IN2000

Sikker systemutvikling i Android

Jamila Mehmandarova

Agenda

- Hvorfor sikkerhet?
  - Sikkerhet på mobile enheter
- Operativsystemet Android
  - Input-validering
  - Kryptografi
  - Autentisering og Autorisering
  - Nettverkskommunikasjon
  - Interaksjon med Mobilplattformen
  - Kodekvalitet og Exploit-Mitigation
  - Anti-Tampering og Anti-Reversing
- Mobilutvikling generelt
  - Sandboxing
- Privacy
- Et eksempel på dårlig håndtert sikkerhet

Hvorfor sikkerhet?

Kilde: https://www.androidauthority.com/fix-samsung-galaxy-a-series-2024-3391657/

OWASP Top Ten != OWASP Mobile Top 10

- Mobilapper kjører i andre miljøer enn web-apper og har derfor andre risikoer (selv om noen er det samme)
- M1: Improper Credential Usage
- M2: Inadequate Supply Chain Security
- M3: Insecure Authentication/Authorization
- M4: Insufficient Input/Output Validation
- M5: Insecure Communication
- M6: Inadequate Privacy Controls
- M7: Insufficient Binary Protections
- M8: Security Misconfiguration
- M9: Insecure Data Storage
- M10: Insufficient Cryptography

Sikkerhet (på mobile enheter)

- Så mye mer enn å sette “et godt passord”
- Man kan bruke kunnskap innenfor f.eks.
  - mobile operativsystemer (Android, iOS)
  - Programmering
  - Nettverksikkerhet
  - Kryptografi
  - Masse mer
- Sikkerhet strekker mye bredere enn kun de tingene man lager som f.eks.
  - Organisatorisk, hvordan håndterer man ting?
  - Informasjon, hva skal vi informere om når noe skjer?
  - Brukbarhet, hvordan kan vi få ting sikkert men også mulig å bruke?
- Det finnes ingen “enkeltkilde” med ting man kan teste for at “appen skal være sikker”

Sikkerhetskomponentene i Android-utvikling

- Input-validering
- Kryptografi
- Autentisering og Autorisering
- Nettverkskommunikasjon
- Interaksjon med Mobilplattformen
- Kodekvalitet og Exploit-Mitigation
- Anti-Tampering og Anti-Reversing

Sikkerhetskomponentene i Android-utvikling

- Input-validering:
  - Prosessen med å sjekke dataene som brukerne legger inn i appen din. Målet er å sikre at dataene er korrekte og ikke inneholder noe skadelig, som kan føre til feil i appen eller sikkerhetsbrudd. Det er som en dørvelger som bare slipper inn de riktige folkene i en klubb.
- Kryptografi:
  - Kryptografi bruker komplekse matematiske algoritmer for å sikre informasjon. Det omdanner data til en kode som ikke kan leses uten en spesiell nøkkel. Dette er viktig for å beskytte sensitiv informasjon, som passord eller personlige meldinger, både når de lagres på enheten og når de sendes over internett.
- Autentisering og Autorisering:
  - Autentisering handler om å bekrefte brukerens identitet, vanligvis ved hjelp av noe de vet (som et passord), noe de har (som en mobiltelefon), eller noe de er (som et fingeravtrykk). Autorisering finner sted etter autentisering og bestemmer hvilke data eller områder i appen brukeren har tilgang til, basert på deres tillatelser.
- Nettverkskommunikasjon:
  - Omhandler sikker overføring av data mellom appen din og servere over internett. Det inkluderer bruk av sikre kommunikasjonsprotokoller som HTTPS, som krypterer data under overføring for å hindre avlytting eller manipulering. Nettverkssikkerhet er også en nøkkelkomponent. I en verden der mobile enheter konstant er koblet til internett, er det avgjørende å forstå hvordan data overføres, og hvordan man kan beskytte disse dataene mot uautorisert tilgang og andre trusler.
- Interaksjon med Mobilplattformen:
  - Fokuserer på hvordan appen din bruker og samhandler med systemressurser og -tjenester på Android-enheter. Det inkluderer å be om og håndtere tillatelser på en sikker måte, slik at appen bare får tilgang til de nødvendige ressursene (som kontakter, kamera, etc.) og unngår overflødige tillatelser som kan utgjøre sikkerhetsrisikoer.
- Kodekvalitet og Exploit-Mitigation:
  - Å opprettholde høy kodekvalitet betyr å skrive kode som er ren, godt organisert, og fri for feil. Dette reduserer sjansen for sikkerhetsproblemer. Exploit-mitigation innebærer å implementere spesifikke sikkerhetstiltak for å beskytte appen mot kjente typer angrep, som bufferoverløp eller SQL-injeksjoner.
- Anti-Tampering og Anti-Reversing:
  - Anti-tampering teknikker sikrer at appen din ikke kan endres eller manipuleres av en angriper etter at den er utgitt. Dette kan inkludere å signere appen din med et sikkert sertifikat. Anti-reversing er rettet mot å forhindre at hackere 'dekomponerer' appen din for å forstå eller stjele koden. Dette kan involvere å obfuskere koden, noe som gjør den vanskelig å lese og forstå for mennesker og maskiner.

