

mobile_security_sop

Mugikorrerako Aplikazioaren Segurtasuna Gogortzea - SOP

Helburua

Mugikorrerako aplikazioa (React Native) segurtasun estandarren arabera gogortzeko prozedura, OWASP Mobile Top 10 arauak kontuan hartuta.

Aurrebaldintzak

- React Native proiektua eskuragarria
- Android Studio eta Xcode instalatuta (berrikuspenetarako)
- MobSF (Mobile Security Framework) edo antzeko tresnak

1. Garapen Ingurune Segurua

1.1 Gradle Seguruak (Android)

Ziurtatu Gradle konfigurazioak bertsio eta SDK seguruak erabiltzen dituela.

```
android {  
    compileSdkVersion 33  
    buildToolsVersion "33.0.0"  
  
    defaultConfig {  
        minSdkVersion 21  
        targetSdkVersion 33  
        versionCode 1  
        versionName "1.0"  
    }  
  
    buildTypes {  
        release {  
            minifyEnabled true  
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'  
            debuggable false  
        }  
    }  
}
```

1.2 Info.plist Segurua (iOS)

Konfiguratu App Transport Security (ATS) HTTPS soilik baimentzeko, salbuespen zehatzekin bakarrik.

```
<key>NSAppTransportSecurity</key>  
<dict>  
    <key>NSAllowsArbitraryLoads</key>  
    <false/>  
    <key>NSExceptionDomains</key>  
    <dict>  
        <key>api.zabala-gailetak.com</key>  
    </dict>  
</dict>
```

```
<key>NSExceptionRequiresForwardSecrecy</key>
<false/>
<key>NSIncludesSubdomains</key>
<true/>
<key>NSTemporaryExceptionAllowsInsecureHTTPLoads</key>
<false/>
</dict>
</dict>
</dict>
```

2. Datu Biltegitratze Segurua

2.1 Ez Gorde Datu Sentikorrak Lokalean

- Ez gorde pasahitzik testu lauan.
- Ez gorde API gako sekreturik iturburu kodean.
- Erabili iraupen laburreko token seguruak.

2.2 AsyncStorage Enkriptatua Erabili

Ez erabili AsyncStorage estandarra datu sentikorreterako. Erabili biltegitratze enkriptatua.

```
import EncryptedStorage from 'react-native-encrypted-storage';
```

```
const storeToken = async (token) => {
  try {
    await EncryptedStorage.setItem('user_token', token);
  } catch (error) {
    console.error('Errorea tokena gordetzerakoan');
  }
};
```

```
const getToken = async () => {
  try {
    return await EncryptedStorage.getItem('user_token');
  } catch (error) {
    console.error('Errorea tokena irakurtzean');
  }
};
```

2.3 Keychain Erabili (iOS)

iOS-en, erabili Keychain zerbitzuak kredentzialak segurtasunez gordetzeko.

```
import * as Keychain from 'react-native-keychain';
```

```
const storeCredentials = async (username, password) => {
  try {
    await Keychain.setGenericPassword(username, password);
  } catch (error) {
    console.error('Errorea kredentzialak gordetzerakoan');
  }
};
```

3. Komunikazio Segurua

3.1 HTTPS Soilik Erabili

Ziurtatu API dei guztiak HTTPS bidez egiten direla eta ziurtagiri baliogabeak baztertzen direla.

```
const apiClient = axios.create({
  baseURL: 'https://zabala-gailetak.infinityfreeapp.com',
  timeout: 10000,
  httpsAgent: new https.Agent({
    rejectUnauthorized: true
  })
});
```

3.2 Certificate Pinning

Implementatu SSL Pinning-a Man-in-the-Middle (MitM) erasoak saihesteko.

```
npm install react-native-ssl-pinning
```

```
import { Pinning } from 'react-native-ssl-pinning';
```

```
Pinning.enableCertificatePinning('api.zabala-gailetak.com', ['cert1'], () => {
  console.log('Certificate pinning gaituta');
});
```

4. Autentifikazio Segurua

4.1 MFA Implementatu

Implementatu Faktore Anitzeko Autentifikazioa (MFA) segurtasuna indartzeko.

```
const handleMFA = async (token) => {
  try {
    const response = await apiClient.post('/auth/mfa/verify', { token });
  } catch (error) {
    Alert.alert('Errorea', 'MFA kodea baliogabea');
  }
};
```

4.2 Biometria Erabili

Erabili hatz-marka edo aurpegi-ezagutza sarbide azkar eta segururako.

```
import TouchID from 'react-native-touch-id';
```

```
const authenticate = async () => {
  try {
    const success = await TouchID.authenticate('Autentifikazioa behar da');
    return success;
  } catch (error) {
    console.error('Autentifikazio biometrikoak huts egin du');
    return false;
  }
};
```

5. Sarbide Kontrola

5.1 Baimenak Balidatu (Permissions)

Eskatu baimenak exekuzio-denboran eta kudeatu ezezkoak.

```
import { PermissionsAndroid, Platform } from 'react-native';
```

```
const requestCameraPermission = async () => {
  if (Platform.OS === 'android') {
```

```

try {
  const granted = await PermissionsAndroid.request(
    PermissionsAndroid.PERMISSIONS.CAMERA
  );
  return granted === PermissionsAndroid.RESULTS.GRANTED;
} catch (err) {
  console.warn(err);
  return false;
}
}
return true;
};

```

5.2 Baimenak Manifest-ean Bakarrik

AndroidManifest.xml eta Info.plist fitxategietan soilik aplikazioak benetan behar dituen baimenak zehaztu. Gutxieneko pribilegioen printzipioa jarraitu.

