LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

VAKCINĀCIJAS SERTIFIKĀTU MOBILO LIETOTŅU DROŠĪBAS NOVĒRTĒŠANA

MAĢISTRA DARBS

Autors: Eduards Blumbergs
Stud. apl. Nr. sf30012
Darba vadītājs: Dr. dat. Pēteris Paikens

RĪGA 2022

ANOTĀCIJA

Maģistra darba tēma "Vakcinācijas sertifikātu mobilo lietotņu drošības novērtēšana" izvēlēta, ņemot vērā tās aktualitāti un saistībā ar izaicinājumiem, ar kuriem saskārusies Pasaules sabiedrība. Populārāko mobilo operētājsistēmu *Android* un *iOS* lietotņu veikalos ir pieejamas daudzas dažādas lietotnes vakcinācijas kvadrātkodu izmantošanai, tomēr nav izdarīts pētījums par šādu Latvijā izplatīto lietotņu kiberapdraudējumu un tehnisko drošības līmeni. Lai pārliecinātos, ka drošība netiek apdraudēta, un identificētu ievainojamības, ir veikta risinājumu potenciālo draudu modelēšana un tehniskās drošības pārbaudes, kā arī sniegti secinājumi un ieteikumi turpmākam darbam.

Maģistra darbs izpildīts Latvijas Universitātē 2022. g.

Atslēgvārdi: drošība, mobilā lietotne, tehniskā drošības pārbaude, programmatūras ievainojamības

ABSTRACT

The Master's thesis “Assessment of Security for Vaccination Certificate Mobile Applications” was chosen for its relevance and in the context of the challenges faced by the Global Society. There are many different apps available in the app stores of the most popular mobile operating systems *Android* and *iOS* for the purpose of using vaccination square codes, however, no research has been done on the cyber threats and technical security posture of such apps common in Latvia. Potential threat modelling and technical security testing of the solutions have been carried out to ensure that security is not being compromised and to identify the vulnerabilities, as well as the conclusions and suggestions for the future work.

Master's thesis completed at the University of Latvia in 2022.

Keywords: Keywords: security, mobile app, technical security testing, software vulnerabilities

AUTOREFERĀTS

Covid-19 ir pirmā pandēmija pasaules vēsturē, kurā situācijas kontrolei papildus ierastai vakcinācijai plaši izmanto mobilās ierīces un lietotnes, lai uzrādītu un pārbaudītu vakcinācijas sertifikātus. Šādu datu apstrāde arvien biežāk ir nepieciešamība ikdienišķās dzīves situācijās, tāpēc gan sertifikātu uzrādītājiem, gan pārbaudītājiem ir svarīgi, lai datu drošība nebūtu kompromitēta. Apple un Google lietotņu veikalos viegli atrast vairākas Latvijā populāras vakcinācijas sertifikātu kvadrātkodu apstrādei paredzētas lietotnes, bet nepieciešams gūt pārliecību par datu pieejamību, konfidencialitāti un integritāti.

Maģistra darba mērķis ir detalizēti izpētīt mobilo lietotņu tehniskās drošības prasības un aprobēt Latvijā populāru vakcinācijas sertifikātu lietotņu drošības līmeni, kā arī konstatēt iespējamās programmatūras ievainojamības.

Darba uzdevumi:

1. Izpētīt, analizēt un apkopot informāciju par mobilajām lietotnēm un to potenciālajām ievainojamībām.
2. Izanalizēt tīmeklī pieejamo izlūkošanas informāciju par COVID-19 vakcinācijas sertifikātu kvadrātkodu mobilajām lietotnēm.
3. Izveidot vakcinācijas sertifikātu lietotnes potenciālo draudu modeli.
4. Apkopot Latvijā izplatītāko COVID-19 vakcinācijas sertifikātu mobilo lietotņu sarakstu
5. Veikt tehnisko drošības pārbaudi un izanalizēt potenciālos draudus Latvijā izplatītākajās COVID-19 vakcinācijas sertifikātu mobilajās lietotnēs.

Darba gaitā autors pētīja literatūru par mobilo lietotņu vēsturi, informācijas drošību, mobilo lietotņu ievainojamību veidiem, draudu modelēšanu, risku izvērtēšanu un vakcinācijas sertifikātu lietotnēm (izmantoti 37 literatūras un informācijas avoti). Darba gaitā izveidots vakcinācijas lietotņu draudu modelis, atlasītas Latvijā izplatītākās lietotnes un veikta šo lietotņu tehniskā drošības līmeņa aprobācija. Darba gaitā iegūti pierādījumi par septiņu dažādu mobilo lietotņu ievainojamībām, kas uzstādītas *iOS* un *Android* operētājsistēmā, kā arī sagatavotas detalizētas konstatējumu atskaites.

Darba struktūru veido ievads, sešas nodaļas ar apakšnodaļām, secinājumi un turpmākā darba virzieni, literatūras un avotu saraksts, kā arī tīmeklī izvietotās detalizētās konstatējumu atskaites.

Darba apjoms: 59 lapas (neskaitot tīmeklī pieejamos 46 pielikumus), 24 attēli, 7 tabulas, izmantoti 37 literatūras un informācijas avoti.

Apzīmējumi un izmantotā terminoloģija

API — Programmas saskarne API (*Application Programming Interface*)

ATT&CK — Draudu novēršanas taktikas un paņēmienu zināšanu bāze

CAPEC — Kopējais uzbrukumu mehānismu uzskaites un klasifikācijas katalogs (*Common Attack Pattern Enumeration and Classification*)

CERT — datortrauksmes reaģēšanas komanda

CISA — Kiberdrošības un infrastruktūras drošības aģentūra

CKC — Kiberuzbrukuma ķēdes modelēšana (*Cyber Kill Chain*)

CNA — CVE numurēšanas autorizētās organizācijas *(*CVA *Numbering Authorities)*

COVID-19 — Koronavīrusu slimība ir infekcijas slimība, ko izraisa SARS-CoV-2 vīruss

CPE — Strukturēta informācijas tehnoloģiju sistēmu, programmatūras un pakešu nosaukumu shēma (*Common Platform Enumeration*)

CVE — Standarta atsauču sistēma CVE (*Common Vulnerabilities and Exposures*)

CVSS — Standarts ievainojamību nopietnības vērtēšanai CVSS (*Common Vulnerability Scoring System*)

CWE — Izplatītāko programmatūras trūkumu saraksts CWE (*Common Weakness Enumeration*)

DHS — ASV Iekšzemes drošības departaments

ENISA — Eiropas Savienības Kiberdrošības aģentūra (*European Union Agency for Cybersecurity*)

FIRST — Incidentu atbildes un drošības komandu globālais forums (*Forum of Incident Response and Security Teams*)

HTTP — Hiperteksta pārsūtīšanas protokols (*Hypertext Transfer Protocol*)

IEC — Starptautiskā elektrotehnikas komisija

ISO — Starptautiskā standartizācijas organizācija

MASVS — Mobilo ievainojamību risku apkopojumu standarts (*Mobile Application Security Verification Standard*)

MITRE — ASV valdības finansētā bezpeļņas korporācija

MSTG — Mobilo lietotņu drošības testēšanas gids (*Mobile Security Testing Guide*)

NIST — ASV Nacionālais standartu un tehnoloģijas institūts (*National Institute of Standarts and Technology*)

NVD — Ievainojamību pārvaldības datu standartu repozitorijs NVD (*National Vulnerability Database)*

OWASP — *Open Web Application Security Project*

PKI — Publiskās atslēgas infrastruktūra (*Public Key Infrastructure*)

URL — Vienotais resursu vietrādis URL (*Uniform Resource Locator*)

WAP — Bezvadu lietojumu protokols (*Wireless Application Protocol*)

SATURS

IEVADS

Gandrīz ikviena programmatūra savā dzīves ciklā saskaras ar dažādiem kiberdrošības apdraudējumiem, un, tehnoloģijai attīstoties, kiberuzbrukumu skaits tikai turpina pieaugt. Lietotāji savus datus uztic simtiem dažādu lietotņu, un tas rada lielu atbildību lietotņu programmatūras izstrādātājiem.

Mūsdienu tehnoloģijām ir daudz reālu pielietojumu COVID-19 pandēmijas seku novēršanai, bet tehnoloģiskais progress rada arī jaunus izaicinājumus. Datu aizsardzības pārkāpumu skaits nepārtraukti pieaug un rada izmaksas. Uzbrucēji izmanto datu zādzības, lai mērķēti kopētu un pārsūtītu sensitīvus datus, pēc tam tos izmantojot, piemēram, identitātes zādzībām. Datu zādzības notiek, apvienojot tādas metodes kā izspedējvīrusus un piegādes ķēžu uzbrukumus, un tās 2021. gadā ir ieņēmušas augstu vietu starp galvenajiem datu aizsardzības apdraudējumiem Eiropā. Uzbrukumi veselības aprūpes sistēmām izraisa datu aizsardzības pārkāpumu pieaugumu, bet uzbrucēji arvien vairāk izmanto dezinformāciju savos uzbrukumos. E-pasta krāpniecībā galvenā izmantotā tēma ir COVID-19 infekcija, kas ir kļuvusi arī par cilvēka izraisīto kļūdu galveno vājo punktu. Pandēmijas laikā ir notikusi sertifikātu kvadrātkodu viltošana un ļaunprātīga aizvietošana [5].

Infekcijas izplatības ierobežošana un COVID-19 vakcinācijas statusa pārbaudes nepieciešamība daudzās valstīs ir veicinājusi kvadrātkodu izmantošanas pieaugumu, izmantojot vakcinācijas sertifikātu mobilās lietotnes. Jau pašreiz populārāko mobilo operētājsistēmu *Android* un *iOS* lietotņu veikalos ir pieejamas daudzas dažādas lietotnes vakcinācijas kvadrātkodu izmantošanai, tomēr nav izdarīts pētījums par šādu Latvijā izplatīto lietotņu kiberapdraudējumu un tehnisko drošības līmeni. Lai pārliecinātos, vai drošība nav kompromitēta, un atklātu ievainojamības, nepieciešama izmantoto risinājumu draudu modelēšana un tehniskā drošības pārbaude.

Darba mērķis

Maģistra darbā sniegts pārskats par kiberdrošības apdraudējumiem mobilajās lietotnēs, īpašu uzmanību veltot lietotnēm, kuras pielieto vakcinācijas sertifikātu kvadrātkodu apstrādei. **Maģistra darba mērķis** ir detalizēti izpētīt mobilo lietotņu tehniskās drošības prasības un aprobēt Latvijā populāru vakcinācijas sertifikātu lietotņu drošības līmeni, kā arī konstatēt iespējamās programmatūras ievainojamības.

Lai sasniegtu darba mērķi, izvirzīti vairāki **uzdevumi**:

1. Izpētīt, analizēt un apkopot informāciju par mobilajām lietotnēm un to potenciālajām ievainojamībām.

2. Izanalizēt tīmeklī pieejamo izlūkošanas informāciju par COVID-19 vakcinācijas sertifikātu kvadrātkodu mobilajām lietotnēm.
3. Izveidot vakcinācijas sertifikātu lietotnes potenciālo draudu modeli.
4. Apkopot Latvijā izplatītāko COVID-19 vakcinācijas sertifikātu mobilo lietotņu sarakstu
5. Veikt tehnisko drošības pārbaudi un izanalizēt potenciālos draudus Latvijā izplatītākajās COVID-19 vakcinācijas sertifikātu mobilajās lietotnēs.

Darba metodes: literatūras izpēte, modelēšana, eksperiments, analīze.

Darba ierobežojumi

Autors vispārīgi identificē iespējamās ievainojamības un svarīgāko praksi, lai nodrošinātu COVID-19 vakcinācijas sertifikātu mobilo lietotņu drošību. Specifisku lietotņu ievainojamības aplūkotas tikai no melnās kastes testēšanas principa, neveicot pirmkoda kvalitātes analīzi. Darbā aplūkota ir datu aizsardzības perspektīva, nevis pašu vakcinācijas sertifikātu ģenerēšana. Aizmugursistēmu API un lietotņu saskarņu darbība ir apskatīta mērā, kādā tas bija iespējams bez izstrādātāju autorizācijas un piekļuves galvenajiem resursiem vai nelabvēlīgas sistēmu ietekmes. Vairums aprakstītā ir piemērojams visu veidu lietotnēm, tomēr plašāk apskatītas lietotnes, kas apstrādā Eiropas digitālo COVID-19 sertifikātu.

Ņemot vērā, ka darba ietvaros nav iegūta vakcinācijas sertifikātu lietotņu izstrādātāju piedāvāta iekšējā informācija vai izdarīta pilnvērtīga ielaušanās testēšana visā sistēmā, modelētos draudus var uzskatīt tikai par teorētiskiem. Pilnvērtīgai iespējamā riska draudu un rīcības ieteikumu novērtēšanai ir nepieciešams plašāks priekšstats par izveidoto risinājumu (infrastruktūru, plānotajām funkcijām, kā arī pieņemamā riska līmeni).

1. ANALĪTISKAIS APSKATS

1.1. Mobilo lietotņu vēsture un attīstība

Tehnoloģiskā attīstība notiek ātri. Lai gan mobilās lietotnes vēl pavisam nesen bija retums, tagad tās ir kļuvušas par mūsu dzīves neatņemamu sastāvdaļu. Mobilā saziņa ir tik ļoti ievijusies mūsu dzīvē, ka daudzi cilvēki jūtas neērti bez mobilā tālruņa.

Pirmās paaudzes mobilos tālruņus projektēja un izstrādāja tālruņu ražotāji. Konkurence bija sīva, un komercnoslēpumi tika rūpīgi sargāti. Ražotāji negribēja izpaust savu ierīču noslēpumus, tāpēc paši izstrādāja tālruņu programmatūru, un izstrādātāji, kas nepiederēja iekšējam lokam, nevarēja veidot tālruņu lietotnes.

Kādreiz populārākās tālruņu funkcijas bija zvanīšana un īsziņu sūtīšana. Mobilajiem tālruņiem varēja būt īpašas funkcijas vai režīmi (piemēram, kalkulators), bet lietotnes, kā tās tagad pazīst, parādījās tikai nesen kopā ar viedtālruņiem. Viedtālrunis ir daudzfunkcionāla ierīce, kas ne tikai nodrošina saziņu, bet arī palīdz mācīties, pelnīt un izklaidēties. Tas ir iespējams, pateicoties mobilo lietotņu attīstībai.

Pirmās mobilās lietotnes radās jau divdesmitā gadsimta beigās. Parasti tās bija nelielas arkādes spēlītes, zvana signālu redaktori, kalkulatori, kalendāri utt. Laika gaitā mobilo tālruņu lietotāji sāka pieprasīt vairāk funkciju, bet ražotājiem nebija ne motivācijas, ne resursu, lai izstrādātu dažādas papildu lietotnes. Bija nepieciešams veids, kā nodrošināt izklaides un informācijas pakalpojumus, nenodrošinot tiešu piekļuvi tālrunim. Kas gan šos pakalpojumus var nodrošināt labāk par internetu?

Izrādījās, ka tieša piekļuve internetam nav piemērota mobilajiem tālruņiem, jo līdz 1990. gadu beigām profesionālās tīmekļa vietnes bija pilnkrāsainas, apgādātas ar tekstu, attēliem un cita veida multivides materiāliem pārsvarā lietošanai tikai 800 × 600 pikseļu ekrānā, bet agrīnajiem tālruņiem bija ļoti mazi, zemas izšķirtspējas ekrāni, kā arī ierobežota atmiņa un datu apstrādes jauda. Arī mobilo datu pārraide lietotājiem izmaksāja pārāk dārgi un darbojās pārāk lēni.

Lai risinātu šīs problēmas, kompānijas *Motorola*, *Ericson* un *Nokia* izstrādāja bezvadu lietojumu protokolu (WAP). WAP izmantoja WML valodu, kas bija HTTP protokola saīsinātā versija, kā arī nodrošināja piekļuvi e-pastam, balss pastam, faksu saņemšanai, bankas transakcijām un nelielu tīmekļa lappušu aplūkošanai, ievērojot tālruņa atmiņas un joslas platuma ierobežojumus.

WAP bija lieliski piemērots tālruņu ražotājiem, jo pietika uzrakstīt vienu WAP pārlūkprogrammu un paļauties uz izstrādātājiem, ka tie piedāvās lietotājiem vēlamo saturu. Tas

bija izdevīgi arī sakaru operatoriem, kuri varēja pielāgot abonentiem paredzēto saturu un iekasēt ar pārlūkošanu saistītās augstās datu pārraides izmaksas. Tomēr bija viena problēma — ierobežotais saturs. Lielākā daļa agrīno WAP vietņu bija populāru ziņu aģentūru un operatoru zīmolu paplašinājumi. Lietotāji tālruņos varēja piekļūt ziņām, fona attēlu un zvana signālu katalogiem, bet WAP pārlūkprogrammas bija lēnas un grūti lietojamas. Garu URL vietrāžu ievade ar ciparu tastatūru bija sarežģīta un izaicinoša, turklāt lielākā daļa WAP vietņu ignorēja dažādu tālruņu specifikācijas, piemēram, ekrāna izmēru vai krāsu atbalstu. Izstrādātājs nevarēja pielāgot lietotāja pieredzi, tādējādi izveidotais rezultāts bija viduvējs un ne pārāk saistošs lietotājam.

Grafiski ietilpīgu videospēļu lietojumprogrammu rakstīšana, izmantojot WAP, bija gandrīz neiespējama. Ja jau tādas ierīces kā *Nintendo Game Boy* varēja nodrošināt stundām ilgu izklaidi, izmantojot tikai piecas pogas, tad kāpēc gan nepievienot tālruņa iespējas? Samazinoties atmiņas izmaksām un palielinoties akumulatoru ietilpībai, mobilajās iegultajās ierīcēs sāka izmantot samazinātas personālo datoru operētājsistēmu versijas (*Windows Mobile*). Tradicionālie datorprogrammatūras izstrādātāji ātri iesaistījās iegulto ierīču tirgū, piedāvājot viedtālruņu tehnoloģijām nepieciešamo programmatūru, bet tālruņu ražotāji aptvēra, ka jāmaina esošā protekcionisma politika attiecībā uz tālruņu dizainu un zināmā mērā jāatklāj to iekšējās darbības principi.

Radās dažādas patentētas viedtālruņu platformas, un izstrādātāji aktīvi sāka radīt tām nepieciešamās lietojumprogrammas. Atšķirībā no sākotnējās mobilo tālruņu programmēšanas vides, ražotāji viedtālruņu operētājsistēmas izveidoja atvērtas trešo pušu programmatūrai. Vienas no pirmajām bija *Palm OS* un *RIM Blackberry OS*. Kompānija *Sun Microsystems* izmantoja *Java* platformu *Java Micro Edition* (*Java ME*) izveidei. *Qualcomm* izstrādāja *Binary Runtime Environment for Wireless* (BREW), bet *Nokia*, *Sony Ericsson*, *Motorola* un *Samsung* izstrādāja *Symbian OS*. Kompānija *Apple* ar *iOS* pievienojās 2007. gadā, bet gadu vēlāk arī *Google Android*.

Pirmais komerciālais 3G mobilais bezvadu tīkls sāka darboties 2001. gadā [7], bet īstu izrāvienu šī tehnoloģija piedzīvoja 2008. gadā, kad kompānija *Apple* izlaida tālruni *iPhone 3G*, palielinot tīmekļa pieejamību un nozīmību un radot lietotājus, kurus vieno vienmēr tīmeklim pieslēgta ierīce.

Mūsdienu mobilo ierīču ražotāji cenšas padarīt savus produktus pievilcīgākus lietotājiem, ieviešot arvien vairāk lietotņu, kā nodrošinot izstrādātājiem programmatūras veikalus un peļņas iespējas. Organizācijas cenšas atvieglot un paātrināt izstrādi, lai lietotāji varētu iegūt pēc iespējas lielāku lietotņu klāstu savu ierīču pielāgošanai. Tomēr svarīga ir arī kvalitāte, un vienlaicīgi izstrādei jābūt vienkāršai un intuitīvai.

