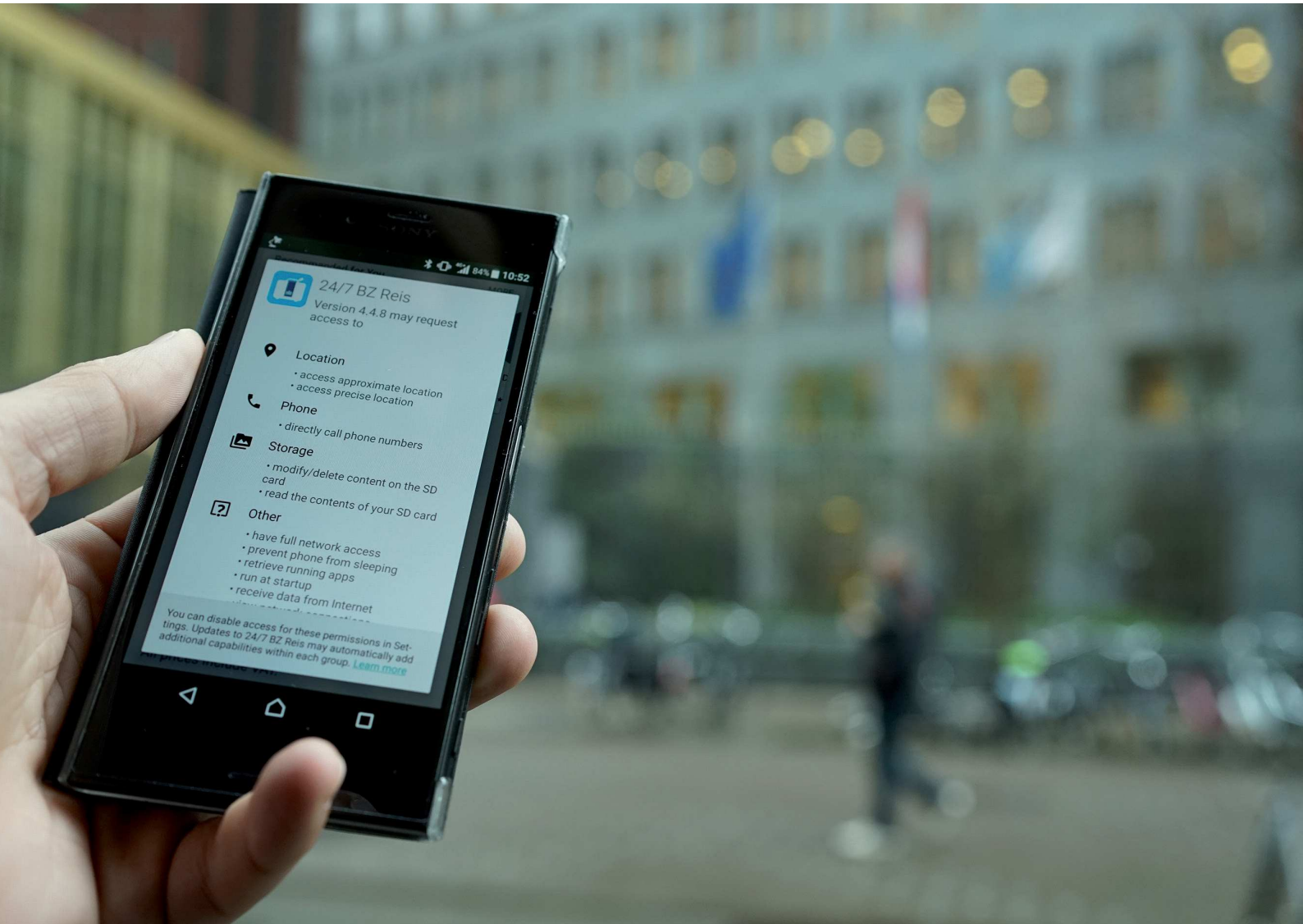
Nationaal Cyber Security Centrum
Ministerie van Justitie en Veiligheid

ICT-beveiligingsrichtlijnen voor mobiele apps



Nationaal Cyber Security Centrum

Het Nationaal Cyber Security Centrum (NCSC) draagt via samenwerking tussen bedrijfsleven, overheid en wetenschap bij aan het vergroten van de weerbaarheid van de Nederlandse samenleving in het digitale domein.

Het NCSC ondersteunt de Rijksoverheid en organisaties met een vitale functie in de samenleving met het bieden van expertise en advies, respons op dreigingen en het versterken van de crisisbeheersing. Daarnaast biedt het NCSC informatie en advies voor burger, overheid en bedrijfsleven ten behoeve van bewustwording en preventie. Het NCSC is daarmee het centrale meld- en informatiepunt voor ICT-dreigingen en -veiligheidsincidenten.

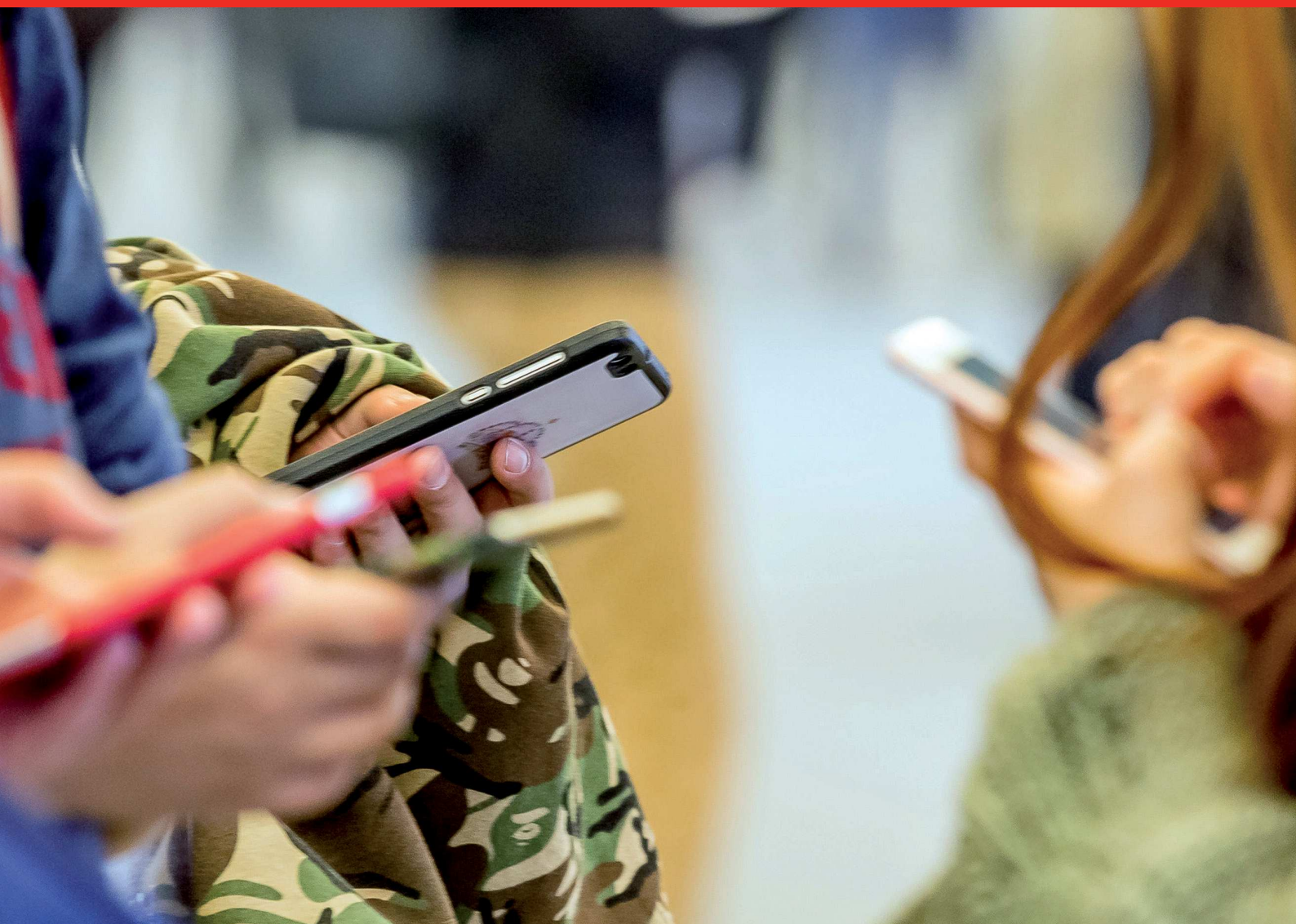
Het NCSC is een onderdeel van de Directie Cyber Security van de Nationaal Coördinator Terrorismebestrijding en Veiligheid.

Samenwerking

Deze publicatie is tot stand gekomen in samenwerking met de volgende partners:

- Centrum voor Informatiebeveiliging en Privacybescherming
- Belastingdienst
- Rabobank
- ING
- Centric
- CERT.be
- Ministerie van Defensie
- ICTU
- Rijkswaterstaat

ICT-beveiligingsrichtlijnen voor mobiele apps



Inhoud


09:41
Home - Living Room
21.8°
52%
indoor
3/18
23 W
lights
on
locked
security
38%
heating
54%
ceiling lights
2/6
3/9
communication

Inleiding

Mobiele apps zijn de afgelopen jaren explosief gegroeid in aantal. Veel organisaties publiceren apps om hun dienstverlening voor klanten te vereenvoudigen, om contact mogelijk te maken, om hun doelgroep te vermaken en nog veel meer. Omdat mobiele apparaten een steeds belangrijkere rol in het leven van mensen spelen, betekent dit dat kwaadwillenden deze platformen steeds aantrekkelijker vinden om digitaal aan te vallen. Om die reden is het belangrijk dat zowel de mobiele apparaten als de apps die daarop draaien veilig zijn. Deze ICT-beveiligingsrichtlijnen voor mobiele apps bieden maatregelen om apps en hun gebruikers te beschermen tegen diverse aanvallen.

Doelgroep

Dit document heeft drie primaire doelgroepen.

- De eerste doelgroep bestaat uit partijen die verantwoordelijk zijn voor het stellen van beveiligingskaders en de controle op naleving hiervan. Hierbij kan worden gedacht aan securitymanagers en systeemeigenaren (de opdrachtgevers) van de te leveren ICT-diensten.
- De tweede doelgroep bestaat uit diegenen die betrokken zijn bij het ontwerp- en ontwikkelproces, de implementatie en het beheer van mobiele apps. Deze doelgroep moet de beveiligingsrichtlijnen toepassen.
- De derde doelgroep bestaat uit de controlerende instanties (IT-auditors) die op basis van deze richtlijnen een objectief ICT-beveiligingsassessment uitvoeren.

Reikwijdte en context

Deze richtlijnen richten zich op de beveiliging van mobiele apps, dat wil zeggen de overdraagbare programmatuur die op een mobiel apparaat wordt uitgevoerd. De richtlijnen richten zich niet op de backend aan de serverzijde van de app of op de configuratie van het mobiele apparaat zelf. Er zijn daarom in deze richtlijnen geen directe beveiligingsmaatregelen opgenomen voor servers of voor de manier waarop een apparaat ingesteld moet zijn om apps veilig te kunnen gebruiken. Hiervoor publiceert het NCSC aparte richtlijnen.[1]

Deze richtlijnen zijn niet alomvattend en kunnen naast beveiligingsvoorschriften en baselines met een bredere scope (zoals BIR en BIG) worden gebruikt. Dergelijke baselines zijn een deelverzameling van de maatregelen uit ISO-standaard 27002. Deze richtlijnen vormen wel een gedeeltelijke overlap met ISO 27002 en baselines, maar zijn ze op bepaalde onderdelen gedetailleerder uitgewerkt.

