ROMÂNIA
MINISTERUL APĂRĂRII NAȚIONALE
ACADEMIA TEHNICA MILITARĂ
„FERDINAND I”

FACULTATEA DE SISTEME INFORMATICE ŞI SECURITATE CIBERNETICĂ
Specializarea: Calculatoare şi sisteme informatice pentru apărare
şi securitate naţională

Aplicație destinată telefoanelor mobile în vederea detecției conectării acestora la stații de bază false care se interpun între terminalul mobil și stațiile de bază ale furnizorilor oficiali de servicii GSM (IMSI CATCHER DETECTOR)

CONDUCĂTOR ȘTIINȚIFIC:
Col. conf. univ. dr. ing. BĂDOI ION

ABSOLVENT:
Ștefan OLTEANU

Conţine _____ file
Inventariat sub nr_____
Poziţia din indicator: ____
Termen de păstrare: _____

BUCUREȘTI
2020

ABSTRACT

IMSI Catchers have become more and more widespread, attacking people's rights to privacy. Even if advances in technology try to stop them from working, hackers will always find a solution to break the security of GSM networks. Therefore, the need of creating an application that will make people aware of the danger they expose to when using their mobile phones, became more and more important. The demonstrative application runs on Android based smartphones and tries to announce users every time their phones connect to a fake cell. All the cells the phone interracts with, are kept in a database so that people will always have a record of the actvity of their phone. Finally, it must be taken into consideration that the current application cannot detect the attack before it happens, cannot protect users from being attacked but only notify them that something abnormal occured.

REZUMAT

Prezenta lucrare își propune descrierea modului de funcționare a dispozitivelor de tip *IMSI Catcher* precum și importanța descoperirii acestor atacuri. Toate persoanele ce folosesc un telefon mobil se expun unui atac ce le amenință dreptul de a avea o viață privată. Mai mult, nimeni nu este conștient de faptul că poate fi urmărit iar problema cea mai mare este că aceste dispozitive sunt utilizate adesea în mod ilegal de către autorități, fără a avea dreptul să le utilizeze în anchetele desfășurate.

Astfel, a apărut nevoia de a avea o aplicație care să fie de partea utilizatorului obișnuit, care să-l anunțe în momentul în care detectează o activitate anormală în apropiere. Din păcate aplicația nu îl va putea feri de eventualele atacuri, dar îl poate face conștient de acest lucru, fapt care nu se întâmplă, de obicei, în utilizarea de zi cu a zi a unui telefon mobil.

În capitolele ce urmează vor fi descrise modul de operare al unui dispozitiv *IMSI Catcher*. Va fi descris, în mare, modul de funcționare al rețelelor GSM dar și cum sunt exploatate urmând ca în final să fie prezentată arhitectura internă a unei aplicații demonstrative, al cărei scop este acela de a detecta astfel de atacuri. Vor fi prezentate metodele de programare utilizate în dezvoltarea, atât a părții vizuale, cât și a părții ce se ocupă de testarea propriu-zisă a celulelor la care se conectează telefonul mobil.

Finalul documentului cuprinde concluziile desprinse în urma realizării acestei lucrări dar și modalități de extindere sau îmbunătățire a proiectului ce pot fi aplicate în viitor.

CUPRINS

LISTĂ DE FIGURI

1 INTRODUCERE

1.1. Importanța temei alese

Dispozitivele de tip *Imsi Catcher* există încă din anul 1993 (1) însă erau mari, greoaie și mai ales scumpe. Datorită acestui lucru și a faptului că existau puțini producători, aceste dispozitive nu au fost folosite decât de unele agenții guvernamentale. În zilele noastre, apariția SDR-urilor[1] a permis crearea unor proiecte de acasă și astfel înmulțirea atacurilor asupra rețelelor GSM. Mai mult, dacă inițial aceste atacuri au fost dezvoltate pentru a putea captura codurile IMSI[2] de la telefoanele din apropiere – de unde vine și numele atacului -, versiunile actuale oferă posibilitatea de a intercepta atât apeluri telefonice cât și mesaje.

Prin urmare, această temă este cu atât mai importantă, cu cât nu există modalități de detectare a acestor atacuri, disponibile publicului larg. De asemenea, intrucât numarul acestora a fost în continuă creștere, este necesară informarea persoanelor cu privire la riscurile la care se expun în momentul în care telefonul se conectează la o stație de bază ce nu aparține unui furnizor de servicii local. Un articol realizat de cei de la Washington Post arată că numărul dispozitivelor de interceptare ilegale, găsite în oraș pe parcursul a mai puțin de două zile, este de aproximativ 18 (2). Așadar, nevoia unei aplicații care să ajute la detectarea acestor atacturi ce pun în pericol drepturile societății, este mai mult decât evidentă.