Lielākajai daļai platformu ir savas piesaistītās izstrādātāju programmas, kas nodrošina, ka izstrādātāju kopienas ir nelielas, pārbaudītas un tām ir līgumattiecības par to, ko tās drīkst un ko nedrīkst darīt. Dalība izstrādātāju programmās bieži ir ražotāju obligāta prasība, un izstrādātājiem par dalību tajās ir jāmaksā.

Saistībā ar dažādo platformu izplatību, mobilo tālruņu tirgus ir kļuvis arvien sadrumstalotāks. Tāpat arī mobilo ierīču izstrādātāju kopiena ir kļuvusi sadrumstalota. Izstrādātāji izmanto dažādas programmēšanas vides, rīkus un programmēšanas valodas. Lietotņu pārnešana starp platformām bieži vien ir dārga un laikietilpīga, bet mobilo tālruņu konfigurācijas un testēšanas prasību pārraudzīšana, dalība parakstīšanas un sertificēšanas programmās, operatoru attiecību vadīšana un lietotņu tirdzniecība ir kļuvušas par sarežģītām blakus uzņēmējdarbībām.

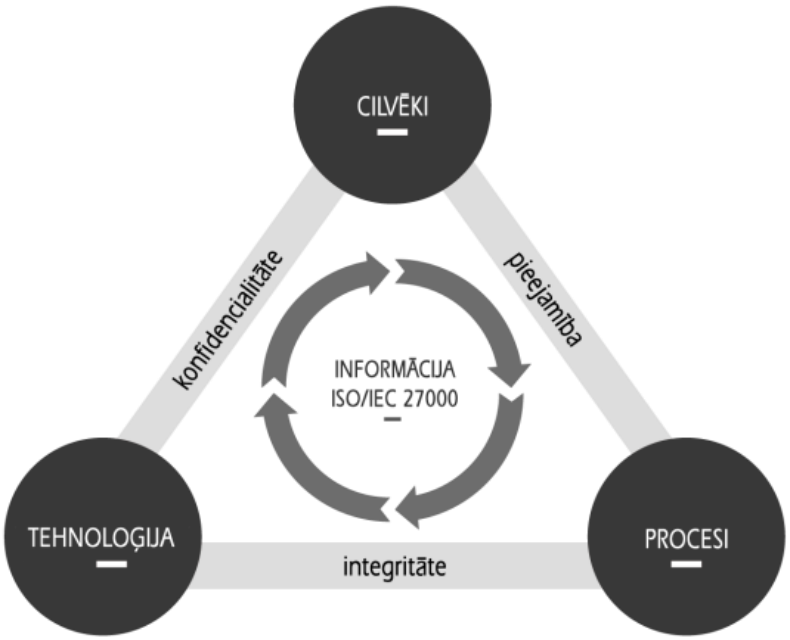
1.2. Drošības raksturojums

Vispārīgā gadījumā drošība ir aizsardzība no draudiem, briesmām, uzbrukuma, iejaukšanās un noklausīšanās. Informācijas tehnoloģiju jomā drošība ir informācijas aizsardzība, izmantojot tehnoloģiju, procesus un apmācību. Informācijas drošības pārvaldība ir aprakstīta ISO/IEC 27000 standarta dokumentos [10].

Drošība ir jebkuras informācijas sistēmas ļoti būtiska nefunkcionālā sastāvdaļa. Drošības pārkāpumi notiek gan iekšēju, gan ārēju iemeslu dēļ un izraisa postošas sekas. Sensitīvu, konfidenciālu vai aizsargātu datu nozagšana neuzticamā vidē var radīt lietotāju uzticības zaudēšanu un juridiskās problēmas. Tā kā ikvienas sistēmas drošības līmeni raksturo tās vājākais posms, drošība attiecināma uz visām sistēmas komponentēm, tajā skaitā fizisko infrastruktūru, aparatūru, tīklu, programmatūru, rīkiem, cilvēkresursiem, apakšuzņēmējiem un klientiem. Ja kādu komponenti ignorē, drošība kļūst apdraudēta.

Infrastruktūru aizsargā ar žogiem, aparatūru ar fiziskās piekļuves kontroli, tomēr grūtāk nosargāt ir programmatūru. Programmatūras problēmas rada bufera pārpildes, nepārbaudītu datu ievade un izvade, nepareiza konfigurācija, kodēšanas defekti, nedroša šifrēšana un datu pārtveršana, uzbrukumi lietotāja sesijām, vāji autentifikācijas mehānismi, nenomainītas vai vājas paroles un kodu injekcijas. Datu aizsardzības apdraudējumus izraisa kiberuzbrukumi, iekšējās datu noplūdes un netīša datu nozaudēšana vai izpaušana.

Apdraudējumu skaits arvien pieaug, un izmantotās uzbrukumu metodes ar katru dienu kļūst arvien izsmalcinātākas un daudzveidīgākas. Uzbrukumus izdara gan atsevišķi uzbrucēji, gan organizētas grupas un pat valdību aģentūras, kas vēlas iegūt nesankcionētu piekļuvi, izpaust sensitīvo informāciju, radīt datu zudumus vai dezinformēt.



1.1. att. **Drošības principi [9]**

Datoru drošībā klasificē trīs galvenos jēdzienus (1.1. att.):

1. Konfidencialitāti — aizsardzību pret neautorizētu (neatļautu) datu atklāšanu un datu avota autentiskuma pārbaudi;
2. Integritāti — aizsardzību pret datu izmainīšanu;
3. Pieejamību — aizsardzību pret datu aizkavēšanu vai dzēšanu.

Saistībā ar drošību apskata:

1. Drošības jautājumus:
   - klienta drošības problēmas — datu autentiskums, integritāte un konfidencialitāte;
   - servera drošības problēmas — datu privātums, pieejamība;
   - drošības politika, risku novērtējums.
2. Autentifikācijas (lietotāja identificēšanas) metodes:
   - kaut kas, ko mēs zinām (piemēram, parole);
   - kaut kas, kas mums ir (piemēram, biometrija);
   - kaut kas, kas mums pieder (piemēram, šifrēšanas sertifikāts);
   - vienfaktora autentifikācija — viens no iepriekš minētajiem veidiem;
   - divfaktoru autentifikācija — divi no iepriekš minētajiem veidiem.
3. Infrastruktūru (ugunsmūrus, starpniekserverus, ielaušanās noteikšanas un aizsardzības sistēmas u.c.).
4. Ievainojamības (programmatūras trūkumus, kas pieļauj pakalpojuma atteices uzbrukumus, vīrusus, tārpus u.c.).

Saskaņā ar ENISA apkopoto informāciju [5] 2021. gadā visnozīmīgākie drošības draudi informācijas tehnoloģijām ir:

- izspiedējprogrammatūra (šifrējošie vīrusi),
- ļaunatūra (neatļautu procesu izpilde pret konfidencialitāti, integritāti, pieejamību),
- kriptouzbrukums (upura datora resursu izmantošana kriptovalūtas ieguvei),
- e-pasta uzbrukums (pikšķerēšanas kampaņas),
- datu aizsardzības pārkāpums (datu izpaušana vai nozaudēšana),
- pieejamības un integritātes pārkāpums (pakalpojuma atteices uzbrukumi),
- dezinformācija (hibrīduzbrukumi, kas mazina vispārējo uzticēšanos kiberdrošībai) un
- nekaitnieciskie draudi (draudi, kurus izraisa nejaušas kļūdas un nepareiza konfigurācija).

1.3. Mobilo lietotņu vides

Mobilās lietojumprogrammas jeb lietotnes programmatūras mērķi var būt ļoti dažādi: pakalpojumi, veikali, izklaide, asistents utt. Visas lietotnes ir veidotas noteiktai mobilai platformai (šobrīd visbiežāk *iOS, Android*). Izvēlētās tehnoloģijas atsaucas uz to, ko lietotne spēj paveikt un cik tā ir droša.

Katra platforma izmanto citu programmēšanas valodu, piemēram, *iOS — Objective C, Swift, Android — Java, Kotlin.* Lai vieglāk un ātrāk pielāgotos dažādām ierīču izšķirtspējām un lietojumiem, izstrādātāji bieži vien veido lietotnes, izmantojot tīmekļa tehnoloģijas un reaģējošo tīmekļa dizainu. Daudzas mobilās lietotnes patiesībā ir pielāgoti tīmekļa pārlūki, kuru darbību vada HTML5 saskarne sadarbībā ar aizmugursistēmas API saskarni, bet datus saglabā pārlūka atmiņā. Tomēr, kad jānodrošina lietotnes darbība bezsaistē, šis risinājums nav piemērots. Šādos gadījumos vairs nepietiek ar vienkāršu tīmekļa lietotni, bet ir jāveido atsevišķa funkcionalitāte, kas ir pielāgota konkrētai mobilai OS. Tādējādi palielinās lietotnes pieejamība un drošība bezsaistē, bet sadārdzinās izstrāde.

Vietēji darbināmai lietotnei ir nepieciešama daudz sarežģītāka izstrāde. Jārūpējas par lokālo datu apstrādi un šifrēšanu gadījumos, ja ierīci attālināti uzlauž vai nozog, kā arī lokālās glabātuves dzēšanu, kad izmantotie dati vairs nav nepieciešami. Sarežģītu lietotņu izveidē bieži izmanto iepriekš sagatavotus satvarus, kas palīdz ātrāk izveidot lietotnes loģiku, nevis nodarboties ar tehnisko uzturēšanu. Satvars ir iepriekš sagatavotu konfigurācijas iestatījumu un bibliotēku krājums, kas palīdz izskaust atkārtotas un vienveidīgas programmu izstrādes darbības, paātrina izstrādes tempu, uzlabo drošību, samazina izmaksas, kā arī atvieglo uzturēšanu [8]. Tomēr izstrādātājiem jāņem vērā arī satvaru radītie drošības riski, tādēļ

nepieciešama kontrole pār izmantoto satvaru un bibliotēku versijām, kā arī aktuālajiem apdraudējumiem.

Bieži lietotnes datus saglabā kādai atbalsta funkcijai (kļūdu ziņojumu sagatavošanai vai atkļūdošanai), tomēr lietotnes datu žurnālfailā dažos gadījumos var atrast arī autorizācijas vai citus sensitīvus datus, kurus pēc darba beigām ir nepieciešams droši šifrēt vai nodzēst. Tāpat izstrādātājiem jārūpējas, lai lietotņu biometriskā autorizācija un datu apmaiņa nedarbotos uzlauztajās mobilajās ierīcēs, kurās ir iegūtas augstākā līmeņa piekļuves tiesības un radītas iespējas viltot drošības sertifikātus, kā arī piekļūt ierobežotajām sistēmas datu glabātuvēm, tādējādi izraisot drošības risku gan lietotājam, gan pakalpojuma sniedzējam.

1.4. Lietotņu drošības jautājumi[1]

Pirms attīstīt jebkuras lietotnes funkcijas, jābūt pārliecībai, cik drošai tai ir jābūt. Mēģinājumi izveidot drošības modeli esošai lietotnei, vai brīdī, kad izveidota lielākā daļa sistēmas, var būt neveiksmīgi. Izstrādes pamatā jābūt precīzai izpratnei par potenciālajām drošības problēmām.

Drošības jautājumus var risināt vismaz trīs pamata jomās. Pirmkārt — aparatūra. Tā bieži tiek neievērota, bet ir kritisks sistēmas elements, jo ietekmē to, ko var panākt ar pārējiem drošības līdzekļiem. Otrkārt — izstrādē izmantotā programmēšanas valoda un rīki. Izvēloties atbilstošu tehnoloģiju, var iegūt drošības priekšrocības, nedarot papildu darbu. Visbeidzot — lietotne. Saprotot, kuras lietotnes daļas ir kritiskas, iespējams izstrādāt visstiprākās drošības iespējas tieši kritiskajām daļām.

Drošības eksperti vienmēr iesaka konfigurēt visus resursus atbilstoši labās prakses drošības pārvaldības principiem, resursu drošības kategorijai un normatīvo aktu prasībām. Parasti drošības riski veidojas no programmētāja, administratora vai mitinātāja pieļautajām kļūdām, nevis no pašas tehnoloģijas.

Lai panāktu fizisko drošību:

1. Tehniskos resursus apsargā un nodrošina piekļuves kontroli.
2. Darbības apstākļus nodrošina saskaņā ar resursu klasifikāciju un riska analīzi.
3. Uztur aktuālu datortīkla shēmu.
4. Nodrošina datortīkla drošības pārbaudes.
5. Nodrošina datu nesēju fizisko drošību.
6. Nepieļauj nesankcionētu resursu pieslēgšanu.

[1] Šajā nodaļā aprakstītais ir autora viedoklis, kura pamatā ir zināšanas un iespaidi, kas radušies, noklausoties tādus kursus kā "E-biznesa sistēmas" Vidzemes Augstskolā 2014. gadā un "Informācijas sistēmu drošība" Ekonomikas un kultūras augstskolā 2019. gadā.

Lai panāktu loģisko drošību:

1. Izmanto drošu identitātes pārbaudi — autentifikācijas mehānismu.
2. Ierobežo tiesības, izmantojot autorizācijas mehānismu.
3. Lieto diagnostikas un iebrukuma noteikšanas rīkus, lai pārraudzītu sistēmu.
4. Atļauj vienīgi funkcionēšanai nepieciešamos pakalpojumus.
5. Veido, saglabā un analizē sistēmas auditācijas pierakstus.
6. Regulāri uzstāda pārbaudītus drošības labojumus.

Pirms sistēmas pamatmodeļa izveides būtu jānoskaidro:

1. Kā un kur glabāt sensitīvo informāciju?
2. Vai lietotāji koplietos informāciju?
3. Vai būs ierobežotas piekļuves sadaļas?
4. Vai būs nepieciešama piekļuve failiem, multimedijiem?
5. Vai pastāv vēl citas drošības prasības saistībā ar lietotnes darbības jomu?

Pats svarīgākais nosacījums — sistēmai ir jādarbojas! Uz sistēmu varēs paļauties, ja:

1. Sistēma pareizi pildīs tai paredzētās funkcijas.
2. Sistēma nebūs izslēgta.
3. Dati nepazudīs.
4. Strādās pietiekoši ātri.
5. Jebkuras problēmas tiks ātri novērstas.
6. Sistēmu varēs attīstīt tālāk.
7. Sistēmu uzreiz varēs integrēt citā sistēmā.
8. Darbības izmaksas būs zināmas.
9. Darbības izmaksas tiks kontrolētas.

2. MOBILĀ DROŠĪBA UN IEVAINOJAMĪBAS

Katra jauna tehnoloģija rada jaunus drošības riskus, un centieni sekot līdzi pārmaiņām ir viens no lielākajiem izaicinājumiem drošības nozarē. Viedtālruņu operētājsistēmas atšķiras no datoru operētājsistēmām, un mobilās lietotnes atšķiras no tīmekļa lietotnēm. Lietotņu izpildes vides (izmēģināšanas jeb "smilšu kastes" režīma) ierobežojumu dēļ bufera pārpildīšanos un starpvietņu skriptēšanu (XSS) mobilajās lietotnēs var uzskatīt par ne tik būtisku ievainojamību kā personālo datoru tīmekļa lietotnēs.

Mobilo lietotņu drošības testēšanā ir joprojām vairāki strīdīgi jautājumi. Piemēram, drošības pētnieki pieslēpšanas (*obfuscation* — angļu val.) vai pilnu tiesību statusa ("root", "jailbreak") noteikšanas trūkumu uzskata par nopietnu ievainojamību, bet virkņu šifrēšanu, atkļūdošanas noteikšanu vai vadības plūsmas pieslēpšanu ne vienmēr uzskata par tik nozīmīgu problēmu [30].

Tomēr ir rūpīgi jāvērtē visas drošības problēmas, kas var rasties, jo katras lietotnes aizsardzības pasākumu pamatā jābūt specifiskam lietotnes draudu modelim. Ideālā pasaulē drošību apsvērtu visos izstrādes posmos, bet realitātē drošība bieži vien tiek apsvērta tikai izstrādes vēlīnā stadijā.

2.1. Datu saglabāšana un privātums

Tā kā mobilās ierīces var viegli pazaudēt, uzlauzt vai nozagt, mobilās drošības pamatjautājums ir rūpēties par lietotnēs glabāto datu aizsardzību. Mobilajās ierīcēs glabājas mūsu personiskā informācija, multimediji, pieraksti, darbam nepieciešamā informācija, autentifikācijas rīki, informācija par atrašanās vietu un daudz kas cits. Lietotnes darbojas kā klienti, kas mūs savieno ar ikdienā izmantotajiem pakalpojumiem, un apstrādā katru ziņu, ar kuru apmaināmies. Kompromitējot viedtālruni, var iegūt nefiltrētu piekļuvi personas privātumam.

Lai apgrūtinātu sensitīvo datu izgūšanu, ir nepieciešama papildu aizsardzība. Par sensitīviem datiem var uzskatīt gan lietotāja autentifikācijas datus, gan jebkurus citus datus, kas konkrētajā kontekstā var būt sensitīvi, piemēram, personu identificējošu informāciju, ko var izmantot identitātes zādzībai, kredītkaršu datus, autorizācijas mehānismus, veselības informāciju vai juridiski aizsargātu informāciju.

Nepareizi izmantojot operētājsistēmas mehānismus, var citām lietotnēm dot iespēju atklāt sensitīvos datus, kas saglabāti ierīcē. Dati var netīši noplūst mākonī, dublējumā, žurnālfailā vai kešatmiņā. Lai novērstu sensitīvo datu noplūdi, tos vajadzētu glabāt sistēmas kriptogrāfiskās informācijas datu glabātuvēs un lietotņu konteinerā ar ierobežotu piekļuvi.

Arī drošā glabātuvē saglabātus sensitīvos datus vajadzētu uzglabāt šifrētā veidā ar atslēgu, kas ir piesaistīta ierīcei vai lietotājam un ne ilgāk, kā tas ir nepieciešams programmas darbībai, jo pastāv iespēja, ka lietotājs modificē savas ierīces programmatūru, lai iegūtu pilnas tiesības ierīcē, un iegūst piekļuvi šai glabātuvei. Labā kriptogrāfijas prakse ir iepriekš atzītu kriptogrāfijas bibliotēku, konfigurāciju un pietiekošas entropijas gadījuma vērtību ģeneratora izmantošana.

Tāpat jāpārrauga mobilās sistēmas starpliktuve, pagaidu failu glabātuve un žurnāla faili, ja tie tiek izmantoti, jo arī tajos var noplūst sensitīvi dati. Piemēram, operētājsistēma *iOS* saglabā lietotnes aktuālo ekrānattēlu, pirms norīko lietotni fonā vai gatavības režīmā. Šādā gadījumā uzlauztas ierīces lietotājs var nokopēt šādus attēlus, iespējami iegūstot tādā veidā arī to sensitīvo informāciju, kas ir redzama uz ekrāna, tāpēc izstrādātājam būtu jāparūpējas par datu slēpšanu arī pirms ieiešanas gatavības režīmā. Arī ekrānattēlu veidošana vai ekrāna ierakstīšana var būt sensitīvo datu noplūdes avots. Piemēram, gadījumos, kad ierīci izmanto, lai pārbaudītu veselības sertifikātu kvadrātkodus, ekrāna attēli un ieraksti var būt pietiekami, lai nozagtu svešu medicīnas informāciju — izstrādātājam būtu jāparūpējas par lietotnes vismaz īslaicīgu darbības apturēšanu, lai aizturētu vairākkārtējus ekrāna attēlu saglabāšanas mēģinājumus.