De richtlijnen bieden specifieke verdieping voor de beveiliging van mobiele apps. De beveiliging van apps moet passen binnen de beveiligingsopzet die organisaties voor hun overige processen en omgeving al ingericht zouden moeten hebben, bijvoorbeeld op basis van ISO 27002.

Deze richtlijnen zijn primair technisch van aard. Dit betekent dat een aantal aspecten van informatiebeveiliging geen onderdeel uitmaakt van het raamwerk dat in deze richtlijnen wordt gehanteerd. Het raamwerk besteedt bijvoorbeeld nauwelijks tot geen aandacht aan zaken als beveiligingsorganisatie, fysieke beveiliging en personeel. Niet-technische maatregelen worden uitsluitend opgenomen wanneer deze noodzakelijk worden geacht voor de technische context of wanneer andere normenkaders of standaarden hier onvoldoende op ingaan. Indien een risicoanalyse aanleiding geeft voor het invullen van deze aanvullende beveiligingsmaatregelen dan wordt verwezen naar andere beveiligingsstandaarden zoals ISO 27001 en ISO 27002.

Deze richtlijnen zijn het uitgangspunt voor de beveiliging van mobiele apps. Een organisatie kan de beveiliging van zijn apps (laten) toetsen op basis van deze richtlijnen. De toetsende organisaties kunnen deze richtlijnen gebruiken om een objectief beveiligingsassessment uit te voeren. Bij het beoordelen van een specifieke situatie en bij het implementeren van de richtlijnen (het oplossen van tekortkomingen) kan naar deze richtlijnen verwezen worden.

Toepassing

Organisaties kunnen (een deel van) deze richtlijnen voor bepaalde toepassingsgebieden verheffen tot een normenkader. In tegenstelling tot de beveiligingsrichtlijnen, die adviserend van aard zijn, is een normenkader dwingend voor het toepassingsgebied. Ook kunnen de richtlijnen worden gebruikt in aanbestedingen, het uitbesteden van dienstverlening en in onderlinge afspraken bij ketenprocessen. Afhankelijk van de aard en de specifieke kenmerken van de betreffende dienst kunnen beveiligingsrichtlijnen worden geselecteerd en kunnen de wegingsfactoren van de individuele beveiligingsrichtlijnen worden aangepast om de gewenste situatie te weerspiegelen.

1. Als de app met de back-end communiceert over https, dan kan de serverzijde worden gezien als webapplicatie. Hiervoor kan worden uitgegaan van de ICT-beveiligingsrichtlijnen voor webapplicaties [1]. Mobiele apparaten vallen meestal buiten het beheer van de uitgevende organisatie en kunnen dan niet centraal worden beveiligd. Als mobiele apparaten wel in het beheer van een organisatie vallen kan gebruik worden gemaakt van de Beveiligingsrichtlijnen voor mobiele apparaten [2].

Prioriteit

De prioriteit van elke beveiligingsrichtlijn wordt in algemene zin gewaardeerd volgens de classificatie Hoog, Midden of Laag. Deze drie classificaties vormen drie punten op een schaal waarbij Hoog de sterkste mate van gewenstheid is (must have), Midden een redelijk sterke mate van gewenstheid is (should have) en Laag een gewenste, maar niet noodzakelijke voorwaarde vormt (nice to have). De drie waarden zijn moeilijk exact te definiëren, maar vormen een functie van kans op optreden van een bedreiging en de mogelijke schade als gevolg hiervan.

De uiteindelijke afweging voor een specifieke app voor een specifieke organisatie is afhankelijk van de weging van risico's die uit een risicoanalyse naar voren komen. Daarbij wordt gekeken naar de kans op optreden van een bedreiging, het te verdedigen belang en de mogelijke impact hiervan op de bedrijfsvoering. De beveiligingsrichtlijnen bieden de maatregelen die genomen kunnen worden om het optreden van bedreigingen terug te dringen of de impact in geval van optreden van een bedreiging te beperken.

Als voorbeeld van een aanpassing van de algemene classificaties in specifieke situaties kan worden gekeken naar beschikbaarheidsmaatregelen. De noodzaak van beschikbaarheidsmaatregelen kan bijvoorbeeld laag zijn in situaties waar het niet beschikbaar zijn van een dienst weinig impact heeft op de bedrijfsvoering. De noodzaak kan juist hoog zijn in situaties waar de impact en de kans op optreden van een bedreiging groot zijn.

Leeswijzer

De richtlijnen zijn ingedeeld volgens het SIVA-raamwerk. [4] De structuur van het SIVA-raamwerk bestaat uit drie domeinen.

Beleidsdomein

Hier bevinden zich elementen die aangeven wat in organisatiebrede zin bereikt kan worden. Het bevat daarom randvoorwaardelijke elementen die van toepassing zijn op de overige lagen, zoals doelstellingen, informatiebeveiligingsbeleid, strategie en vernieuwing, organisatiestructuur en architectuur.

Uitvoeringsdomein

In dit domein wordt de implementatie van de mobiele app uiteengezet. Dit document behandelt van de drie domeinen het uitvoeringsdomein. Het beleidsdomein en beheersingsdomein worden in een apart document behandeld. [3]

Beheersingsdomein (control)

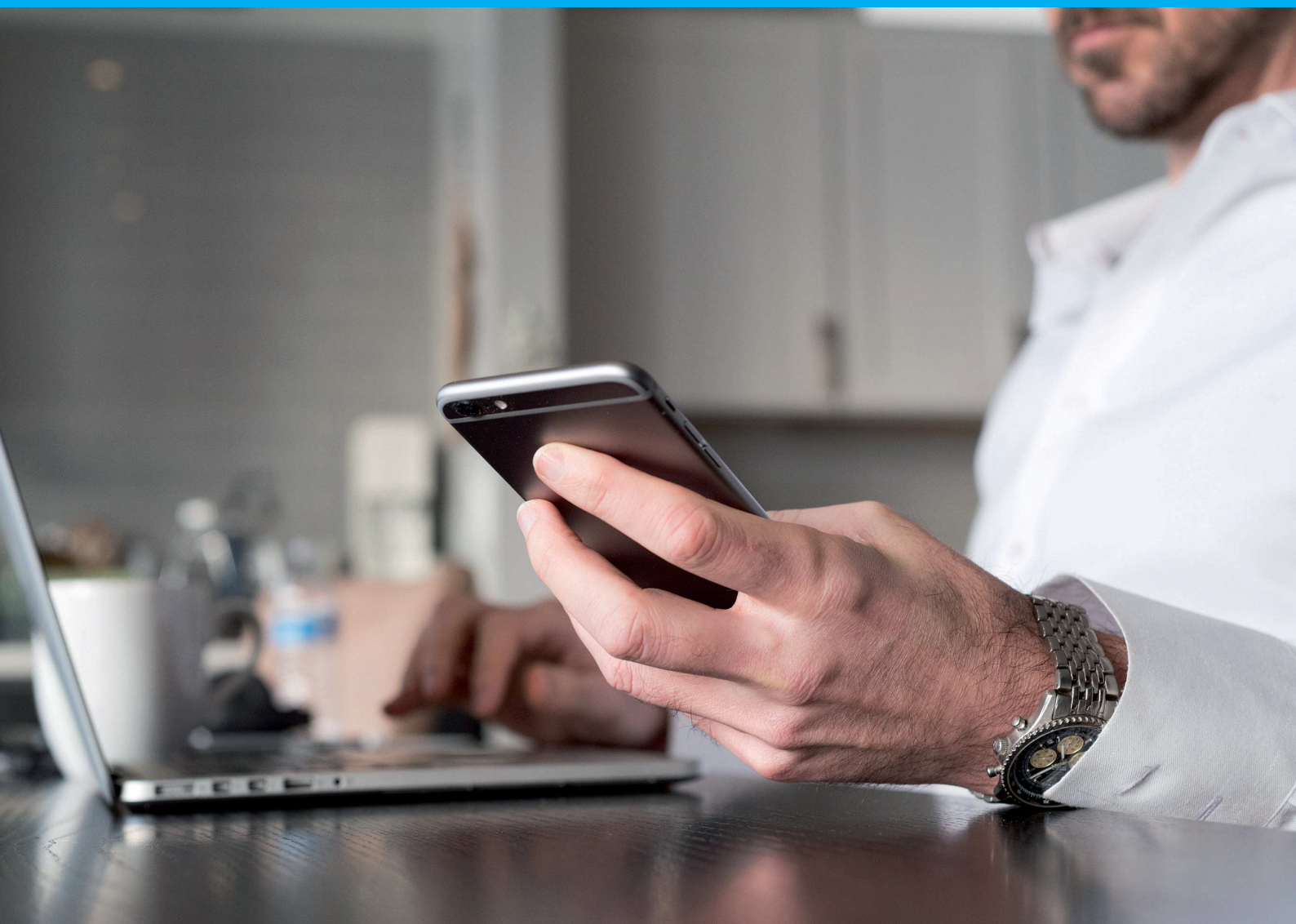
Evaluatieaspecten en meetaspecten zijn in dit domein opgenomen. Daarnaast staan hier ook de beheerprocessen beschreven, die noodzakelijk zijn voor de instandhouding van ICT-diensten. De informatie uit de evaluaties en de beheerprocessen is niet alleen gericht op het bijsturen van de geïmplementeerde mobiele apps, maar ook op het bijsturen of aanpassen van de eerder geformuleerde conditionele elementen.

Relatie met andere documenten

Deze richtlijnen zijn samengesteld op basis van het document Grip op SSD, de Normen voor Mobiele Apps van het Centrum voor Informatiebeveiliging en Privacybescherming (CIP). [5] Het Uitvoeringsdomein van dit document komt overeen met het normenkader van het CIP. Het NCSC en CIP zorgen voor gezamenlijk en gecoördineerd onderhoud van de richtlijnen zodat mobiele apps, wanneer zij aan een van de richtlijnensets voldoen, inherent technisch ook aan de andere voldoen.

De ICT-beveiligingsrichtlijnen voor webapplicaties van het NCSC [1] staan met deze richtlijnen in verbinding. Zij putten uit een gezamenlijk beleids- en beheersingsdomein; de uitvoeringsdomeinen spitsen zicht toe op de verschillende architecturele onderdelen van een applicatieomgeving.

Uitvoeringsdomein: Mobiele apps



U/MA.01 Operationeel beleid voor mobiele apps

Het operationeel beleid voor mobiele apps beschrijft de manier waarop de organisatie omgaat met het inrichten en beschikbaar stellen van apps. Het operationeel beleid is een concretere uitwerking van het bovenliggende beleid. Een solide operationeel beleid is daarom een randvoorwaarde voor een veilige inrichting van een app en zijn omgeving.

U/MA.01 Operationeel beleid voor mobiele apps

Criterium *(wie en wat)*	Het operationeel beleid voor mobiele apps bevat **richtlijnen** en **instructies en procedures** met betrekking tot ontwikkeling, onderhoud en uitfasering van apps.
Doelstelling *(waarom)*	De ontwikkeling van de app optimaal ondersteunen en de klant betrouwbare diensten bieden.
Risico	Geen eenduidige richting voor apps, waardoor de beveiliging van de app los staat van zijn omgeving. Dit vergroot de kans op beveiligingsincidenten.
Classificatie	Hoog

Maatregelen

Richtlijnen

01 Stel richtlijnen op voor:
- ontwikkeling, onderhoud en uitfasering van apps;
- beveiliging van apps;
- verwerking van gegevens;
- koppelingen met achterliggende systemen.

Gebruik bij de ontwikkeling van apps methodes voor secure software development.
Achterliggende systemen zijn systemen die direct of indirect door de app ontsloten of bereikt worden.

Instructies en procedures

02 Stel instructies en procedures op voor:
- het werken met gescheiden ontwikkel-, test-, acceptatie- en productie-omgevingen (OTAP);
- contentmanagement.