1.2. Scopul și obiectivele lucrării

Scopul acestei lucrări de licență este acela de a expune problematica atacurilor de tip *Imsi Catcher*, de a evidenția modalități de detecție și nu în ultimul rând de a crea o aplicație Android funcțională care să pună în practică modalitățile precizate mai sus.

Printre obiectivele care trebuiesc atinse pe parcursul rezolvării acestei teme se numără un număr de informații cu ajutorul cărora cititorul își poate face o idee

[1] Software Defined Radio – sistem de comunicare radio. Componente care în mod normal ar fi implementate hardware, precum amplificatoare, modulatoare sunt implementate pe partea de software.

[2] International Mobile Subscriber Identity – cod alocat utilizatorului de către operatorii de telefonie, ce le permite să identifice abonatul in funcție de numărul de telefon.

complexă despre modul de funcționare atât a dispozitivului cu care se realizează atacul cât și a dispozitivului de interceptare. De exemplu:

- informații despre modul în care o rețea GSM functionează și ce puncte slabe are;
- prezentarea pe larg a modalităților prin care un dispozitiv *Imsi Catcher* exploatează rețeaua și reușește să capteze atât locația utilizatorului cât și mesaje sau apeluri telefonice;
- descrierea metodelor folosite în cadrul aplicației pentru a detecta atacul;
- s.a.m.d.

Obiectivul principal este ca la finalul lucrării să fie posibilă prezentarea unei aplicații demonstrative, capabilă să distingă antenele ce nu aparțin furnizorilor de servicii de telefonie mobilă de cele de bază. Aplicația va trebui să ofere informații utilizatorului despre celula la care telefonul acestuia s-a conectat, să îi indice pe harta o localizare aproximativă a acesteia și să efectueze o serie de teste, care să demonstreze că celula curentă nu reprezintă un pericol pentru securitatea utilizatorului. Va exista posibilitatea începerii acestor teste atât manual de către posesorul telefonului, cât și automat la schimbarea antenei de către dispozitivul mobil.

1.3. Prezentarea metodologiei de cercetare

Pentru a redacta prezenta lucrare și pentru dezvoltarea aplicației demonstrative s-au consultat diferite articole științifice și site-uri care conțineau informații despre subiectul curent. S-a dorit realizarea unei imagini de ansamblu asupra a tot ceea ce înseamnă dispozitivele malițioase de tip *IMSI Catcher*, a rețelei GSM asupra căreia acestea acționează dar și a modalităților existente de dectare.

Dezvoltarea aplicației demonstrative a început cu cercetarea modului în care un dispozitiv malițios exploatează rețeaua dar și a modalităților prin care acesta poate fi detectat. Au fost utilizate atât materiale online sub formă de documente sau articole de pe diferite site-uri cu ajutorul cărora s-au elaborat diferitele teste pe care aplicația le utilizează.

Scopul final al acestei documentații este de a prezenta o aplicație funcțională ce reușește să intercepteze dispozitivele malițioase ce exploatează rețeaua.

1.4. Rezumatul lucrării pe capitole

Prezenta lucrare conține un număr de șase capitole, după cum urmează:

- primul capitol prezintă importanța temei alese împreună cu scopul și obiectivele acestei lucrări. Mai mult, este realizată o scurtă descriere a metodologiei de cercetare și un rezumat pe capitole al întregului document.
- cel de-al doilea capitol își propune descrierea reglementărilor în vigoare cu privire la dispozitivele de tip *IMSI Catcher* dar și o descriere a rețelei GSM pe care acestea o exploatează. În final, este prezentat modul în care un dispozitiv malițios desfășoară atacul.
- al treilea capitol al lucrării prezintă o analiză a modalităților existente de detectare pentru astfel de atacuri dar și o descriere a unei baze de date publice, pusă la dispoziție pentru verificarea celulelor pe care dispozitivul mobil le întâlnește.
- al patrulea capitol prezintă atât mediul de lucru pentru dezvoltarea aplicației demonstrative, cât și arhitectura internă a acesteia. În partea finală, este descrisă și arhitectura vizuală, capitolul urmând să se încheie cu o simulare, în care se demonstrează că celula la care era conectat telefonul aparține unui furnizor de telefonie mobilă.
- ultimele două capitole conțin principalele concluzii ale acestei lucrări, direcții pentru continuarea cercetării și îmbunătățirea aplicației demonstrative și în final referințele bibliografice ale lucrării.

2 CLASIFICAREA STAȚIILOR DE BAZĂ ÎN FUNCȚIE DE LEGALITATE

2.1. Reglementări cu privire la *Imsi Catcher*

Aceste dispozitive folosite în mare parte de agențiile de securitate statală au ridicat probleme în legătură cu libertatea și intimitatea civilă. De aceea folosirea acestora se realizează numai conform regulilor scrise în *Codurile de Procedură Penală* (3).