Lai informētu par sensitīvu datu noplūdes riskiem, izstrādātājam vajadzētu pašā lietotnes saskarnē informēt par visiem ar privātumu saistītajiem aspektiem. Tāpat nevajadzētu izplatīt programmu, kas jau satur sensitīvus datus. Vispirms dati būtu jānokopē atmiņā no attāla servera pa drošu kanālu, bet pēc pārāk daudziem neveiksmīgas autentifikācijas mēģinājumiem lietotnes lokālā datu krātuve jābloķē un / vai jādzēš. Piemēram, lai nebūtu iespējas pārtvert un modificēt saņemtos datus vai izmantot novecojušus medicīniskos datus.

2.2. Autentificēšanās un sesiju pārvaldība

Lai pārvaldītu lietotāju kontus un sesijas mobilajās lietotnēs, parasti tiek izmantotas iespējas, kuras nodrošina serverī izvietots galapunkts. Autentifikācija var notikt, izmantojot gan daudzfaktoru autentifikāciju, gan drošības sertifikātus, gan uzģenerētu paroli. Izmantojot paroli, parasti nosaka paroles stiprības prasības, lai tā nebūtu viegli atminama vai atlaužama. Biometrisko autentifikāciju izmanto, lai dotu piekļuvi atslēgu krātuvei, kas pēc tam autorizē pieprasīto pakalpojumu, tādējādi dodot iespēju viegli pārvaldīt lietotājus, izmantojot jau esošus autentifikācijas līdzekļus. Parasti vismaz viens no autorizācijas faktoriem noteikti tiek izpildīts arī attālajā galapunktā.

Gadījumos, kad ir nepieciešama lietotāja sesijas pārvaldība, parasti attālais galapunkts ģenerē nejaušus unikālus identifikatorus, kas nodrošina turpmāko klienta pieprasījumu

autentiskumu bez atkārtotas lietotāja autentifikācijas. Beidzoties lietotāja sesijai (noilguma vai lietotāja atteikšanās gadījumā), attālais galapunkts pārtrauc identifikatoru autorizēšanu. Lai pasargātu galapunktus pret pakalpojuma atteici un pārlases uzbrukumiem, parasti izmanto mehānismus, kas nepieļauj neveiksmīgu autentifikācijas mēģinājumu izpildīšanu vairākas reizes pēc kārtas īsā laika periodā.

Lai nodrošinātu lietotāja konta aizsardzību, kad tiek izmantotas vairākas ierīces, parasti lietotājam dod iespēju saņemt informāciju par visām sensitīvajām darbībām savā kontā. Piemēram, informācija par jaunu ierīci nosūta uz e-pastu vai parādās mobilajā ierīcē, pieprasot apstiprināšanu. Lietotājam var nodot informāciju par veiktajām darbībām, izmantoto IP adresi un aktīvajām ierīcēm, kā arī iespēju bloķēt nevēlamās ierīces.

2.3. Informācijas apmaiņa ar attālo galapunktu

Lai nodrošinātu konfidencialitāti un integritāti informācijai, kas tiek apmainīta starp mobilo lietotni un attālināto pakalpojumu galapunktiem, ir nepieciešams drošs, šifrēts kanāls visai tīkla saziņai, izmantojot TLS protokolu ar labai praksei atbilstošiem iestatījumiem — lietotne pārbauda un pieņem tikai tādus sertifikātus, kurus ir parakstījusi uzticama centrālā iestāde. Lai uzlabotu lietotnes aizsardzības līmeni, tiek izmantota sertifikāta piespraušana — lietotne izmanto savu sertifikātu krātuvi, piesprauž galapunkta sertifikātu vai publisko atslēgu un pēc tam vairs neveido savienojumus ar citiem galapunktiem, kas piedāvā atšķirīgu sertifikātu, pat ja to ir parakstījusi uzticama centrālā iestāde.

2.4. Informācijas drošība mobilajā platformā

Lai pasargātu lietotāju datus, lietotnes veido tā, lai tās pieprasītu minimālo nepieciešamo tiesību kopumu attiecībā uz ierīces sniegtajām iespējām. Visas ievades no ārējiem avotiem un lietotāja pārbauda un, ja nepieciešams, desensitivizē. Tāpat visas lietotnes iekšējās funkcionalitātes netiek eksportētas, izmantojot starpprocesu komunikāciju, ja vien tās nav pienācīgi aizsargātas.

Saskaņā ar labo praksi, lai pasargātu skata klases *WebView* komponentes, tajās pēc noklusējuma ir atslēgts *JavaScript* un atļauts tikai HTTPS saturs, bet, ja tas ir nepieciešams programmas darbam, tad pieļaujama tikai tādu skriptu izpilde, kas ir iekļauti lietotnes pakotnē. Pirms *WebView* iznīcināšanas jādzēš arī izmantotā kešatmiņa, krātuve un ielādētie resursi.

Objektu deserializāciju, ja tāda ir nepieciešama, vajadzētu izpildīt, izmantojot drošas serializācijas API saskarni. Nedroša deserializācija bieži noved pie attālā koda izpildes. Pat ja deserializācijas nepilnības nenoved pie attālā koda izpildes, tās var izmantot uzbrukumos, tajā

skaitā var izpildīt atkārtošanas, injekcijas un tiesību eskalācijas uzbrukumus. Lai novērstu nedrošu serializāciju, nedrīkst pieņemt serializētus objektus no neuzticamiem un nedrošiem avotiem, kā arī jānodrošina integritātes pārbaude un deserializēšanas pārraudzība [33]

2.5. Droša produkcijas versija

Lai sagatavotu drošu lietotnes produkcijas versiju izplatīšanai, ir vairāki pasākumi, ko veic izstrādātāji. Veidojot mobilās lietotnes produkcijas laidienu nepieciešams:

1. Aktivizēt kompilatora un izmantoto rīku piedāvātos iebūvētos drošības līdzekļus.
2. Parakstīt lietotni ar derīgu aizsargātu sertifikātu.
3. Atslēgt atkļūdošanas režīmu, aizvācot atkļūdošanas failus un testa kodus.
4. Pārliecināties, ka lietotne neizpilda izvērstus atkļūdošanas informācijas kļūdu ziņojumus.
5. Atjaunināt un pārbaudīt visus izmantotos trešo pušu komponentus, vai tie nesatur zināmas ievainojamības.
6. Pārliecināties, ka pareizi tiek apstrādāti iespējamie izņēmumi un pēc noklusējuma liegta piekļuve, ja tā nav atļauta.
7. Pārliecināties par atmiņas drošu piešķiršanu un atbrīvošanu.

2.6. Aizsardzība pret izmainīšanu

Lietotnēs var izmantot līdzekļus, kas palielina noturību pret reverso inženieriju un noteiktiem klienta puses uzbrukumiem. Lai pasargātu lietotnē apstrādātos sensitīvos datus, daudzi izstrādātāji lietotnēs ietver dažādus aizsargmehānismus, kas neļauj izmantot lietotni ierīcēs, kurās parastam lietotājam ir iespēja darbināt lietotnes ar pilnām tiesībām un apiet ražotāju noteiktos ierobežojumus. Izstrādātājs, izmantojot definēto draudu modeli, var nodrošināties pret konkrētiem draudiem. Piemēram, likt drošības pētniekiem pielikt lielas manuālas inženierijas pūles, lai īstenotu noteiktus mērķus.

Lai pasargātu lietotni pret izmainīšanu, var apturēt lietotnes darbību, ja ir noteikts, ka:

1. Ierīce izmanto pilno tiesību statusu.
2. Lietotnei ir pievienots atkļūdotājs (piemēram, pārbaudot visus pieejamos atkļūdošanas protokolus).
3. Notiek lietotnes izmēģināšanas (“smilšu kastes”) režīmā esošo vai izmantotajā atmiņas apgabalā ielādēto datu izmanīšana.
4. Mobilajā ierīcē ir uzstādīti plaši zināmi reversās inženierijas vai piedares rīki.
5. Lietotne ir palaista emulatorā.

Jo vairāk pretpasākumu īsteno, jo lielāka ir noturība pret reverso inženieriju. Lai apgrūtinātu nelūgto drošības pētnieku darbu, lietotnē var izmantot arī koda pieslēpšanu un lietotnes piesaistīšanu konkrētas ierīces unikālajām īpašībām.

2.7. Izplatītākās mobilo lietotņu ievainojamības

Mobilo lietotņu programmatūras satura jomā drošības draudi galvenokārt izpaužas kā ļaunprātīgā koda izplatīšana ar sociālās inženierijas metodēm (piemēram, pikšķerēšanu), lai pēc tam izmantotu operētājsistēmas un lietotņu ievainojamības, tādējādi nopludinot personisko informāciju vai gūstot finansiālu labumu.

Drošības draudi, kas minēti 2.1. tabulā, var izraisīt ievainojamību lielākajā daļā lietotņu. Izmantojot lietotnes ievainojamības, var apiet autentifikāciju un neautorizēti piekļūt informācijai, kā arī veikt ļaunprātīgas darbības, piemēram, uzstādīt atsevišķas lietotnes, kuras var darboties ar augstākā līmeņa tiesībām, izmantojot jau zināmās ievainojamības, vai veikt lietotņu pārpakošanu, pārveidojot sākotnējās lietotnes tā, lai tās varētu izpildīt ļaunprātīgo kodu, apietu iestrādātās drošības funkcijas, pēc tam šādu lietotni atkārtoti izplatot lietotņu veikalos, lai panāktu neapzinātu viltotās lietotnes uzstādīšanu lietotāju ierīcēs. Viltotu lietotņu izplatīšana ne tikai apdraud lietotājus, bet arī grauj izstrādātāja uzticamību, tēlu un rada peļņas zaudējumus.

2.1. tabula

Nozīmīgākie drošības draudi mobilajās lietotnēs pēc OWASP [15]

Drošības draudi	Skaidrojums
M:1 Neatbilstoša platformas lietošana	Draudi, kas rodas no kādas funkcijas ļaunprātīgas izmantošanas vai drošības kontroles neizmantošanas
M2: Nedroša datu glabāšana	Draudi, kas rodas, iegūstot datus no nozagtas ierīces vai pārpakotas lietotnes, kura piekļūst citas lietotnes datiem, lai nosūtu tos uzbrucējam
M3: Nedroša saziņa	Draudi, kas attiecas uz nešifrētu datu apmaiņu ar lietotnes tiešsaistes aizmugursistēmu, izmantojot neuzticamu datortīklu
M4: Nedroša autentifikācija	Draudi, kas rodas apiešanas vai rekvizītu zādzības gadījumos, ja lietotne izmanto autentifikāciju
M5: Neatbilstoša šifrēšana	Draudi, kas rodas, piekļūstot neatbilstoši šifrētiem datiem
M6: Nedroša autorizācija	Draudi, kas rodas, apejot lietotnes autorizācijas mehānismu
M7: Nekvalitatīvs kods	Draudi, kas rodas no nepietiekamas ievades validācijas un var izraisīt bufera pārpildīšanu
M8: Koda sagrozīšana	Draudi, kurus rada modificēta lietotne, kas iegūta trešās puses lietotņu veikalā
M9: Reversā inženierija	Draudi, kas rodas, uzbrucējam analizējot lietotni lokālā vidē, izmantojot dažādu rīku kopu
M10: Nezināma funkcionalitāte	Draudi, kas rodas aizmugursistēmai no nepazīstamas izcelsmes funkcionalitātes slēpto funkciju izzināšanas

Palielinoties viedtālruņu lietotāju skaitam, pieaug tādu prasmīgo lietotāju skaits, kas labprāt uzlauž savu ierīču aizsardzību, lai palielinātu tās iespēju klāstu, apietu reklāmas vai uzstādītu nelicenzētu programmatūru, tādējādi neierobežoti piekļūstot sistēmas funkcijām, kā

arī pakļaujot riskam savas viedierīces drošību. Lai apmierinātu šādu lietotāju vajadzības, ir pieejamas arī mobilo tālruņu antivīrusa programmas un ielaušanās atklāšanas lietotnes.

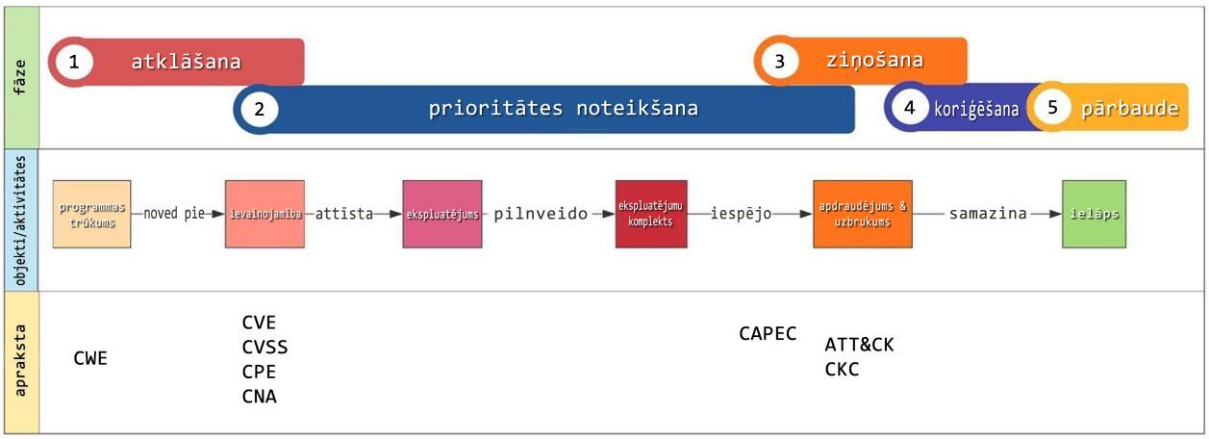
Mobilās drošības kompānijas *Lookout* pētnieki pēc 100000 lietotņu pilnīgas analīzes apgalvo, ka daudzas ievainojamības *Android* un *iOS* lietotnēs rodas no nejaušām kodēšanas kļūdām trešās puses lietotnēs [25]. Piemēram, mobilo datu noplūde rodas no lietotnes darbības žurnālā reģistrētiem sensitīviem datiem, kuriem var piekļūt citas lietotnes, kas uzstādītas ierīcē. Ļaunprātīgas lietotnes šādus datus pēc tam var pārsūtīt uz pašu kontrolētu serveri, lai vāktu lietotāju sensitīvo informāciju.

Vēl viena nozīmīga lietotnes ievainojamība ir bibliotēku ar zināmām ievainojamībām izmantošana. Piemēram, izmantojot salauztu transporta slāņa drošības (TLS) šifrēšanas bibliotēku, jebkura mobilā lietotne, kas izmantos šādu bibliotēku, pakļaus lietotājus pārtveršanas uzbrukumiem, un uzbrucēji varēs atšifrēt un nolasīt attiecīgo datplūsmu. Izlabot esošu ievainojamību mobilajā lietotnē ir grūtāk, jo labojumu izplatīšana notiek lietotņu veikalā, kur ir nepieciešams ražotāja apstiprinājums un jāgaida, līdz lietotāji izdomās atjaunināt.

2.8. Ievainojamību un trūkumu izvērtēšana

Saskaņā ar standartiem (2.1. att.) ievainojamībām piešķir CVSS punktus atkarībā no to nopietnības. CVSS standartu uztur FIRST, bet izmanto DHS, CISA un MITRE, lai veidotu programmatūras trūkumu klasifikatoru CWE [3].

Pēc tehnoloģijas ievainojamību skaita un izmantojamības līmeņa var spriest par risku un programmatūras drošību. Dažādu produktu ievainojamības var atrast, izmantojot tīmeklī pieejamās ievainojamību datubāzes, piemēram, https://www.opencve.io/cve, https://www.exploit-db.com, kā arī atvērtā pirmkoda repozitorijus un forumus.



2.1. att. **Ievainojamības dzīves cikls un standarti [18]**

CVE standarta atsauču sistēma nodrošina atklāto ievainojamību klasifikatorus un identifikāciju. Izmantojot CVE var noskaidrot, kuram produktam vai bibliotēkai piemīt konkrētā ievainojamība, turpretī CWE sistēma tikai apraksta trūkuma būtību, nevis specifisku produktu. CVE identifikatorus piešķir CNA autorizētās organizācijas. Ievainojamības, kuras publiski nav zināmas, var būt bez CVE numuriem. Publiski neatklātas ievainojamības sauc par nulles dienas (*zero-day*) ievainojamībām. Lai identificētu zināmās CVE ievainojamības konkrētās produktu versijās, korporācija MITRE uztur CPE datubāzi.

2.2. tabula

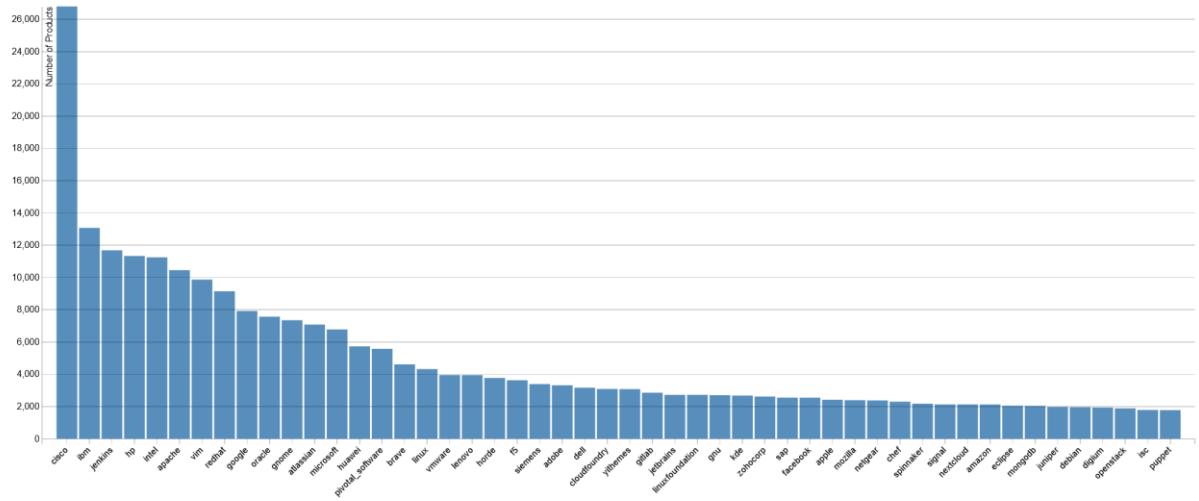
CWE Top 25 bīstamāko programmatūras trūkumu saraksts 2021. gadā [1]

Vieta	ID	Nosaukums	CWE punkti
1	CWE-787	Rakstīšana ārpus robežas apgabala	65.93
2	CWE-79	Nepareiza ievades neitralizēšana tīmekļa lapas ģenerēšanas laikā (Starpvietņu skriptēšana)	46.84
3	CWE-125	Nepareiza ievades neitralizēšana tīmekļa lapas ģenerēšanas laikā (Starpvietņu skriptēšana)	24.9
4	CWE-20	Nepareiza ievades validācija	20.47
5	CWE-78	Nepareiza OS komandas speciālo elementu neitralizācija (OS komandas injekcija)	19.55
6	CWE-89	Nepareiza SQL komandā izmantoto elementu neitralizācija (SQL injekcija)	19.54
7	CWE-416	Atmiņas rādītāja izmantošana pēc atmiņas atbrīvošanas	16.83
8	CWE-22	Nepareizs piekļuves ceļa ierobežojums (Ceļa apvade)	14.69
9	CWE-352	Vairāku vietņu pieprasījuma viltošana (CSRF)	14.46
10	CWE-434	Neierobežota bīstama veida failu augšupielāde	8.45
11	CWE-306	Trūkst kritiskas funkcijas autentifikācijas	7.93
12	CWE-190	Vesela skaitļa pārpilde vai aplauzums	7.12
13	CWE-502	Neuzticamu datu deserializācija	6.71
14	CWE-287	Nepareiza autentifikācija	6.58
15	CWE-476	Piekļuve NULL rādītājam	6.54
16	CWE-798	Cieti iekodētu pilnvaru izmantošana	6.27
17	CWE-119	Nepareizs operāciju ierobežojums atmiņas bufera robežās	5.84
18	CWE-862	Trūkst autorizācijas	5.47
19	CWE-276	Nepareizas noklusējuma atļaujas	5.09
20	CWE-200	Informācijas eksponēšana	4.74
21	CWE-522	Nepietiekami aizsargāti akreditācijas dati	4.21
22	CWE-732	Nepareiza atļaujas piešķiršana kritiskam resursam	4.2
23	CWE-611	Nepareizs XML ārējās entitātes atsauces ierobežojums	4.02
24	CWE-918	Servera puses pieprasījuma viltošana (SSRF)	3.78
25	CWE-77	Komandā izmantoto īpašo elementu nepareiza neitralizēšana ("Komandas injekcija")	3.58

Veidojot CWE sarakstu 2021. gadā (2.2. tab.), izmantota NIST apkopotā statistika par reālo ievainojamību skaitu NVD (*U.S. Department of Commerce National Institute of Standards and Technology: National Vulnerability Database*: [14]) repozitorijā un iekļauj 2019. un 2020. gada datu stāvokli 2021. gada 18. martā. CWE punktu aprēķinā izmantota vērtēšanas formula, kas apvieno biežumu, cik bieži CVE ir ievainojamības pamatcēlonis, ar

prognozēto tās izmantošanas nopietnību. Gan biežums, gan nopietnība ir normalizēti attiecībā pret minimālo un maksimālo novēroto vērtību. CWE biežumu aprēķina pēc tā, cik reižu CWE ir piesaistīts CVE NVD, iekļaujot tikai tos gadījumus, kur CVE ir piesaistīts CWE trūkums.

