Asclepio VA & Audit
Mobisec 05/2020

Copyright Mobisec Italia s.r.l.

Author contacts
zannol@mobisec.com, poffo@mobisec.com, suppressa@mobisec.com, camuffo@mobisec.com, tech@mobisec.com

Distribution lists
Mobisec

Date	Author	Revisor	Version	Validity
22/05/2020	R. Poffo	A. Zannol	0.1, 0.2	RESERVED CONFIDENTIAL
24/05/2020	A. Zannol	A. Zannol	0.3	RESERVED CONFIDENTIAL
25/05/2020	D. Violo, A. Zannol	A. Zannol	0.4, 0.5	RESERVED CONFIDENTIAL
26/05/2020	R. Poffo, D. Violo, A. Suppressa, F. Camuffo, A. Zannol	A. Zannol	0.6	RESERVED CONFIDENTIAL
27/05/2020	R. Poffo, D. Violo, A. Suppressa, F. Camuffo, A. Zannol	A. Zannol	0.7	RESERVED CONFIDENTIAL
28/05/2020	R. Poffo, A. Suppressa, F. Camuffo, A. Zannol	A. Zannol	0.8	RESERVED CONFIDENTIAL
29/05/2020	R. Poffo, A. Suppressa, F. Camuffo, A. Zannol	A. Zannol	0.9	RESERVED CONFIDENTIAL
02/06/2020	A. Zannol	A. Zannol	1.0	FINAL

Sommario

1. Abbreviazioni

CA	Certificate Authority
CIA	Confidentiality – Integrity – Availability
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability scoring system
DB	Database
GUI	Graphical User Interface
HTTPS	Hyper Text Transfer Protocol Secure
IPC	Inter Process Communications
MITM	Man In The Middle
OS	Operative system
OSSTM	Open Source Security Testing Methodology
OWASP	Open Web Application Security Project
PT	Penetration Testing
RPI	Rotating (Rolling) Proximity Identifier
TEK	Temporary Exposure Key
VA	Vulnerability Assessment

2. Mobisec methodology

La procedura di analisi proprietaria Mobisec inizia con una prima fase di raccolta delle informazioni, tracciamento dei dati e architettura applicativa utilizzando dispositivi altamente personalizzati con le estensioni del kernel iOS e Android di **Mobisec Dynamic Mobile Security Analysis**.

Questi dispositivi sono in grado di raccogliere le informazioni delle applicazioni selezionate durante un loro utilizzo attivo, operando sugli strati più profondi del sistema operativo e recuperando i dati inviati dall'utente o scambiati tra le diverse fasi del suo utilizzo (*use cases*). I dati raccolti non sono dati contestuali del solo perimetro dell'applicazione (sandbox) ma le comunicazioni, qualsiasi trigger innescato dalle azioni utente nella GUI, vengono raccolte anche nell'environment in cui l'applicazione viene eseguita, tracciando quindi anche IPC, interazioni fra le app, accessi a librerie di file system, database e *data source* locali, file system, accessi a componenti hardware, etc.

Le sessioni di test Mobisec vengono elaborate più volte in modalità *fuzzy* in diversi momenti del giorno e con diversi parametri ambientali come il tipo di connessione di rete, la batteria

residua o le percentuali di CPU e RAM in modo da impostare lo stato del dispositivo in più combinazioni differenti.

Tutti i test Mobisec sono realizzati con un approccio *black box*, il che significa che nessuna informazione tecnica o documentazione è stata condivisa prima della valutazione della sicurezza. La raccolta di informazioni e il *reverse engineering* dell'applicazione che eseguiamo durante le fasi di analisi sono parte integrante dell'analisi stessa.

2.1 Strumenti e metodologie

Le metodologie alla base dell'analisi Mobisec sono le metodologie di mercato globalmente riconosciute come standard di *trustability* e *reliability*. Per i relativi compendi di:

- Compliancy
- Policies, standards and baselines
- Scoring and evaluation

Le definizioni e gli schemi sono ereditati da OWASP e OSSTM mentre domini e sottodomini sono acquisiti dai documenti standard di Web Security Analysis e trasposti in ambiente mobile.

Per ogni vulnerabilità riscontrata, Mobisec assegna un punteggio (da 0,1 a 10,0) basato sul sistema di punteggio CVSS 3.0, per ponderare correttamente l'*exploitability* sulle sue metriche di base, su metriche temporali e sul contesto ambientale in cui viene applicata l'analisi dell'app.

2.1.1 Information assurance objectives

La sicurezza si basa semplicemente sul controllare **chi** può interagire con le tue informazioni, **cosa** può fare con esse e **quando** ci può interagire. Queste caratteristiche vengono riassunte nella triade detta **CIA**.

CIA sta per *Confidentiality*, *Integrity*, *Availability* ed è comunemente raffigurata come un triangolo che rappresenta i legami forti tra i suoi tre principi.

- **Confidentiality**: si tratta della protezione dei dati da *disclosure* non autorizzate, o comunque di assicurare che solamente coloro che dispongono della corretta autorizzazione possono accedere ai dati.
- **Integrity**: si tratta della protezione dei dati dalle modifiche non autorizzate o di garantirne l'affidabilità. Il concetto contiene in sé le nozioni di *data integrity* (i.e. i dati non sono stati modificati accidentalmente o deliberatamente) e di *source integrity* (i.e. i dati sono stati modificati o ricevuti da una fonte legittima e autorizzata).
- **Availability**: si tratta di garantire la presenza di informazioni e risorse. Questo concetto non si basa solamente sulla mera protezione dei dati ma anche sulla protezione del servizio che fornisce l'accesso ai dati stessi.

In aggiunta, questa triade è spesso estesa ad altri due concetti, *Authentication* e *Authorization*, data la loro stretta connessione con i concetti della **CIA**. Meglio ancora, questa triade ha una dipendenza così forte dall'autenticazione e dall'autorizzazione che la riservatezza dei dati in questione non può essere garantita senza di loro.

- **Authentication**: la conferma dell'identità dell'entità che intende interagire con un sistema sicuro.
- **Authorization**: specificare i diritti di accesso per proteggere le risorse (dati, servizi, file, applicazioni). Questi diritti descrivono i privilegi o i livelli di accesso relativi alle risorse in questione. Normalmente è preceduto dall'autenticazione.

Auditing (non-repudiation): consiste nel tenere traccia degli eventi a livello di implementazione, nonché degli eventi a livello di dominio che si svolgono in un sistema. Può fornire non solo informazioni tecniche sul sistema in esecuzione, ma anche la prova che sono state eseguite azioni particolari. Le domande tipiche alle quali viene data una risposta sono "***Chi*** ha fatto ***Cosa***? ***Quando?*** E potenzialmente ***Come***?".

2.2 Security analysis

L'analisi di sicurezza si riferisce all'abilità di trasformare le informazioni in *security intelligence*. Ciò richiede comprensione non solo delle informazioni ma anche della loro provenienza, come e quando sono state raccolte e gli eventuali vincoli del processo di raccolta. La parte finale del processo di analisi consiste nel creare *actionable intelligence*, informazioni derivanti dal fatto che possono essere utilizzate per prendere decisioni. Questa è la chiara distinzione tra **sicurezza** e **analisi del rischio**. In *security analysis* vengono prodotti fatti concreti anche se gli stessi indicherebbero che qualcosa non può essere conosciuto dalle informazioni fornite. Nell'analisi del rischio invece si ipotizzano e si ricavano opinioni basate sulle informazioni.

L'analisi del rischio può utilizzare l'analisi della sicurezza per fornire risposte migliori e più accurate, tuttavia l'analisi della sicurezza non può utilizzare l'analisi del rischio per migliorare l'accuratezza.

2.2.1 Fasi della security analysis

Il metodo migliore di analisi di può essere suddiviso in quattro fasi, che devono essere eseguite nell'ordine corretto; unendo i dati di ciascuna fase in maniera iterativa con quelli delle fasi successive si realizza una profondità di analisi ottimale.

2.2.1.1 Induction phase

Durante la prima fase, l'analista inizia la sua indagine in base ai requisiti di *audit*, al loro *scope* e ai loro vincoli. L'elenco dei test viene spesso definito dopo questa fase.