Besteed aandacht aan het regelmatig evalueren en bijstellen van de richtlijnen. Indien deze instructies en procedures ruimte bieden om af te wijken van de richtlijn, dient hieraan de vereiste van 'pas toe of leg uit' gekoppeld te zijn.

U/MA.02 Veilige server-side applicatie

De app op het mobiele apparaat vormt doorgaans samen met de applicatie aan de serverzijde een keten. Veilig gebruik van de app is alleen mogelijk in combinatie met een veilig ingerichte server-side applicatie.

Het ontbreken van een veilige server-side applicatie of een veilige server leidt tot het niet kunnen opbouwen van een veilige keten en daarmee tot een omgeving die niet geschikt is om vertrouwelijke informatie op te slaan en uit te wisselen.

Aandacht voor de informatieveiligheid van de serverapplicatie voorkomt dat bij de bouw van de app eisen worden gesteld aan de serverapplicatie die strijdig zijn met de informatieveiligheid van de gehele keten van mobiel apparaat tot server.

U/MA.02 Veilige server-side applicatie

Criterium *(wie en wat)*	Bij de bouw van de app worden de **beveiligingseisen** van de applicatie op de server door de ontwikkelaar opgevolgd.
Doelstelling *(waarom)*	Bij de ontwikkeling van de app worden alleen inrichtingskeuzen gemaakt die de veiligheid van de server(applicatie) niet in het geding brengen. Zodoende blijft de veiligheid van de keten van app tot en met server-side applicatie gewaarborgd.
Risico	De bouw van een app maakt een veilige inrichting van de server-side applicatie onmogelijk of moeilijker.
Classificatie	Hoog

Maatregelen

Beveiligingseisen

01 Voorkom dat eisen aan de server-side applicatie ertoe leiden dat deze niet meer voldoet aan de ICT-beveiligingsrichtlijnen voor webapplicaties.

02 Waarborg als opdrachtgever dat de server-side applicaties die door de app worden gebruikt voldoen aan de ICT-beveiligingsrichtlijnen voor webapplicaties.

Het hanteren van de richtlijnen voor webapplicaties op de server is een vereiste voor een veilige keten.

U/MA.03 Apps van derden

Apps werken vaak samen met andere apps, zoals viewers en toetsenborden. Deze apps zijn doorgaans afkomstig van andere leveranciers en worden aangeduid als third-party-apps. Deze apps verwerken informatie buiten de app, waardoor de bescherming van de informatie buiten de invloed van de ontwikkelaar terechtkomt. Deze apps kunnen verborgen functionaliteit hebben, waarmee toegang verkregen kan worden tot vertrouwelijke informatie, ook als deze informatie door de app is afgeschermd.

Bij de keuze voor dergelijke apps moeten daarom de voordelen en de risico's tegen elkaar worden afgezet. Een voordeel kan bijvoorbeeld kostenbesparing zijn. Het zelf bouwen van functionaliteit hoeft niet per definitie te leiden tot een veiligere app. Het bouwen van een veilige app is in veel situaties, zoals cryptografische hulpmiddelen, geen sinecure.

Third-party-apps kunnen permissies vragen of verborgen functionaliteit hebben waarmee toegang verkregen wordt tot vertrouwelijke informatie, ook als deze is afgeschermd. Ook zouden zij gebruik kunnen maken van onveilige API's en kwetsbaarheden in apps en het besturingssysteem. Het uitgangspunt is dat apps van derden zonder inspectie van het maakproces en de code in principe als onveilig moeten worden beschouwd. Dit geldt in het bijzonder wanneer gebruikers zelf third-party-apps gebruiken in combinatie met de app, zoals third-partytoetsenborden.

U/MA.03 Apps van derden

Criterium *(wie en wat)*	Het gebruik van third-party-apps is gebaseerd op een **risicoafweging**.
Doelstelling *(waarom)*	Voorkomen dat third-party-apps ongewenst toegang krijgen tot informatie die door de app moet worden afgeschermd.
Risico	Vertrouwelijke informatie wordt via een third-party-app toegankelijk gemaakt voor aanvallers.
Classificatie	Hoog

Maatregelen

Risicoafweging

01 Controleer de third-party-apps op kwetsbaarheden, zoals verborgen functionaliteit en onveilige API's. Stel de risico's vast.

De keuze voor third-party-apps is gebaseerd op de afweging tussen een veilige werking van de app en de voordelen van het gebruik van bestaande third-party-apps.

Bij het bepalen van de risico's wordt gekeken naar:
- *Zijn de third-party-apps en hun code te testen bij de oplevering van de app?*
- *Zijn patches op of nieuwe versies van (de code van) de third-party-apps ook te testen na de eerste ingebruikname van de third-party-app?*
- *Zijn afspraken over het up-to-date houden van de app contractueel vastgelegd en is hierbij een adequaat governanceproces ingericht?*
- *Voldoet de third-party-app aan de beveiligingsrichtlijnen?*

02 Baseer de keuze voor third-party-apps op een risicoanalyse. Leg de resultaten van de risicoanalyse vast.

Bij de risicoanalyse is gebruik gemaakt van de resultaten van de controle op kwetsbaarheden (maatregel 01). Neem in de risicoafweging ook de voordelen van third-party-apps mee, zoals de in de third party app ingebouwde beveiligingsvoorzieningen, wanneer die in de praktijk bewezen effectief zijn.

03 Besteed bij de keuze van third-party-apps speciale aandacht aan third-partytoetsenborden.

Op het mobiele apparaat kan naast het standaardtoetsenbord dat bij het besturingssysteem hoort of dat door de leverancier is voorzien, vaak ook gebruik gemaakt worden van alternatieven van derden (third-partytoetsenborden). Deze third-partytoetsenborden kunnen toetsaanslagen vastleggen en daardoor gegevens lekken of misbruiken.
Gebruikers kunnen vaak zelf alternatieve third-partytoetsenborden kiezen. Indien het gebruik van third-partytoetsenborden niet voorkomen kan worden, kies er dan voor om zeer vertrouwelijke informatie binnen de gebruikersinterface van de app in te laten voeren.

U/MA.04 Veilige code bij oplevering

Bij de ontwikkeling van apps kan de ontwikkeling worden versneld door gebruik te maken van (externe) codebibliotheken. Deze kunnen echter kwetsbaarheden of malware bevatten. Informatie over de gebruikte bibliotheken en de werking van de app geeft de aanvaller informatie over zwakheden in de app.

Bij gebruik van bestaande code of elders ontwikkelde code wordt zekerheid verkregen over de veiligheid van de code.

U/MA.04	Veilige code bij oplevering
Criterium *(wie en wat)*	De ontwikkelaar gebruikt alleen **vertrouwde broncode**bibliotheken en de broncode bevat geen **technische informatie** over de werking van de app.
Doelstelling *(waarom)*	De gebruikte broncode is veilig en geeft geen informatie vrij over de interne werking van de app.
Risico	Broncode van derden bevat kwetsbaarheden, malware of informatie die een aanvaller toegang geven tot de werking van de app of de vertrouwelijke gegevens.
Classificatie	Midden

Maatregelen

Vertrouwde broncode

01 Gebruik alleen code waarvan de oorsprong bekend en de veiligheid vastgesteld is.

De bron is bekend en er zijn van de gebruikte versie binnen het vakgebied geen zwakheden bekend die een bedreiging vormen.
Vertrouw geen closed-sourcecomponenten van derde partijen, tenzij precies bekend is wat de componenten doen. Van open-sourcecomponenten moet een analyse zijn uitgevoerd, waaruit blijkt dat de software als veilig kan worden gezien.

02 Houd de gebruikte code en codebibliotheken up-to-date zodat deze geen bekende kwetsbaarheden bevatten.

03 Stel de oorsprong en veiligheid vast van mobiele code die door de app wordt gedownload.

Voor de software die bij de bouw van een applicatie is gedownload geldt ook maatregel 01 ten aanzien van bekendheid met de herkomst.
De gebruikelijke methode om dit te waarborgen is code-signing. Dit is een beveiligingsfunctie op het apparaat die pogingen om ongeautoriseerde applicaties op het apparaat te draaien voorkomt door het valideren van de handtekening op de app. Elke keer als de app wordt aangeroepen wordt de validatie uitgevoerd. Hierdoor kunnen apps alleen code uitvoeren met een geldige, vertrouwde handtekening.

04 Zorg ervoor dat de instellingen in het configuratiebestand van de aangeboden app de veiligheid optimaal waarborgen.

Hierbij wordt gekeken naar de instellingen voor debugging, permissies en de beveiligingsopties van de configuratie-instellingen.

05 Publiceer de hash van de gedownloade app-binary en onderteken de hash en app-binary, zodat duidelijk is van wie de app afkomstig is.

Een hash wordt berekend op basis van de binary. Wijziging van de binary levert doorgaans altijd (afhankelijk van de kwaliteit van het hashalgoritme) een andere hashwaarde op. Zo kan gecontroleerd worden of de binary is gewijzigd. De hash en de binary kunnen zelf weer worden beveiligd door deze te ondertekenen (signen). Hierbij worden de binary en de hash door middel van een cryptografische sleutel voorzien van een voor die hash unieke digitale handtekening.
Niet elke gebruiker controleert de digitale handtekening en daarmee de authenticiteit van de app. Het is daarom nuttig om periodiek naar geïmiteerde of gekopieerde en gemodificeerde apps in de appstores te zoeken.

Technische informatie

06 Sla geen gevoelige technische informatie op in de app. Als het wel opslaan als noodzakelijk wordt gezien, dan gebeurt dit op basis van een risicoafweging.

Denk daarbij aan: cryptografische sleutels, wachtwoorden, interne url's, etc. Omdat een binary nooit volledig kan worden afgeschermd, mogen beveiligingsmaatregelen, alsook beveiligingsinstellingen nooit worden opgeslagen in de code zelf.

07 Sla beveiligingsinstellingen en -maatregelen niet op in bestanden die buiten de scope van de ondertekening van de app en buiten de afscherming van het besturingssysteem liggen.

08 Informatie over de werking van de app en relevante en te beschermen vertrouwelijke informatie in de binary is afgeschermd ter voorkoming van reverse engineering en manipulatie.

Als regel geldt: obfuscatie geldt niet als vervanging daar waar versleuteling vereist wordt.
Bij oplevering van de app aan de appstore wordt de app digitaal ondertekend en via een beveiligde verbinding en een account aangeboden. De publieke sleutel (en het algoritme) moet bij de appstore bekend zijn, zodat de appstore de app kan inspecteren en na goedkeuring kan publiceren. In het Apple-certificaat wordt door Apple ook de identiteit van de aanbieder vastgelegd. Dit heeft Apple de mogelijkheid gegeven om het aanbieden van de apps te beperken tot iOS program enabled ontwikkelaars.
Bedenk dat in een gejailbreakt mobiel apparaat de binary onversleuteld in het geheugen staat en daardoor toegankelijk is voor kwaadwillenden.

U/MA.05 Integere werking van de app

In tegenstelling tot server-side applicaties, draaien apps niet in een vertrouwde omgeving, maar op het mobiele apparaat zelf. Hierdoor kan een aanvaller volledige code in handen krijgen: de binary en de werkende app. Zodoende kan de aanvaller controle krijgen over elk stukje code tijdens elke fase van het programma van de app plus de informatie die is opgeslagen in de binaire bestanden van de app, inclusief de configuratiebestanden. Manipulatie van de werking door kwaadwillenden kan daarom alleen worden beperkt door de binary en de werkende app elektronisch te beschermen.