Institūta NIST tīmekļvietnē https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cwe-over-time ir iespēja interaktīvi izsekot aktuālajām programmatūras trūkumu tendencēm, bet vietnē https://nvd.nist.gov/general/visualizations/cpe-products-distribution uzzināt, kuri programmatūras izstrādātāji ražo visvairāk produktus ar atklātām ievainojamībām (2.2. att.).



2.2. att. **Produkti ar vislielāko atklāto ievainojamību skaitu pēc ražotāja 2021. gada decembrī (U.S. Department of Commerce National Institute of Standards and Technology: National Vulnerability Database)**

ATT&CK draudu novēršanas taktikas un paņēmienu zināšanu bāzi izmanto problēmu izpratnes veicināšanai [19]. Kiberuzbrukuma ķēdes modelēšanu CKC veic, lai labāk izprastu ievainojamības radītos draudus un izvēlētos labāko stratēģiju [22]. MITRE uztur kopējo uzbrukumu mehānismu uzskaites un klasifikācijas katalogu CAPEC, kuru izmanto, lai pētītu aizsardzību, pamatojoties uz uzbrucēju rīcības scenārijiem.

2.9. Tehniskās drošības novērtēšanas plānošana

Tehniskās drošības novērtēšana ir process, kura laikā pārliecinās par produkta loģisko aizsardzību, meklējot tādus scenārijus, kuros varētu apiet sistēmas drošības uzstādījumus. Lai atklātu šādus scenārijus, var pētīt zināmās drošības problēmas līdzīgās sistēmās un veidot tādus testa scenārijus, kas atklātu līdzīgas problēmas testējamā sistēmā. Lai varētu pārbaudīt ievainojamības, nepieciešamas plašas zināšanas par dažādu protokolu pieprasījumiem un

atbildēm, kā arī jāprot nosūtīt iepriekš neparedzētus pieprasījumus tādās vietās, kas veiksmīgi ļautu izpildīt uzbrukumu.

Lai atvieglotu drošības pārbaudes, var izmantot drošības pārbaudes rīkus. Drošības pārbaudes rīki ietaupa laiku un ir nozīmīgi satura atklāšanā un izlūkošanā. Lai sameklētu atbilstošākos rīkus, var izmantot OWASP Nozīmīgāko mobilo drošības pārbaudes rīku sarakstu [32], kā arī drošības testēšanas rīku apkopojumu *Mobexler* operētājsistēmas tīmekļa vietnē [26].

3. KIBERDROŠĪBAS NOVĒRTĒŠANA

Ikvienu datortīklā pieslēgtu IT infrastruktūras iekārtu apdraud kiberuzbrukums. Lai saprastu, cik liels ir draudošais risks, un spētu pārvaldīt risku, veic kiberdrošības riska novērtēšanu, kurā nosaka, kuri resursi ir visneaizsargātākie. Kiberdrošības riska novērtēšanā apzina organizācijas galvenos darbības mērķus un identificē tehnoloģiskos resursus, kas ir nozīmīgi darbības mērķu sasniegšanai. Resursu kiberapdraudējuma iespējamības, ietekmes un uzbrukumu veidu identificēšana palīdz samazināt drošības riskus līdz pieņemamam līmenim, novērst datu aizsardzības pārkāpumus un neatbilstību normatīvajiem aktiem, kā arī pieņemt faktos pamatotus lēmumus situācijas kontrolei.

Kiberdrošības riska novērtēšana ir apjomīga un pastāvīgi nepieciešama darbība, tāpēc, lai uzlabotu organizācijas drošību nākotnē, tai jāparedz laiks un resursi. Novērtēšanu parasti atkārto, tiklīdz parādās jauni draudi un ievieš jaunas sistēmas vai procedūras. Laba sākotnējā novērtēšana veido pamatu un palīdz turpmākajiem novērtējumiem, vienlaikus samazinot iespēju, ka kiberuzbrukums negatīvi ietekmēs organizācijas darbības mērķus.

Kiberdrošības riska novēršanā parasti izšķir vairākus svarīgus soļus:

1. Riska novērtējuma darbības jomas noteikšana;
2. Riska identifikācija;
3. Riska analīze un iespējamās ietekmes klasifikācija;
4. Riska prioritātes noteikšana;
5. Visu risku dokumentēšana.

3.1. Pirmais solis: riska novērtējuma darbības jomas noteikšana

Riska novērtēšana sākas ar lēmuma pieņemšanu par to, kas ietilpst novērtējuma darbības jomā. Tā var būt organizācija, struktūrvienība, atrašanās vieta vai konkrēts darbības aspekts, piemēram, maksājumu apstrāde vai mobilā lietotne.

Riska novērtēšanai izmanto drošības standartus, piemēram, ISO/IEC 27001, vai risku satvarus, piemēram, NIST SP 800-37 [28], kas palīdz strukturēti novērtēt informācijas drošības riskus un nodrošināt piemērotas un efektīvas riska kontroles. Dažādi standarti un tiesību akti pieprasa organizācijām novērtēt riskus un bieži vien sniedz vadlīnijas vai ieteikumus, tomēr formāla atbilstības prasību izpilde ne vienmēr nozīmē, ka sistēma nav pakļauta nekādiem riskiem.

3.2. Otrais solis: riska identifikācija

Resursu identificēšana

Nevarat aizsargāt kaut ko nezināmu, tāpēc nepieciešams saraksts ar visiem fiziskajiem un loģiskajiem resursiem, kas ietilpst riska novērtējuma darbības jomā. Svarīgi identificēt ne tikai kritiskos, bet arī mazāk svarīgos resursus, kurus varētu apdraudēt noziedznieki, lai izmantotu tos kā atspēriena punktu uzbrukuma paplašināšanai. Lai pārskatītu savstarpējo savienojamību un komunikāciju starp resursiem, kā arī tīkla ieejas punktus tīklā, parasti veido tīkla arhitektūras diagrammas, kas atvieglo draudu identificēšanu.

Draudu identificēšana

Draudi ir taktika, paņēmieni un metodes, kurus izmanto apdraudējuma aktori, lai nodarītu kaitējumu organizācijas resursiem. Ievērojamākā tīmekļa lietotņu drošības un aizsardzības pētnieku organizācija globālā mērogā ir OWASP, kas veido metodoloģiju un dokumentāciju lietotņu drošības jomā. Organizācija ir apkopojusi aktuālo pasaulē nozīmīgāko mobilo drošības ievainojamības risku sarakstu *OWASP: Top 10 Mobile Application Security Risks:* [15]. Ievainojamību sarakstā riski ir sakārtoti, ņemot vērā spēju ekspluatēt, izplatības biežumu, kā arī to, cik viegli ir detektēt ievainojamību.

Pašreiz OWASP strādā pie tādiem projektiem kā mobilo ievainojamību risku apkopojumu standarts MASVS [30], kas nosaka mobilo lietotņu drošības testēšanas scenāriju pamatprasības, un MSTG [31], kas visaptveroši apraksta MASVS standarta drošības testēšanas tehniskos procesus un reverso inženieriju.

Lai palīdzētu identificēt katra resursa potenciālos draudus un piemērotāko aizsardzību, var izmantot arī draudu bibliotēku, piemēram, MITRE ATT&CK zināšanu bāzi [13], un apsvērt, kurā vietā kibernoziegumu ķēdē [2] resurss varētu atrasties. Kiberuzbrukumu ķēde kartē tipiskus reāla uzbrukuma posmus un mērķus.

Seku identificēšana

Apkopojot iespējamās sekas vienkāršu riska scenāriju (novērojumu) veidā, visām ieinteresētajām pusēm ir vieglāk izprast pastāvošos riskus, bet drošībniekiem ir vieglāk izdarīt atbilstošus pretpasākumus. Novērojumā var precizēt identificēto draudu sekas, ja notiktu uzbrukums tvērumā esošam resursam. Piemēram:

Draudi: Uzbrucējs veic injekcijas uzbrukumu API

Ievainojamība: zināma, neizlabota

Resurss: tīmekļa serveris

Sekas: privāto datu zādzība.

3.3. Trešais solis: riska analīze un iespējamās ietekmes klasifikācija

Klasificējot iespējamo sagaidāmo kiberdrošības riska ietekmi (draudu seku radītā kaitējuma apmēru), nosaka riska scenārija reālas iestāšanās iespējamību. Kiberdrošības riska ietekmi nosaka tā īstenošanas varbūtība, t.i., ka konkrētais apdraudējums notiks, izmantojot atrasto ievainojamību, un tas ir atkarīgs no tā, cik viegli šo ievainojamību ir atklāt un izmantot, nevis pēc vēsturiskajiem ievainojamības rādītājiem. Klasifikācijā ņem vērtā ietekmi uz resursa konfidencialitāti, integritāti un pieejamību, atbilstoši esošajām sistēmas funkcijām un ieviestajām kontrolēm. Riska gala vērtējumā rezultātā izmanto lielāko iespējamo ietekmi. Šis novērtējuma aspekts pēc būtības ir riska subjektīva interpretācija, tāpēc ļoti svarīgs ir ieinteresēto personu un drošības ekspertu ieguldījums.

Resursu apdraudējuma riska varbūtību un kaitējumu parasti klasificē skalā no 1: "Mazs" līdz 5: "Liels" [17]. Pēc šīs klasifikācijas var viegli izveidot riska matricu, kas palīdz prioritizēt riskus. Ņemot iepriekš minēto injekcijas gadījuma piemēru, ietekmi uz konfidencialitāti varētu novērtēt kā "Lielu" ietekmi.

3.4. Ceturtais solis: riska prioritātes noteikšana

Neviena sistēma vai vide nevar būt 100 % droša, jo vienmēr pastāv zināms atlikušais risks, kas atbilst pieņemamam riska līmenim. Ja risks pārsniedz pieņemama riska līmeni, tad tam jānosaka tāda prioritāte, kas risku novērš atkarībā no prioritātes un pieejamiem resursiem.

Lai klasificētu riska prioritāti, izmanto riska matricu (3.1. tab.), kurā katra riska līmeni nosaka kā "varbūtību, reizinātu ar ietekmi". Ja iepriekš piemērā minētais injekcijas uzbrukuma risks būtu klasificēts kā "Ievērojams" vai "Liels", šādu riska scenāriju varētu klasificēt kā "Kritisks".

Lai novērsu riskus, parasti izvēlas trīs iespējamos scenārijus:

1. Izvairīšanās. Ja risks ir lielāks par ieguvumu, vislabāk ir uzreiz pārtraukt šādu darbību, ja tas ļauj izvairīties no šāda riska.
2. Nodošana. Risku var dalīt ar citām personām, piemēram, izmantot apdrošināšanu vai nodot riskanto resursu ārpakalpojumā.
3. Novēršana. Var izveidot drošības kontroles un citus pasākumus, kas samazina riska iespējamību un / vai ietekmi, tādējādi samazinot arī riska līmeni.

3.1. tabula

5x5 risku prioritātes matrica

Prioritāte = Varbūtība × Ietekme	1:Maza	2:Neliela	3:Vidēja	4:Ievērojama	5:Liela
5:Liela	5:Vidēja	10:Vidēji augsta	15:Augsta	20:Kritiska	25:Kritiska
4:Ievērojama	4:Zema	8:Vidēja	12:Vidēji augsta	16:Augsta	20:Kritiska
3:Vidēja	3:Zema	6:Vidēja	9:Vidēja	12:Vidēji augsta	15:Augsta
2:Neliela	2:Zema	4:Zema	6:Vidēja	8:Vidēja	10:Vidēji augsta
1:Maza	1:Zema	2:Zema	3:Zema	4:Zema	5:Vidēja

3.5. Piektais solis: visu risku dokumentēšana

Svarīgi ir dokumentēt un reģistrēt visus identificētos riska scenārijus. Risku reģistrā vienmēr jābūt patiesai un aktuālai informācijai par kiberdrošības riska stāvokli. Risku reģistrā iekļauj:

1. Riska scenāriju (novērojumu);
2. Identifikācijas datumu;
3. Esošās drošības kontroles;
4. Pašreizējo riska līmeni;
5. Novēršanas plānu — plānotās darbības un laika grafiku, kas paredzēts riska samazināšanai līdz pieņemamam līmenim;
6. Novēršanas stāvokli — kādā stāvoklī pašreiz ir novēršanas plāna izpilde;
7. Atlikušo risku — kāds būs riska līmenis pēc novēršanas;
8. Riska īpašnieku — kas atbild par atlikušā riska līmeņa nepārsniegšanu.

4. VAKCINĀCIJAS SERTIFIKĀTU LIETOTNES

4.1. COVID-19 sertifikāti un krāpšana

Saskaņā ar Amerikas globālās datoru drošības programmatūras uzņēmuma *McAfee* "Mobilo draudu ziņojumu" 2021. gadā [12] COVID-19 pandēmija un vakcīnas pagaidām ir lielākais globālais dezinformācijas un ļaunprātīgas programmatūras uzbrukumu mērķis. Kibernoziedznieki veido mērķētas viltotas lietotnes dažādu valstu vakcīnu programmām, lai pārbaudītu uzbrukumu ienesību. Veiksmīgi uzbrukumi vakcinācijas programmām ir notikuši Indijā un Čīlē, tāpēc ir sagaidāms, ka tos īstenos arī pret citām valstīm. Zinoši cilvēki ar pareizajiem sakariem izmanto arī Eiropas vakcinācijas sertifikāta drošības nepilnības, lai viltotu un piesavinātos vakcinācijas sertifikātus. Uzbrukumos galvenokārt vainojami organizētās noziedzības tīkli, korumpēti veselības aprūpes darbinieki, kā arī pret vakcināciju noskaņoti cilvēki.

Vēl nav zināms, vai Eiropas digitālais COVID-19 sertifikāts palīdzēs samazināt ar pandēmiju saistītos krāpniecības gadījumus, jo pagaidām rezultāti nav viennozīmīgi. Pieaugot krāpšanas gadījumu skaitam [29], pieaug arī bažas par sertifikātu drošību.

2021. gadā Itālijā tika atklātas vairākas tiešsaistes krāpšanas shēmas [11], kurās tirgoja viltotus vakcinācijas sertifikātus ar viltotiem kvardātkodiem un vakcīnu sērijas numuriem. Francijā tika tirgoti īsti sertifikāti un īsti kvadrātkodi, kas, iespējams, tika iegūti no korumpētiem veselības aprūpes darbiniekiem, kuriem bija oficiāla piekļuve veselības aprūpes sistēmas datu bāzēm [6]. Grieķijā kāds pret vakcināciju un COVID-19 draudu atzīšanu noskaņots ārsts veica viltotas injekcijas, lai iegūtu sertifikātus līdzīgi domājošiem biedriem [23].

4.2. Vakcinācijas sertifikātu standarti un iniciatīvas

COVID-19 digitālo sertifikātu izstrāde ir piesaistījusi vairāku ievērojamu organizāciju un aliansu interesi visā pasaulē. Papīra formāta vakcinācijas sertifikāti, kas neizmanto kvadrātkodu, ir mazāk aizsargāti pret krāpšanu, jo tos ir iespējams vieglāk viltot un pārveidot, tāpēc rakstā [24] ir izteikti dažādi priekšlikumi, lai izveidotu vienotus standartus digitālo vakcinācijas sertifikātu ieviešanai visā pasaulē. Tomēr pašreiz vēl nav izveidota globāla digitālo vakcinācijas sertifikātu uzticības sistēma [37].

4.2.1. Pasaules veselības organizācija

Pasaules Veselības organizācija (PVO) ir 2021. gada augustā ir izstrādājusi vadlīnijas COVID-19 digitālo sertifikātu ieviešanai [37]. Sākotnēji PVO strādāja pie viedo vakcinācijas sertifikātu (*smart vaccination certificate* SVC) vadlīnijām, lai tās varētu pielietot jebkuru vakcīnu digitālajiem sertifikātiem. Tika aprakstīti divi SVC pamatscenāriji, proti, aprūpes nepārtrauktība, lai reģistrētu vakcinācijas ierakstus, un vakcinācijas apliecinājums, lai pierādītu vakcinācijas faktu. 2021. gada jūnijā paplašinājās SVC darba grupas darbības joma [36] un viedā vakcinācijas sertifikāta specifikāciju pārdēvēja par "COVID-19 sertifikātu digitālās dokumentācijas (DDCC)" specifikāciju, iekļaujot tajā COVID-19 vakcinācijas statusu, SARS-CoV-2 testu rezultātus un COVID-19 atveseļošanās statusu.

4.2.2. Eiropas Savienība

Lai COVID-19 pandēmijas laikā veicinātu drošu un brīvu pārvietošanos ES teritorijā, Eiropas Komisija 2021. gada martā ierosināja ES digitālo COVID-19 sertifikātu (EUDCC), iepriekš dēvētu arī par ES digitālo zaļo sertifikātu [16]. Šāds sertifikāts ir digitāls pierādījums, ka persona vai nu ir vakcinēta pret COVID-19, vai saņēmusi negatīvu testa rezultātu, vai izārstējusies no COVID-19. Medicīnas iestādes sertifikātus izsniedz bez maksas digitālā vai papīra formātā. Kvadrātkoda sertifikātā ietilpst akreditācijas dati, kas satur būtisku medicīnisko informāciju par mērķa slimību vai ierosinātāju, vakcīnas veidu, zāļu nosaukumu, ražotāju, dozas kārtas numuru, kopējo paredzēto devu skaitu, vakcinācijas datumu, valsti, sertifikāta izsniedzēju, unikālo identifikatoru, testa veidu un nosaukumu, testa ražotāju, parauga ņemšanas datumu un laiku, testa rezultāta datumu un laiku, testa rezultātu, testēšanas centru, testa valsti, pirmā pozitīvā testa datumu, testēšanas sertifikāta derīgumu [4]. Izmantojot lietotni var noteikt kvadrātkoda derīgumu un autentiskumu. Tas notiek, pārbaudot parakstu, kas pievienots sertifikāta izdošanas procesā.

Lai nodrošinātu starpvalstu validāciju, ir izstrādāta ES vārteja, kas izmanto PKI infrastruktūru. Vārteja ir galvenais komponents sertifikātu sadarbspējas nodrošināšanā. Tā sāka darboties 2021. gada 1. jūlijā. Eiropas Savienība ir sagatavojusi un arī regulāri papildina vadlīnijas par savu digitālo COVID-19 sertifikātu tehniskajām specifikācijām [4].