Partendo da un primo approccio ad ampio spettro l'analista comincia poi a stabilire quale tipologia di test sia maggiormente funzionale per la specifica *run*.

2.2.1.2 Interaction phase

Durante questa fase, le operazioni dell'analista definiscono il perimetro dell'analisi e le relazioni tra le interazioni al suo interno. La fase di interazione definisce lo scopo principale dell'analisi.

Viene inoltre censita l'applicazione insieme ai suoi componenti andando così ad approfondire nel dettaglio, sempre attraverso un approccio *black box*, l'architettura dell'app

e il funzionamento delle sue funzionalità. A tal proposito vengono esplorati tutti gli aspetti e tutte le casistiche riscontrabili da un utilizzo definito *fuzzy* (i.e. da utente medio).

Qui l'analista comincia a adattare strategie, pattern e metodi mirandoli per la singola app in questione, in modo tale che l'analisi sia finalizzata per il caso specifico e andando così a fornire il miglior risultato possibile.

2.2.1.3 Inquest phase

Durante questa fase, i dati trovati vengono esposti e vengono riportati tutti i rischi relativi alla loro gestione. Ciò include la mancanza di protezione durante la trasmissione, l'utilizzo, il processo e la conservazione delle informazioni.

A tal proposito è compito dell'analista individuare le vulnerabilità reali e scartare i falsi positivi riscontrati in fase di test.

2.2.1.4 Intervention phase

Durante questa fase, i test si focalizzano sulle risorse che le applicazioni in oggetto richiedono durante il loro utilizzo e che utilizzano per il loro *scope*. Queste risorse vengono attaccate dall'analista che cerca di cambiare, rubare e/o iniettare informazioni nel tentativo di verificare la possibilità di una penetrazione o un'interruzione dell'applicazione al fine di creare un'*exploitability* praticabile empiricamente.

2.3 Ambiti e domini

I domini su cui si articola e struttura un'analisi di *actual security* sono:

- sensitive data;
- operations;
- network;
- system;
- untrusted input;
- broken cryptography;
- code compliance

La copertura di questi domini assicura la possibilità di analizzare in dettaglio:

- vulnerabilità di rete;
- archiviazione insicura di dati sensibili o mancata protezione degli stessi;
- mancata implementazione della crittografia nella trasmissione di dati e/o nella loro archiviazione in locale;
- gestione debole della sessione;
- accesso non autorizzato ad altri account utente;
- immissione di dati non attendibile o non autorizzata;
- vulnerabilità note e di pubblico dominio;

- errori che visualizzano informazioni sensibili (information disclosure);
- liste corrotte di controllo degli accessi e/o password deboli.

2.3.1 Oggetti di analisi

L'analisi di questi domini consente di definire e delineare gli ambiti di applicabilità delle verifiche di sicurezza. Il modello di analisi di Mobisec definisce cinque ambiti di indagine: *data flows*, *data stores*, *processes*, *interactors* e *trust system boundaries*.

- **data flows**: rappresentano i dati in comunicazione attraverso connessioni di rete, canali di trasmissione, slot mail, canali SMS, telefonate, etc.;
- **data stores**: rappresentano i file, database, properties, le risorse usate e come sono usate;
- **processes**: sono le computazioni o i programmi eseguiti dal kernel del sistema operativo o dall'utente;
- **interactors**: sono gli endpoint del sistema e possono essere interni come ad esempio le interazioni dell'utente con gli oggetti della UI, i sensori di geolocalizzazione, gli accelerometri, la rubrica contatti, il telefono, etc; oppure possono essere esterni come web services, data stream, etc. In generale sono *data providers* e *consumers* che non rientrano nello *scope* dell'applicazione da analizzare ma sono evidentemente relazionati ad essa e possono concorrere nel pregiudicarne la sicurezza;
- **trust boundaries**: sono forse i più soggettivi di tutti perché rappresentano il confine fra elementi *trusted* e *untrusted* nel sistema operativo mobile e il suo *environment* di *trusted execution*.

2.4 Mobisec components

Mobisec è una piattaforma che si struttura principalmente in quattro componenti:

- kext device agent (client);
- service/events handler and data collector (services);
- pattern matching engine (server – definitions, knwoledge, matching maps);
- reporting master (server).

2.4.1 Agent

È il componente client e si installa nel dispositivo in cui è residente l'app mobile. È un'estensione del kernel del sistema operativo che si occupa di registrare eventi nell'interfaccia e nelle componenti dell'applicazione oltre a raccogliere gli output dei trigger sollecitati. È sviluppato in tecnologia nativa a seconda del sistema operativo su cui deve essere installato, e nello specifico:

- **iOS agent**: C/Objective-C;
- **Android agent**: Java.

2.4.2 Services

Servizi di comunicazione tra **agent** e **server**. Fondamentalmente è un bus di comunicazione bidirezionale, *full-duplex*, che consente di raccogliere e inviare al server i dati raccolti dall'agent durante l'esecuzione delle routine impostate; invia inoltre, in modalità push dal server all'agent, le istruzioni e le direttive per modificare il runtime dell'agent in termini di *security hooks enable/disable requests*, *fuzzy test*, *scenario testcase*, modifica di frequenza, etc.

2.4.3 Matching engine e Analysis manager

È la logica di interpretazione del sistema che contiene, per ogni dominio di applicazione, tutte le casistiche di vulnerabilità note e i pattern relazionali che le combinano per ottenere il livello di rischio della coesistenza dei singoli fattori. È di fatto un database dinamico e relazionale che viene esteso sulla base dei risultati di ogni *security analysis*. Combina euristicamente i dati raccolti dall'agent e decide se una casistica, che presa singolarmente può non rappresentare una minaccia, diventa rilevante nel momento in cui è combinata ad altre condizioni del sistema.

Le analisi vengono affrontate su tre *pillar* di ricerca:

- **pattern**: possono essere pattern generali, specifici per tipologia di app (home banking, gaming, insurance, service, etc) o pattern specifici del dominio applicativo dell'app analizzata (username e password, metodi, classi, regular expression, array, indici, etc);
- **CVE di mercato**: sia hardware che software rilasciate da Apple, Google, NSA, etc. Sono le CVE note di cui è stata data disclosure al mercato e che sono quindi di pubblico dominio;
- **strategie**: items proprietari del sistema Mobisec, sono pattern complessi di casistiche che prese singolarmente non rappresentano vulnerabilità, ma che, combinate fra loro, possono costituire un fattore di rischio. Le strategie sono una componente fondamentale, dinamica e in continua evoluzione del sistema di *knowledge base* della piattaforma e crescono all'aumentare delle analisi effettuate;

2.4.4 Report Manager

Contiene la grammatica di rappresentazione dei dati raccolti dall'**agent**, trasmessi dal **service bus** e combinati dal **pattern matching engine**. È un motore di query JSON-style che estrae i record e li compone con un connettore Jit per generare report dinamici secondo le grammatiche e i template di presentazione impostati, siano essi sistemi terzi, *web documents*, database o *API based services*.

Figura 1- Un esempio di evidenza presente nel report di analisi Mobisec.

3.Asclepio Analysis

3.1. Premesse

Il giorno 21 maggio 2020 Mobisec si occupa di un vulnerability assessment sull'app Immuni, sviluppata da Bending Spoons S.p.A.

Mobisec opera in regime black box, un approccio per i VA per cui non vengono condivise documentazione tecnica o progettuale internamente, ma solamente le distribuzioni dell'app da testare, lasciando ai tester l'intera fase di recupero di informazioni e studio di architettura del progetto in modo da valutare la quantità e qualità delle informazioni che un attaccante esterno all'azienda sarebbe in grado di ottenere autonomamente. In questa specifica casistica c'è da ricordare che l'applicazione è stata sviluppata come progetto open source, per cui codice e documentazione sono disponibili pubblicamente alle pagine di progetto del profilo GitHub creato appositamente per l'app.

Lo scopo del VA e dell'audit di sicurezza è di verificare e certificare, ad opera di un ente terzo (Mobisec) la sicurezza dell'applicazione, la gestione dei dati sia in fase di gathering che di distribuzione, comunicazione agli end point server, sia dell'eventuale storicizzazione locale degli stessi e la solidità dei criteri di actual security nell'utilizzo dell'app.