Orice utilizare a unui dispozitiv de tip *Imsi Catcher* se realizează sub jurisdicția *Convenției Europene a Drepturilor Omului,* care stabilește un nivel minim de protecție atât pentru viața privată a individului cât și pentru corespondența acestuia (4). Toate aceste legi ar trebui să analizeze foarte bine comportamentul dispozitivelor, întrucât acestea nu sunt folosite numai pentru monitorizarea prezenței telefoanelor mobile într-o anume zonă, având ca scop identificarea locației pe hartă a acestora, ci și pentru a aduna informații despre utilizatori sau despre dispozitivele pe care aceștia le folosesc.

Pe lânga drepturile descrise mai sus, pe care un astfel de atac ar putea să le încalce, se mai adaugă și încălcarea dreptului la libertatea întrunilor pașnice. Există numeroase modalități prin care acest lucru este posibil, cum ar fi (5):

- persoanele ce cunosc modul de operarare ale acestor tipuri de dispozitive, nu vor mai participa la întalniri, proteste sau orice altă manifestare pașnică deoarece vor ști că prin captarea comunicațiilor mobile, aceștia pot fi urmăriți;
- prin captarea, editarea și rutarea comunicațiilor, abilitatea persoanelor de a comunica între ele va fi alterată;
- în unele cazuri, este chiar posibil ca unele guverne să trimită mesaje tuturor telefoanelor mobile aflate în zona în care se desfășoară o întalnire pentru a intimida, manipula sau pentru a împrăștia utilizatorii.

Deși există o lege europeană care face referire la protejarea individului, nu există legi specifice pentru utilizarea acestor dispozitive. Guverne din Statele Unite ale Americii și Germania au stabilit legi și politici care limitează modalitățile în care agențiile statatale pot utiliza în mod legal dispozitive de tip *Imsi Catcher.* Mai mult, Judecătorii din SUA au impus restricții suplimentare. Câteva exemple referitoare la legile adoptate de Ministerul de Justiție din SUA cu privire la utilizarea *Imsi Catcher* vor fi expuse mai jos: (6)

- limitarea folorii acestor dispozitive numai pentru identificare;
- cererea supervizării interne pentru utilizarea acestora;
- nevoia unui mandat pentru utilizarea acestora în situații care nu sunt urgente sau excepționale.

În cazul Germaniei, *Imsi Catcher-urile* au fost supuse legilor federale încă din anul 2002, atunci când utilizarea acestora a fost autorizată ca urmare a atacurilor de pe 9/11 din Statele Unite ale Americii. Curtea de Justiție Germană are dreptul de a autoriza folosirea unui astfel de dispozitiv numai în cazul în care există dovezi clare că o crimă serioasă are loc. Mai mult, persoanele care sunt vizate de aceste dispozitive trebuie anunțate că au fost subiecții unui *Imsi Catcher*, fără a pune în pericol anumite aspecte ale investigației (6).

2.1.1. Utilizarea IMSI Catcher în afara legii

Deși atât în Europa cât și în Statele Unite ale Americii există reglementări în legătură cu utilizarea acestui tip de dispozitive, guvernele, agențiile de securitate și anumite persoane rău intenționate le utilizează fără un mandat. Spre exemplu, în 2014, poliția din Florida a recunoscut că a folosit *Imsi Catcher-uri* de cel puțin 200 de ori fără a recunoaște acest lucru în fața judecătorilor sau a avea un mandat care să le permită folosirea acestui dispozitiv (7).

Ideea care ridică probleme în acest sens, este aceea că poliția nu folosește acest tip de dispozive numai pentru prindere criminalilor dar și pentru spionarea populației. Chiar dacă în SUA exista numeroase state care dețin astfel de dispozitive, utilizarea lor nu este una transparentă, fapt care a ridicat multe semne de întrebare, deoarece nu se știe dacă acestea au fost folosite cu un mandat sau nu (8).

Pe lângă poliție sau guverne, dispozitivele de tip *IMSI Catcher* sunt adesea folosite si de spioni statali pentru a intra în posesia unor date despre ofițerii de inteligență. Acest lucru este posibil prin identificarea codurilor IMSI ale telefoanelor acestor ofițeri, care sunt localizate 8 ore pe zi în același loc. (9)

2.2. Modalități prin care IMSI Catcher exploatează rețeaua

Ținta unui *IMSI Catcher* este rețeaua GSM. Acest dispozitiv se infiltrează in rețeaua GSM printr-un atac de tip *man-in-the-middle*[3]. Practic, acesta reprezintă o antena falsă, ce este plasată între telefonul țintă si antena adevărată, a furnizorului de servicii. Chiar dacă standardele GSM oferă protecție împotriva riscurilor, datorită autentificării mutuale între dispozitiv și rețea, protocolul nu este sută la sută sigur, deoarece atacatorii pot declasa servicii cum ar fi 3G sau LTE în servicii de rețea non-LTE care nu au nevoie de această autentificare mutuală specificată mai sus.