4.2.3. Citas nozīmīgas iniciatīvas

Savu artavu COVID-19 pandēmijas seku mazināšanā ir devušas arī vairākas citas starptautiskas organizācijas. Starptautiskā gaisa transporta asociācija (*International Air Transport Association* IATA). IATA *Travel Pass* nodrošina mobilo lietotni

(https://play.google.com/store/apps/details?id=org.iata.tpa), ko var izmantot, lai uzglabātu un pārvaldītu COVID-19 testus vai vakcinēšanās sertifikātus. Tas ļauj pilnvarotajiem testēšanas centriem nosūtīt testu rezultātus un vakcinēšanās sertifikātus pasažieriem, lai pierādītu savu COVID-19 veselības stāvokli attiecīgajām iestādēm ceļojuma laikā. Turklāt IATA *Travel Pass* izmanto decentralizētas pārbaudāmas pilnvaras (*Verifiable Credentials*), tāpēc lietotāju dati glabājas tikai viņu ierīces. Pārbaudāmās pilnvaras izmanto arī COVID-19 akreditācijas iniciatīva (CCI, https://www.covidcreds.org/), kurā apvienojušās vairākas organizācijas un individuāli biedri, lai atbalstītu pārbaudāmo pilnvaru izmantošanu vīrusa izplatības mazināšanai.

4.3. Uzticamība

Lai nodrošinātu COVID-19 pandēmijas kontroles pasākumus un novērstu krāpšanos ar vakcinācijas uzskaiti, vairākas struktūras, tostarp valdības un privātpersonas, ir izstrādājušas vakcinācijas sertifikātu kvadrātkodu glabāšanas un pārbaudes lietotnes. Vakcinācijas lietotnes izmanto, lai atvieglotu privātpersonu vakcinācijas un izslimošanas statusa kontroli un veidotu atsevišķas zonas, kurās samazināt iespēju inficēties ar vīrusu.

Lai veicinātu uzticību izveidotajai sistēmai, svarīgs faktors ir uzticēšanās lietotnē apstrādāto datu privātumam. Vienotas sertifikātu sistēmas neesamība visā pasaulē ir apgrūtinājusi robežsargu iespējas pārbaudīt sertifikātus un ir ļāvusi vieglāk izplatīties viltotiem sertifikātiem [21]. Kibernoziedzības pieaugums palielina risku, ka pārāk liela paļaušanās uz nepārbaudītām un nedrošām tehnoloģijām var ļaut noziedzniekiem vieglāk iefiltrēties COVID-19 vakcīnu sertifikācijas sistēmā.

4.4. Vispārīgā darbība

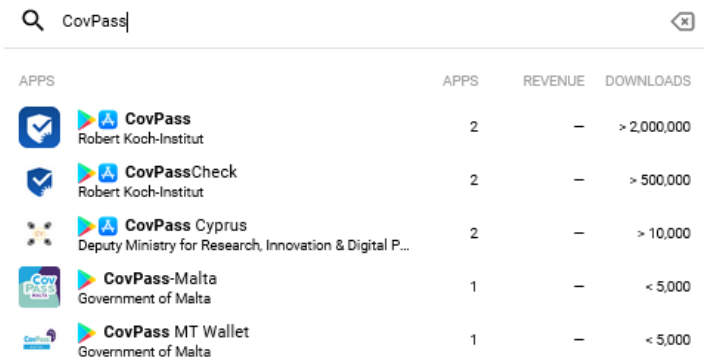
Vakcinācijas sertifikātu lietotnes vispārīgās darbības pamatā ir princips, kur regulāri no centralizēta servera iegūst informāciju par personas imunitātes statusu, un to izmanto mobilajā ierīcē, lai pārbaudītu iesniegtā kvadrātkoda derīgumu. Vakcinācijas lietotņu izmantošana ir brīvprātīga, tomēr tā ir daudz ātrāka un parocīgāka, nekā izdrukāts apliecinājums, tādēļ to praksē plaši izmanto gan uzlabotai ceļošanas kontrolei, gan iekštelpu kontrolei.

4.5. Populārākās lietotnes Latvijā

Palielinoties izveidoto lietotņu skaitam, kā arī lietotņu statistikas tirdzniecības biznesa dēļ, drošības ekspertiem un uzbrucējiem ir salīdzinoši grūti bez maksas noskaidrot, kuras

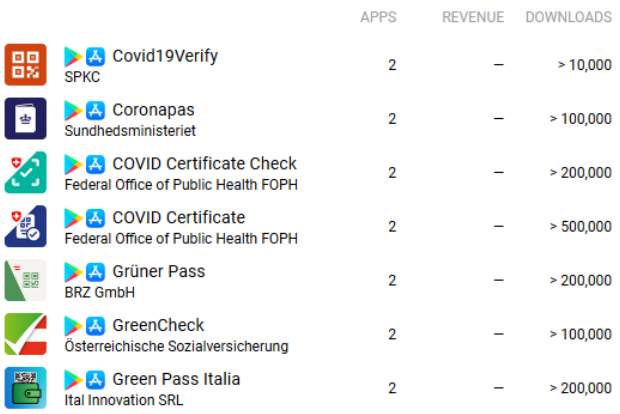
lietotnes ir pašreiz populārākas. Tomēr tas ir nepieciešams, lai varētu aptvert pēc iespējas lielāku lietotāju loku.

Izvēloties lietotni, svarīga ir tās pieejamība konkrētajā valstī, kuras mērķauditoriju ir nepieciešams aptvert. Piemēram, pašreiz Latvijas lietotņu veikalos nav pieejamas tādas Vācijā populāras lietotnes kā *Robert Koch-Institut* izstrādātā *CovPass* (2 miljoni lejupielāžu mēneša laikā) un *CovPassCheck* (500 tūkstoši lejupielāžu mēneša laikā). Taisnības labad jāsaka, ka pašreiz šī lietotne ir pieejama tikai divu valstu lietotņu veikalos — Vācijā un ASV, taču šī pieejamība nav atkarīga no reālās atrašanās vietas vai lietotāja pilsonības, bet gan no tā, kuras valsts iestatījumus lietotājs izmanto savā viedierīcē. Tas nozīmē, ka izstrādātāji, kas šādi ierobežo COVID-19 sertifikātu lietotņu pieejamību, patiesībā pasliktina vīrusa izplatības kontroli, jo šādas lietotnes vairāk izmanto tieši ceļotāji, kuri ne vienmēr iestata ierīci uz savas dzimtenes lietotņu veikalu.



4.1. att. **Lietotņu meklēšana *Appmagic* (autora ekrānattēls)**

Par laimi ir pietiekoši daudz izstrādātāju, kas ļauj lietotnes uzstādīt lielākā skaitā valstu veikalu. Lai atlasītu lietotnes, var izmantot lietotņu veikalus, uzstādot visas lietotnes, kuras atbilst aprakstam. Tomēr, iegūstot klāt statistikas datus, iespējams noteikt, kuras lietotnes izmanto biežāk, kā arī uzzināt par citām populārām lietotnēm, kas nav pieejamas konkrētās valsts lietotņu veikalā (4.1. att.).



4.2. att. **Lietotņu atlase pēc lejupielāžu skaita *Appmagic* (autora ekrānattēls)**

Izmantojot statistikas rīku *Appmagic* (https://appmagic.rocks), viegli pārliecināties par pēdējā laikā visbiežāk lejupielādētajām lietotnēm, kā arī to pieejamību (4.2. att.). Tāpat *Appmagic* tiešsaistes rīks ļauj bez maksas aplūkot grafiku ar lietotņu popularitātes izmaiņām (4.3. att.) gan pēc mobilajām platformām, gan pieejamības valstu veikalos.

4.3. att. **Lietotņu popularitāte *Appmagic* (autora ekrānattēls)**

Lietotņu popularitāti raksturo dažādi parametri, piemēram, popularitātes indekss, ienākumi no lietotnes tirdzniecības vai reklāmām, kā arī pieejamība dažādās valstīs un lejupielāžu skaits pēdējā mēneša laikā. Izmantojot šos parametrus, viegli atlasīt populārākas lietotnes.

Lai izveidotu populārāko vakcinācijas sertifikātu mobilo lietotņu sarakstu (4.1. tab.), autors uzstādīja 20 dažādas lietotnes, kas pieejamas *App Store*, un pārliecinājās par to atbilstību Eiropas vakcinācijas sertifikāta atbalstam. Izmantojot *Appmagic*, autors atlasīja populārākās lietotnes, kuras bija pieejamas gan *Android*, gan *iOS* platformā. Sarakstā tika iekļauta arī oficiālā Latvijā izveidotā lietotne *Covid19Verify*.

4.1. tabula

Latvijas veikalos pieejamās populārākās vakcinācijas sertifikātu lietotnes (autora apkopojums)

N.p.k.	Nosaukums	Kopējā veiktspēja (*Appmagic*)	Reģistrācija	Izstrādātāja valsts	Sertifikāta glabāšana	Sertifikāta pārbaude
1.	*Grüner Pass*	49.041	NAV	Austrija	IR	IR
2.	*Coronapass*	8.774	IR (NemID)	Dānija	Tikai ar NemID	IR
3.	*COVID Certificate*	7.352	NAV	Šveice	IR	IR
4.	*COVID Certificate Check*	4.448	NAV	Šveice	NAV	IR
5.	*GreenCheck*	1.158	NAV	Austrija	NAV	IR
6.	*GreenPass Italia*	1.048	IR	Itālija	IR	NAV
7.	*Covid19Verify*	186	NAV	Lavija	NAV	IR

5. DRAUDU MODELĒŠANA

5.1. Draudu modelēšanas pieeja

Vakcinācijas sertifikātu mobilu lietotņu draudu modelēšanas pieeja sniedz atbildi uz jautājumu "Kādām drošības prasībām jāatbilst vakcinācijas sertifikātu lietotnei?", t.i., ko var uzskatīt par drošu vakcinācijas sertifikātu lietotni. Lai gan kvadrātkodi ar digitālajiem parakstiem ievērojami apgrūtina vakcīnu sertifikātu viltošanu, tie nav pilnībā nevainojami. Pastāv iespējamība, ka pandēmijas situācijas ar cita veida vīrusiem var atkārtoties arī nākotnē, tādēļ drošības jautājumu detalizēta apsvēršana ir nepieciešama, lai varētu novērtēt dažādās drošības un privātuma kontroles, kuras jāiestrādā vakcinācijas sertifikātu mobilajās lietotnēs.

5.2. Draudu modelēšanas soļi

Drošas sistēmas projektēšana ir sarežģīts uzdevums. Bieži vien sistēmas izstrādā tikai biznesa prasību apmierināšanai. Draudu modelēšana ir pieeja, kas palīdz identificēt lietojumprogrammas potenciālos drošības draudus un ievainojamības jau projektēšanas posmā, kā arī izveidot atbilstošas drošības prasības, lai novērstu draudus. Tas ir svarīgi, jo testēšanas posmā atklāto drošības problēmu novēršana ir laikietilpīga un dārga.

Iedomāsimies, ka izstrādātāju komandu strādā pie jaunas mobilās lietotnes. Šķiet, ka viss norit gludi, un, tuvojoties darba nodošanai, mārketinga komanda ir gatava paziņot pasaulei par lietotni un gaidīt veiksmīgu atklāšanu. Tad notiek kaut kas negaidīts — sistēmas API saskarne nav pieejama nezināma avota izplatītā pakalpojuma atteikuma (DDoS) uzbrukuma dēļ. Lai gan lietotnes izveide bijusi veiksmīga, komanda nav spējusi to pasargāt no ievainojamības, jo tā nebija iepriekš paredzēta. Lai novērstu šādu scenāriju nākotnē, ir nepieciešama draudu modelēšana.

Pirms draudu modeļa definēšanas ir būtiski saprast, kas ir draudi un kāpēc modelēšanas process ir noderīgs. Saskaņā ar NIST definīciju publicētajās riska novērtēšanas vadlīnijās [27] draudi ir apstākļi vai notikumi, kas var negatīvi ietekmēt organizācijas darbību (tostarp misiju, funkcijas, tēlu vai reputāciju), organizācijas resursus, indivīdus, citas organizācijas vai valsti, izmantojot informācijas sistēmu, kur notikusi nesankcionēta piekļuve, datu iznīcināšana, informācijas izpaušana, modifikācija vai pakalpojuma atteikums. Savukārt modelis ir konkrētas jomas abstrakts attēlojums, kas veidots no cilvēka zināšanām, kuru var izmantot, lai strukturētu zināšanas, nodrošinātu kopīgu valodu šo zināšanu apspriešanai un veiktu jomas analīzi [20].

Apvienojot iepriekšminētos jēdzienus, izriet, ka kiberdraudu modelēšana ir process, kurā izstrādā un piemēro kiberdraudu (avotu, scenāriju un konkrētu notikumu) reprezentāciju [20].

Draudu modelēšana var palīdzēt uzlabot kiberdrošību un noturību daudzos veidos, tostarp riska pārvaldībā, kiberdrošības simulācijās, tehnoloģiju profilēšanā un meklēšanā, sistēmu drošības inženierijā, drošības operācijās un analīzē. Apdraudējumu modelēšanas process ietver piemērotas draudu modelēšanas sistēmas izvēli un tās aizpildīšanu ar novērotiem vai hipotētiskiem datiem.

Aplūkosim vēl vienu piemēru. Pieņemot, ka mobilai lietotnei nav veikta draudu modelēšana, testētājs ielaušanās testēšanas posmā konstatē, ka uzbrucējs var manipulēt ar pieprasījumu, izmainot datus sistēmā. Izplatītākie iemesli, kāpēc tas varētu būt iespējams, ir šādi:

1. Lietotājiem ir atļauts veikt kritiskas operācijas bez atkārtotas autentifikācijas.
2. Pirms apstrādes nenotiek ievades datu validācija.
3. Kļūdu atbildēs tiek atklāta sensitīva informācija, piemēram, sistēmas dati, sesijas identifikatori vai cita nozīmīga informācija.

Draudu modelēšanai nav vienotas pieejas. Katram no mums var būt atšķirīga pieeja draudu modelēšanai atkarībā no projekta prasībām vai mūsu individuālās pieredzes. Tomēr draudu modelēšanā var izšķirt vismaz 5 soļus, lai panāktu optimālu drošību.

5.2.1. Pirmais solis: drošības mērķu identificēšana

Jāizprot drošības prasības un jāidentificē iespējamie draudi biznesa plūsmās. Jāapsver, vai pastāv kādas īpašas atbilstības vai drošības prasības saistībā ar organizācijas darbības mērķiem. Piemēram, neatbilstoša sensitīvas informācijas vākšana, žurnāla faila pieejamība noteiktam lietotāju lokam.

5.2.2. Otrais solis: resursu un ārējo atkarību apzināšana

Draudu rašanās iemesls ir nesankcionēta piekļuve tādiem resursiem kā dati, pirmkods un sistēmas informācija. Ir jāidentificē to resursu saraksts, kas jāaizsargā no potenciālajiem uzbrucējiem. Jāidentificē arī ārējās atkarības, kas nav koda daļa, bet var radīt draudus sistēmai (piemēram, ārējās bibliotēkas, savienojuma šifrēšana).

5.2.3. Trešais solis: uzticamības zonu noteikšana

Ieejas un izejas punktu, kā arī uzticamības zonas identificēšana. Šo informāciju izmanto, lai izstrādātu datplūsmas diagrammas ar privilēģiju robežām, kas palīdz noteikt pieeju lietotāju autentifikācijai, ievades datu validācijai un kļūdu apstrādei.

5.2.4. Ceturtais solis: iespējamo draudu un ievainojamības identificēšana

Papildus plašai draudu meklēšanai atbilstīgi iepriekš noteiktai draudu modelēšanas pieejai, jāapsver draudi, kas parasti ietekmētu jūsu sistēmu. Daži piemēri varētu būt — koda injekcija, bojāta autentifikācija un sesiju pārvaldības ievainojamības. Riskam pakļauto jomu noteikšana, piemēram, nepietiekama ievades validācija, pārmērīgi privileģēti konti, vāja paroļu politika, vāji šifrēšanas algoritmi, neatbilstoša auditēšana, informācijas eksponēšana kļūdu vai izņēmumu ziņojumos.

5.2.5. Piektais solis: draudu modeļa dokumentēšana

Draudu modelēšana ir iteratīvs process. Var izmantot dokumentāciju, lai veidotu drošu arhitektūru un mazinātu apdraudējumus. Draudu modeļa dokumentāciju izstrādātāji izmanto apdraudējumu mazināšanai, bet testētāji, lai veidotu testēšanas gadījumus un atrastu sistēmas ievainojamības uzticamības zonai.

5.3. Potenciālo draudu modelēšana

Potenciālo draudu modelēšana ir nepieciešama, lai apzinātu potenciālos draudus. Potenciālie draudi kļūs par risku tikai tad, ja būs atklāta ievainojamība, kas šos draudus apstiprina. Pēc potenciālo draudu apzināšanas var apsvērt draudu iestāšanās iespējamības potenciālā riska ietekmi. Pēc tam var noteikt, vai riska līmenis ir pieņemams, vai tālāk pārbaudāms (jāveic ielaušanās testi un pirmkoda caurskatīšana), vai arī nekavējoši novēršams (pārtraucams).

5.3.1. Potenciālie drošības mērķi un augsta līmeņa dizains

Vakcinācijas sertifikātu mobilajām lietotnēm var identificēt vairākas kopīgas biznesa un funkcionālās prasības, kas nosaka, kādi varētu būt potenciālie draudi:

1. Uzstādīšana — lielākā daļa lietotņu ir publicēta veikalos *Google Play* un *App Store*.
2. Reģistrācija — lietotnēs, kurās izmanto atsevišķu lietotāja reģistrāciju un autentifikāciju.
3. Derīgo un atcelto sertifikātu informācijas nodošana — PKI uztur sertifikātu un tajos ietverto publisko atslēgu sarakstu, kā arī sertifikātu atsaukuma sarakstu, kuru pēc lietotnes pieprasījuma šifrētā veidā nosūta veselības jomas institūcijas kontrolētais API serveris [4].

4. Kvadrātkodu pārbaude — kvadrātkodu pārbaude var notikt lokāli, izmantojot mobilās ierīces operētājsistēmu, vai arī ar tīmekļa API saskarnes palīdzību.
5. Kvadrātkodu saglabāšana — lietotnē glabātie vai pārbaudītie kvadrātkodi var būt saglabāti mobilās ierīces lokālajā datubāzē un darbības žurnālā.
6. Sertifikātu statusa maiņa — automātisks (piemēram, derīguma termiņš) vai manuāls (piemēram, jauna vakcīna, balstvakcīna) process, kuru kontrolē veselības jomas institūcija.

5.3.2. Potenciālie draudu aktori

Var identificēt vairākus iespējamos aktorus un mērķus:

1. Saslimušais — lai pārkāptu noteiktos ierobežojumus.
2. Vakcinācijas procesa pretinieks — lai slēptu savu vakcinācijas statusu un radītu dezinformāciju.
3. Vakcinācijas sertifikātu pretinieks, privātuma aktīvists — lai diskreditētu sertifikātu izmantošanas procesu un grautu sistēmas reputāciju.
4. Negodīgs veselības jomu kontrolējošās institūcijas darbinieks, administrators — lai iegūtu sensitīvus datus vai materiālu labumu.
5. Ļaunatūras veidotājs, uzbrucējs — lai veiktu pārtveršanas uzbrukumus, nozagtu, viltotu vai izmanītu nosūtītos datus, iegūtu plašākas lietotāja tiesības, apdraudētu iekārtu integritāti, konfidencialitāti vai pieejamību un iegūtu materiālo labumu.
6. Trešās puses — lai izmantotu noplūdušos sensitīvos datus.
7. Konkurējošās lietotnes veidotājs — lai palielinātu savas lietotnes izmantošanu un gūtu ienākumus no reklāmas.

