Seguretat de l’app Versió Sprint 3.0

VIA – Vilanova Intelligent Airport

Index

Introducció:

En aquest informe s'indican els requeriments de seguretat aplicables a la app del projecte VIA – Vilanova Intelligent Airport (PTIN 19-2020). Aquest document s'anirà modificant al rebre una resposta de l'equip.

Actualment per fer l'app es farà servir Android Studio i java script.

Mètodes consensuats:

- L'ús d'HTTPS [Pendent d'implementar]
- Emmagatzematge intern de dades confidencials [pendent d'implementar]
- Dades personals [pendent d'implementar]
- Validació de formulari.
- Confirmació per correu electrònic.
- Un formulari de protecció de dades.

Prioritat respecte a Seguretat:

En aquest apartat hi haurà una llista del més importat a implantar de cara als següents sprints

- Validació de formulari
- Confirmació per correu electrònic.
- Un formulari de protecció de dades

Validació del formulari:

- Els camps han de ser obligatoris, no es poden deixar en blanc.
- Una adreça de correu electrònic vàlida.
- La seva contrasenya ha de tenir entre 8 i 30 caràcters i contenir una lletra majúscula, lletres minúscules, un símbol i un nombre. També seria recomanable que es vagi mostrant colors que indiquin el nivell de seguretat : groc-dèbil, taronja=mig, verd: segur)
- A més la contrasenya a d'estar encriptada.

Confirmació per correu electrònic:

S'enviarà un missatge al correu electrònic per confirmar que es la persona correcta i també per informar-li.

Formulari de protecció de dades:

Ha de tindrà una espai per marcar si s'accepta o no. En aquest formulari se li informarà al client del que es farà amb les seves dades.

Seguretat en una aplicació Android:

- **Ús d'emmagatzematge intern de dades confidencials**

Cada aplicació per Android té un directori d'emmagatzematge intern associat a ell, la trajectòria es basa en el nom del paquet de l'aplicació.

Els arxius dins d'aquest directori utilitzen la manera de creació d'arxius MODE_PRIVATE per defecte. Per tant als arxius no pot accedir qualsevol altra aplicació en el dispositiu.

Per determinar la ruta absoluta de directori d'emmagatzematge intern de la seva aplicació, es recomana que s'utilitzi el mètode getFilesDir ().

Una vegada que conegui la seva ruta, fent referència als arxius dins. Per exemple, d'aquesta forma es podria fer referència a un arxiu anomenat myfile.dat en el directori d'emmagatzematge intern de la seva aplicació:

```
File myFile = new File (getFilesDir (), "myfile.dat");
```

- **L'ús de HTTPS**

Fent ús de HTTPS, sempre que el servidor està configurat amb un certificat expedit per una autoritat de certificació de confiança, com DigiCert o GlobalSign, pot estar segur que el trànsit de la seva xarxa és segura contra l'espionatge.

- Let's encrypt

Per habilitar HTTPS s'ha d'obtenir un certificat d'una autoritat de certificació (CA). Let 's Encrypt és una CA.

Per obtenir un certificat de Let 's Encrypt, ha de demostrar el control sobre el domini.

Amb Let 's Encrypt, ho fa utilitzant un programari que utilitza el protocol ACME.

Per a veure quin mètode funcionarà millor es necessitarà saber si té accés shell (també conegut com accés SSH) al seu servidor.

- Amb accés a Shell Recomanació: Fer servir el client Certbot ACME. Pot automatitzar l'emissió i instal·lació de certificats sense temps d'inactivitat. També té maneres experts per a persones que no desitgen l'autoconfiguració. És fàcil d'usar, funciona en molts sistemes operatius.

o **No sol·licitar dades personals**

Es demana el mínim de dades necessàries i a més hi haurà perfils d'usuaris que han de garantir que les poques dades que es demanin no estiguin a disposició d'usuaris sense permís.

A menys que es tingui una bona per recollir, emmagatzemar i transmetre informació personal de l'usuari, s'ha d'evitar demanar directament en les seves aplicacions.

Es pot fer servir: la Plataforma d'Identitat Google, la qual permet als usuaris registrar-se de forma ràpida en la seva aplicació a través del compte de Google. També podeu utilitzar els serveis com Firebase que pot administrar l'autenticació d'usuari per a vostè.