U/MA.05	Integere werking van de app
Criterium *(wie en wat)*	De app op het mobiele apparaat is afgeschermd tegen ongewenste of onbedoelde **manipulatie** en de uitkomsten van de **bedrijfslogica** worden altijd gecontroleerd op de server.
Doelstelling *(waarom)*	Voorkomen dat de werking van de app onbedoeld kan worden beïnvloed.
Risico	De vertrouwelijkheid van de informatie, de correcte werking van de app en de integriteit van de output komen onder invloed van een aanvaller.
Classificatie	Hoog

Maatregelen

Manipulatie

01 Gebruik Position Independent Code (PIC) en Address Space Layout Randomisation (ASLR).

Het niet meenemen van ASLR bij de code of delen ervan is gebaseerd op een risicoafweging.
Position independent code (PIC) maakt het mogelijk de code te draaien vanuit elke plek in het geheugen. PIC maakt het mogelijk de code een steeds wisselende plaats te geven, waardoor een aanvaller geen houvast heeft aan de geheugenlocatie van de code. Apps die PIC hebben ingeschakeld worden geconfigureerd als een position independent executable (PIE).
Address Space Layout randomisatie (ASLR) is een beveiligingsfunctie die er voor zorgt dat de code van een PIE een steeds wisselende plaats in het procesgeheugen krijgt. ASLR inschakelen maakt het moeilijker om aanvallen uit te voeren die gebruik maken van vaste, voorspelbare locaties van de code in het geheugen, zoals bufferoverflows.

Bedrijfslogica

02 Controleer beslissingen gebaseerd op kritische bedrijfslogica in de app in een veilige omgeving op de server.

Hoewel het beschermen van de app de kans op wijzigen van de bedrijfslogica minimaliseert, is er geen garantie dat de logica in de app bij een aanval door een hacker intact blijft.

U/MA.06 Locatie voor de opslag

De keuze van de locatie voor de opslag van gegevens bepaalt in belangrijke mate de mogelijkheden om de gegevens te kunnen afschermen tegen ongewenste toegang. De keuze voor de opslag van ieder gegeven of set van gegevens moet daarom steeds bewust worden gemaakt. Omdat veiligere locaties meer zekerheid bieden wordt bij deze beveiligingsrichtlijn als uitgangspunt aangehouden dat voor de opslag gekozen wordt voor de veiligste locatie, tenzij zekerheid bestaat dat een minder veilige locatie aantoonbaar passend beveiligd kan worden.

Opslag op het mobiele apparaat door middel van de mogelijkheden van het apparaat is moeilijker af te schermen voor een ieder die toegang (fysiek of via malware) heeft tot het apparaat dan een plaats die elektronisch en fysiek is beschermd.

Een passende afscherming van gegevens kan worden geboden wanneer de vertrouwelijkheid ervan is vastgesteld.

U/MA.06 Locatie voor de opslag

Criterium *(wie en wat)*	De keuze van de **locatie** waar de gegevens en informatie van de logica van de app worden opgeslagen is gebaseerd op het principe van de **vertrouwelijkheid**.
Doelstelling *(waarom)*	Vertrouwelijke informatie en kritische logica worden alleen opgeslagen op locaties waar doeltreffende beveiligingsmaatregelen genomen kunnen worden.
Risico	Vertrouwelijke informatie of kritische logica komt in handen van onbevoegden.
Classificatie	Midden

Maatregelen

Locatie

01 Sla uitsluitend gevoelige informatie over of gevoelige logica op het mobiele apparaat op als dit noodzakelijk is voor de werking van de app.

02 Sla vertrouwelijke gegevens en gevoelige informatie over de logica van de app standaard op de serverzijde in een veilige omgeving op.

Opslag van alle gegevens en logica vindt plaats op de backend als met een risicoanalyse is bepaald dat de gevolgen van het openbaar worden niet acceptabel zijn.

03 Hanteer als uitgangspunt bij het ontwerp dat lokaal opslaan op het mobiele apparaat zo veel mogelijk wordt voorkomen. Hierbij wordt per gegeven of set van gegevens de afweging gemaakt of offline lokale beschikbaarheid een vereiste is of vermeden kan worden.

04 Bepaal in een risicoanalyse welke afscherming vereist is indien vertrouwelijke informatie op het mobiele apparaat wordt opgeslagen.

In de risicoanalyse wordt ook de werking van door de app gebruikte componenten meegenomen en daarmee ook de gegevens die achter de schermen, onmerkbaar voor de gebruiker, worden opgeslagen (caching, logging etc.).
De risicoanalyse kan dan bijvoorbeeld leiden tot het versleuteld opslaan van informatie.

05 Laat de opslag van vertrouwelijke gegevens op het apparaat plaatsvinden in de interne opslag van de app, tenzij de vereiste afscherming (maatregel 04) anders aangeeft. Hierbij is er rekening mee gehouden dat de standaardmappen onderdeel kunnen uitmaken van back-ups of synchronisatie met de cloud of een gekoppelde computer.

Door een sandbox te gebruiken is de app en de informatie in de app afgeschermd. Een sandbox is een container op het mobiele apparaat die door versleuteling is afgeschermd. Ook als een aanvaller toegang tot het apparaat heeft, kan hij niet bij de gegevens zolang de sleutel waarmee de sandbox is versleuteld is afgeschermd.
Op het mobiele apparaat is de 'interne opslag' de opslag binnen de map waarin de app is geïnstalleerd. Deze map is samen met de app door de sandbox beveiligd. Andere apps hebben geen toegang tot deze bestanden, tenzij het apparaat gejailbreakt of geroot is, of als er kwetsbaarheden in het besturingssysteem worden misbruikt.
Standaard worden bepaalde mappen binnen de interne opslag meegenomen in back-ups of synchronisatie met de cloud of een vertrouwde computer. Raadpleeg de platformdocumentatie om een map te kiezen binnen de interne opslag, die op basis van functionele en risicoafwegingen in aanmerking komt.
Indien de sleutel van een sandbox veilig is opgeslagen op een backend (maatregel 01), dan is deze ook afgeschermd als het apparaat gejailbreakt of geroot is. Dit biedt daardoor de mogelijkheid van een betere beveiliging.

06 Beperk de opslag van gegevens op het apparaat buiten de interne opslag van de app tot niet-vertrouwelijke informatie, of tot informatie waar de gebruiker van weet dat hij die moet beheren en dus moet afschermen.

07 Laat de opslag van vertrouwelijke informatie pas in een publieke cloud plaatsvinden als opslag op de server (maatregel 02) niet mogelijk is, en altijd pas nadat hiervan vooraf de voor- en nadelen voor de opdrachtgever en gebruiker, inclusief de privacyaspecten, zijn vastgesteld en leiden tot een afweging met een positief resultaat. Dit geldt evenzeer voor back-ups.

Opslag op het mobiele apparaat en in de publieke cloud, zoals sociale media, maar ook andere opslag bij derden wordt voor vertrouwelijke informatie als onveilig gezien. In alle omstandigheden zouden gegevens voor opslag versleuteld moeten zijn.

Vertrouwelijkheid

08 Specificeert (de classificatie van) de vertrouwelijkheid van de gegevens in of bij de app.

De specificatie geeft duidelijkheid over welke gegevens afgeschermd moeten worden.
Indien gebruikgemaakt wordt van classificaties, is per classificatie de beveiligingsvereiste vastgelegd.
In de analyse worden ook de back-ups van gegevens meegenomen.
Niet voor ieder gegeven zal in de praktijk bekend zijn wat de classificatie is, zeker niet als het een gegeven is dat gebruikt wordt voor de technische werking van de app.

09 Behandel gegevens als vertrouwelijk en beveilig ze als zodanig indien de vertrouwelijkheid niet is vastgesteld.

In de analyse wordt ook de bedrijfslogica in de software meegenomen.

U/MA.07 Opslag op het mobiele apparaat

Gevoelige gegevens kunnen worden beschermd door gebruik te maken van cryptografische technieken. Cryptografische technieken zijn de enige efficiënte manier om informatie af te schermen als deze informatie fysiek toegankelijk is.

Welke gegevens gevoelig of vertrouwelijk zijn, moet door de organisatie worden vastgesteld. Als onderdeel van het bepalen van de locatie van de opslag is de vertrouwelijkheid bepaald (zie U.06 Locatie voor de opslag).

U/MA.07 Opslag op het mobiele apparaat

Criterium *(wie en wat)*	Bij het opslaan van vertrouwelijke informatie op het mobiele apparaat zijn de vertrouwelijke gegevens afgeschermd door toepassing van **cryptografische technieken**.
Doelstelling *(waarom)*	Voorkomen van het onbevoegd gebruik, inzien, wijzigen en verlies van vertrouwelijke gegevens in een onveilige opslag.
Risico	De vertrouwelijke gegevens in de mobiele app komen in handen van onbevoegden.
Classificatie	Hoog

Maatregelen

Cryptografische technieken

01 Scherm vertrouwelijke informatie minimaal af door middel van een pincode.

Afscherming gebaseerd op de gebruikelijke pincode-schermbeveiliging is betrekkelijk eenvoudig te doorbreken door middel van brute force. Voor zeer vertrouwelijke informatie, zoals medische gegevens en identiteitsgegevens, zijn aanvullende maatregelen vereist. Geheime sleutels en identiteitsgegevens zijn daarom ook niet in de code opgeslagen.
De app en de informatie kunnen worden afgeschermd met een sleutel die wordt samengesteld uit een combinatie van de pincode en unieke kenmerken van het apparaat. Met deze afgeleide sleutel wordt de werkelijke sleutel versleuteld.

02 Sla encryptiesleutels niet op het mobiele apparaat op.

Om opgeslagen gegevens te beschermen met behulp van encryptie moet een geheime encryptiesleutel op een veilige locatie worden opgeslagen.
Bij opslag van de encryptiesleutel op het mobiele apparaat kan de afscherming ervan weer eenvoudig worden doorbroken. Opslag van de sleutel moet daarom op een andere locatie, bijvoorbeeld op een ander apparaat van de gebruiker of op een server, worden geregeld. Dit kan bijvoorbeeld door het op te slaan in het profiel van de gebruiker.

03 Overweeg het gebruik van een passphrase of vertragende keystretching-algoritmen.

De tijd om door middel van brute force binnen te komen wordt vergroot door het toepassen van een passphrase of vertragende keystretching-algorithmen, zoals PBKDF2 of bcrypt. Een passphrase is een wachtzin in plaats van een wachtwoord.

04 Genereer voor tijdelijke bestanden een tijdelijke sleutel en houd de sleutel in het (interne) geheugen tot het afsluiten van de app.

Gebruik voor het genereren van de sleutel een cryptografisch sterke random-numbergenerator.

05 Sla vertrouwelijke informatie alleen op worden in een database als die database versleuteld is.

De SQLite-database wordt vaak gebruikt zonder encryptie. Het is mogelijk de gehele database te versleutelen. Het heeft echter de voorkeur de datavelden per stuk te versleutelen en de ontsleutelde gegevens niet langer dan voor de verwerking nodig is in het geheugen te houden.

06 Gebruik bekende en bewezen algoritmes en implementaties daarvan voor versleuteling. Gebruik geen zelfontwikkelde crypto-functies.

Het robuust implementeren van cryptografische functies is erg complex. Doorgaans is het meer verantwoord om gebruik te maken van cryptografische functies van het besturingssysteem, of van bekende en bewezen third-partycryptografiebibliotheken.

U/MA.08 Onnodige informatie in het werkgeheugen

Het tijdelijk opslaan van vertrouwelijke informatie in het geheugen van het mobiele apparaat is voor de meeste apps onvermijdelijk. Er zijn echter aanvallen bekend, waarbij een geheugendump wordt gemaakt, terwijl het toestel is vergrendeld. Vertrouwelijke informatie die op dat moment in het geheugen staat (zoals een pincode), kan dan door een aanvaller worden bemachtigd. Ook na bijvoorbeeld een crash is vertrouwelijke informatie toegankelijk.