5.3.3. Potenciālie galvenie resursi

Lokālie resursi — viedierīce:

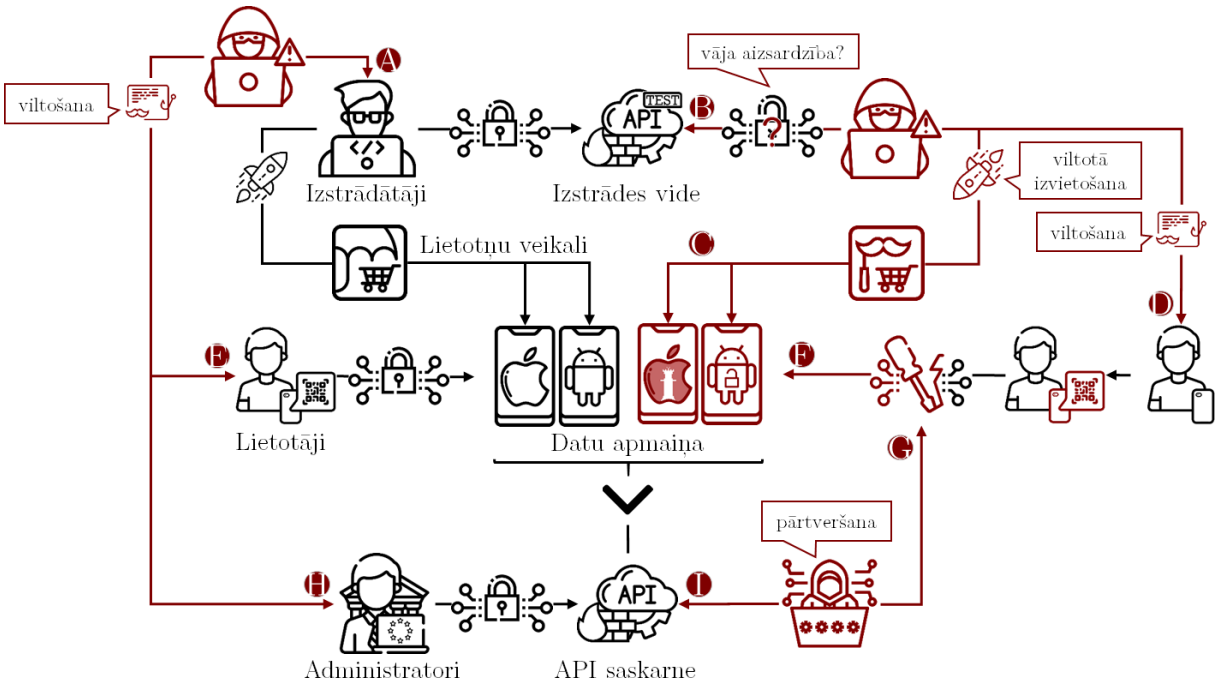
1. Derīgo un atcelto sertifikātu (vai publisko atslēgu) saraksts.
2. Pārbaudīto sertifikātu datubāze.
3. Personu identificējoša informācija — tālruņa numurs, e-pasts u.c. dati.

Attālā infrastruktūra — API saskarnes serveris, izstrādes vides serveris:

1. Personu identificējoša informācija — tālruņa numurs, e-pasts u.c. dati.
2. Vakcinēto lietotāju derīgo un atcelto sertifikātu datubāze.
3. TLS sertifikāti, lai pieslēgtos augstāka līmeņa vārtejai.

4. Piekļuve nacionālai infrastruktūrai un akreditācijas dati tās pārvaldībai.
5. *Google* un *Apple* izstrādātāju akreditācijas dati jaunu lietotnes versiju izlaišanai.
6. Atbildīgās veselības jomu kontrolējošās institūcijas akreditācijas dati, kas dod tiesības papildināt sertifikātu datubāzi.

5.3.4. Potenciālās ievainojamības, draudi un uzbrukuma vektori



5.1. att. **Potenciālās ievainojamības, draudi un uzbrukuma vektori (autora zīmējums)**

Lai uzskatāmāk aplūkotu potenciālās vakcinācijas sertifikātu lietotņu ievainojamības, draudus un uzbrukuma vektorus, aplūkosim autora veidotu zīmējumu (5.1. att.). Uzbrukuma draudu vektori, aktori un ievainojamības ir attēlotas tumši sarkanā krāsā, bet augsta līmeņa dizaina elementi — melnā. Aplīšos attēlotos burtus raksturo dažādas ievainojamības, draudi un uzbrukuma vektori:

A.

1. Uzbrukumi izstrādātāju kontiem vai izmantotajām bibliotēkām un satvariem, lai *Google Play* vai *App Store* veikalos izvietotu modificētas lietotnes, kas nokompilētas ar izmainītu pirmkodu, vai pārpakotas, lai saturētu slēptu ļaunatūru (piemēram, nolūkā vākt lietotāja datus).
2. Pikšķerēšana ar saiti uz ļaunatūru.

B.

1. Sensitīvu datu noplūde, izmantojot publiski pieejamu testa vidi, piemēram, izmantojot vājākas drošības kontroles un patiesos (nevis testa) datus.

2. Lietotnes un tās infrastruktūras vai izstrādes infrastruktūras pārņemšana, kompromitējot izstrādātāja vai administratora priviliģētos lietotāja kontus, tai skaitā, uzbrūkot veselības jomu kontrolējošās institūcijas, izstrādātāju vai apakšuzņēmēju infrastruktūrai.

C.

1. Uzbrukumi lietotājiem, kuri izmanto trešās puses lietotņu veikalus, izvietojot modificētas lietotnes, kas nokompilētas ar izmainītu pirmkodu, vai pārpakotas, lai saturētu slēptu ļaunatūru (piemēram, ar nolūku vākt lietotāja datus).
2. Sensitīvu datu noplūde, izmantojot citas lietotnes (ļaunatūras), kas uzstādītas viedierīcē.
3. Pikšķerēšana, izmainot lietotnē attēloto paziņojumu, iekļaujot saiti uz ļaunatūru.

D.

1. Lietotājs neizmanto lietotni, jo izstrādātājs to tīši vai netīši ir padarījis grūti lietojamu, piemēram, izveidojis neērtu saskarni, nav novērsis būtiskas funkcionālās kļūdas, neoptimāli izmanto aparatūras resursus vai pieprasa aizdomīgi plašas tiesības.
2. Negatīvi komentāri un vērtējumi lietotņu veikalos, lai radītu šaubas par lietotnēs apstrādāto datu konfidencialitāti.
3. Viltus ziņu publicēšana par kritisku ievainojamību atklāšanu.
4. Zema riska ievainojamības atklāšana, kuru mediji apraksta kā kritisku, lai veidotu nomelnošanas kampaņas.
5. Pikšķerēšana, izmainot lietotnē attēloto paziņojumu, iekļaujot saiti uz ļaunatūru.

E.

1. Sensitīvu datu noplūde, izmantojot trešās puses, piemēram, atkļūdošanas vai lietojuma statistikas pakalpojumus.
2. Viltus ziņu publicēšana par kritisku ievainojamību atklāšanu.
3. Lietotnē izmantotās trešās puses bibliotēkas kompromitēšana — modificējot bibliotēku, lai izmainītu vai nozagtu sensitīvus datus.
4. Vakcinācijas statusa izmanīšana, apejot veselības jomu kontrolējošās institūcijas apstiprinājumu.
5. Pikšķerēšana ar saiti uz ļaunatūru.

F.

1. Izmainītu kvadrātkodu pārbaude, lai viltotu vakcinācijas statusu, izraisītu lietotnes izņēmumus vai atklātu iekšējo informāciju.
2. Pikšķerēšana, izmainot lietotnē attēloto paziņojumu, iekļaujot saiti uz ļaunatūru.
3. Vakcinācijas statusa viltošana, attēlojot “zaļu” statusu, kad patiesībā sertifikāts vairs nav derīgs, vai otrādi.
4. Lokāla sertifikāta datu iegūšana no nozagtas ierīces, lai to izmantotu krāpšanā.

G.

1. Lietotnē izmantotās trešās puses bibliotēkas kompromitēšana — modificējot bibliotēku, lai izmainītu vai nozagtu sensitīvus datus.
2. Vakcinācijas statusa izmanīšana, apejot veselības jomu kontrolējošās institūcijas apstiprinājumu.
3. Datu pārtveršanas uzbrukumi (*Man in the Middle*) un datu izmainīšana starp lietotni un API saskarni.

H.

1. Lietotnes un tās infrastruktūras vai testa infrastruktūras pārņemšana, kompromitējot izstrādātāja vai administratora priviliģētos lietotāja kontus, tai skaitā uzbrūkot veselības jomu kontrolējošās institūcijas, izstrādātāju vai apakšuzņēmēju infrastruktūrai.
2. Viltus ziņu publicēšana par kritisku ievainojamību atklāšanu.
3. Pikšķerēšana ar saiti uz ļaunatūru.

I.

1. Datu pārtveršanas uzbrukumi (*Man in the Middle*) un datu izmainīšana starp lietotni un API saskarni.
2. Sensitīvo datu noplūde, atklājot ievainojamības API saskarnē vai tās infrastruktūrā, piemēram, piekļuves kontroles apiešanu, slēptu funkcionalitāti, koda injekcijas, attālā koda izpildīšanu.
3. Pakalpojuma atteikuma uzbrukumi API saskarnei vai tās infrastruktūrai, piemēram, izmantojot API saskarnes datu apmaiņas ierobežojumu trūkumus, padarot nepieejamas svarīgas sistēmas funkcijas.
4. Datu integritātes ietekmēšana, atklājot ievainojamības API saskarnē vai tās infrastruktūrā.

6. DRAUDU ANALĪZE

6.1. Rīku sagatavošana un lietotņu ieguve

Lai iegūtu iespējas pārtvert datus, aplūkot iekšējos tālruņa procesus un iegūt lietotņu failus drošības pārbaudēm, nozīmīga loma ir "root" statusa (*Android*) un "jailbreak" statusa (*iOS*) iegūšanai. Izmantojot augstākā līmeņa pilnvaras, iespējams piekļūt pilnīgi visām tālruņa operētājsistēmas darbībām. Tomēr jāatceras, ka tas rada ne tikai ieguvumus testētājam, bet arī drošības riskus, ja šādu ierīci izmanto ikdienā, tāpēc testam tika izmantotas atsevišķas fiziskas ierīces (*iPhone 6s* tālruņi ar *iOS* versijām 14.4 un 14.8, kā arī *Samsung Galaxy S8* ar *Android* 9. versiju).

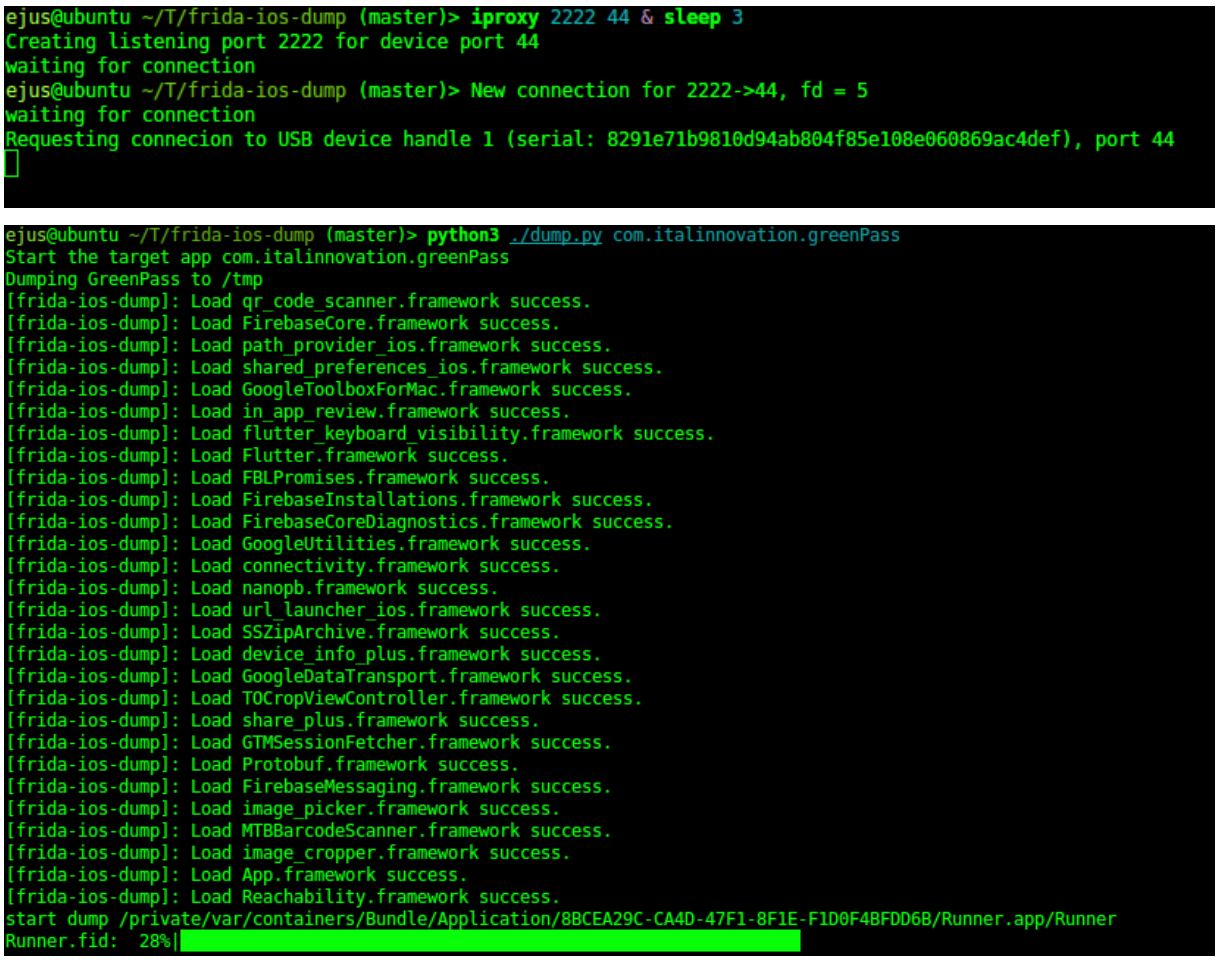
6.1. tabula

Tehniskajā drošības pārbaudē izmantoto lietotņu versijas un identifikatori

Nr.p.k.	Nosaukums	Identifikatori	Versija (*iOS*)	Versija (*Android*)
1.	*Coronapas*	dk.sum.ssicpas	1.4.6	1.4.6
2.	*Covid Cert*	ch.admin.bag.covidcertificate.wallet	4.1.0	4.1.0
3.	*Covid Check*	ch.admin.bag.covidcertificate.verifier	4.0.0	4.0.0
4.	*Covid19Verify*	lv.verification.dgc	1.3.10	1.3.10
5.	*GreenCheck*	at.itsv.mobile.cochap	1.18	1.18
6.	*GreenPass Italia*	com.italinnovation.green_pass, com.italinnovation.greenPass	1.4.2	2.0.1
7.	*Grüner Pass*	at.gv.brz.wallet	2.3.2	2.3.2

Pārbaudot analīzei izvēlētās Latvijā populārās lietotnes (6.1. tab.), viedtelefonos ar pilnām lietotāja tiesībām bija novērojams, ka visas atlasītās lietotnes var uzstādīt un darbināt bez izrobežojumiem. Tas nozīmē, ka šī darba ietvaros nebūs nepieciešamība slēpt lietotnei ierīces pilno tiesību statusu, ko parasti panāk ar tādām programmām kā *ih8sn*, *Shamiko*, *Zygsik* (*Android*) vai *A-Bypass*, *iHide*, *JailProtect*, *KernBypass*, *LibertyLite*, *NoJail*, *Shadow* (*iOS*). *Google* veikalā iegūtās lietotnes parasti izmanto *SafetyNet* atslēgas atestācijas API, lai attālināti pārbaudītu ierīces aizsardzības statusu, bet *iOS* atlaušanas statusu pārbauda ar vairākām metodēm, pārliecinoties par lietotnes pieprasītajiem neparedzētajiem vai neatļautajiem operētājsistēmas resursiem, kurus var uzstādīt tikai atlauztās ierīcēs.

Lai iegūtu analīzei nepieciešamās lietotnes, *Android* tika izmantota programma *Android Debug Bridge* (*adb*). Lietotnes bināro failu var iegūt, izmantojot komandu "`adb pull`" un norādot attiecīgo *.APK failu, kas glabājas mapē `/data/app/{id}` (lai noskaidrotu ceļus, tika izmantota komanda "`adb shell pm list packages -f -3`"). Operētājsistēmai *iOS* tika izmantoti rīki *iproxy* (*itunnel SSH over USB*), *frida-ios-dump* (https://github.com/AloneMonkey/frida-ios-dump), kā arī *Frida* (https://github.com/frida) serveris (6.1. att.).



6.1. att. **Programmas iegūšana no *iOS* ierīces, izmantojot *iproxy* un *frida-ios-dump* rīkus (autora ekrānattēls)**

6.2. Tehniskās drošības pārbaude

Lai analizētu izvēlēto lietotņu koda līmeņa kļūdas, kas varētu radīt potenciālus drošības riskus, tika izmantota statiskā un dinamiskā analīze. Statisko analīzi veic, faktiski neizpildot programmu, bet izmantojot skeneri — tāpēc šīs analīzes veikšanai bija nepieciešams iegūt programmas bināro kodu. Statiskās analīzes galvenais ierobežojums ir viltus pozitīvo rezultātu īpatsvars, kas padara šo analīzi neefektīvu. Viltus pozitīvie rezultāti rodas, jo tiek analizēti visi programmas ceļi, bet daži no tiem tiek kļūdaini noteikti kā nedroši. Dinamiskās analīzes laikā lietotājs rīkojas ar lietotni un novēro darbību izraisītās izmaiņas. Dinamiskajā analīzē ir mazāk viltus pozitīvo rezultātu, bet pastāv iespēja, ka kādas ievainojamības nav pamanītas, ja kāds programmas izpildes ceļš paliek nepamanīts.

Ar mobilo drošības satvaru *MobSF* (https://github.com/MobSF/Mobile-Security-Framework-MobSF) var veikt gan statisko, gan dinamisko analīzi, kā arī meklēt ļaunprātīgu programmatūru un testēt API. Detalizētas informācijas atskaites par veiktās statiskās lietotņu analīzes konstatējumiem, kas ir apkopoti 6.2. tabulā, ir atrodami tīmeklī —

iOS: https://github.com/Ejuc/covid_apps/tree/main/Static%20Analysis%20IPA, bet *Android*: https://github.com/Ejuc/covid_apps/tree/main/Static%20Analysis%20APK.

6.2. tabula

Mobilo lietotņu statistiskās analīzes salīdzinājums

Nr. p.k.	Nosaukums	Drošības līmenis	Riska novērtējums	Ierīču izsekošana	Augsta prioritāte	Brīdinājumi	OS
1.	*Coronapas*	64/100	A (zems risks)	0	0	4	*Android*
2.	*Covid Cert*	64/100	A (zems risks)	0	0	8	*Android*
3.	*Covid Check*	70/100	A (zems risks)	0	0	5	*Android*
4.	*Covid19Verify*	61/100	A (zems risks)	0	1	5	*Android*
5.	*GreenCheck*	53/100	B (vidējs risks)	0	2	9	*Android*
6.	*GreenPass Italia*	52/100	B (vidējs risks)	1	2	13	*Android*
7.	*Grüner Pass*	58/100	B (vidējs risks)	0	1	8	*Android*
8.	*Coronapas*	36/100	C (augsts risks)	0	3	2	*iOS*
9.	*Covid Cert*	44/100	B (vidējs risks)	0	2	2	*iOS*
10.	*Covid Check*	44/100	B (vidējs risks)	0	2	2	*iOS*
11.	*Covid19Verify*	44/100	B (vidējs risks)	0	2	2	*iOS*
12.	*GreenCheck*	44/100	B (vidējs risks)	0	2	2	*iOS*
13.	*GreenPass Italia*	25/100	F (kritisks risks)	2	3	3	*iOS*
14.	*Grüner Pass*	44/100	B (vidējs risks)	0	2	2	*iOS*

6.2. Attālo galapunktu testēšana

Lai saskaņotu un autorizētu tehniskos drošības testus, darba autors sagatavoja informāciju par plānotajiem testiem, to mērķi un nosūtīja vēstules uz lietotnēs un to tīmekļa vietnēs norādītajām e-pasta adresēm: Dānijas Veselības ministrijas Nacionālajam COVID-19 komunikācijas centram (lietotne *Coronapas*), Šveices Federālajam sabiedrības veselības birojam (lietotnes *Covid Cert* un *Covid Check*), Latvijas Veselības ministrijas Slimību profilakses un kontroles centram (lietotne *Covid19Verify*), Austrijas Federālajam skaitļošanas centram (lietotne *Grüner Pass*), Austrijas uzņēmumam *ITSV GmbH* (lietotne *GreenCheck*) un Itālijas uzņēmumam *Ital Innovation SRL* (lietotne *GreenPass Italia*). No adresātiem tika saņemtas divas atbildes — automātisks apliecinājums no Dānijas Veselības ministrijas (*Coronapas*) un noraidoša atbilde no Austrijas uzņēmuma *ITSV GmbH* (*GreenCheck*). Iepriekš

aprakstīto apstākļu dēļ, darba autors tehniskās drošības pārbaudes API galapunktiem izdarīja ierobežotā apjomā, vairāk koncentrējoties uz lietotņu uzbūves statisko un dinamisko pārbaudi.