3.1.1. App Overview

3.1.1.1. Introduzione

L'applicazione oggetto dell'analisi è l'app Immuni, sviluppata da Bending Spoons S.p.A.

Questa applicazione è stata sviluppata per smartphone iOS e Android, in modo da affrontare e tentare di contenere l'epidemia COVID-19, permettendo agli utenti che la utilizzano di sapere se siano o meno a rischio di aver contratto il virus, così che possano isolarsi ed evitare di infettare gli altri, riducendo al minimo la diffusione del virus ed accelerando, nello stesso momento, il ritorno alla vita normale per la maggior parte delle persone.

Per funzionare, Immuni utilizza un sistema di notifica che usa il cosiddetto "Bluetooth Low Energy", in maniera da salvare il codice di prossimità a rotazione di chiunque sia stato vicino all'utente (anche un estraneo, sempre che sia provvisto dell'applicazione) per un tempo sufficiente, nella memoria locale. I codici di prossimità a rotazione sono generati da chiavi di esposizione temporanee e cambiano più volte ogni ora. Le chiavi di esposizione temporanee vengono generate in modo casuale e cambiano una volta al giorno.

Ogni qualvolta un utente risulti positivo alla COVID-19, può decidere se caricare su un server le sue più recenti chiavi di esposizione temporanea, sempre dopo la convalida di un operatore sanitario. L'app scarica dal server le chiavi periodicamente, in modo da metterle a confronto con quelle degli altri utenti vicini, e avvisare se si sia verificata un'esposizione rischiosa. Non viene utilizzata la geolocalizzazione, per cui l'applicazione non può sapere dove sia avvenuto il rischio o da chi, visto il fatto che i dati personali dell'utente, come il nome, l'età, l'indirizzo, l'e-mail o il numero di telefono, non vengono inseriti. Quindi, l'app sa semplicemente che ha avuto luogo il contatto con un utente infetto, quanto è durato e può stimare la distanza che separava i due utenti.

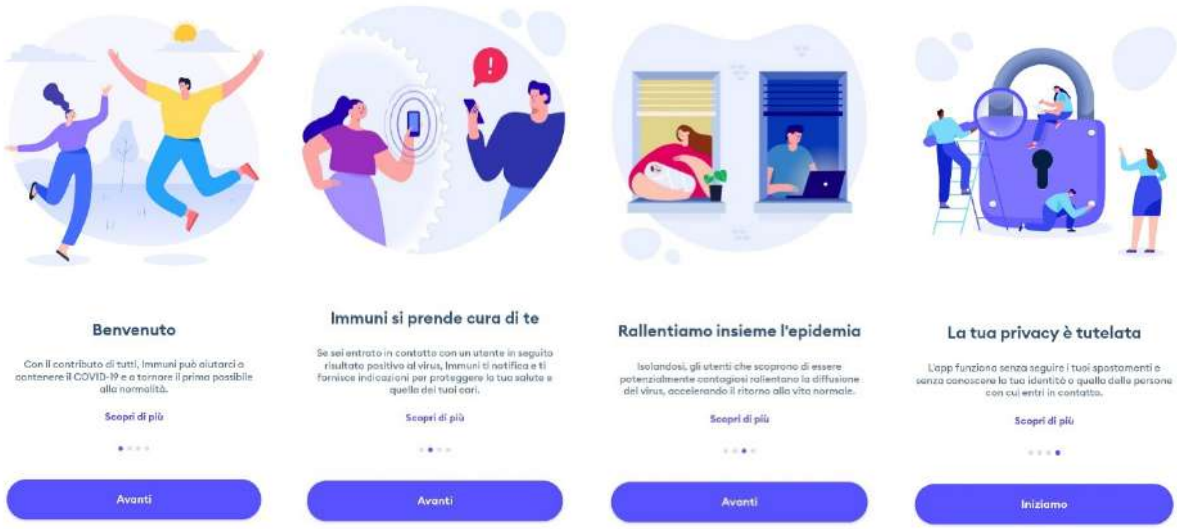
Se l'applicazione di un ipotetico utente rilevasse un rischio, gli vengono subito forniti consigli dall'applicazione stessa, che possono essere l'auto-isolamento (che aiuta a ridurre al minimo la diffusione della malattia) o contattare il proprio medico di medicina generale (in modo che l'utente possa ricevere le cure più appropriate e ridurre la probabilità di sviluppare la malattia).

Tutti i dati, archiviati sul dispositivo o sul server, verranno eliminati quando non saranno più necessari e comunque entro il 31 dicembre 2020.

3.1.1.2 Funzionamento

3.1.1.2.1 Spiegazione iniziale

All'apertura dell'applicazione viene mostrata all'utente la spiegazione di come funziona la piattaforma, suddivisa in quattro semplici schermate.



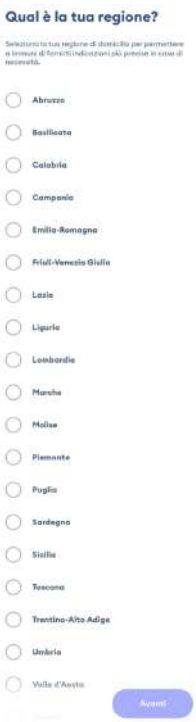
Cliccando sulla scritta "Scopri di più", l'utente può ottenere maggiori informazioni sull'applicazione Immuni, in modo da comprendere meglio la sua funzionalità, ed esplorare le FAQ (domande frequenti), così da eliminare ogni dubbio.

3.1.1.2.2 Informativa sulla Privacy

Selezionando il pulsante “Iniziamo”, appare una schermata denominata “La tua privacy è al sicuro”, in cui viene spiegato come vengono trattati i dati ed in cui all’utente viene assicurato che la sua privacy è tutelata. Dopo aver letto queste informazioni, egli deve dichiarare di avere almeno 14 anni e di aver letto e accettato l’informativa sulla privacy. Proseguendo, inoltre, l’utente dichiara automaticamente di aver letto ed accettato i termini di utilizzo, visualizzabili cliccando sopra alla relativa scritta.

3.1.1.2.3 Luogo di domicilio

Dopo aver cliccato sul pulsante “Avanti”, all’utente viene richiesto di selezionare la regione di domicilio, in modo che Immuni possa fornirgli informazioni in base al regolamento vigente in quella selezionata.



Immediatamente dopo la selezione della regione di appartenenza, viene richiesto quale sia la provincia dell'utente, così da seguire ulteriormente il regolamento di provincia, basandosi appunto anche su questo.

Si presume che le indicazioni di regione e provincia siano funzionali alle disposizioni di controllo sanitario locali (ASL) ed alle operazioni di certificazione dell'operatore sanitario, nel momento in cui l'utente scegliesse volontariamente di segnalare la propria positività.

3.1.1.2.4 Attenzioni particolari

Infine, selezionando di nuovo il pulsante "Avanti", appare una schermata che comunica all'utente di impostare un codice di protezione al suo smartphone, in modo da tutelare i

propri dati e le proprie applicazioni, inclusa Immuni. Non è obbligatorio, è solo un'attenzione in più consigliata all'utente.

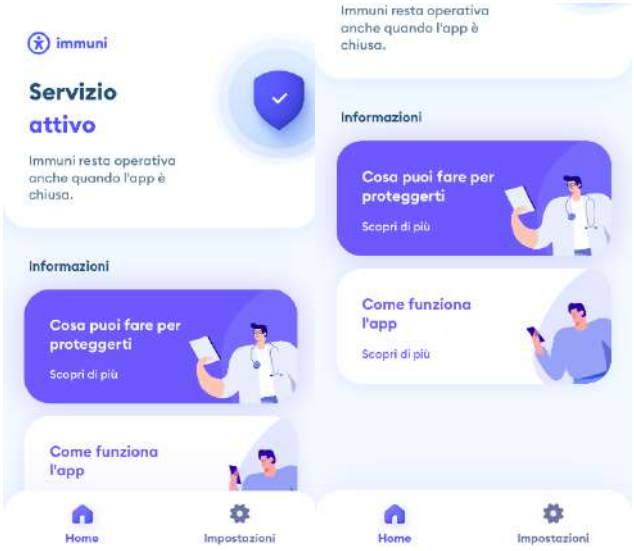
Infine, dopo aver cliccato sul pulsante "Ho capito", appare una schermata con un ulteriore avviso, che informa l'utente di non fidarsi di nessun SMS, telefonata, e-mail o qualsiasi altro tipo di comunicazione che sembri provenire da Immuni, a causa del fatto che l'applicazione comunica con l'utente solo attraverso la piattaforma stessa e le relative notifiche.