U/MA.08	Onnodige informatie in het werkgeheugen
Criterium *(wie en wat)*	Vertrouwelijke informatie is in het cachegeheugen op basis van een **risicoafweging** per gegeven **tot een minimum beperkt**; dit geldt zowel binnen als buiten de eigen app.
Doelstelling *(waarom)*	Het voorkomen dat gevoelige informatie via het cachegeheugen toegankelijk wordt (en blijft).
Risico	Tijdelijk opgeslagen gevoelige informatie komt in handen van onbevoegden.
Classificatie	Midden

Maatregelen

Risicoafweging

01 Beperk het (tijdelijk) beschikbaar hebben van vertrouwelijke informatie in het werkgeheugen tot een minimum en op basis van een risicoafweging per gegeven.

Neem in de risicoafweging mee of de duur van de tijdelijke opslag wordt verlengd, doordat bijvoorbeeld een service of de toegang tot een bestand of netwerk langere tijd niet beschikbaar is.

02 Bepaal hoe lang of kort informatie in het (cache)geheugen wordt bewaard en of deze informatie onleesbaar wordt gemaakt, bijvoorbeeld door te overschrijven met data wiping, op basis van de vertrouwelijkheid van de informatie.

Java heeft een garbage-collection-mechanisme voor het opruimen van achtergebleven gegevens. Doordat het enige tijd kan duren voordat de garbage collector geactiveerd wordt, wordt dit als niet afdoende gezien voor het verwijderen van vertrouwelijke gegevens. Dit betekent dat in Java een tekstveld niet naar een String geconverteerd dient te worden als dit tekstveld vertrouwelijke gegevens bevat, maar dat modificeerbare sequences of arrays gebruikt moeten worden.

03 Maskeer daar waar mogelijk gegevens door middel van truncating.

Met truncating wordt een deel van de karakters van het vertrouwelijke gegeven niet opgeslagen, waardoor de resterende karakters geen vertrouwelijkheid weggeven, bijvoorbeeld door van een creditcardnummer alleen de laatste vier cijfers op te slaan.

Tot een minimum beperkt

04 Sla alleen gevoelige informatie op die nodig is voor de werking van de app.

Sla geen vertrouwelijke informatie op in bijvoorbeeld de aanwezige cookies, webgeschiedenis, webcache, property-bestanden en SQLite-bestanden.

05 Sla vertrouwelijke gegevens die zijn opgeslagen op een server niet (ook) op het mobiele apparaat op.

Hierbij kan gebruikgemaakt worden van webviews en JSON parsing. De gegevens blijven dan zo veel mogelijk op de server en worden ingezien met webviews en JSON parsing.
Ook is het mogelijk vertrouwelijke gegevens niet binnen te halen in de app, maar de vertrouwelijke gegevens te verwerken aan de serverzijde en alleen de resultaten te tonen in de app. Deze maatregel is goed te combineren met de maatregel U/MA.04/02.

06 Voorkom de opslag van vertrouwelijke informatie ten behoeve van autocomplete-functionaliteit.

Bij het intypen van een tekstveld kunnen de gegevens worden opgeslagen en gebruikt door de autocomplete-functionaliteit. Hierdoor kan ook vertrouwelijke informatie worden voorgesorteerd en daarmee zichtbaar worden gemaakt. Om dit te voorkomen kan een invoerveld als 'password' worden getypeerd.
Bij gebruik van third-partytoetsenborden kan dit niet altijd worden voorkomen.

07 Voorkom caching van het https-verkeer.

08 Verwijder zeer vertrouwelijke informatie of maak het onleesbaar in de screenshot wanneer de applicatie naar de achtergrond wordt verplaatst.

Veel besturingssystemen bewaren een screenshot van een app, zodra deze naar de achtergrond wordt gestuurd. Hierdoor kunnen vertrouwelijke gegevens die op dat moment op het scherm werden vertoond worden bekeken. Het is daarom aan te raden om het nemen van 'screen captures on backgrounding' uit te schakelen. Dit is echter vaak door de gebruiker instelbaar. Zeer vertrouwelijke informatie kan worden afgeschermd door middel van een overlay of door de screenshot te vervangen met het splash-beeld van de app.

U/MA.09 Sessietime-out

Tijdens het gebruik van de app door de gebruiker staat een gebruikersessie met een app open. Tijdens deze sessie is informatie via de app toegankelijk. Andere personen kunnen hiervan misbruik maken. Sessiebeëindiging zorgt ervoor dat de sessie wordt beëindigd na een voorgeschreven tijdsinterval van inactiviteit.

Inactiviteit in de gebruikersessie is een periode zonder interactie van de gebruiker met de app.

U/MA.09	Sessietime-out
Criterium *(wie en wat)*	De app beëindigt een gebruikerssessie na een **vooringestelde periode** van inactiviteit van de gebruiker via **automatische sessiebeëindiging**.
Doelstelling *(waarom)*	Het voorkomen dat een sessie slechts een beperkte tijd toegankelijk is voor andere personen wanneer de gebruiker de sessie (op het apparaat) onbeheerd heeft achtergelaten.
Risico	De openstaande sessie wordt door kwaadwillenden misbruikt.
Classificatie	Midden

Maatregelen

Vooringestelde periode

01 Stel de periode voor sessietime-out vast aan de hand van een risico-analyse.

Normaliter wordt voor gevoelige apps een time-out van 2 tot 5 minuten gehanteerd. Bij een laag risico is dit 15 tot 30 minuten.

02 Onderbouw de noodzaak voor een langere vooringestelde periode.

De onderbouwing door de opdrachtgever opgesteld en met de ontwikkelaar afgestemd.

Automatische sessiebeëindiging

03 De app beëindigt de sessie automatisch na een door de opdrachtgever voorgeschreven periode van inactiviteit.

04 Sessiebeëindiging komt overeen met het uitloggen van de gebruiker.

05 Na sessiebeëindiging wordt bij het weer starten van een (scherm)activiteit het inlogscherm van de app getoond en moet de gebruiker zich opnieuw authenticeren.

U/MA.1 o Logging

Tijdens het loggen kunnen handelingen van gebruikers en meldingen over de werking van de app in logbestanden worden vastgelegd. De logging kan gebruikt worden voor het bijhouden van beveiligingsincidenten en fouten in de werking van de app, maar ook gevoelige informatie kan in de logging terechtkomen. Indien deze informatie in verkeerde handen valt dan loopt niet alleen de privacy van de gebruiker gevaar, maar kan de informatie ook gebruikt worden om zwakheden in de app te vinden. Toegang tot deze gevoelige loginformatie moet daarom worden voorkomen.

Debuglogging is een functie voor het registreren van activiteiten en gebeurtenissen in de app. Debugging biedt de mogelijkheid gebeurtenissen en fouten op te sporen tijdens het ontwikkelen, testen of gebruiken van de app. Activiteiten en de gebeurtenissen worden opgenomen in een debuglogboek.

U/MA.1 o Logging

Criterium *(wie en wat)*	**Logging voor debugging** is **vóór inproductiename** uitgeschakeld en de logbestanden zijn verwijderd. Wanneer **statistische logging** over het gebruik van de app plaatsvindt, dan bevat deze geen persoonsgegevens.
Doelstelling *(waarom)*	Voorkomen dat onnodig gedetailleerde informatie over de werking van de app of over de gebruiker beschikbaar komt voor een aanvaller.
Risico	Gevoelige informatie of informatie over de zwakheden (in de beveiliging) van de app komt in handen van een aanvaller.
Classificatie	Laag

Maatregelen

Logging voor debugging

01 Schakel loggingmechanismen voor debugging zijn bij oplevering uit.

Logging voor debugging dient voor foutzoeken. Ter voorkoming van diverse privacy- en veiligheidsrisico's moet deze logging bij inproductiename van de app uitgeschakeld zijn.
Bedenk daarbij dat ook voor alle gebruikte bibliotheken de debuglogging moet zijn uitgeschakeld.

02 Registreer geen gevoelige informatie over de werking of de gebruiker.

Het advies is geen enkele informatie te loggen. Als de ontwikkelaar besluit wel informatie te loggen geeft hij aan de opdrachtgever aan dat daarin geen gevoelige informatie wordt meegenomen.
Indien wel informatie wordt gelogd, dan is hiervan nagegaan of dit gedetailleerde technische informatie vrijgeeft, waardoor de beveiliging van de app in gevaar zou kunnen komen. Ook het eventueel vrijkomen van privacygevoelige informatie over de gebruiker is getest en uitgesloten.

03 Voorkom dat de app bij oplevering logbestanden bevat.

04 Schakel de functies voor de opslag van crashlogs en dumps uit.

Voorkom dat bij het vastlopen van de app een crashlog of geheugendump wordt gemaakt. Bij voorkeur wordt een foutmelding afgehandeld door de app.

05 Voorkom dat de app bij oplevering testdata bevat.

Testdata kan zich bijvoorbeeld bevinden in bestanden met de extenties .ipa, .apk en .jar.

Vóór inproductiename

06 Voorkom dat de app die door de ontwikkelaar of de tester aan de gebruiker wordt aangeboden logbestanden of functies ten behoeve van logging bevat.

Door alle log- en debugoperaties te koppelen aan de release-configuratie wordt in de app, als deze voor productie wordt gebouwd, automatisch alle code voor log en debug-operaties uitgeschakeld. Dit voorkomt dat per ongeluk toch debugcode en loginstructies in de productieversie belanden.

07 Zorg ervoor dat apps die ten behoeve van test en acceptatie worden aangeboden aan een bredere groep gebruikers alleen die functies ten behoeve van logging bevatten, waarvan vastligt dat deze noodzakelijk zijn voor test en acceptatie (beperk logging in productie tot een minimum, zie hiervoor maatregel 02).

In specifieke gevallen kan het nuttig zijn logging-functies ten behoeve van test en acceptatie beschikbaar te hebben. Dit ligt dan in een testplan vast.

08 Controleer het aanwezig zijn van (informatie in) logbestanden bij oplevering, voorafgaand aan het openbaar aanbieden van de app in de appstore.

09 Zorg ervoor dat van een meegeleverd toetsenbord bekend is dat het geen inloggegevens, financiële informatie of andere vertrouwelijke informatie vastlegt.

Statistische logging

10 Laat de app en eventueel de third-partyprogrammabibliotheken in de logbestanden geen gevoelige informatie opnemen.

Logging van activiteiten en gebeurtenissen voor functionele doeleinden ten behoeve van de gebruiker zijn gebonden aan de eisen die voor opslag gelden.

11 Zorg ervoor dat logging minimaal is en geen persoonsgegevens bevat.

Statistische informatie over het gebruik van de app bevat nooit naar personen herleidbare informatie.
Houd er rekening mee dat het loggen van ogenschijnlijk onschuldige requests ook kan resulteren in het lekken van gebruikersgegevens, via bijvoorbeeld de querystring.

U/MA.11 Transportversleuteling

Encryptie van het transport van gegevens tussen het serverdeel van de applicatie en het werkplekdeel van de applicatie beschermt vertrouwelijke gegevens. Mobiele apparaten maken vaak gebruik van open en daarmee onveilige wifi-netwerken. De communicatie is daardoor gevoelig voor man-in-the-middle-aanvallen. Het afschermen door versleuteling van de sessie door middel van TLS, ook wanneer de uitgewisselde informatie niet vertrouwelijk is, is tegenwoordig een standaardvereiste voor het beveiligen voor mobiele apparaten.

Encryptie is nodig over netwerken die als niet veilig moeten worden bestempeld. Onveilige netwerken zijn netwerken die niet afgeschermd zijn tegen ongeautoriseerde toegang. Dit zijn ook bedrijfsnetwerken op kantoren die niet aantoonbaar fysiek en elektronisch zijn afgeschermd. Omdat mobiele apparaten ook via niet-veilige en publieke netwerken worden gebruikt geldt voor de apps dat zij gebruikmaken van met encryptie beveiligde communicatie.