Versió Sprint 2
Index

Introducció:

En aquest informe s'indican els requeriments de seguretat aplicables a la app del projecte VIA – Vilanova Intelligent Airport (PTIN 19-2020). Aquest document s'anirà modificant al rebre una resposta de l'equip.

Actualment per fer l'app es farà servir Android Studio i java script.

Mètodes consensuats:

- Emmagatzematge intern de dades confidencials [pendents d'implementar]

- L'ús de HTTPS [pendents d'implementar]

- Dades personals [pendents d'implementar]

Encara estan per consensuar:

- Ús de GCM

- Javascript

Seguretat en una aplicació Android:

o **<u>Ús d'emmagatzematge intern de dades confidencials</u>**

Cada aplicació per Android té un directori d'emmagatzematge intern associat a ell, la trajectòria es basa en el nom del paquet de l'aplicació.

Els arxius dins d'aquest directori utilitzen la manera de creació d'arxius MODE_PRIVATE per defecte. Per tant als arxius no pot accedir qualsevol altra aplicació en el dispositiu.

Per determinar la ruta absoluta de directori d'emmagatzematge intern de la seva aplicació, es recomana que s'utilitzi el mètode getFilesDir ().

Una vegada que conegui la seva ruta, fent referència als arxius dins. Per exemple, d'aquesta forma es podria fer referència a un arxiu anomenat myfile.dat en el directori d'emmagatzematge intern de la seva aplicació:

```
File myFile = new File (getFilesDir (), "myfile.dat");
```

o **L'ús de HTTPS**

Fent ús de HTTPS, sempre que el servidor està configurat amb un certificat expedit per una autoritat de certificació de confiança, com DigiCert o GlobalSign, pot estar segur que el trànsit de la seva xarxa és segura contra l'espionatge.

- Let's encrypt

Per habilitar HTTPS s'ha d'obtenir un certificat d'una autoritat de certificació (CA). Let 's Encrypt és una CA.

Per obtenir un certificat de Let 's Encrypt, ha de demostrar el control sobre el domini.

Amb Let 's Encrypt, ho fa utilitzant un programari que utilitza el protocol ACME.

Per a veure quin mètode funcionarà millor es necessitarà saber si té accés shell (també conegut com accés SSH) al seu servidor.

● Amb accés a Shell Recomanació: Fer servir el client Certbot ACME. Pot automatitzar l'emissió i instal·lació de certificats sense temps d'inactivitat. També té maneres experts per a persones que no desitgen l'autoconfiguració. És fàcil d'usar, funciona en molts sistemes operatius.

o **Ús de GCM**

GCM, abreviatura de Google Cloud Messaging, "Google missatgeria al núvol", serveix per enviar dades dels servidors per a les seves aplicacions.

A més serveix per enviar missatges a una aplicació, ja que totes les comunicacions GCM estan encriptades. Aquests s'autentiquen utilitzant fitxes de registre, s'actualitza de forma regular al costat del client i una clau d'API única al costat de servidor.

o **No sol·licitar dades personals**

Es demana el mínim de dades necessàries i a més hi haurà perfils d'usuaris que han de garantir que les poques dades que es demanin no estiguin a disposició d'usuaris sense permís.

A menys que es tingui una bona per recollir, emmagatzemar i transmetre informació personal de l'usuari, s'ha d'evitar demanar directament en les seves aplicacions.

Es pot fer servir: la Plataforma d'Identitat Google, la qual permet als usuaris registrar-se de forma ràpida en la seva aplicació a través del compte de Google. També podeu utilitzar els serveis com Firebase que pot administrar l'autenticació d'usuari per a vostè.

o Javascript

En programació sempre és molt important tenir en compte la seguretat a l'hora d'escriure el codi.

En javascript necessitem evitar la injecció directa en el codi. El més adequat és crear elements DOM de manera programàtica i anar-los inserint en el DOM.

Versió Sprint 1

Index

Introducció:

En aquest informe s'indican els requeriments de seguretat aplicables a la app del projecte VIA – Vilanova Intelligent Airport (PTIN 19-2020).

Seguretat en una aplicació Android:

- **Ús d'emmagatzematge intern de dades confidencials**

Cada aplicació per Android té un directori d'emmagatzematge intern associat a ell, la trajectòria es basa en el nom del paquet de l'aplicació.

Els arxius dins d'aquest directori utilitzen la manera de creació d'arxius MODE_PRIVATE per defecte. Per tant als arxius no pot accedir qualsevol altra aplicació en el dispositiu.