3.1.1.2.5 Schermata Home

Cliccando di nuovo sul pulsante "Ho capito", l'utente si ritrova nella schermata "Home". In basso, l'applicazione viene divisa in due sezioni, ovvero "Home" e "Impostazioni".

Qui, l'utente noterà la scritta "servizio attivo", e sotto viene spiegato che l'app rimane operativa anche quando viene chiusa.

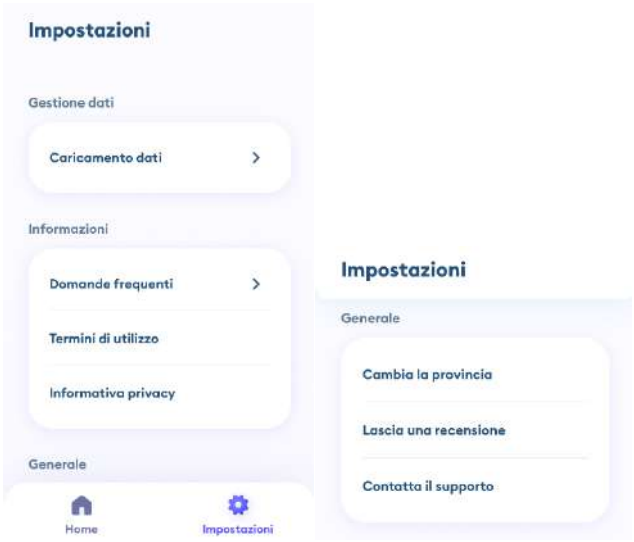
Sotto, si trova una sezione chiamata “Informazioni”, in cui vengono mostrati due pulsanti: uno denominato “Cosa puoi fare per proteggerti” e l’altro “Come funziona l’app”.



Selezionando il primo pulsante, all’utente appaiono alcuni consigli che gli permettono di sentirsi maggiormente al sicuro e comprendere quali misure deve adottare per tutelare sé stesso e gli altri dal COVID-19.

Selezionando il secondo pulsante, all’utente appaiono le stesse informazioni che sono state visualizzate nella spiegazione iniziale, cliccando sulla scritta “Scopri di più”. (vd. punto 3.1.1.2.1)

3.1.1.2.6 Impostazioni



Andando alla sezione “impostazioni”, appaiono tre voci, ovvero “Gestione dati”, “Informazioni” e “Generale”.

Nella prima, è inserita una sola sottosezione, chiamata “Caricamento dati”.

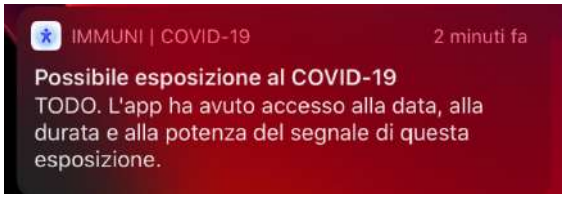
Cliccando su questa voce, l’utente può comunicare il proprio codice all’operatore sanitario nel caso in cui sia infetto. Questa operazione può essere effettuata solo previa autorizzazione di un operatore.

All’interno della seconda voce, sono inserite tre sottosezioni, ovvero “Domande frequenti”, “Termini di utilizzo”, “Informativa privacy”.

Nella sezione “Generale”, l’utente trova tre sottosezioni, da cui può cambiare la propria provincia, lasciare una recensione o contattare il supporto.

3.1.1.2.7 Device “infetto”

Nel caso in cui l’utente si trovi in una posizione rischiosa e abbia avuto contatti con un utente Immuni infetto, l’applicazione invia un messaggio di notifica sul dispositivo mobile della persona a rischio.



Inoltre, appare all’interno dell’applicazione, nella sezione “Home”, un avviso di colore rosso in alto, con su scritto “Rilevato contatto con una persona positiva al COVID-19”.

Cliccando sull’avviso, all’utente vengono date diverse indicazioni da seguire. La prima fra tutti, sarà “Contatta immediatamente il Dipartimento di Prevenzione della tua ASL, se non lo hai già fatto”.

In basso a questo, vengono elencate diverse misure ed accorgimenti da seguire in attesa della comunicazione all’ASL da parte dell’utente, ad esempio rispettare le misure di sicurezza, lavarsi frequentemente le mani, fino all’auto-isolamento.

L’utente può decidere di nascondere l’avviso, cliccando sul relativo pulsante. Ovviamente, gli viene suggerito di farlo solo se si ha una ragione valida (nei casi in cui sia un operatore sanitario a contatto con casi di COVID-19 o in cui l’utente sia già in contatto con la sua ASL, ecc…), altrimenti è vivamente sconsigliato.

Nel caso in cui l’utente si sia già messo in contatto con la sua ASL, può cliccare sulla scritta “Sono in contatto con la ASL”.

3.2. Analysis scope

3.2.1. Perimetro e oggetti di analisi

L'analisi viene condotta su entrambe le distribuzioni iOS ed Android dell'app, nelle versioni di 1.0 iOS e 0.0.1 Android dedicate agli ambienti di test e QA. L'identificazione delle app può essere riconducibile ai seguenti pacchetti:

- it.ministerodellasalute.immuni – Android v. Name 0.0.1 (Code version 35);
- it.ministerodellasalute.immuni – iOS 1.0 (Build 999).

Per prendere visione della lista generale degli oggetti di analisi Mobisec v. 2.3.1.

L'audit è riservato alla componente client dell'applicazione e non agli end point server, tuttavia i test riguardano anche le interazioni fra client e server, per cui sono oggetto dell'analisi anche le modalità di comunicazione e scambio dei dati, la sicurezza e riservatezza degli handshake e delle trasmissioni, nonché la conformità delle risposte lato server e gli eventuali information disclosure o le possibilità di injection sui data source o breach sui sistemi.

Gli oggetti principali dell'audit sono i dati sanitari (o potenziali) dell'utente (lo status di positivo o meno), le logiche di individuazione (che devono avvenire nella totale anonimità e non tracciabilità anagrafica del soggetto), l'ingegnerizzazione della meccanica di marking del dispositivo, le comunicazioni fra i dispositivi e lo scambio informativo fra i device e con gli endpoint server.

Nella convenzione dell'audit sono inoltre definite "Comunicazioni" tutte le chiamate eseguite dall'applicazione che sia diretta verso funzioni e metodi esterni, come ad esempio IPC, lettura e scrittura di dati sul filesystem, accesso a DB locali, ecc.

Tutti questi oggetti di indagine devono, per la natura stessa dell'applicazione, preservare completamente l'anonimità del dato, la non tracciabilità geografica del device e dell'utente, la preservazione dell'integrità del dato stesso e la non possibilità di manomissione, nonché i criteri di availability e di persistenza del servizio.

L'analisi viene eseguita a livello di kernel dell'OS (sia iOS che Android) e non solo in sandbox dell'app; il perimetro quindi non riguarda solo cosa avviene all'interno della sandbox dell'app Immuni e del suo relativo spazio di memoria a runtime, ma anche di tutte le eventuali

interazioni dell'app con il sistema operativo, le periferiche hardware e le possibili interazioni con qualsiasi altra app (legittima o malevola) installata nei dispositivi.

Durante l'esecuzione dei test vengono tracciati tutti i comportamenti dell'app e i risultanti trigger scatenati nelle funzioni dell'OS.

3.3. Audit

3.3.1. Preparazione all'audit

3.3.1.1. Strumenti

L'audit è stato condotto con strumenti classici di VA e PT e con la piattaforma proprietaria Mobisec MSDAPaaS.