Dinamisko testu laikā tika atrasti vairāki attāli galapunkti, kuru datplūsma lietotnēs nebija pienācīgi aizsargāta ar drošības sertifikātu piespraušanu vai arī bija viegli apejama, izmantojot starpniekserveri *BurpSuite* (https://portswigger.net/burp), serveri *Frida* un dinamiskās analīzes komandrindas rīkkopu *objection* (https://github.com/sensepost/objection). Rīkkopa *objection* tika izmantota dažādām reāllaika manipulācijām, lai novērotu satura izmaiņas atslēgu un failu glabātuvēs, starpliktuvē un datubāzēs. Izmantojot iepriekš minētos rīkus, tika noklausīta un atšifrēta datplūsma lietotnēs *Covid19Verify*, *Grüner Pass*, *GreenPass* un *GreenCheck*, bet lietotnēs *Coronapas*, *Covid Cert* un *Covid Check* pārtvert un izmanīt datplūsmu ar iepriekš aprakstīto rīku palīdzību neizdevās.

6.3. tabula

Lietotņu datplūsmas galapunkti un konstatējumu prioritāte

Nr.p.k.	Nosaukums	Galapunkti	Vidēja prioritāte	Zema prioritāte
1.	*Coronapas*	https://api.coronapas.sunhedsdata.dk	n/a	n/a
2.	*Covid Cert*	https://www.cc.bit.admin.ch	n/a	n/a
3.	*Covid Check*	https://www.cc.bit.admin.ch	n/a	n/a
4.	*Covid19Verify*	https://api.covid19sertifikats.lv	0	41
5.	*GreenCheck*	http://greencheck.gv.at	0	33
6.	*GreenPass Italia*	https://firebaseinstalations.googleapis.com	1	1
7.	*Grüner Pass*	https://dgc-trust.qr.gv.at	0	6

Visi reģistrētie konstatējumi bija šādās kategorijās:

1. Drošības galveņu trūkums vai nepareiza konfigurācija.
2. Satura drošības politikā nav prasīta apakšresursu integritāte.
3. Satura noklausīšanās nav atspējota.
4. Pārlūkprogrammas starpvietņu skriptēšanas filtru nepareiza konfigurācija.
5. JavaScript elementiem trūkst apakšresursu integritātes atribūtu.

Datplūsmas pārbaužu rezultāti, kas apkopoti 6.3. tabulā detalizēti ir pieejami tiešsaistē https://github.com/Ejuc/covid_apps/tree/main/Burp%20Reports. Latvijas Veselības ministrijas Slimību profilakses un kontroles centra lietotnē *Covid19Verify* notika datplūsma ar galapunktu https://api.covid19sertifikats.lv un tika reģistrēts 41 konstatējums ar zemu prioritāti. Austrijas uzņēmuma *ITSV GmbH* lietotnē *GreenCheck* notika datplūsma ar galapunktu http://greencheck.gv.at un reģistrēti 33 zemas prioritātes konstatējumi. Itālijas uzņēmuma *Ital Innovation SRL* lietotnē *GreenPass Italia* notika datplūsma ar galapunktu https://firebaseinstalations.googleapis.com un reģistrēti 2 konstatējumi — viens ar vidēju, bet otrs ar zemu prioritāti. Austrijas Federālā skaitļošanas centra lietotnē *Grüner Pass* notika datplūsma ar galapunktu https://dgc-trust.qr.gv.at un reģistrēti 6 konstatējumi ar zemu prioritāti.

6.3. Nozīmīgākie konstatējumi

6.3.1. Lietotne Coronapas

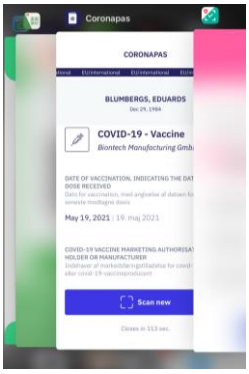
Identifikators: dk.sum.ssicpas

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.
2. Lietotne izmanto nedrošas gadījuma skaitļu funkcijas.

Citi nozīmīgi konstatējumi:

1. *iOS* lietotne *Coronapas* fona režīmā neslēpj informāciju (6.2. att.)



6.2. att. **Lietotne *Coronapas* fona režīmā (autora ekrānattēls)**

2. TLS savienojums ir pakļauts ievainojamībai (6.3. att.), ja klients atjauno savienojumu, nevis izveido to no jauna, izraisot pakalpojumatteici vai pārtverot šifrētus datus.

```
                                 -->> 87.48.150.232:443 (api.coronapas.sundhedsdata.dk) <<--

rDNS (87.48.150.232):   --
Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      not offered
 TLS 1.1    not offered
 TLS 1.2    offered (OK)
 TLS 1.3    not offered and downgraded to a weaker protocol
 NPN/SPDY   not offered
 ALPN/HTTP2 http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                      not offered (OK)
 Anonymous NULL Ciphers (no authentication)        not offered (OK)
 Export ciphers (w/o ADH+NULL)                     not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)      not offered (OK)
 Triple DES Ciphers / IDEA                         not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)            not offered
 Strong encryption (AEAD ciphers) with no FS       not offered
 Forward Secrecy strong encryption (AEAD ciphers)  offered (OK)


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session ticket extension
 ROBOT                                     Server does not support any cipher suites that use RSA key transport
 Secure Renegotiation (RFC 5746)           supported (OK)
 Secure Client-Initiated Renegotiation     VULNERABLE (NOT ok), DoS threat (6 attempts)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    no gzip/deflate/compress/br HTTP compression (OK)  - only supplied "/" tested
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no SSLv3 support
 TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible (OK), no protocol below TLS 1.2 offered
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           make sure you don't use this certificate elsewhere with SSLv2 enabled services, see
                                           https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=F41E37FBA2D078B9FE4342
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
 BEAST (CVE-2011-3389)                     not vulnerable (OK), no SSL3 or TLS1
 LUCKY13 (CVE-2013-0169), experimental     not vulnerable (OK)
 Winshock (CVE-2014-6321), experimental    not vulnerable (OK) - ARIA, CHACHA or CCM ciphers found
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)
```

6.3. att. **TLS konfigurācijas un šifru pārbaude (autora ekrānattēls)**

6.3.2. Lietotne COVID Certificate

Identifikators: ch.admin.bag.covidcertificate.wallet

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.
2. Lietotne izmanto funkciju "malloc".

Citi nozīmīgi konstatējumi:

1. *iOS* saskarnē *NSuserDefaults*, parametrā "lastConfigURL" (6.4. att.) glabājas lietotnes konfigurācijas saite https://www.cc.bit.admin.ch/app/wallet/v1/config?appversion=ios-4.1.0&buildnr=ios=220429.1021.99279&osversion=ios14.4.
2. Kešatmiņas datu bāzē ir apskatāmi galapunkti (6.4. att.).

6.4. att. **Kešatmiņas apskate (autora ekrānattēls)**

6.3.3. Lietotne COVID Certificate Check

Identifikators: ch.admin.bag.covidcertificate.verifier

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.
2. Lietotne izmanto funkciju "malloc".

Citi nozīmīgi konstatējumi:

1. *iOS* saskarnē *NSuserDefaults*, parametrā "lastConfigURL" glabājas lietotnes konfigurācijas saite https://www.cc.bit.admin.ch/app/verifier/v1/config?appversion=ios-4.0.0&buildnr=ios=220411.1280.99268&osversion=ios14.4.
2. Kešatmiņas datu bāzē ir apskatāmi galapunkti (6.5. att.)

request_key
https://www.cc.bit.admin.ch/trust/v1/metadata
https://www.cc.bit.admin.ch/trust/v2/keys/list
https://www.cc.bit.admin.ch/trust/v2/verificationRules
https://www.cc.bit.admin.ch/app/verifier/v1/config?appversion=ios-4.0.0&buildnr=ios-220411.1820.99268&osversion=ios14.8
https://www.cc.bit.admin.ch/trust/v2/keys/updates?certFormat=IOS&since=&upTo=12307
https://www.cc.bit.admin.ch/trust/v2/keys/updates?certFormat=IOS&since=12307&upTo=12307
https://www.cc.bit.admin.ch/trust/v2/revocationList?since=11822103

6.5. att. **Kešatmiņas apskate (autora ekrānattēls)**

6.3.4. Lietotne Covid19Verify

Identifikators: lv.verification.dgc

iOS statiskās analīzes augstas prioritātes konstatējumi:

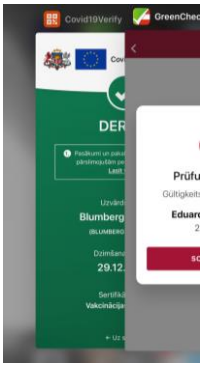
1. Lietotne pieļauj ATS (*App Transport Security*) nedroša savienojuma izmantošanu domēniem, kas nav iekļauti saskarnē *NSExceptionDomains*.
2. Lietotne neizmanto funkciju "-fstack-protector-all", kas pasargātu no steka adreses pārrakstīšanas.

Android statiskās analīzes augstas prioritātes konstatējumi:

1. Serviss "com.google.android.play.core.AssetPackExtractionService" nav aizsargāts no citu lietotņu piekļuves.

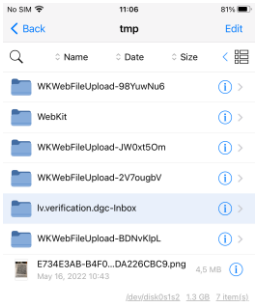
Citi nozīmīgi konstatējumi:

1. *iOS* lietotne *Covid19Verify* fona režīmā neslēpj informāciju (6.6. att.)



6.6. att. **Lietotne fona režīmā (autora ekrānattēls)**

2. *iOS* lietotne *Covid19Verify* pagaidu failu mapē "tmp" (6.7. att.) saglabā visus augšupielādētos attēlus, PDF failus un ar kameru uzņemtās fotogrāfijas. Šie faili automātiski netiek dzēsti. Laika gaitā mapes izmērs kļūst ļoti liels. Ja programmu izmanto publiskai sertifikātu pārbaudei, ir lielas iespējas nozagt sensitīvu informāciju, ja tiek iegūtas pilnas tiesības ierīcē.



6.7. att. **Pagaidfailu mape (autora ekrānattēls)**

3. Lietotne lejupielādē konfigurāciju no galapunkta https://covid19sertifikats.lv/verify/rules.json, kas izmanto nedrošu TLS konfigurāciju un šifrus, kas ir pakļauti zināmām ievainojamībām (6.8. att.).

```
                                 -->> 104.21.8.55:443 (covid19sertifikats.lv) <<--

Further IP addresses:   172.67.156.224 2606:4700:3036::6815:837 2606:4700:3034::ac43:9ce0
rDNS (104.21.8.55):     --
Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

 SSLv2      not offered (OK)
 SSLv3      not offered (OK)
 TLS 1      offered (deprecated)
 TLS 1.1    offered (deprecated)
 TLS 1.2    offered (OK)
 TLS 1.3    offered (OK): final
 NPN/SPDY   h2, http/1.1 (advertised)
 ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

 NULL ciphers (no encryption)                     not offered (OK)
 Anonymous NULL Ciphers (no authentication)       not offered (OK)
 Export ciphers (w/o ADH+NULL)                    not offered (OK)
 LOW: 64 Bit + DES, RC[2,4], MD5 (w/o export)     not offered (OK)
 Triple DES Ciphers / IDEA                        not offered
 Obsoleted CBC ciphers (AES, ARIA etc.)           offered
 Strong encryption (AEAD ciphers) with no FS      not offered
 Forward Secrecy strong encryption (AEAD ciphers) offered (OK)


 Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                not vulnerable (OK), no heartbeat extension
 CCS (CVE-2014-0224)                       not vulnerable (OK)
 Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no session tickets
 ROBOT                                     Server does not support any cipher suites that use RSA key transport
 Secure Renegotiation (RFC 5746)           OpenSSL handshake didn't succeed
 Secure Client-Initiated Renegotiation     not vulnerable (OK)
 CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
 BREACH (CVE-2013-3587)                    potentially NOT ok, "br gzip" HTTP compression detected. - only supplied "/" tested
                                           Can be ignored for static pages or if no secrets in the page
 POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no SSLv3 support
 TLS_FALLBACK_SCSV (RFC 7507)              Downgrade attack prevention supported (OK)
 SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
 FREAK (CVE-2015-0204)                     not vulnerable (OK)
 DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this host and port (OK)
                                           no RSA certificate, thus certificate can't be used with SSLv2 elsewhere
 LOGJAM (CVE-2015-4000), experimental      not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
 BEAST (CVE-2011-3389)                     TLS1: ECDHE-ECDSA-AES128-SHA ECDHE-ECDSA-AES256-SHA
                                           VULNERABLE -- but also supports higher protocols  TLSv1.1 TLSv1.2 (likely mitigated)
 LUCKY13 (CVE-2013-0169), experimental     potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches
 Winshock (CVE-2014-6321), experimental    not vulnerable (OK)
 RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (OK)
```

6.8. att. **TLS konfigurācijas un šifru pārbaude (autora ekrānattēls)**

6.3.5. Lietotne GreenCheck

Identifikators: at.itsv.mobile.cochap

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.
2. Lietotne izmanto funkciju “malloc”.

Android statiskās analīzes augstas prioritātes konstatējumi:

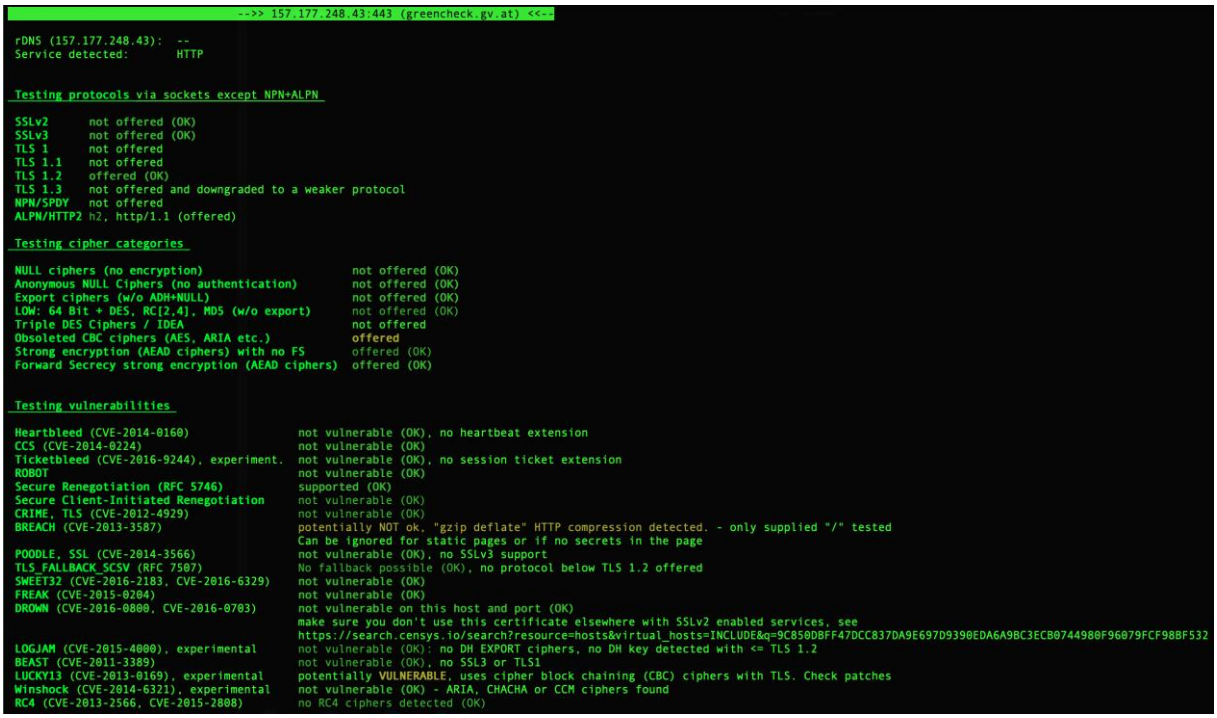
1. Darbība “at.itsv.mobile.cochap.MainActivity” nav aizsargāta no citu lietotņu piekļuves.
2. Lietotne izmanto nedrošu CBC kriptogrāfisko šifru.

Citi nozīmīgi konstatējumi:

1. Lietotne *GreenCheck* fona režīmā neslēpj informāciju (6.9. att.)

6.9. att. **Lietotne fona režīmā (autora ekrānattēls)**

2. Lietotne izmanto nedrošu CBC šifrus, kas ir pakļauti zināmām ievainojamībām (6.10. att.).



6.10. att. **TLS konfigurācijas un šifru pārbaude (autora ekrānattēls)**

6.3.6. Lietotne GreenPass Italia

Identifikators: com.italinnovation.green_pass (*Android*), com.italinnovation.greenPass (*iOS*)

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.

2. Lietotne izmanto nedrošas gadījuma skaitļu funkcijas.
3. Lietotne izmanto funkciju “malloc”.

Android statiskās analīzes augstas prioritātes konstatējumi:

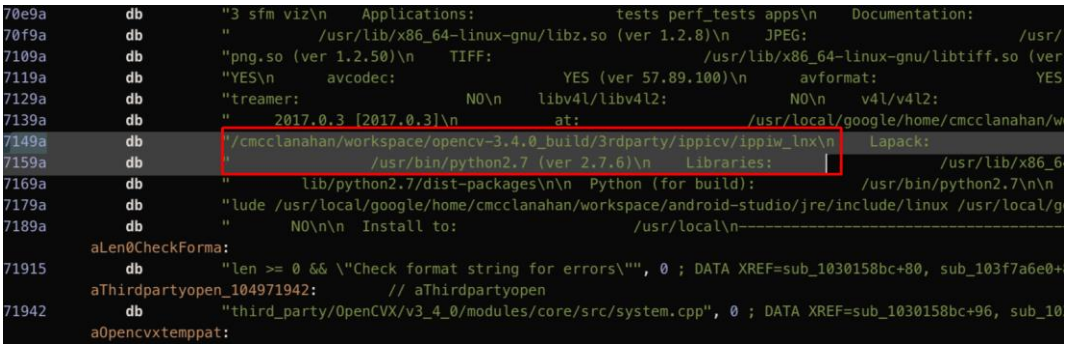
2. Lietotne var sazināties, izmantojot atvērtu tekstu.
3. Serviss “com.google.android.play.core.AssetPackExtractionService” nav aizsargāts no citu lietotņu piekļuves.