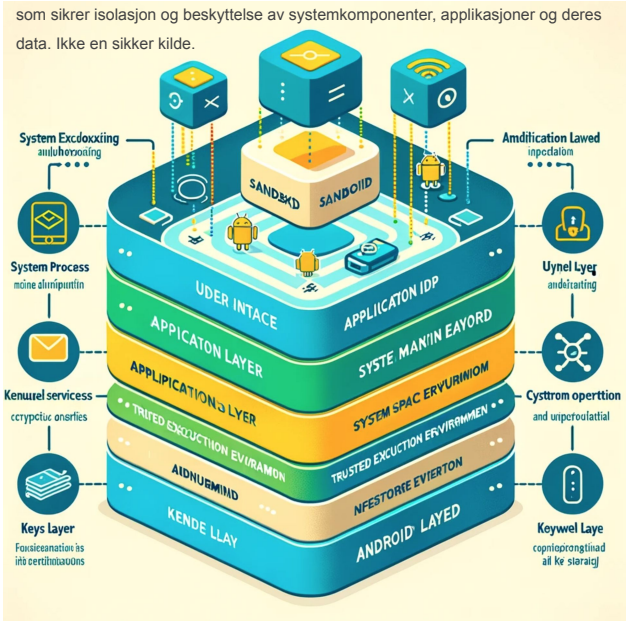
Sikkerhet (på mobile enheter)

- Så mye mer enn å sette “et godt passord”
- Man kan bruke kunnskap innenfor f.eks.
  - mobile operativsystemer (Android, iOS)
  - Programmering
  - Nettverksikkerhet
  - Kryptografi
  - Masse mer
- Sikkerhet strekker mye bredere enn kun de tingene man lager som f.eks.
  - Organisatorisk, hvordan håndterer man ting?
  - Informasjon, hva skal vi informere om når noe skjer?
  - Brukbarhet, hvordan kan vi få ting sikkert men også mulig å bruke?
- Det finnes ingen “enkeltkilde” med ting man kan teste for at “appen skal være sikker”

Sandboxing

Sandboxing er en viktig del av Android sin sikkerhetsmodell, designet for å fjerne isolasjon mellom applikasjoner, systemtjenester og selve operativsystemet. Denne prosessisoleringen er grunnleggende for å opprettholde integriteten og konfidensialiteten til appdata og systemoperasjoner. Denne flerlagrede tilnærmingen til sandboxing i Android sikrer at apper og systemprosesser fungerer sikkert og uavhengig av hverandre, og gir en robusthet mot uautorisert tilgang og ondsinnede aktiviteter.

OBS! Figuren er generert av GPT kun for å illustrere lagdelte sikkerhetsarkitekturen som sikrer isolasjon og beskyttelse av systemkomponenter, applikasjoner og deres data. Ikke en sikker kilde.



Se Android dokumentasjonen:

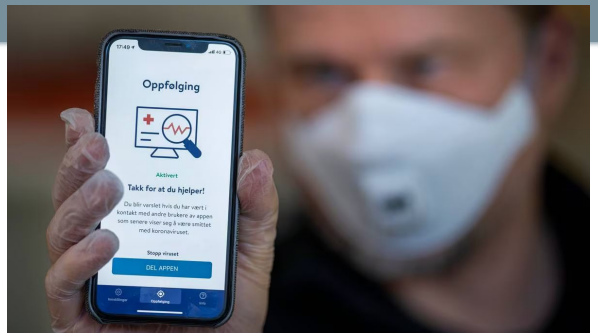
- Android Security Features: https://source.android.com/docs/security/features
- Design for Safety: https://developer.android.com/quality/privacy-and-security

Privacy

- Dataen vår har blitt en råvare
  - Skapt en ny industri
- Tjenesten selger dataen videre til dataforhandlere
  - disse selger videre til aktører, myndigheter, osv.
  - dataforhandlere påstår at dataene er anonymisert før videre salg
  - i realiteten er den knyttet til identifikatorer,
  - og man kan spore det tilbake til personen
- Ved å søke videre på nettet og sosiale medier finner vi likheter til informasjonen, og har skapt et bilde på personen og deres handlingsmønster

Kilde: https://www.nrk.no/norge/xl/avslort-av-mobilen-1.14911685

Eksempel på dårlig håndtert sikkerhet



Smittestopp-app for sporing av COVID-19 spredningen i 2020

- **Sentralisert lagring av data:** Appen samlet inn store mengder personlig informasjon om brukerne, inkludert GPS-posisjon, kontakter og helseinformasjon. Dataene ble lagret sentralt på servere, noe som bekymret mange for personvernet og muligheten for misbruk av informasjonen.
- **Mangelfull anonymisering:** Selv om appen hevdet å anonymisere dataene, viste det seg at det var mulig å koble brukerdata til individuelle personer, noe som svekket personvernet.
- **Svakheter i kryptering:** Sikkerhetsanalyser av appen avdekket svakheter i måten data ble kryptert og overført, noe som kunne gjøre det enklere for uautoriserte personer å få tilgang til brukerdata.
- **Utilstrekkelig testing:** Det ble også hevdet at appen ikke hadde gjennomgått tilstrekkelig testing før den ble lansert, noe som kunne føre til uoppdagede sikkerhetsproblemer.

Ta gjerne kontakt!

jamilakm@uio.no