U/MA.11	Sessieversleuteling
Criterium *(wie en wat)*	De applicatie past **encryptie** toe op alle **communicatie** via netwerken.
Doelstelling *(waarom)*	Het waarborgen van de vertrouwelijkheid en integriteit van gegevensleveringen en transacties.
Risico	De uitgewisselde informatie komt onder invloed (lezen en muteren) van een aanvaller.
Classificatie	Hoog

Maatregelen

Encryptie

01 Versleutel de communicatie tussen de app en de server.

02 Gebruik alleen protocollen en cryptografische technieken die als veilig worden bestempeld.
Van de gebruikte versies zijn binnen het vakgebied geen zwakheden bekend die aantoonbaar een bedreiging vormen.
Op X.509-certificaten (of vergelijkbaar) gebaseerde encryptie biedt momenteel in de meeste gevallen voldoende bescherming.
Richtlijnen voor het veilig inzetten van TLS worden besproken in de ICT-beveiligingsrichtlijnen voor Transport Layer Security (TLS). [6]

03 Versleutel bij de aanroep van de diensten alle informatie in de aanroep.
Dit geldt bijvoorbeeld voor privacygevoelige informatie in url's en cookies.

Communicatie

04 Versleutel standaard alle communicatie over het netwerk.
Er wordt van uitgegaan dat communicatie over niet-beveiligde netwerken kan plaatvinden. Houd er rekening mee dat er altijd door de gebruiker ook onbedoeld een onveilig netwerk gekozen kan worden.

U/MA.12 Certificaatpinning

Eén van de belangrijkste veiligheidsmaatregelen voor apps is de beveiliging van de communicatie met de server door middel van versleuteling (U/MA.11).

Het uitgeven van certificaten voor versleuteling verloopt via een vertrouwde Public Key Infrastructure (PKI), waarbij een certificaat worden uitgegeven door een certificaatautoriteit (CA). Meestal is er een root-CA en verschillende intermediate CA's die de certificaten uitgeven. Het uiteindelijk verstrekte certificaat bevat het specifieke domein waarvoor het certificaat is verstrekt.

Zolang een certificaatautoriteit niet gecompromitteerd is, zijn de door hem uitgegeven certificaten veilig. De praktijk laat echter zien dat ook CA's gecompromitteerd kunnen raken. Het is daarom nodig om een aanvullende beveiligingsmaatregel toe te passen: certificaatpinning.

Bij certificaatverificatie wordt door de app gecontroleerd of het gebruikte certificaat door een vertrouwde certificaatautoriteit is uitgegeven en of het certificaat ongewijzigd (niet gecompromitteerd) is. Met certificaatpinning wordt daarbij het certificaat gevalideerd tegen één specifieke CA of eindcertificaat. Hiermee wordt voorkomen dat door een hack bij een niet-gebruikte CA toch een risico kan optreden. Als het certificaat door een niet-vertrouwde certificaatautoriteit is uitgegeven kan de app de verbinding weigeren.

U/MA.12 Certificaatpinning

Criterium *(wie en wat)*	De app controleert tijdens het opzetten van een versleutelde verbinding of het servercertificaat **vertrouwd** is en neemt de nodige **maatregelen**.
Doelstelling *(waarom)*	Het waarborgen van de vertrouwelijkheid, de integriteit en desgewenst de onweerlegbaarheid van gegevens en transacties door middel van een veilige sleutel c.q. veilig certificaat.
Risico	De uitgewisselde informatie komt onder invloed (lezen en muteren) van een aanvaller, hoewel gebruik wordt gemaakt van een versleutelde verbinding.
Classificatie	Midden

Maatregelen

Vertrouwd

01 Voorzie de app van certificaatpinning voor alle communicatie over het netwerk.

Standaard vindt certificaatpinning plaats door het certificaat hardcoded op te nemen in de app.

Maatregelen

02 Controleer het certificaat. Bij een onjuist certificaat wordt de communicatie gestopt.

03 Zorg voor procedures ingeval een certificaat onvertrouwd is geworden.

Dit houdt de gevolgen van het niet veilig zijn van het certificaat of de sleutel en daarmee van het niet veilig zijn van de communicatie of onbeschikbaarheid van de app beperkt. De ontwikkelaar heeft dan bijvoorbeeld een tweede certificaat van een andere certificaatautoriteit beschikbaar en gepind. Voor het niet meer vertrouwde certificaat wordt een nieuwe aangemaakt en gepind welke zo snel mogelijk als update van de app beschikbaar komt (zie U/MA.19).

U/MA.13 Hardening van de apps

Bij het hardenen van de app worden de communicatiemogelijkheden van de app tot een minimum (het strikt noodzakelijke) beperkt. Eén van de manieren om dit te bereiken is door onnodige interfaces te verwijderen of uit te schakelen. Door benodigde interfaces in kaart te brengen en vervolgens de afhankelijkheden te bepalen, kan een minimale lijst van interfaces worden samengesteld waarover de app moet beschikken. Alle overige interfaces kunnen weg. Bedenk dat niet-actieve maar wel aanwezige interfaces op een systeem uiteindelijk toch tot een kwetsbare app kunnen leiden. Het is daarom veiliger om het aanvalsoppervlak zo klein mogelijk te houden. Overbodige interfaces en toegangsrechten worden daarom bij voorkeur verwijderd.

Het is mogelijk om het aantal potentiële aanvallen te verminderen door de app te ontdoen van functionaliteit die niet gerelateerd en vereist (strikt noodzakelijk) is voor het functioneren van de app. Wanneer verwijderen niet mogelijk is, moet alle niet strikt noodzakelijke functionaliteit zijn uitgeschakeld.

De meeste bibliotheken die gebruikt worden bij de ontwikkeling van apps bevatten meer functionaliteit dan de app nodig heeft voor het doel waarvoor deze is ontwikkeld. Om te voorkomen dat onveilige (verouderde) protocollen of functies actief zijn, wordt bijgehouden welke interfaces worden gebruikt. Als onderdeel van U/MA.19 Up-to-date app wordt gewaarborgd dat de gebruikte versies van de app up-to-date zijn.

U/MA.13 Hardening van de apps

Criterium *(wie en wat)*	De ontwikkelaar waarborgt dat toegang tot de app alleen mogelijk is via **interacties** die **noodzakelijk** zijn voor het correct functioneren van de applicatie.
Doelstelling *(waarom)*	De app en de gegevens zijn beveiligd tegen extra kans op misbruik via zwakheden, door het beperken van de functionaliteit tot wat noodzakelijk is voor het correct functioneren van de app.
Risico	Een aanvaller breekt in via ongebruikte ingangen op de app, waardoor de vertrouwelijke gegevens en de app onder invloed komen van een aanvaller (lezen en muteren).
Classificatie	Hoog

Maatregelen

Interacties

01 Deactiveer programmacode, bibliotheken en componenten die niet worden gebruikt en verwijder deze waar mogelijk.

Een interface van de app voor communicatie buiten de app is alleen beschikbaar en toegankelijk indien het doel en daarmee het belang voor de gebruiker onderbouwd is.

02 Deactiveer niet-noodzakelijke protocollen of functies waarmee de app kan worden benaderd (bijvoorbeeld door andere apps) en verwijder deze waar mogelijk.

03 Verleen autorisaties alleen als dat noodzakelijk is voor het correct functioneren van de app.

Noodzakelijk

04 Gebruik interacties uitsluitend wanneer het doel ervan en daarmee het belang voor de gebruiker onderbouwd is.

Na de bouw wordt bij iedere wijziging van de app getoetst of de gewijzigde app niet meer dan de vanuit het ontwerp noodzakelijke interactiemogelijkheden biedt.

05 Stel de app beschikbaar met een actueel overzicht van de noodzakelijke interactiemogelijkheden en lever daarbij de onderbouwing.

06 Stel de app beschikbaar met een actueel overzicht van de gebruikte bibliotheken.

07 Voorzie de app van de meest recente, relevante versies van de communicatiemogelijkheden.

08 Gebruik geen verouderde functionaliteit in de app.

Verouderde ('deprecated') functionaliteit is software die door de oorspronkelijke ontwikkelaar (of op de platforms voor ontwikkelaars) wordt afgekeurd, bijvoorbeeld doordat het kwetsbaarheden bevat of tot kwetsbaarheden in de werking van de app leidt.
Delen van API's (klassen of functies) kunnen afgekeurd zijn. Dat geldt voor API's van het platform, maar kan ook gelden voor third-partybibliotheken. De ontwikkelaar van die API raadt dan af om bepaalde functies nog te gebruiken. Meestal omdat die functies verouderd zijn en risico met zich meebrengen. Er moet rekening mee worden gehouden dat afgekeurde functionaliteit na verloop van tijd wordt verwijderd.
Als bepaalde functies afgekeurd zijn, zijn doorgaans nieuwe faciliteiten beschikbaar om de beoogde functionaliteit te kunnen realiseren. Daar waar afgekeurde code bij een oude versie aanwezig was wordt deze vervangen door de meest actuele versie of wordt deze functie niet meer gebruikt.

09 Maak een risicoanalyse indien verouderde functionaliteit in de app moet worden gebruikt, omdat anders oudere versies van het besturingssysteem niet meer ondersteund kunnen worden. Leg de resultaten van de analyse ter afweging en acceptatie aan de opdrachtgever voor.

Afhankelijk van de oudste versie van het besturingssysteem dat ondersteund moet worden, kan het zijn dat nieuwe functionaliteit pas vanaf een bepaalde API beschikbaar is. Voor het kunnen draaien op oudere versies is dan handhaven van de afgekeurde logica de enige manier. Het is een risicoafweging of zowel de oude als de nieuwe manier ondersteund moet worden, of dat ervoor gekozen wordt de API en daarmee een andere versie van het besturingssysteem te vereisen, waarin de betreffende nieuwe functionaliteit beschikbaar is gekomen.

Verdieping

Interacties kunnen synchroon of asynchroon werken, op de achtergrond beschikbaar zijn of een interactie met de gebruiker bieden.

- Services: een service is een onderdeel van een app dat op de achtergrond draait zonder dat er interactie nodig is met de gebruiker en dat de functionaliteit levert voor interactie met andere toepassingen. Services kunnen andere achtergrondprocessen starten en via zogenaamde contentproviders gegevens uitwisselen met andere apps. Naast services zijn er binnen Android ook zogenaamde intents, waarmee asynchrone communicatie mogelijk is.
- Publish/subscribe services: dit zijn services die in vergelijking met de gewone services zeer geschikt zijn voor vraag- en antwoordinteracties in een (al dan niet over meerdere apparaten gedistribueerd) systeem waarin apps/services niet worden aangeboden door één partij. De vragen en antwoorden gaan via knooppunten die vraag en antwoord op elkaar afstemmen. Een voorbeeld bij Android hiervan is een 'broadcast receiver'. Bij iOS is dat een 'Notification center service'.

U/MA.1 4 Least privilege voor andere apps

Risico's van systeemmisbruik kunnen aanzienlijk worden verminderd door rechten op de app te beperken. Gangbare principes voor het beleid zijn bijvoorbeeld gebaseerd op 'standaard geen toegang', 'least privilege' en 'need-to-know'. Dit geldt voor gebruikers, maar ook voor apps onderling. Volgens het principe van 'least privilege' worden de rechten op een app gelimiteerd tot de minimale set van rechten die nodig is voor het correct functioneren.

Bij een remote hack van de app kunnen aanvallers alles doen wat de app ook mag. Heeft de app bijvoorbeeld onnodig rechten voor het gebruik van de camera, dan kunnen aanvallers dit overbodige recht misbruiken om op afstand foto's en video's maken. Beperk de rechten dus tot het minimale om het risicoprofiel van de app zo laag mogelijk te houden. In het ontwerp van de applicatie moet daarom rekening gehouden zijn met het principe van least privilege.