Di seguito la lista degli strumenti utilizzati:

Tool ID	Vendor	Name	Version	Type	URL
MsT01	Mobisec	Mobisec Static and Dynamic Security Analysis Platform	3.5.5	Analysis Platform	www.mobisec.com
MsT02	Offensive Security	Kali	2020-04-03	OS	www.kali.org
MsT03	Apple	macOS	Many	OS	www.apple.com
MsT04	Microsoft	Windows	Many	OS	www.microsoft.com/it-it/windows
MsT05	Apple	iOS	Many	OS	www.apple.com/ios
MsT06	Google	Android	Many	OS	www.android.com
MsT07	Jay Freeman (saurik)	Cydia Impactor	1.1.33	iPhone APT frontend	www.saurik.com/id/1
MsT08	Unc0ver team	Unc0ver	5.0.1	Jailbreak tool	unc0ver.dev
MsT09	topjohnwu	Magisk	20.4	Root access provider	github.com/topjohnwu/Magisk
MsT10	rovo89	Xposed Framework	90-beta3	Modules framework	repo.xposed.info
MsT11	OpenSecurity	MobSF	3.0.7	Security assessment framework	mobsf.github.io/Mobile-Security-Framework-MobSF
MsT12	oleavr	Frida server	12.9.4	Dynamic code instrumentation toolkit	frida.re
MsT13	MitmProxy	mitmproxy	5.1.1	Proxy	mitmproxy.org
MsT14	OWASP	ZAP Proxy	2.9.0	Proxy	www.zaproxy.org/
MsT15	The Wireshark team	Wireshark	3.2.2	Network protocol analyzer	www.wireshark.org

Tabella 1 - Test tools

3.3.1.2. Configurazioni

L'audit è stato eseguito su diverse configurazioni OS/HW per garantire la più completa copertura del model office compatibile con l'utilizzo dell'app Immuni.

Di seguito la lista dei dispositivi utilizzati:

Device ID	Vendor	Model	OS version
MsD01	Apple	iPhone XR	13.5
MsD02	Apple	iPhone 6S	13.5
MsD03	Apple	iPhone 6S	13.2.2
MsD04	Apple	iPhone X	13.5
MsD05	Xiaomi	Mi A2	9.0
MsD06	Samsung	Galaxy Note 8	8.0.0
MsD07	Samsung	Galaxy J5 (2016)	7.0
MsD08	Samsung	Galaxy S9	9.0

Tabella 2 - Test Devices model office

I device su cui è stata eseguita l'analisi sono dispositivi a cui sono stati concessi i permessi di super-admin (altresì noti come permessi di *root*), garantendo così un accesso a possibilità di esecuzione di codici e operazioni normalmente difficili da ottenere per un utente medio, che rendono il device *compromesso* in termini di integrità; i device Android assumono la condizione definita *rooted*, mentre quelli iOS si definiscono *jailbroken* a causa della nomenclatura del metodo di ottenimento di tali permessi che viene chiamato *jailbreak*.

I dispositivi compromessi sono quindi predisposti con delle estensioni del kernel e dei metodi e funzioni standard del sistema operativo, definiti *hook*, al fine di intercettare le Comunicazioni e i loro contenuti che vengono effettuate durante il normale utilizzo dell'applicazione.

Si è provveduto quindi a configurare i device con il client Mobisec (una kernel extension nativa installata sui client), e si è installato il package di riferimento per l'analisi, indicandolo al client Mobisec come oggetto dei test ed iniziando così a registrare e controllare tutte le migliaia di operazioni che venivano eseguite dall'app durante la sua esecuzione, sia in foreground che in background.

3.3.2. Esecuzione dell'audit

3.3.2.1. Test chain e test case

Per ogni dispositivo si è seguita questa catena di test per verificare l'integrità dell'app e dei dati trattati anche in funzione della variazione dell'hardware e della versione dell'OS, nel caso di Android anche passibile di pesanti modifiche e personalizzazioni rispetto alla versione pubblica (nota come *Android One*) rilasciata in modalità open source da Google.

Ogni punto della lista della test chain è stato effettuato su ogni singolo model office definito in Tabella 2 al capitolo 3.3.1.2 e si sono verificate le condizioni di messa in sicurezza di quel determinato rischio indicando con OK/KO l'effettiva esposizione agli exploit che possono interessarlo ed indicando in una colonna a lato eventuali dettagli necessari a fornire in maniera più puntuale eventuali informazioni emerse durante i test.

Di seguito presentata la *test chain* Mobisec seguita nell'esecuzione dell'analisi suddivisa per i domini di ricerca Mobisec.

ID	DESCRIZIONE TEST
	BROKEN CRYPTOGRAPHY

BC01	Presence of coded and unencrypted data (e.g. Base64)
BC02	Presence of weak or insecure encryption methods
BC03	Presence of enforceable encryption methods
BC04	Presence of keys not properly protected
BC05	Presence of salt not properly protected
BC06	App signed with weak certificate (MD5/SHA-1, Android only)
	UNTRUSTED INPUT
UI01	Presence of clipboard functions on sensitive fields
UI02	Screenshots in clear once the app goes background
UI03	Ability to take screenshots of the application on windows containing sensitive data
UI04	Numeric native keyboard for PIN input/access codes
UI05	Bypass methods of root/jailbreak detection
	OPERATIONS
OP01	Presence of system logs
OP02	Variables containing sensitive data in memory (RAM dump)
OP03	Opening web services on browsers instead of WebView
OP04	WebView free navigation enabled (lack of domain whitelist)
	FILESYSTEM
FS01	Weak save of "Remember me" options
FS02	Preferences file (.xml/.plist) in plaintext
FS03	Readable log files
FS04	Database in plaintext
FS05	Cache in plaintext
FS06	Presence of non-production/environment configuration files
FS07	Unprotected customer media files
FS08	Use of common memory areas for saving information
FS09	Temporary files not discarded after use
FS10	Use of temporary files for containing sensitive/personal data
FS11	Weak consistency of data in the filesystem
FS12	Sensitive data saved in cookies
FS13	Incorrect deletion of data after logging out (e.g. profile files)
FS14	Download of files containing sensitive data in the common area of the filesystem
	NETWORK / CLIENT
NC01	Lack of end-to-end encryption of communications
NC02	Lack of certificate pinning
NC03	Acceptance of a proxy without certificate
NC04	Acceptance of a proxy with self-signed certificate
NC05	Acceptance of a proxy with trusted CA certificate

NC06	Possibility of repetition of the call (lack of CSRF)
NC07	Possibility of forging calls (lack of one-time CSRF token)
	NETWORK / SERVER
NS01	Incorrect deletion of data after logging out (e.g. tokens)
NS02	Anti-weakness password policy is missing
NS03	Use of insecure HTTP connections
NS04	Incorrect use of insecure GET HTTP method
NS05	Unused HTTP methods allowed
NS06	Failure on server certificate chaintrust
NS07	Information disclosure of the server in correct calls (e.g. headers)
NS08	Information disclosure of the server in uncorrect calls (e.g. stack traces)
NS09	Non-presence of request control methods (open REST API)
NS10	Failure to check consistency of the token-user binding
NS11	Failure to check consistency of the token-operation binding
NS12	Lack of transaction signature
NS13	Information disclosure in error pages (40X, 50X)
NS14	Disclosure of sensitive data in case of errors in operations
NS15	Lack of brute-force attack checks at login
	INJECTION
IN01	Possibility of REST API injection
IN02	Possibility of HTML/JavaScript injection
IN03	Possibility of WebView phishing (e.g. DNS hijacking, URL change, ...)
IN04	Possibility of injection on local SQLite databases
IN05	Possibility of injection on local configuration/settings files
	STATIC ANALYSIS
SA01	Banned, weak or deprecated APIs / methods
SA02	Weak management of flags for allocation of memory areas
SA03	Request of unused permits
SA04	Request of non-optimized permits
SA05	Lack of obfuscation of the application code
SA06	Data backup enabled without adequate protection of the information saved by the app
SA07	Disabled ATS options (Apple only)

Tabella 3 - Test chain & Test Cases Mobisec, versione 1.3

Oltre alla normale *test chain*, vengono effettuati laddove necessario dei controlli personalizzati sulle singole app che variano a seconda delle funzioni che è in grado di eseguire. Nello specifico per l'app Immuni sono stati aggiunti alla test chain i seguenti controlli specifici:

- broadcast receivers optimization (Android);
- Bluetooth injection;

- Bluetooth sniffing.

Nel capitolo successivo verranno illustrate nel dettaglio le attività effettuate per l'esecuzione dei test.