Citi nozīmīgi konstatējumi:

1. Lietotne fona režīmā neslēpj informāciju (6.11. att.)

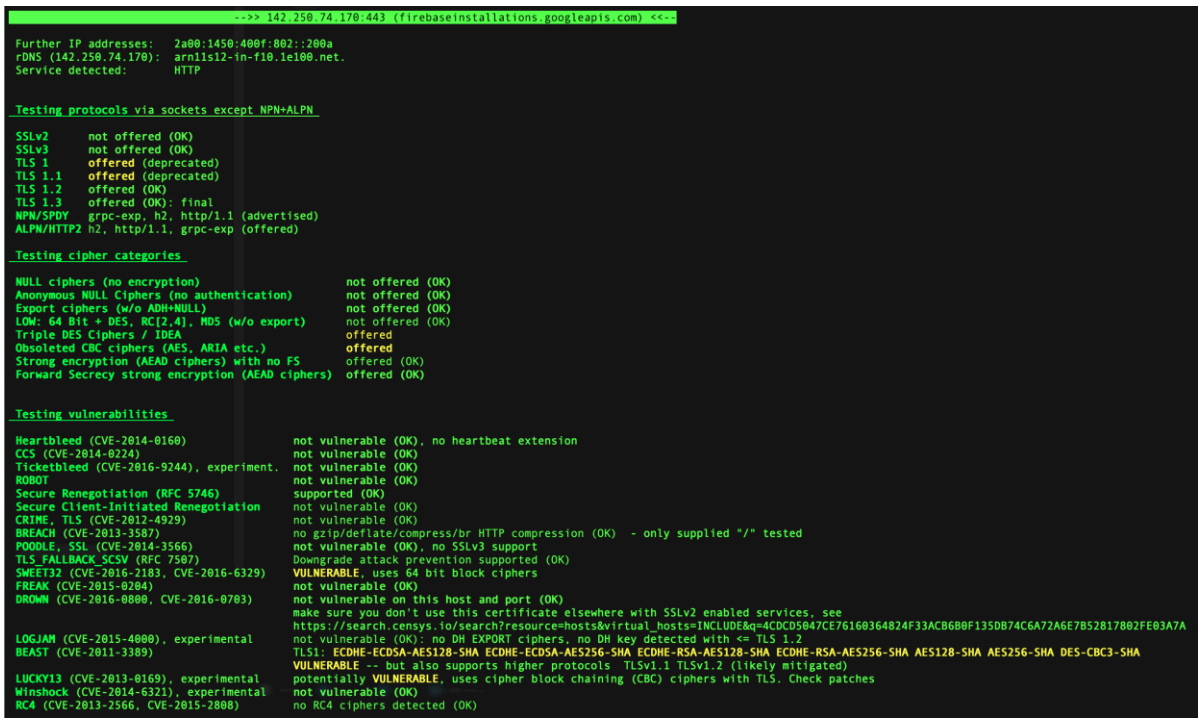
6.11. att. **Lietotne fona režīmā (autora ekrānattēls)**

2. *iOS* lietotne izmanto (6.12. att.) novecojušu un nedrošu *OpenCV* bibliotēkas versiju 3.4.0, kā arī neatbalstītu *Python* valodas versiju 2.7.6.



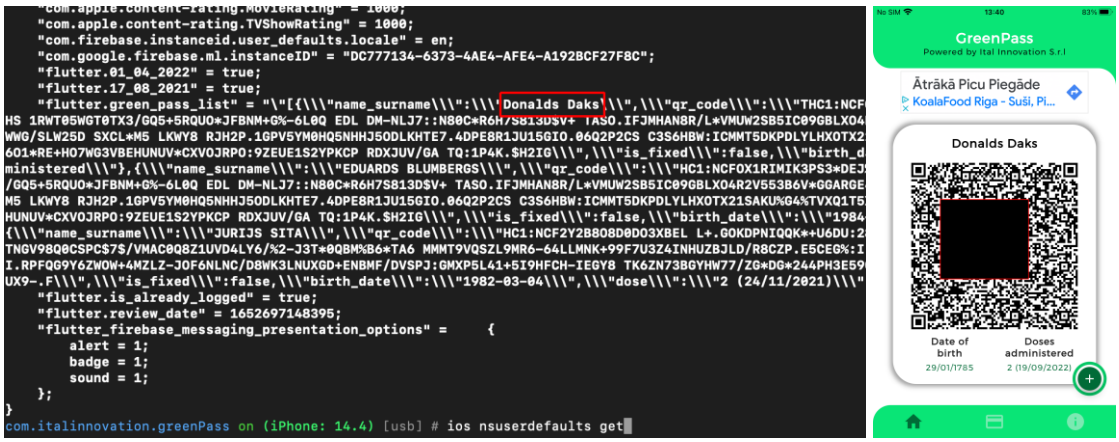
6.12. att. ***iOS* IPA lietotnes apskate reversās inženierijas programmā *Hopper Disassabler* (autora ekrānattēls)**

3. Lietotne sazinās ar galasistēmu https://firebaseinstalations.googleapis.com (6.13. att.), kura izmanto nedrošu konfigurāciju: novecojušus TLS protokolus 1.0 un 1.1, kā arī šifrus, kas ir pakļauti vairākām zināmām ievainojamībām: SWEET32, BEAST, LUCKY13.



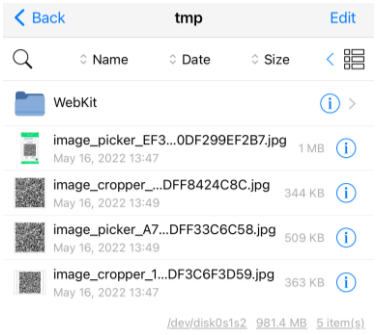
6.13. att. **TLS konfigurācijas un šifru pārbaude (autora ekrānattēls)**

4. *iOS* lietotne visu sertifikātu informāciju nešifrēti glabā saskarnē *NSuserDefaults*, parametrā "flutter.gree_pass_list", kurā ir iespējams izmanīt lietotāja personas datus (6.14. att.), jo sertifikāta informācija tiek nolasīta tikai pirmajā pievienošanas reizē, kā arī netiek validēta ar aizmugursistēmu. Atlauztā tālrunī šo parametru var izmainīt, izmantojot citu lietotni, piemēram, *Apps Manager* (https://moreinfo.thebigboss.org/moreinfo/depiction.php?file=appsmanagerDp).



6.14. att. **Lietotnes datu izmainīšana (autora ekrānattēls)**

5. iOS lietotne pagaidu failu mapē "tmp" (6.15. att.) saglabā visus augšupielādētos kvadrātkodu attēlus un ar kameru uzņemtās fotogrāfijas. Šie faili automātiski netiek dzēsti. Laika gaitā mapes izmērs kļūst ļoti liels. Ja programmu izmanto publiskai sertifikātu pārbaudei, ir lielas iespējas nozagt sensitīvu informāciju, ja tiek iegūtas pilnas tiesības ierīcē.



6.15. att. **Pagaidfailu mape (autora ekrānattēls)**

6.3.7. Lietotne Grüner Pass

Identifikators: at.gv.brz.wallet

iOS statiskās analīzes augstas prioritātes konstatējumi:

1. Lietotne izmanto nedrošus API.
2. Lietotne izmanto funkciju "malloc".

Android statiskās analīzes augstas prioritātes konstatējumi:

1. Serviss "com.google.android.play.core.AssetPackExtractionService" nav aizsargāts no citu lietotņu piekļuves.

Citi nozīmīgi konstatējumi:

1. Lietotnes tīmekļa datplūsmu var noklausīties un izmainīt (6.16. att.).

6.16. att. **Loģikas konfigurācijas pieprasījuma pārtveršana (autora ekrānattēls)**

2. iOS lietotnē failā Info.plist ir pieejams API marķieris (6.17. att.). Failā esošā informācija nav šifrēta un tas ir iekļauts lietotnes uzstādīšanas pakotnē. API atslēgas nav ieteicams izplatīt un uzglabāt klientā. API atslēgu vajadzētu glabāt attālā serverī un lejupielādēt pēc nepieciešamības.

```
    UISupportedInterfaceOrientations =     (
        UIInterfaceOrientationPortrait
    );
    UISupportsDocumentBrowser = 1;
    UIUserInterfaceStyle = Light;
    "WALLET_APP_SDK_API_TOKEN" = aknZGsFD9qCNmCm4NzFYfcK7WWbBeTFF;
}
at.gv.brz.wallet on (iPhone: 14.4) [usb] # ios plist cat Info.plist
```

6.17. att. **API marķiera apskate rīkkopā *objection* (autora ekrānattēls)**

SECINĀJUMI UN TURPMĀKAIS DARBS

Darba gaitā tika īstenoti galvenie izvirzītie daba uzdevumi un mērķi, kā arī izdarīti vairāki secinājumi.

1. Vakcinācijas sertifikātu kvadrātkodu apstrādei izmantotās lietotnes ir plaši izplatītas, tomēr publiski pieejams maz informācijas par to drošības novērtējumu.
2. Tīmeklī ir pieejama plaša informācija par mobilo lietotņu potenciālajām ievainojamībām, kā arī dažādas specifikācijas, kas palīdz potenciālo draudu modelēšanā.
3. Potenciālo draudu modelēšana ļauj vieglāk apkopot prasības vakcinācijas sertifikātu lietotņu drošības līmeņa aprobācijai un programmatūras ievainojamību konstatēšanai.
4. Pilnvērtīgas biznesa informācijas apkopošana par trešās puses COVID-19 lietotņu popularitāti ir sarežģīts un dārgs process, jo pieejamie rīki nav pieejami bez reģistrācijas un tie piedāvā ļoti nelielu bezmaksas funkcionalitāti.
5. Veicot tehnisko drošības pārbaudi izdevās atklāt vairākas ievainojamības, kas saistītas ar nepienācīgu datu saglabāšanu un pārsūtīšanu, kā arī nesavlaicīgu datu dzēšanu.
6. Vairāku lietotņu datu datplūsmu bija iespējams noklausīties un izmainīt.
7. Ņemot vērā, ka lietotnes varēja darbināt ar augstākā līmeņa pilnvarām un piekļūt pilnīgi visiem sensitīvajiem datiem, kas nebija saglabāti vai pārsūtīti šifrētā veidā, var secināt, ka vakcinācijas lietotnēm nepieciešamas augstākas prasības gan tiesību līmeņa kontrolei, gan datu šifrēšanai.
8. Lai mazinātu datu pārtveršanas uzbrukumus, izstrādātājiem vienmēr ieteicams izmantot sertifikātu piespraušanu, gan lai verificētu atzītus sertifikātus, gan ļautu droši izmantot pašparakstītus sertifikātus.
9. Dažas lietotnes joprojām izmanto neaizsargātus TLS savienojumus, ļaujot uzbrucējiem pārtvert datplūsmu, jo tās vai nu akceptē visus pašparakstītos sertifikātus, vai arī nepārbauda domēna vārdu.

Paveiktajā darbā ir apskatīta tikai neliela daļa lietotņu, koncentrējoties uz lietotnēm, kas tiek visvairāk izmantotas Latvijā, tomēr nepieciešami tālāki pētījumi, iesaistot un aktivizējot izstrādātājus, kurus šoreiz ieinteresēt neizdevās, lai varētu pilnvērtīgi pārbaudīt aizmugursistēmas, kā arī iegūt pirmkodus kvalitātes analīzei un caurskatīšanai.

Nepieciešams celt arī sabiedrības zināšanu līmeni par niansēm, kas ir atklājušās tehnisko drošības testu laikā, kā arī par veidiem, kas jaunajiem lietotājiem palīdzētu izvēlēties visdrošākās un kvalitatīvākās lietotnes no pieejamā klāsta. Pieaugot lietotņu skaitam, kas apstrādā sensitīvu informāciju, obligāti jāanalizē, kā lietotnes piekļūst un pārsūta informāciju.

IZMANTOTĀ LITERATŪRA UN AVOTI

[1] "2021 CWE Top 25 Most Dangerous Software Weaknesses", Online article, 2021., Pieejams: https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html

[2] "Cyber Kill Chain framework", Online information, Lockheed Martin Corp., Pieejams: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[3] "Common Weakness Enumeration Overview", Online information, Pieejams: http://cwe.mitre.org/about/index.html

[4] "eHealth Network Guidelines on Technical Specifications for EU Digital COVID Certificates", Volume 2, EU Digital COVID Certificate Gateway, Version 1.4, 2022., Pieejams: https://ec.europa.eu/health/system/files/2022-02/digital-covid-certificates_v2_en_0.pdf

[5] "ENISA Threat Landscape 2021", Online Publication, The European Union Agency for Cybersecurity, 2021., pp. 8., 56-61., Pieejams: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2021

[6] "EU's health pass aims to crack down on fake Covid certificates", Online news, 2021., Pieejams: https://www.connexionfrance.com/French-news/EU-s-health-pass-aims-to-crack-down-on-fake-Covid-certificates

[7] "First 3G mobiles launched in Japan", BBC News, 2001., Pieejams: http://news.bbc.co.uk/2/hi/business/1572372.stm

[8] "Framework – GitHub Topics", Online information, GitHub, Inc., 2021, Pieejams: https://github.com/topics/framework

[9] "Free PNG Download - Transparent PNG Images Free Download", Online clipart, Pieejams: https://www.subpng.com

[10] "ISO/IEC 27001 Information security management", Online information, Pieejams: https://www.iso.org/isoiec-27001-information-security.html

[11] "Italy Cracks Down on Schemes Selling Fake EU Digital COVID Certificates & Vaccines", Online news, 2021., Pieejams: https://www.schengenvisainfo.com/news/italy-cracks-down-on-schemes-selling-fake-eu-digital-covid-certificates-vaccines/

[12] "McAfee Mobile Threat Report 2021", Online Publication, McAfee, 2021., Pieejams: https://www.mcafee.com/content/dam/global/infographics/McAfeeMobileThreatReport2021.pdf

[13] "MITRE ATT&CK" Pieejams: https://attack.mitre.org

[14] "National Vulnerability Database", Online database, Pieejams: https://nvd.nist.gov

[15] "OWASP Mobile Top 10", Online information, 2016., Pieejams: https://owasp.org/www-project-mobile-top-10/

[16] “Priekšlikums EIROPAS PARLAMENTA UN PADOMES REGULA par sadarbspējīgu vakcinācijas, testēšanas un pārslimošanas sertifikātu izdošanas, verifikācijas un akceptēšanas satvaru nolūkā atvieglot brīvu pārvietošanos Covid-19 pandēmijas laikā (digitālais zaļais sertifikāts)”, Tiešsaistes publikācija, Eiropas Komisija, Briselē, 2021., Pieejams: https://eur-lex.europa.eu/legal-content/LV/TXT/?uri=CELEX:52021PC0130

[17] “Resursu klasifikācijas un risku analīzes piemērs virtuālajai iestādei”, CERT.LV, 2020., Pieejams: https://cert.lv/uploads/iestadem/resursu_klasifikacija_risku_parvaldiba.docx

[18] “State of Vulnerabilities 2018/2019 – Analysis of Events in the life of Vulnerabilities”, ENISA, Online publication, 2019., p. 9, Pieejams: https://www.enisa.europa.eu/publications/technical-reports-on-cybersecurity-situation-the-state-of-cyber-security-vulnerabilities

[19] “What Is MITRE ATT&CK and How Is It Useful?”, Online article, Anomali, Pieejams: https://www.anomali.com/resources/what-mitre-attck-is-and-how-it-is-useful

[20] Bodeau D. J., McCollum C. D, Fox D. B. “Cyber threat modeling: survey, assessment, and representative framework,” Technical report, The Homeland Security Systems Engineering and Development Institute, 2018., Pieejams: https://www.mitre.org/sites/default/files/publications/pr_18-1174-ngci-cyber-threat-modeling.pdf

[21] Grierson J. “Fake Covid vaccine and test certificate market is growing, researchers say”, Online news, 2021., Pieejams: https://www.theguardian.com/world/2021/may/16/fake-covid-vaccine-and-test-certificate-market-is-growing-researchers-say

[22] Hospelhorn S. “What is The Cyber Kill Chain and How to Use it Effectively”, Online blog, 2016., Pieejams: https://www.varonis.com/blog/cyber-kill-chain

[23] Kontogianni B. “Greek Doctor Fakes Vaccine Injections For Covid Deniers”, Online news, 2021., Pieejams: https://greekreporter.com/2021/07/08/doctor-athens-attempted-give-out-fake-vaccination-certificates/

[24] Mbunge E., Dzinamarira T., Fashoto S. G., Batani J. “Emerging technologies and COVID-19 digital vaccination certificates and passports,” Public Health in Practice, Volume 2, 2021., 100136, ISSN 2666-5352, Pieejams: https://doi.org/10.1016/j.puhip.2021.100136

[25] Mills E. “App Genome Project eyes iPhone, Android security”, CNET, 2010., Pieejams: https://www.cnet.com/tech/services-and-software/app-genome-project-eyes-iphone-android-security/

[26] Mobexler “Awesome Tools”, Online information, 2020., Pieejams: https://mobexler.com/tools.htm

[27] NIST Joint Task Force (Ross R. S. *Et al.*) “Guide for conducting risk assessments,” Special Publication (SP) 800-30 Rev. 1, The National Institute of Standards and Technology (NIST),

2012., B-13, Pieejams: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[28] NIST Joint Task Force "Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy", Online publication, 2018., Pieejams: https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

[29] Orlandi G., Joly J. "Fake COVID-19 certificate sales rise as health passes become mandatory", Online news, 2021., Pieejams: https://www.euronews.com/2021/08/04/fake-covid-19-certificate-sales-rise-as-health-passes-become-mandatory

[30] OWASP "Mobile Application Security Verification Standard", Online documentation, 2022., Pieejams: https://github.com/OWASP/owasp-masvs

[31] OWASP "Mobile Security Testing Guide", Online documentation, 2022., Pieejams: https://github.com/OWASP/owasp-mstg/

[32] OWASP MSTG "Testing Tools", OWASP, 2022., Pieejams: https://github.com/OWASP/owasp-mstg/blob/master/Document/0x08-Testing-Tools.md

[33] Phonlawat Khunphet P. "Insecure Deserialization", Online blog, 2019., Pieejams: https://medium.com/blog-blog/insecure-deserialization-e5398e83defe

[34] The Ada Lovelace Institute "International monitor: vaccine passports and COVID status apps", Online Publication, 2020., Pieejams: https://www.adalovelaceinstitute.org/project/international-monitor-vaccine-passports-covid-status-apps/

[35] World Health Organization "Digital documentation of COVID-19 certificates: vaccination status: technical specifications and implementation guidance", Online Publication, 2021., Pieejams: https://www.who.int/publications/i/item/WHO-2019-nCoV-Digital_certificates-vaccination-2021.1

[36] World Health Organization "Interim guidance for developing a Smart Vaccination Certificate", Online Publication, 2021., Pieejams: https://cdn.who.int/media/docs/default-source/documents/interim-guidance-svc_20210319_final.pdf

[37] World Health Organization "Revised scope and direction for the Smart Vaccination Certificate and WHO's role in the Global Health Trust Framework", Online news, 2021., Pieejams: https://www.who.int/news/item/04-06-2021-revised-scope-and-direction-for-the-smart-vaccination-certificate-and-who-s-role-in-the-global-health-trust-framework

Maģistra darbs "**Vakcinācijas sertifikātu mobilo lietotņu drošības novērtēšana**" izstrādāts LU Datorikas fakultātē.

Darba teksta galīgā versija izgatavota **23.05.2022.**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, un tajā tiek tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst

Signature not validated

Autors: Eduards Blumbergs
(Autora paraksts un datums)

Digitally signed by EDUARDS BLUMBERGS
Date: 2022.05.24 00:45:22 EEST

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **p i e m ē r o t u / n e p i e m ē r o t u** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: Dr. dat. Pēteris Paikens
(Vadītāja paraksts un datums)

Darbs iesniegts **maģistratūras sekretariātā** _____.
(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____.
(Metodiķes paraksts)

Recenzents: profesors Dr. dat. Uldis Straujums
(Akad.amats, zin.grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)