U/MA.1 4 Least privilege voor andere apps

Criterium *(wie en wat)*	De **toegangsrechten** van de app tot functies en data zijn alleen uitgegeven waar zij het onderbouwde doel en belang van de gebruiker dienen.
Doelstelling *(waarom)*	Een andere app en gebruikers krijgen alleen die rechten op een app die noodzakelijk zijn voor de werking van de app.
Risico	Een aanvaller, gebruiker of gecompromitteerde app maakt misbruik van rechten, waardoor de werking van de app en de gevoelige informatie onder invloed komt van onbevoegden (lezen en muteren).
Classificatie	Midden

Maatregelen

Toegangsrechten

01 Houd in het ontwerp rekening met het voorkomen van het toekennen van onnodige rechten aan andere apps.

In het ontwerp kan al voorkomen worden dat rechten onnodig worden afgegeven aan andere apps door de benodigde rechten vast te stellen en zo de minst benodigde te bepalen.

02 Stel op basis van een risicoanalyse vast welke toegangsrechten verleend moeten worden die noodzakelijk zijn voor een correcte werking van de app.

03 Maak de technische interface van de app door middel van toegangsrechten uitsluitend voor vertrouwde apps toegankelijk.

04 Ken autorisaties voor gebruik van een functie en/of toegang tot data van de app expliciet toe.

De door de app benodigde rechten worden bijgehouden in een app manifest-bestand in het permissies-element. Zo wordt voorkomen dat een onveilige app toegang krijgt tot informatie die door de gebruiker wordt ingevoerd in de gebruikersinterface van de app.

U/MA.1 5 Invoernormalisatie

Zoals alle software zijn ook apps sterk afhankelijk van allerlei invoer, zoals de invoer door de gebruiker, data ontvangen van externe servers, andere apps en lokale bestanden. Apps zijn daarbij sterk afhankelijk van internetstandaarden zoals JSON, XML, SQL, html en JavaScript. Invoer kan karakters of commando's bevatten die de werking van de app beïnvloeden. Deze invoer voldoet dan niet aan de regels voor een veilige invoer.

Zoals (web-)applicaties aan de serverzijde moeten worden afgeschermd, zo moeten ook apps worden afgeschermd. Wanneer de app kwaadaardige invoer niet goed afhandelt, kan met bepaalde invoer toegang worden verkregen tot de gegevens die door de app moeten worden afgeschermd.

Normaliseren van inhoud betekent dat de inhoud gaat voldoen aan een aantal beperkende regels. Via normalisatie voorkomt de app dat malafide verzoeken abusievelijk de filters voor validatie kunnen passeren. Bovendien wordt de invoer in een zodanige representatie gebracht dat deze op alle plaatsen in de app veilig verwerkt kan worden (eliminatie van SQL- en html-injecties). Normalisatie staat ook wel bekend als anti-evasion of canonicalization.

De app ontvangt invoer van de gebruiker, van andere apps, de backend en services op het mobiele apparaat en via het netwerk. Deze invoer kan verschillende vormen hebben. De app moet de invoer normaliseren.

U/MA.1 5 Invoernormalisatie

Criterium *(wie en wat)*	De app voorkomt manipulatie door alle **ontvangen invoer** te **normaliseren**, voordat deze invoer wordt verwerkt.
Doelstelling *(waarom)*	Voorkomen van inzage, wijziging, verlies en misbruik van gegevens via daartoe geëigende karakters of commando's in de invoer van de app.
Risico	Aanvallers kunnen de werking van de app beïnvloeden en vertrouwelijke gegevens bemachtigen, muteren of toevoegen.
Classificatie	Hoog

Maatregelen

Ontvangen invoer

01 Inventariseer alle ingangen van de app.
De ontwikkelaar houdt een inventaris bij van alle plaatsen waar de app invoer kan verwachten.

02 Pas normalisatie toe op alle ingangen, van gebruikersinterfaces, andere apps en bestanden op het mobiele apparaat tot externe netwerkbronnen.
Naast antwoorden van de serverzijde die veranderd zijn door een man-in-the-middle of serverhack kan ook de invoer van de gebruiker afkomstig zijn van een aanvaller. Een aanvaller kan zich immers fysieke toegang hebben verschaft tot het mobiele apparaat. Ook andere apps, intents of bestanden op het mobiele apparaat, inclusief de SD-kaart, of communicatie op openbare locaties kunnen gemanipuleerde input genereren. Daarom moet op alle ingangen normalisatie plaatsvinden.

03 Whitelist alle vertrouwde bronnen in de app.
Op de ingangen wordt gecontroleerd of de bron een vertrouwde bron is.

Normaliseren

04 Verzorg normalisatie naar wat nodig is voor het type invoer, onder andere (maar in ieder geval):
- het omzetten van null-karakters naar spaties;
- het normaliseren van padverwijzingen zoals '/./' en '/../';
- het verwijderen van overbodige spaties en regeleinden;
- het verwijderen van onnodige witruimtes;
- het omzetten van hoofdletters naar kleine letters of andersom.

Maak ten minste gebruik van de daarvoor bestaande standaardservices. Converteer alle risicovolle tekens naar een veilig formaat door middel van escaping. Houd rekening met verschillen tussen tekensets, bijvoorbeeld ASCII en UTF-8.
Als voorbeeld: De escape (\) voorafgaand aan een teken geeft aan dat het teken letterlijk moet worden geïnterpreteerd, \" wordt ". Escaping kan ook door gebruik te maken van de hexadecimale waarde van een teken, \x22 wordt ". Dit levert echter niet de gewenste tekst op indien deze wordt geïnterpreteerd met een ASCII-incompatibele tekenset.
Tekens uit de invoer die verwerkbaar en niet ongewenst zijn kunnen nog steeds risicovol zijn bij het gebruik hiervan binnen de programmalogica. Risicovolle tekens kunnen onderdeel uitmaken van legitieme invoer. De plaatsnaam 's-Hertogenbosch leidt dan bijvoorbeeld tot een syntactisch incorrecte query. Door een escape voor de apostrof te plaatsen, beschouwt de database de apostrof als onderdeel van de invoerstring en niet als onderdeel van de query. Veel programmeertalen ondersteunen standaardservices voor het escapen van gevaarlijke tekens.
Soortgelijke conversies gelden voor bijvoorbeeld html, XML, etc. Voer escaping uit op de invoer na het toepassen van whitelists en eventueel blacklists. Escaping moet toegespitst zijn op de programmaonderdelen waar invoer wordt verwerkt.

U/MA.1 6 Invoervalidatie

De belangrijkste vuistregel voor invoer in een app is dat de applicatie geen enkele invoer mag vertrouwen en daarom alle invoer moet valideren op juistheid, volledigheid en geldigheid. Daarbij dient de invoer minimaal gevalideerd te worden op waarden die buiten het geldige bereik vallen (grenswaarden), ongeldige tekens, ontbrekende of onvolledige gegevens, gegevens die niet aan het juiste formaat voldoen en inconsistentie van gegevens ten opzichte van andere gegevens binnen de invoer dan wel in andere gegevensbestanden. Niet-vertrouwde invoer kan afkomstig zijn van allerlei toegangspaden tot de app, zoals de intents, de services, het netwerkverkeer, binding-interfaces en toegang tot bestanden. Invoervalidatie is de belangrijkste voorwaarde voor betrouwbare gegevensverwerking en ongeldige invoer moet door de app worden geweigerd.

Valideren van de inhoud zorgt ervoor dat alleen geldige gegevens verwerkt worden. Validatie vindt zowel op protocolniveau (meestal http) als op applicatieniveau plaats. Het doel is te voorkomen dat de software op applicatieniveau misbruikt wordt of faalt door invoer van de gebruiker.

U/MA.1 6 Invoervalidatie

Criterium *(wie en wat)*	De app voorkomt de mogelijkheid tot manipulatie door alle **ontvangen invoer** te **valideren**, voordat deze invoer wordt verwerkt.
Doelstelling *(waarom)*	Voorkomen van manipulatie van de app via de invoer van de app.
Risico	Aanvallers kunnen de werking van de app beïnvloeden en vertrouwelijke gegevens bemachtigen, muteren of toevoegen.
Classificatie	Hoog

Maatregelen

Ontvangen invoer

01 Inventariseer alle ingangen van de app.
De app ontvangt invoer van de gebruiker, andere apps, de backend en services op het mobiele apparaat en via het netwerk. Deze invoer kan verschillende vormen hebben.
Naast antwoorden van de server die veranderd zijn door een man-in-the-middle of een serverhack, kan ook de gebruikersinvoer afkomstig zijn van een aanvaller. Een aanvaller kan zich immers fysieke toegang hebben verschaft tot het mobiele apparaat. Ook andere apps, waaronder intents of bestanden op het mobiele apparaat, inclusief de SD-kaart, en communicatie op openbare locaties kunnen een gemanipuleerde input genereren.

02 Valideer op alle ingangen de invoer die afkomstig is van gebruikersinterfaces, andere apps en bestanden op het mobiele apparaat, en vanaf externe netwerkbronnen.

03 Whitelist alle vertrouwde bronnen in de app.
Als een app wel invoervalidatie en filtering uitvoert, blijkt deze filtering vaak niet voldoende effectief om alle mogelijke aanvallen op de app te blokkeren. Dit is voornamelijk het geval op het moment dat de app gebruik maakt van blacklisting. Bij blacklisting moeten de gevaarlijke bronnen bekend zijn om ze te blacklisten. Bij whitelisting hoeven niet alle gevaarlijke bronnen bekend te zijn, maar worden alleen vertrouwde bronnen toegelaten.

Valideren

04 Valideer alle invoer.
Valideer de inhoud van een http-request op basis van verwerkbare invoer (whitelist). Valideer de invoer op malafide sleutelwoorden, tekens en patronen (blacklist).

05 Weiger foute, ongeldige of verboden invoer.

06 Voer WebView-aanroepen pas uit als de invoervalidatie is uitgevoerd.
WebViews zorgen er voor dat apps inhoud van online bronnen binnen de app kunnen openen.
WebViews zijn qua risico's in feite vergelijkbaar met webbrowsers, doordat ze kwetsbaar zijn voor allerlei browsergerelateerde acties zoals origin-policy-bypasses, JavaScriptparser-bufferoverflows en cross-site scripting.
Vanuit het oogpunt van informatiebeveiliging vormen WebViews risico's die niet gemakkelijk opgelost kunnen worden. Deze risico's leiden ertoe dat een aanvaller in staat is om in de WebView geladen informatie onder controle te krijgen. Dit kan zich op allerlei manieren voordoen, zoals man-in-the-middle, client-side injectie of met cross-site scripting.

07 Whitelist de toegestane methoden in en bronnen voor de WebViews.

08 Gebruik indien JavaScript nodig is addJavaScriptInterface() voor alleen de webpagina's van waaruit alle invoer betrouwbaar is.
In het algemeen is alleen de JavaScript van de eigen app betrouwbaar.

Verdieping

Ongecontroleerde (niet-gevalideerde) invoer van gebruikers is een belangrijke bedreiging voor een app. Als invoer van gebruikers rechtstreeks wordt gebruikt in onder meer html-uitvoer, cookie-waarden of SQL-query's, dan is de kans groot dat een kwaadwillende de app compromitteert. Een gebrek aan invoervalidatie kan tot cross-site scripting-, commando- en SQL-injectie-kwetsbaarheden leiden:

- Cross-site scripting (XSS): met XSS kunnen aanvallers, naast de traditionele XSS-exploits, door WebViews gebruikte applicatiecode starten.

- SQL-injectie: bij gebruik van een lokale database maakt SQL-injectie het mogelijk om de lokaal opgeslagen data te lezen, te wijzigen en te verwijderen.
- Commando-injectie: hierbij wordt door het manipuleren van de invoer voor een commando de logica van de app gewijzigd, waardoor het mogelijk wordt om de lokaal opgeslagen data te lezen, te wijzigen en te verwijderen.