3.3.2.2. Test activities

La metodologia di seguito descritta è stata svolta in maniera paritaria sia sui device Android che su quelli iOS; laddove ci sia una differenza sostanziale tra le operazioni di test con discriminante il sistema operativo, sarà esplicitamente descritto il differenziale delle due situazioni.

Sul dispositivo debitamente configurato (jailbroken e rooted con kernel extension e Mobisec client onboard) è stato installato il pacchetto dell'applicazione Immuni, reso poi target dei test nell'interfaccia del client Mobisec precedentemente caricato.

Nella configurazione sono stati abilitati tutti gli *hook* in modo da garantire una copertura completa di tutte le chiamate eseguite dall'applicazione verso classi e metodi standard dell'OS, quindi il device è stato riavviato in modo da ottenere anche le chiamate svolte in fase di avvio del sistema, se presenti.

L'app è stata lanciata manualmente, iniziando così una sessione standard al pari di quella che eseguirebbe un qualsiasi utente.

La prima parte di test (information gathering) richiede di carpire il funzionamento dell'app nella maniera il più possibilmente dettagliata in modo da capire quali possono essere i punti deboli e di attacco su cui un hacker può agire e su cui gli stessi test Mobisec si possono focalizzare alla ricerca di un exploit, sia esso empirico o teorizzato sulla base delle potenzialità di malware e CWE di mercato note.

Una volta che l'app è stata eseguita, si è esplorata nelle sue sezioni (vedi 3.1.1), andando a studiarne il funzionamento indicato nella presentazione introduttiva in modo sia da sapere cosa aspettarsi durante l'esecuzione dell'app, che per poterlo confrontare con l'effettivo funzionamento tracciato dal client Mobisec, riuscendo così a cercare un riscontro tra quanto dichiarato e quanto realmente avviene nel device.

Vista la presenza di un certificate pinning, ovvero un metodo di protezione delle comunicazioni di rete client-server che permette i dati vengano trasmessi solo se non ci sono altri apparati di rete nel mezzo (come ad esempio i proxy, i quali permettono di eseguire una serie di cyber-attacchi come MITM, i quali possono prevedere sniffing, injection, DoS, ecc.), Mobisec ha eseguito grazie al suo client una pratica definita "*unpinning*", eseguibile solo su device compromessi grazie ai permessi di root, che consiste nel far credere al dispositivo che stia comunicando con il server anche se viene posto un proxy nel mezzo, permettendo così agli operatori di avere facile accesso a tutte le comunicazioni di rete che avvengono con l'app e che un normale utente non riuscirebbe altrimenti in alcun modo a ottenere.

Di norma, come indicato al Capitolo 2, la metodologia Mobisec prevede la registrazione dei casi d'uso dell'app e la loro riesecuzione in automatico in modalità *fuzzy*. Tuttavia, essendo quella in oggetto un'applicazione la cui principale funzione è quella di comunicare con altri device nelle vicinanze che fanno uso della stessa app, l'analisi è stata condotta con l'ausilio di più operatori in contemporanea con situazioni applicative, posizionamento fisico (vista la

variabilità dei dati epidemiologici in funzione della distanza e durata dell'esposizione a soggetti poi indicati come infetti) ed istanze diverse.

I dispositivi sono così stati esposti ad un device che ad inizio sessione di test è stato marchiato come quello che nella successiva simulazione sarebbe stato quello in possesso di un utente infetto dal virus, sfruttando cicli di durata e ripetizione diversi per assicurarsi che la variabilità dei dati fornisse risultati pesati e diversi e che la qualità e la quantità di dati non creasse overflow di informazioni che potessero risultare in problemi di gestione sicura dei dati.

In questi periodi pre-comunicazione dell'esposizione sono state monitorate le comunicazioni di rete, la lettura e scrittura dei dati nel filesystem e in memoria e le funzioni di sistema sfruttate dall'app, ponendo particolare attenzione ai momenti di conferma e modifica delle informazioni relative alla regione e provincia di domicilio, di fatto gli unici dati dell'utente.

Nei test è emerso come la maggior parte delle operazioni dell'app vengono svolte direttamente con una serie di comunicazioni con i framework Apple e Google (nelle relative distribuzioni) che opera in background nel sistema operativo e che si occupa di generare, calcolare e gestire i codici anonimi usati dal sistema di controllo dell'esposizione. L'app funge principalmente da punto di collegamento e abilitazione alla condivisione dei dati tra il framework, l'utente e l'end point server nazionale che poi si occupa di permettere il caricamento dei codici degli utenti che vengono certificati come positivi al virus.

Questa comunicazione tra l'app e il server centrale necessaria alla comunicazione dei token degli utenti positivi è stata quindi oggetto di due diversi tipi di test:

- un primo test di sicurezza procedurale, in cui si è verificato in che occasione un utente potesse venire abilitato all'upload dei suoi token, ovvero solamente un operatore sanitario che identifica una persona come positiva al virus a seguito di un tampone può procedere a convalidare il codice OTP necessario all'upload;
- un test classico dei VA che prevede dei tentativi di SQL e data injection nel tentativo di portare in remoto dei falsi positivi, codici che possono creare un'inconsistenza nei dati verso il server o i device che li scaricassero.

Una volta comunicati in remoto i token dell'utente si è atteso il download delle informazioni aggiornate dal server centrale e si sono registrate le operazioni svolte dall'app durante il controllo dei token e la segnalazione (se presente) del rischio all'esposizione.

Si è tentato quindi in più riprese di andare a leggere i dati all'interno della *sandbox* dell'applicazione in cerca di informazioni personali o di altri utenti, sia su device di infetti che non, valutando il livello di criptazione delle stringhe nel filesystem per ottenere un parametro di rischio in caso di esfiltrazione delle stesse.

Le comunicazioni Bluetooth sono stato oggetto di attenta analisi e tentativo di injection di informazioni potenzialmente dannose o malevole, ma i dati trattati sono sanificati e controllati dai framework Apple e Google, evitando quindi che potenziali stream di informazioni riescano ad essere salvati ed usati in maniera dannosa nel dispositivo.

Si è considerata nei test la possibilità di generare un alto numero di parametri da inviare al server in modo da creare una situazione di "data overflow" che può causare falsi positivi in un ricalcolo degli RPI a più utenti, ma si sono verificati 3 importanti fattori che rendono

questo tipo di *bad data injection* nel sistema molto complicato ed improbabile da portare a termine:

1. l'estrema casualità dei TEK e la variabilità nei parametri di ricalcolo degli RPI;
2. il fatto che il sistema accetti solamente 14 TEK alla volta, legati agli ultimi 14 giorni di spostamento del device;
3. la procedura di push dei token nel sistema che prevede che l'attaccante debba essere indicato come positivo al virus da un operatore sanitario per poter inserire questi dati nel sistema, con una procedura che prevede un'operazione umana di un operatore qualificato.

Terminati i test, questi sono stati ripetuti resettando i device alle impostazioni di fabbrica e ricominciando tutta la procedura di analisi, variando le postazioni di test e le tempistiche di esposizione, lanciando in background alcuni processi che saturassero memoria e processori e ripetendo tutti i precedenti punti per cercare di creare una situazione di stress hardware che portasse a problemi di gestione della memoria o di funzionamento, ma l'unico impatto risultante è stato sulle performance e non ha fatto rilevare modifiche sostanziali alla solidità del runtime dell'app.

Questa ripetizione ha anche potuto assicurare che i TEK e gli RPI gestiti dai framework Apple e Google sono casuali e non legati all'utenza o al device; la riprova si è certificata dal momento che dopo la re-inizializzazione dei contenuti, i nuovi token creati dal framework sono stati confrontati con quelli delle precedenti sessioni e sono risultati completamente diversi e privi di alcun dato in comune – anche parziale – che potesse far risalire ad un tracciamento o identificazione del device a cui sono stati associati.

3.3.3. Audit results

I risultati dell'audit vergono su cinque componenti fondamentali di sicurezza che sono definiti *pillar*. Ognuno di questi pilastri definisce la quantità e la qualità dei dati presenti nell'applicazione, suddividendo la sua capacità di mantenerli a sicuro durante il ciclo di vita dell'esecuzione stessa dei processi.