6. HTTPS eta Certificate Pinning Konfigurazioa (Android)

6.1 Network Security Config

Konfiguratu network-security-config.xml ziurtagiriak ainguratzeko.

```

<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config>
    <domain includeSubdomains="true">api.zabala-gailetak.com</domain>
    <pin-set>
      <pin digest="SHA-256">BASE64_CERT_HASH</pin>
    </pin-set>
  </domain-config>
</network-security-config>

```

7. Interceptor Seguruak

7.1 Axios Interceptor

Erabili interzeptoreak tokenak automatikoki gehitzeko eta 401 erroreak kudeatzeko (saioa ixteko).

```

apiClient.interceptors.request.use(async (config) => {
  const token = await getToken();
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

```

```

apiClient.interceptors.response.use(
  (response) => response,
  async (error) => {
    if (error.response?.status === 401) {
      await clearToken();
      navigation.navigate('Login');
    }
    return Promise.reject(error);
  }
);

```

8. Deep Link Segurua

8.1 Universal Links (iOS) eta App Links (Android)

- Erabili HTTPS soilik deep link-etarako.
- Balidatu URLaren jatorria datuak prozesatu aurretik.

```
import { Linking } from 'react-native';

const handleOpenURL = async (url) => {
  if (url.startsWith('https://zabala-gailetak.com/')) {
  } else {
    Alert.alert('Abisua', 'Esteka ez da segurua');
  }
};

Linking.addEventListener('url', (event) => {
  handleOpenURL(event.url);
});
```

9. Debug Modua Ezabatu Produkzioan

Ziurtatu console.log eta garapeneko beste tresnak desgaituta daudela produkzio bertsioan.

```
if (__DEV__) {
  console.log('Debug mode - ez erabili produkzioan');
} else {
  console.log = () => {};
  console.warn = () => {};
  console.error = () => {};
}
```

10. Segurtasun Auditoretza Tresnak

10.1 MobSF Erabili

Exekutatu analisi estatikoa eta dinamikoa MobSF erabiliz.

```
docker run -it -p 8000:8000 opensecurity/mobile-security-framework-mobsf
```

- Kargatu APK/IPA fitxategia.
- Analizatu txostena.
- Konpondu detektatutako ahultasunak (arrisku altukoak eta ertainak).

10.2 Frida Erabili

Erabili Frida aplikazioaren portaera dinamikoa aztertzeko eta funtzioak hook egiteko.

```
npm install -g frida
```

11. Kodearen Lausotzea (Obfuscation)

11.1 ProGuard / R8 (Android)

Gaitu kodearen murrizketa eta lausotzea build.gradle fitxategian.

```
android {
  buildTypes {
```

```

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

```

11.2 Hermes (JavaScript)

Erabili Hermes motorra JavaScript bytecode-a konpilatzeko, errendimendua hobetzeko eta iturburu kodea zuzenean ez erakusteko.

```

project.ext.react = [
    enableHermes: true
]

```

12. Probak (Test Cases)

12.1 Unitate Probak

Egiaztatu segurtasun funtzioak.

```

describe('Segurtasun Probak', () => {
    it('ez luke pasahitza testu lauan gorde behar', () => {
        const password = 'Test123456';
        const storedPassword = storePassword(password);
        expect(storedPassword).not.toBe(password);
    });
});

```

12.2 E2E Probak

Simulatu segurtasun eszenatokiak, hala nola tasa-mugatzea (rate limiting).

```

describe('Segurtasun E2E', () => {
    it('tasa-mugatzea behartu beharko luke', async () => {
        for (let i = 0; i < 20; i++) {
            await loginAttempt('test', 'wrong');
        }
        const response = await loginAttempt('test', 'wrong');
        expect(response.status).toBe(429);
    });
});

```

13. Berrikuspen eta Eguneratzeak

- Egin segurtasun probak bertsio nagusi bakoitzean.
- Exekutatu MobSF eskaneatzeak hilerok.
- Mantendu dependentziak eguneratuta (npm audit).
- Jarraitu React Native eta plataformen segurtasun buletinak.

Erreferentziak

- OWASP Mobile Top 10: <https://owasp.org/www-project-mobile-top-10/>
- OWASP Mobile Security Testing Guide: <https://owasp.org/www-project-mobile-security-testing-guide/>
- React Native Security: <https://reactnative.dev/docs/security>