Er geldt een aantal uitgangspunten met betrekking tot invoer bij het ontwikkelen van apps, deze zijn:

a. Andere apps zijn niet vertrouwd, dus de invoer die ervandaan komt ook niet.
b. De invoer die niet aan één of meerdere controles voldoet wordt verwijderd of geweigerd. Alleen de lettertekens die voorkomen in een vooraf gedefinieerde lijst worden toegelaten. Deze aanpak wordt ook wel aangeduid als de whitelistaanpak.

In het ontwikkelproces van de app zal de software expliciet op een correcte invulling van deze uitgangspunten onderzocht moeten worden. Dit vraagt om uitgebreide testen of gerichte codereviews.

SQL-injectie
Bij het ontwikkelen van de apps kan informatie lokaal op verschillende manieren worden opgeslagen, zoals in een SQLite-database. In het algemeen zijn dergelijke databases alleen toegankelijk voor de app waarvoor ze zijn ingezet, of alleen vanuit een specifieke sandbox.

Voor het delen van gegevens tussen apps biedt het model met een contentprovider de aanbevolen manier. De interfaces die gebruik maken van SQL zijn gevoelig voor SQL-injecties, indien deze niet veilig worden gebruikt. De interfaces kunnen bestaan uit native code in de apps of uit webcomponenten (html5).

Met SQL-injectie kan een aanvaller de input van toelaatbare SQL-opdrachten uitbreiden met eigen SQL-opdrachten. Daardoor kan de aanvaller mogelijk in de database gegevens creëren, wijzigen, bewerken, lezen of verwijderen. Zulke niet-vertrouwde input kan afkomstig zijn van allerlei toegangspaden tot de app, zoals de intents, de services, het netwerkverkeer, binding-interfaces en toegang tot bestanden. Android-contentproviders en de SQLite Query Builder bieden volledige ondersteuning voor geparametriseerde query's. Door de parametrisering bestaat de input alleen nog uit de specifieke variabelen of parameters van de query. De code van de query zelf ligt vast in de app. Door de validatie van de ingevoerde variabelen en parameters wordt SQL-injectie voorkomen.

U/MA.17 Http-methoden

De back-endwebserver ondersteunt het http-protocol. Http kent methoden, headers en foutinformatie, die mogelijk misbruikt kunnen worden. Daarom is het gebruik hiervan beperkt tot het minimum dat noodzakelijk is voor een goede werking van de ontsloten apps.

Http 1.1 en 2.0 ondersteunen verschillende functies. In de praktijk gebruikt een app alleen de methoden GET en POST. Voor veel scripts en objecten geldt zelfs dat alleen GET nodig is. De overige functionaliteit is vrijwel nooit nodig binnen traditionele apps en vormen een extra beveiligingsrisico.

U/MA.17 Http-methoden

Criterium *(wie en wat)*	De app gebruikt alleen de **http-methoden** die nodig zijn voor het communiceren met andere apps en services, al dan niet over het netwerk.
Doelstelling *(waarom)*	Voorkomen van het gebruik van niet noodzakelijke methoden, die gebruikt kunnen worden voor de manipulatie van de logica van de app.
Risico	De werking van de app wordt gemanipuleerd, waardoor deze onder controle komt van een aanvaller.
Classificatie	Midden

Maatregelen

Http-methoden

01 Gebruik alleen GET en POST als http-methode. Onderbouw en beschrijf eventuele noodzakelijke methoden anders dan GET en POST en leg dit vast in de ontwerpdocumentatie.

In de ontwerpdocumentatie is vastgelegd:
- *welke http-requestmethoden (GET, POST) voor de ondersteunde apps benodigd zijn;*
- *welke informatie in de http-headers voor het functioneren van belang is;*
- *welke standaardfoutmeldingen worden getoond of verstuurd;*
- *op welke wijze bovenstaande is gerealiseerd, denk hierbij aan de configuratie van de webserver en, indien van toepassing, de applicatiefirewall.*

Methoden anders dan GET en POST zijn vrijwel nooit nodig binnen traditionele apps en vormen alleen een extra beveiligingsrisico (misbruik). Eventuele afwijkingen van bovenstaande die noodzakelijk zijn, omdat de app anders niet kan functioneren, zijn onderbouwd. Dit kan bijvoorbeeld het geval zijn bij het gebruik van PUT en DELETE wanneer REST gebruikt wordt.

In een CORS-preflight-scenario is het gebruik van de OPTIONS-header acceptabel. Belangrijk is dat CSRF-aanvallen worden voorkomen door het gebruik van expliciete Origins waar mogelijk. Indien geen expliciete Origins worden gebruikt is in een risicoanalyse bepaald, dat dit geen nadelige gevolgen heeft voor de beveiliging.

02 Leg in de configuratiedocumentatie vast welke http-methoden worden gebruikt.

U/MA.1 8 XML-externe-entiteitinjectie

XML is naast JSON een veelgebruikt formaat om gegevens in te lezen in de app. De op het XML-formaat gebaseerde invoer bevat gegevens, waar in een bepaalde structuur codes omheen staan: bij XML worden de entiteiten (gegevens) weergegeven tussen een openings- en een sluittag.

Met de codes wordt de structuur en betekenis van de gegevens bepaald. Door de beschrijving van structuur en de betekenis van gegevens kunnen de gegevens op een aantal manieren hergebruikt worden. Deze mogelijkheid van portabiliteit heeft XML tot één van de meestgebruikte technologieën voor het uitwisselen van gegevens gemaakt. XML wordt daarom veel gebruikt in apps.

Voor het uitlezen van de gegevens uit de XML-invoer wordt veelal gebruik gemaakt van parsers. Iedere parser is gevoelig voor aanvallen door middel van externe-entiteitinjectie (XXE) als de juiste maatregelen niet zijn getroffen.

U/MA.1 8 XML-externe-entiteitinjectie

Criterium *(wie en wat)*	De app beperkt de mogelijkheid tot manipulatie door alle **externe XML-invoer** te beschermen tegen **entiteitinjectie**.
Doelstelling *(waarom)*	Voorkomen van informatielekkage of een DoS-aanval door manipulatie van de externe invoer die op XML is gebaseerd.
Risico	Een aanvaller leest interne gegevens van de app uit of maakt de app onbeschikbaar voor de gebruiker.
Classificatie	Laag

Maatregelen

Externe XML-invoer

01 Valideer alle invoer die niet via een vertrouwde bron komt.

02 Weiger foute, ongeldige of verboden invoer.

Entiteitinjectie

03 Zet de entityresolver uit bij het aanroepen van de parser voor externe XML-bronnen. Zodoende zijn dan het parsen van de namespace en documentdefinities uitgeschakeld.

De entiteiten in XML kunnen eindeloos in elkaar worden genest. Bij XXE kan het eindeloos nesten leiden tot een denial-of-service van de parser. De DoS wordt veroorzaakt doordat het probleem niet in de parser lokaal wordt opgelost, maar leidt tot extra netwerkverkeer voor het steeds weer benaderen van de XML-bron.

U/MA.1 9 Up-to-date apps

Oudere versies van apps kunnen kwetsbaarheden bevatten die veelal breed bij aanvallers bekend zijn en die kunnen worden misbruikt. Het up-to-date houden van apps vormt daarmee een belangrijke voorwaarde voor het veilig houden van de app.

U/MA.1 9	Up-to-date apps
Criterium *(wie en wat)*	De leverancier past **lifecyclemanagement** toe op de apps die hij levert, waardoor gebruikers altijd over de **veiligste versie** kunnen beschikken.
Doelstelling *(waarom)*	Beperken dat onveilige versies worden gebruikt.
Risico	Bekende kwetsbaarheden in oude versies worden misbruikt door een aanvaller.
Classificatie	Hoog

Maatregelen

Lifecyclemanagement

01 Maak als opdrachtgever afspraken met de ontwikkelaar over het up-to-date houden van de app.

Hierbij wordt rekening gehouden met de verwachte levensduur van de app.

02 Richt lifecyclemanagement in voor de app: van het constateren of detecteren van een dreiging tot de installatie en het opruimen van de app en bijbehorende informatie.

De kwaliteit van het lifecyclemanagementproces komt tot uiting in de tijd die verstrijkt vanaf het moment van signaleren van een dreiging tot het uitbrengen van een patch.
Een app die is ontwikkeld volgens de geldende vereisten kan als gevolg van nieuwe ontwikkelingen onveilig worden. Een mechanisme voor gedwongen updates kan ervoor zorgen dat gebruikers werken met de meest actuele versie.

03 Zorg voor een risicogebaseerd lifecyclemanagementproces. Houd hierbij rekening met de niveaus van vertrouwelijkheid en integriteit waaraan de app moet voldoen.

De niveaus van vertrouwelijkheid en de robuustheid waarmee zij worden gerealiseerd en gehandhaafd zijn in het risicomanagementproces bepalend voor de vaststelling van de actuele veiligheid van de apps en services. Onveilige services, API's en apps worden verwijderd (of de appkey wordt verwijderd) en services en apps worden vervangen door nieuwe.
Bedenk dat de appstore niet vanzelfsprekend hogere eisen stelt aan het publiceren van apps die moeten voldoen aan een hoge vertrouwelijkheid. Voor dergelijke apps moet daarom vooraf nagegaan worden of dit extra maatregelen vereist.
Controleer het functioneren van het risicomanagementproces met periodieke audits.

04 Dwing updates bij voorkeur af als een update beschikbaar is. Minimaal krijgt de gebruiker een melding als een update beschikbaar is.

Bijvoorbeeld door bij opstarten van de app een waarschuwing en update-verzoek te tonen.

Veiligste versie

05 Gebruik een up-to-date versie van alle (third-party)bibliotheken, waarvan geen kwetsbaarheden bekend zijn.

06 Sluit ~~actief aan bij platforms waar dreigingen worden gemonitord om te anticiperen op~~ aanvallen en voorbereid en in staat te zijn deze af te weren.

07 Dwing de nieuwste versie af voor zowel de app zelf, als de third-partyservices, -API's en -apps die voor de app in kwestie worden gebruikt.

Bij het borgen van de veiligheid wordt gekeken naar:
- *contractuele garanties;*
- *conformiteit met de beveiligingseisen;*
- *de toegang (least privilege);*
- *informatie die gedeeld wordt (de transacties via de API's).*

Bijlage A **Referenties**

[1] NCSC: ICT-beveiligingsrichtlijnen voor webapplicaties.
https://www.ncsc.nl/actueel/whitepapers/ict-beveiligingsrichtlijnen-voor-webapplicaties.html

[2] NCSC: Beveiligingsrichtlijnen voor mobiele apparaten.
https://www.ncsc.nl/actueel/whitepapers/beveiligingsrichtlijnen-voor-mobiele-apparaten.html

[3] NCSC: Beleids- en beheersrichtlijnen voor de ontwikkeling van veilige software.
https://www.ncsc.nl/actueel/whitepapers/beleids--en-beheersingsrichtlijnen-voor-de-ontwikkeling-van-veilige-software.html

[4] W. Tewarie: SIVA – Methodiek voor de ontwikkeling van auditreferentiekaders.

[5] CIP: Grip op SSD, de Normen voor Mobiele Apps.
https://www.cip-overheid.nl/downloads/grip-op-ssd/

[6] NCSC: ICT-beveiligingsrichtlijnen voor Transport Layer Security (TLS).
https://www.ncsc.nl/actueel/whitepapers/ict-beveiligingsrichtlijnen-voor-transport-layer-security-tls.html

Uitgave
Nationaal Cyber Security Centrum (NCSC)
Postbus 117, 2501 CC Den Haag
Turfmarkt 147, 2511 DP Den Haag
070 751 5555

Meer informatie
www.ncsc.nl
richtlijnen@ncsc.nl
@ncsc_nl

December 2017