Il design dell'applicazione testata fonda la propria ingegnerizzazione su quattro concetti fondamentali di preservazione della privacy:

- la mancanza di accesso e di raccolta dei dati personali;
- l'astrazione del tracing del dispositivo dalla persona fisica;
- la casualità e non rintracciabilità nella generazione e nell'assegnazione dei codici di tracing;
- la segregazione fra le logiche di tracciamento degli avvenuti contatti e di identificazione dell'utente risultato positivo al virus;

I test effettuati sono stati volti alla verifica formale e sostanziale che tali requisiti siano rispettati e ottimizzati in ognuno dei pillar al fine di ridurre al minimo ogni possibile rischio che l'app possa essere oggetto di exploit, siano essi destinati al furto diretto delle informazioni in utilizzo o sfruttata come tramite da parte di altri malware per ottenere dati e accessi che altrimenti non sarebbero loro garantiti.

3.3.3.1. Presence of user data

L'applicazione non richiede, né raccoglie in alcun dato informazioni anagrafiche o di alcun tipo che possano collegare gli id univoci assegnati al device con la persona che utilizza il dispositivo. All'avvio l'unico dato richiesto è relativo alla regione ed alla provincia di residenza della persona.

Già dal momento dell'installazione si verifica e certifica che non viene richiesto all'utente alcun dato personale, né in maniera opt-in (volontaria dal cliente), né con richieste al Keychain/Keystore dell'utente o a profili diversi (Google, Apple, etc.) presenti nel dispositivo.

Il Chip Bluetooth del telefono è identificato con un codice univoco (UUID).

L'identificazione del dispositivo è gestita con un sistema di codici definiti "TEK", generati casualmente e che non contengono alcuna informazione sull'individuo.

Tali codici TEK sono rigenerati ogni 24 ore.

A partire dai codici TEK, vengono generati altri codici, gli RPI, che vengono continuamente inviati tramite il Bluetooth. Quest'ultimo, oltre ad inviare i codici RPI, è in continuo ascolto di codici in arrivo da altri dispositivi. Anche gli RPI sono codici randomici che non presentano alcun dato utente, ma soprattutto non racchiudono nessuna informazione relativa alla geolocalizzazione dell'utente.

Gli RPI vengono aggiornati ogni 15 minuti per impedire il tracciamento dell'utente.

Nel caso in cui un utente risulti infetto tramite conferma di un test del tampone, questi può volontariamente avviare la procedura di comunicazione dell'infezione con la supervisione di un operatore sanitario, il quale si assicurerà dell'effettivo stato di infezione e che a quel punto richiederà all'utente una OTP, fornita dall'app al primo step della procedura di segnalazione della propria positività.

Anche questa OTP come tutti gli altri codici del sistema di tracciamento dell'esposizione è generata in maniera casuale e si rigenera ogni volta che viene aperta la schermata di comunicazione del contagio; una volta validata dall'operatore sanitario, l'utente può utilizzarla, come per definizione, per una singola comunicazione che invia al sistema i seguenti dati:

- gli ultimi 14 codici TEK, equivalenti agli identificativi base dell'utente degli ultimi 14 giorni (ricordiamo, non tracciabili né legati alla persona);
- i parametri necessari per il ricalcolo degli RPI a partire dai codici TEK, necessari per permettere al framework di ricalcolare l'effettiva possibile esposizione ad un infetto;
- le informazioni epidemiologiche, parametri di potenza di segnale Bluetooth rilevato (con cui si stima una distanza) e durata dell'esposizione utili ad eseguire il calcolo di un livello di rischio e ridurre il più possibile falsi allarmi;
- la provincia di domicilio indicata nella fase di setup dell'app. Questo dato, che comunque rimane anonimo e non legato alla persona, non viene condiviso con i framework di Apple e Google ma è richiesto e fornito solamente al Governo per permettere un miglior controllo dei rischi epidemiologici alle singole province.

Nel dettaglio, le informazioni epidemiologiche calcolate sono set di riepiloghi delle esposizioni di un singolo device. Ogni singolo riepilogo contiene:

- il numero di TEK che corrispondono ad uno degli RPI salvati nel device;
- il numero di giorni passati dall'ultima rilevazione di TEK corrispondente ad un RPI infetto;
- la somma delle durate di ogni esposizione;
- il punteggio di rischio totale più alto delle ultime esposizioni.

La procedura comunica in alcun modo alcun dato anagrafico o personale, ma esclusivamente questa OTP random.

Una volta segnalata la positività il sistema si occupa di distribuire a tutti i device la lista di TEK infetti e l'applicazione eseguirà un controllo per verificare se tra gli RPI memorizzati nel telefono ci sono corrispondenze con i TEK appena scaricati.

Se uno degli RPI presenti nell'applicazione corrisponde ad un TEK infetto, l'applicazione avviserà l'utente che è stato in contatto con una persona risultata infetta e cambierà lo stato dell'applicazione visualizzando nella schermata di Home il seguente messaggio "Rilevato contatto con una persona positiva al COVID-19"

I TEK e gli RPI hanno una validità di 14 giorni, in modo da gestire in maniera corretta il tempo indicativo del rischio di incubazione (un RPI o un TEK di 3 mesi infatti non avrebbe alcuna utilità).

I TEK associati ad un dispositivo sono generati casualmente ed anonimamente. Gli RPI sono derivati dai codici TEK ed anch'essi randomici ed anonimi.

Non risulta esserci alcuna componente anagrafica, né dato di geolocalizzazione nella formulazione dei codici TEK ed RPI.

3.3.3.2. Network communications

L'applicazione esegue comunicazioni di rete solamente nei momenti necessari di scambio dati per la funzionalità stessa dell'applicazione.

Precisamente avvengono comunicazioni di rete nei seguenti momenti:

- al primo onboarding dell'utente per scaricare la configurazione originale e successivamente, nel caso in cui la configurazione sia cambiata, per gli update della stessa
- per il download dei TEK infetti (non siamo riusciti a ricostruire il periodo esatto. Sembra vari in funzione del fatto che il device sia in carica o in modalità batteria)
- In caso di notifica della positività al COVID-19, volontariamente da parte dell'utente, tramite l'apposita procedura in app

Oltre a quanto sopra indicato ci sono dei collegamenti dall'app verso pagine informative del Ministero che vengono aperte con tecnologia *webview*, ovvero all'interno di una finestra di browser implementata nativamente nell'app. Questi collegamenti sono comunque protetti da due fattori fondamentali che ne assicurano l'integrità: la presenza di una *whitelist* di domini accettati dall'app e la completa disabilitazione dei JavaScript nelle pagine web che vengono aperte, notoriamente la principale fonte di rischio degli attacchi web.

Tutte queste comunicazioni di rete avvengono in un canale sicuro sicuri utilizzando come protocollo di sicurezza il TLS alla sua versione 1.3; i certificati installati sul server sono validi

e la presenza di certificate pinning lato device rende le connessioni ancora più sicure evitando ogni possibilità di semplici attacchi di tipo Man In The Middle (MITM) che altrimenti con la semplice abilitazione di un proxy di rete sarebbero in grado di avere accesso ai contenuti delle comunicazioni, permettendo sniffing ed injection di dati. Sono previste navigazioni esterne all'app (pagine informative su browser) in cui la sicurezza è demandata al sito in questione ed al browser web, ma non ha alcun effetto sulla applicazione o i suoi dati.

L'applicazione presenta comunicazioni sicure e protette, con il più adeguato standard di securizzazione del canale. Il contenuto non è criptato end to end, ma i dati trasmessi non sono da ritenersi sensibili o riservati, anzi anonimi, ed in situazioni normali è impossibile effettuare un attacco MITM nella comunicazione di rete, per cui è impossibile che vengano intercettati.

Le comunicazioni Bluetooth avvengono in maniera automatica quando due device con installata l'applicazione sono in prossimità l'uno dell'altro; quando ciò accade i device si scambiano degli hash univoci che in caso di contrazione del Covid-19 vengono poi riutilizzati dal sistema di tracciamento dei contatti; questi hash sono stringhe alfanumeriche calcolate tramite algoritmi matematici in una sequenza di lettere e numeri che, per definizione, non può essere poi calcolata al contrario per riottenere i dati di partenza, rendendo di fatto ogni dato scambiato completamente anonimo.

