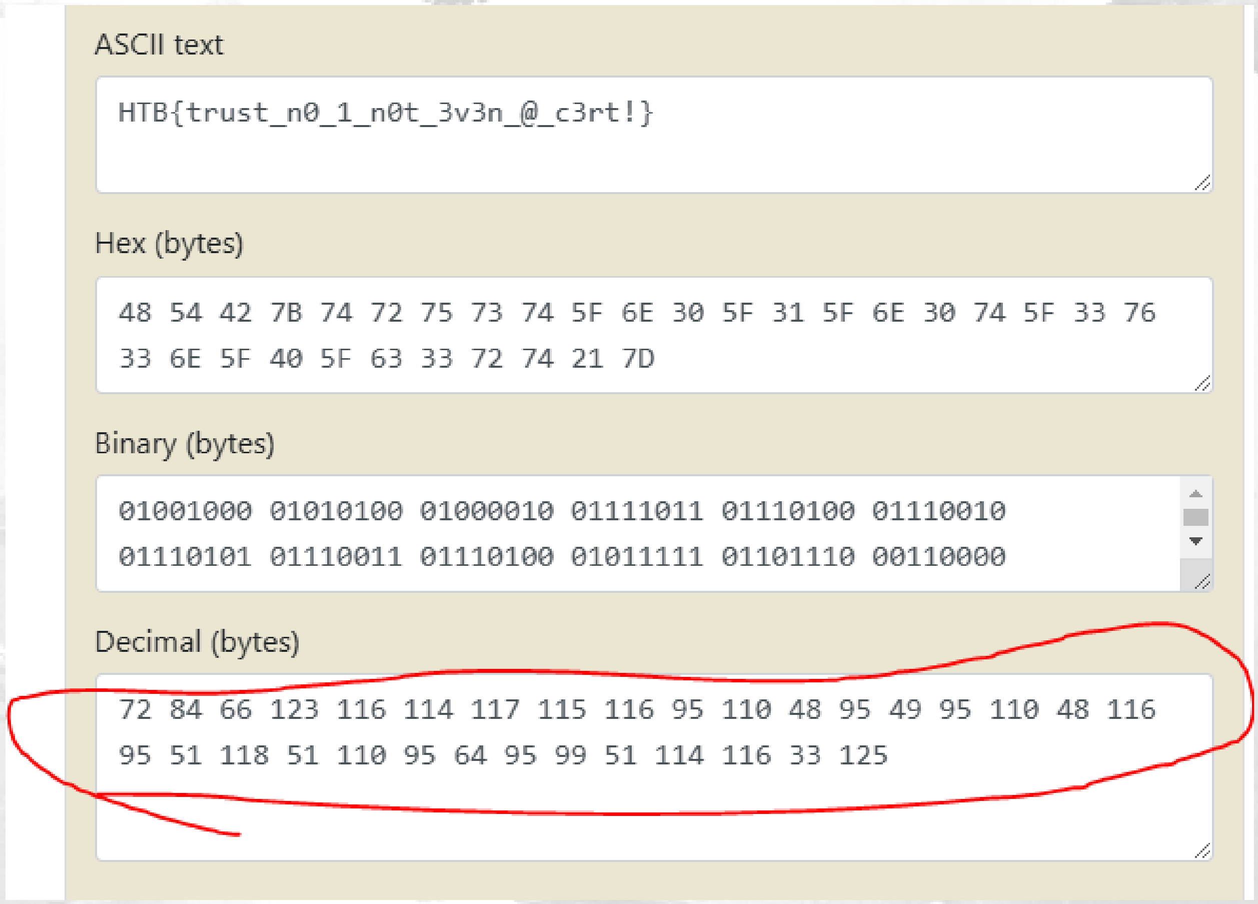
  The evidence for this is that the result of [72,84,66,123,116] converted into text will be [HTB{T],

  The full string of this strings is:



```
356-82556811969-17-63-23124-46-811249-5109-4351-2310-62-4926-34-73-10515-8406-5555166390-5516-10414103-40-97-829-12075-31"},
ame":"Crypto","class":"javax.crypto.Cipher","method":"doFinal","arguments":[[-50,88,56,-82,55,68,119,69,-17,-63,-23,124,-46,-81,124,9,-5
,15,-84,86,-55,55,16,63,90,-55,16,-104,14,103,-40,-97,-8,29,-128,75,-
],"result":"72,84,66,123,116,114,117,115,116,95,110,48,95,49,95,110,48,116,95,51,118,51,110,95,64,95,99,51,114,116,33,125","calledFrom":
nValue":"728466123116114117115116951104895499511048116955111851110956495995111411633125"},
```

  Fully Decoded we get :



ASCII text

```
HTB{trust_n0_1_n0t_3v3n_@_c3rt!}
```

Hex (bytes)

```
48 54 42 7B 74 72 75 73 74 5F 6E 30 5F 31 5F 6E 30 74 5F 33 76
33 6E 5F 40 5F 63 33 72 74 21 7D
```

Binary (bytes)

```
01001000 01010100 01000010 01111011 01110100 01110010
01110101 01110011 01110100 01011111 01101110 00110000
```

Decimal (bytes)

```
72 84 66 123 116 114 117 115 116 95 110 48 95 49 95 110 48 116
95 51 118 51 110 95 64 95 99 51 114 116 33 125
```

- We get the flag : HTB{trust_n0_1_n0t_3v3n_@_c3rt!}

  From Birdo