Per determinar la ruta absoluta de directori d'emmagatzematge intern de la seva aplicació, es recomana que s'utilitzi el mètode getFilesDir ().

Una vegada que conegui la seva ruta, fent referència als arxius dins. Per exemple, d'aquesta forma es podria fer referència a un arxiu anomenat myfile.dat en el directori d'emmagatzematge intern de la seva aplicació:

```
File myFile = new File (getFilesDir (), "myfile.dat");
```

- **Xifrar les dades en emmagatzematge extern**

Com la capacitat d'emmagatzematge intern d'un dispositiu Android pot ser limitada a vegades, és possible que no hi hagi més opció que emmagatzemar dades sensibles en mitjans d'emmagatzematge externs, com ara una targeta SD extraïble.

A causa de que les dades en mitjans d'emmagatzematge extern es pot accedir directament pels usuaris i altres aplicacions en el dispositiu, és important que s'emmagatzemi en un format codificat.

Algoritmes de xifrat: AES, sigles d'Advanced Encryption Standard, amb una mida de clau de 256 bits.

Quan administris dades des de l'emmagatzematge extern, hauries de fer una validació d'entrada. No hauries emmagatzemar arxius executables ni de classe en un mitjà d'emmagatzematge extern abans de realitzar una càrrega dinàmica.

Si l'app obté arxius executables de mitjans d'emmagatzematge extern, els arxius haurien d'estar signats i verificats criptogràficament abans de la càrrega dinàmica

- **L'ús de HTTPS**

Fent ús de HTTPS, sempre que el servidor està configurat amb un certificat expedit per una autoritat de certificació de confiança, com DigiCert o GlobalSign, pot estar segur que el trànsit de la seva xarxa és segura tant contra l'espionatge i els atacs "man in the middle".

- **Ús de GCM**

GCM, abreviatura de Google Cloud Messaging, "Google missatgeria al núvol", serveix per enviar dades dels servidors per a les seves aplicacions.

A més serveix per enviar missatges a una aplicació, ja que totes les comunicacions GCM estan encriptades. Aquests s'autentiquen utilitzant fitxes de registre, s'actualitza de forma regular al costat del client i una clau d'API única al costat de servidor.

- **No sol·licitar dades personals**

A menys que es tingui una bona per recollir, emmagatzemar i transmetre informació personal de l'usuari, s'ha d'evitar demanar directament en les seves aplicacions.

Es pot fer servir: la Plataforma d'Identitat Google, la qual permet als usuaris registrar-se de forma ràpida en la seva aplicació a través del compte de Google. També podeu utilitzar els serveis com Firebase que pot administrar l'autenticació d'usuari per a vostè.

- **Abans Publicar**

Les mesures de seguretat integrades en una aplicació per Android es poden comprometre si els atacants són capaços d'arribar al codi font.

Abans de publicar la seva aplicació: es pot fer utilizar Proguard, que s'inclou en el SDK d'Android i serveix per ofuscar i compactar el codi font.

Android Studio:

Android Studio inclou automàticament Proguard en el procés de construcció si el buildtype està llest per llançar.

La configuració per defecte Proguard es troben disponibles a l'arxiu Proguard-android.txt de l'SDK d'Android.

- **Afegir regles personalitzades per a la configuració:** Es pot fer dins d'un arxiu anomenat proguard-rules.pro

<module-dir>/proguard-rules.pro

Quan crees un mòdul nou amb Android Studio, l'IDE crea un arxiu proguard-rules.pro al directori arrel d'aquest mòdul.

Per defecte, aquest arxiu no aplica cap regla. Per tant, s'ha d'incloure les aquí les regles Proguard.

Seguretat en MongoDB:

o **Habilitar auth**

Ofereix la "Defensa en profunditat" si la xarxa està en perill. Edita el seu mongodb fitxer de configuració per habilitar l'autenticació.

o **L'ús de tallafocs**

Utilitzar tallafocs per a restringir que altres entitats se'ls permeti connectar al servidor de MongoDB.

- Si estas allotjat en AWS es pot utilitzar grups de seguretat 'per a restringir l'accés.
- Si estas allotjat en un proveïdor que no admet tallafocs es pot configurar utilitzant "iptables".

o **Habilitar SSL**

Utilitza SSL per evitar que les seves dades viatgen entre el seu client Mongo i el servidor Mongo sense xifrar.

Seguretat en el núvol:

o **establir contrasenyes**

o **xifrar dades**