L'applicazione non fa uso o richiesta d'uso della tecnologia NFC.

3.3.3.3. Data exchange

Lo scambio dei codici RPI funziona tramite Bluetooth; una distinzione nel funzionamento dei framework Apple e Google va fatta per Android vista la necessità dell'OS (legata all'architettura dei servizi) che il device abbia attivi i servizi di localizzazione per garantire il corretto funzionamento del framework di tracciamento all'esposizione. Durante i test effettuati sull'app non è stato trovato infatti alcun dato relativo alla geolocalizzazione del dispositivo. In ogni caso l'app Immuni per Android non ha nel framework nemmeno i permessi per garantire l'utilizzo delle informazioni derivate dai servizi di localizzazione, il che renderebbe impossibile l'accesso a qualsiasi dato derivante dal modulo GPS o simili sistemi.

Per lo scambio dei codici RPI tramite Bluetooth, l'SDK necessita dei permessi di localizzazione (solo Android). Durante i test effettuati non sono stati trovati dati relativi alla geolocalizzazione del dispositivo. In ogni caso l'app Immuni non ha i permessi per utilizzare la localizzazione, utilizzata infatti solo dal sistema per il trasferimento dei codici TEK e RPI che restano comunque anonimi e privi di informazioni di localizzazione.

Nella RAM del dispositivo, durante l'esecuzione non persistono dati che possano essere raccolti con un dump di memoria.

Non sono previsti URI scheme per l'apertura di smartUrl con l'applicazione Immuni.

Le comunicazioni intra-processo avvengono solamente tra l'app e librerie del sistema operativo, nello specifico con i framework Apple e Google, segregando così l'app da attacchi provenienti da altre fonti.

Non essendo previsti data input dagli utenti non viene abilitata la clipboard di sistema e non vengono salvati screenshot dell'app nella galleria di sistema nell'albero delle app attive.

I dati di esposizione, di tracing e di contatto sono tutti scambiati via Bluetooth, anonimizzati e sfruttando le librerie dei sistemi operativi proprietari Apple/Google per la gestione dei token (v. 3.3.3.2).

L'integrità delle comunicazioni client-server per la segnalazione del contagio è assicurata da un'autorizzazione rilasciata da un operatore sanitario, rendendo pressoché impossibile l'injection di dati fittizi nel database dei positivi.

3.3.3.4. Filesystem

L'applicazione non ha persistenza locale di dati sensibili o riconducibili alla persona fisica.

La gestione del filesystem dell'applicazione si limita alla sandbox applicativa, ovvero un'area di memoria del device riservata alla singola app a cui nessun'altra applicazione ha accesso.

In questo spazio dedicato l'app comunque non salva alcun dato personale dell'utente in maniera esplicita, limitandosi ad avere in chiaro al suo interno solamente una cache tecnica con alcuni parametri di setup, utile ad ottimizzare il quantitativo di traffico internet generato dal device.

Il dato relativo alla provincia di residenza invece è salvato in sandbox ma risulta criptato, pertanto inaccessibile, se non utilizzando una chiave di decifratura che solamente l'app possiede e che sfrutta quando l'utente cambia il dato della provincia o quando deve comunicarla in remoto per la segnalazione dell'infezione.

Vengono salvati in locale i TEK, che vengono utilizzati per derivare i codici RPI che vengono trasmessi via Bluetooth.

Il codice TEK viene rinnovato ogni 24 ore (v. 3.3.3.1).

Anche i codici RPI altrui (ricevuti durante un contatto) vengono storicizzati per essere poi confrontati con gli RPI dei TEK infetti scaricati dal server e verificare se si è stati in contatto con un positivo negli ultimi 14 gg.

L'RPI proprio del dispositivo dell'utente invece non viene storicizzato, ma viene mantenuto solamente l'RPI attivo in quello specifico momento.

I codici TEK ed RPI non contengono informazioni personali, né di geolocalizzazione, ma sono generati casualmente dal sistema (v. 3.3.3.1).

TEK ed RPI vengono cancellati dopo 14 giorni.

3.3.3.5. Codebase

3.3.3.5.1. Android

Il codice dell'applicazione Android presenta pochissimi rischi per la sicurezza. Tutti i permessi richiesti dall'app (accesso alla rete, bluetooth, esecuzione in primo piano e prevenzione dallo spegnimento dello schermo) sono tutti fondamentali al corretto funzionamento della stessa e ne viene fatto un uso ponderato e pesato sulle reali necessità.

L'applicazione espone anche alcuni receiver, servizi interni esposti ad altri processi in esecuzione sullo stesso sistema, che vengono richiamati in particolari momenti (ad esempio all'accensione o al riavvio del device) o su richiesta di altre app e che eseguono

operazioni a nome dell'app stessa. Questi receiver risultano adeguatamente protetti e confinati ai casi d'uso previsti grazie ad una corretta implementazione dei controlli nel codice.

L'app non fa uso di metodi o classi deprecate, implementa solo gli URL necessari al suo corretto funzionamento ed è priva di librerie di tracking ed analytics, assicurando così per definizione che nessun dato dell'utente sia esposto a terze parti.

3.3.3.5.2. iOS

Il codice scritto per l'app iOS non presenta rischi per la sicurezza.

L'applicazione non richiede alcun permesso legato all'hardware del device, limitandosi ad assicurarsi che l'utente accetti nella privacy la condivisione dei dati per il tracciamento dell'esposizione al Covid-19.

Non ci sono eccezioni alla sicurezza legate ai requisiti minimi di connessione, e l'app è stata compilata con tutti i possibili flag di protezione agli attacchi di basso livello legati allo sfruttamento della memoria RAM, limitando i possibili danni che un exploit di quel tipo potrebbe generare.

Tutti i domini presenti nel codice di cui l'app fa uso sono stati controllati e risultano sicuri, così come le librerie e i framework.

4. Conclusioni

L'audit di sicurezza effettuato su Immuni di Bending Spoons S.p.A. è stato eseguito nel periodo che va dal 21/05/2020 al 01/06/2020.

I risultati dell'analisi certificano che l'applicazione è coerente con la relativa documentazione pubblica.

Durante le sessioni di utilizzo si è potuto verificare che i seguenti principi sono preservati:

- mancanza di accesso e di raccolta dei dati personali;
- astrazione del tracing del dispositivo dalla persona fisica;
- casualità e non rintracciabilità nella generazione e nell'assegnazione dei codici di tracing;
- segregazione fra le logiche di tracciamento degli avvenuti contatti e di identificazione dell'utente risultato positivo al virus

In dettaglio **l'applicazione non raccoglie** né in opt-in dell'utente, né in meccaniche automatiche **alcun dato personale, privato o sensibile relativo all'identità anagrafica** ad eccezione della provincia di domicilio, utilizzata per fini di gestione del processo di notifica presso l'unità sanitaria locale dell'utente.

Non vengono raccolte informazioni sull'identità dei soggetti né durante il processo di onboarding dell'utente, né durante la registrazione dei contatti fra i dispositivi.

Non c'è commistione di dati con altre applicazioni installate nel dispositivo (account Google o iCloud del device, librerie e servizi dei sistemi operativi, applicazioni legittime o malevole di terze parti) **che possano in qualche modo permettere l'identificazione del soggetto**.

L'applicazione integra le SDK Apple e Google (pubblicamente documentate) e riporta nativamente alcune funzioni di carattere puramente informativo e didascalico per le operazioni di supporto all'utente in caso di infezione o di potenziale contatto, oppure sezioni informative della tutela della privacy dell'utente e dei termini e condizioni di utilizzo del software.

In caso di **contagio** o **potenziale contatto** con un positivo il **dato viene elaborato dal sistema e condiviso con l'amministrazione centrale** e gli enti preposti alle operazioni di supporto sanitario al cittadino e **mai** con le SDK Apple e Google.

Le **comunicazioni di rete** TCP/IP risultano **sicure** ed effettuate in un canale **HTTPS** in cui è correttamente implementato il **certificate pinning** per tutta una serie di domini e sottodomini definiti nei file di configurazione dell'app, nello specifico qualsiasi indirizzo web che termini con immuni.gov.it (wildcard **.immuni.gov.it*).