În capitolele ce urmează vor fi prezentate detalii despre rețelele GSM, modul acestora de funcționare, securitatea lor cât și despre câteva tipuri de atacuri cunoscute. În final, după ce s-a descris cum funcționează rețelele GSM, se vor prezenta modalități prin care *IMSI Catcher-urile* exploatează rețeaua.

2.2.1. Rețele de comunicații GSM

GSM sau *Global System for Mobile Communications,* standard folosit pentru a descrie protcoalele din a doua generație(2G) de rețele celulare digitale, a fost pentru prima oara folosit în anul 1991 în Finlanda. În jurul anului 2010, acesta a devenit un standard global pentru comunicațiile mobile, operând în peste 193 de țări și teritorii. Pentru o bună funcționare a tuturor echipamentelor compatibile cu această rețea dar și interconectarea acestora într-un mod corespunzător, standardul pune la dispoziție 161 de recomandări. Toate aceste reguli privind sistemele și serviciile GSM sunt guvernate de ETSI(*European Telecommunications Standards Insitute)* (10).

În Europa, standardul GSM folosește benzi de frecvență între 900 MHz și 1800 MHz pe când în Statele Unite ale Americii este folosită banda de 1900 MHz. Din această cauză, telefoanele mobile care pot opera pe ambele continente sunt numite *tri-band,* în timp ce acelea care operează doar în Europa sunt cunoscute sub

[3] Atac man-in-the-middle – tip de atac în care o persoană rău intenționată se infilitrează între furnizorul de date și țintă. Acesta poate fi atât pasiv, doar pentru monitorizare, cât și activ prin alterarea datelor.

denumirea de *dual band*. Deoarece standardul acceptă un maxim de 9,6 kbps se pot transmite doar voce sau un volum mic de date, cum ar fi mesaje text sau mesaje multimedia (11).

2.2.2. Securitatea în GSM

Deși standardul GSM este considerat a fi cel mai sigur sistem de telecomunicații celulare din ziua de azi deoarece folosește autentificarea folosind o cheie pre-partajată și autentificare de tip întrebare-răspuns, acesta este vulnerabil la diferite tipuri de atacuri, fiecare din ele țintind o altă parte a rețelei.

Operatorii de telefonie mobilă sunt astfel obligați să asigure atât securitatea propriilor servicii dar și a clienților întrucât acestora le trebuie oferită garanția că nimeni nu le poate intercepta conversațiile sau să le detecteze locația. Cu toate acestea, o rețea sigură nu înseamnă o rețea care să îngreuneze apelurile telefonice sau transmisia de date. Astfel, principalele mecanisme de securitate folosite în standardul GSM sunt împărțite în 4 mari categorii (12):

Autentificarea utilizatorului se realizează printr-un mecanism de securitate de tipul întrebare-răspuns. Un număr random de 128 de biți este trimis către stația mobilă care la rândul ei calculează un răspuns pe 32 de biți bazat pe numărul primit dar și pe cheia de autentificare individuală. Criptarea acestui răspuns este realizată cu ajutorul algoritmului A3.

Criptarea datelor și semnalului se asigură că datele utilizatorilor, cum ar fi mesaje text dar și voce, sunt protejate împotriva interceptării. Procesul de criptare este îndeplinit cu ajutorul algoritmului A5. Tot procesul de comunicare începe, însă, printr-o cerere la rețeaua GSM pentru a specifica modul de criptare. Astfel, operatorul de rețea este cel care decide modul în care se va desfășura întregul proces de comunicare. Mai mult, cheia de criptare este schimbată la intervale regulate de timp pentru a face rețeaua chiar mai rezistentă la interceptări. Calculul acestei chei are loc în interiorul SIM-ului și astfel această informație nu va fi niciodată divulgată de SIM.

Confidențialitatea identității utilizatorului asigură securitatea IMSI[4] prin faptul că într-o comunicație GSM, codul IMSI este comunicat rar. Este mult mai

[4] International Mobile Subscriber Identity – identifică în mod unic fiecare utilizator dintr-o rețea GSM

sigur să se folosească un cod TMSI[5] pentru a evita descoperirea identității unui utilizator al rețelei. Acest lucru presupune că un atacator ce interceptează comunicația nu va putea afla dacă un anume telefon mobil se află în aria căutată.

Cartela SIM asigură securitatea prin simplul fapt că în cazul pierderii, aceasta va cere un cod PIN pentru a putea fi folosită. Metoda este utilă doar în cazul în care codul PIN este activat și cel implicit a fost schimbat de către deținătorul cartelei.

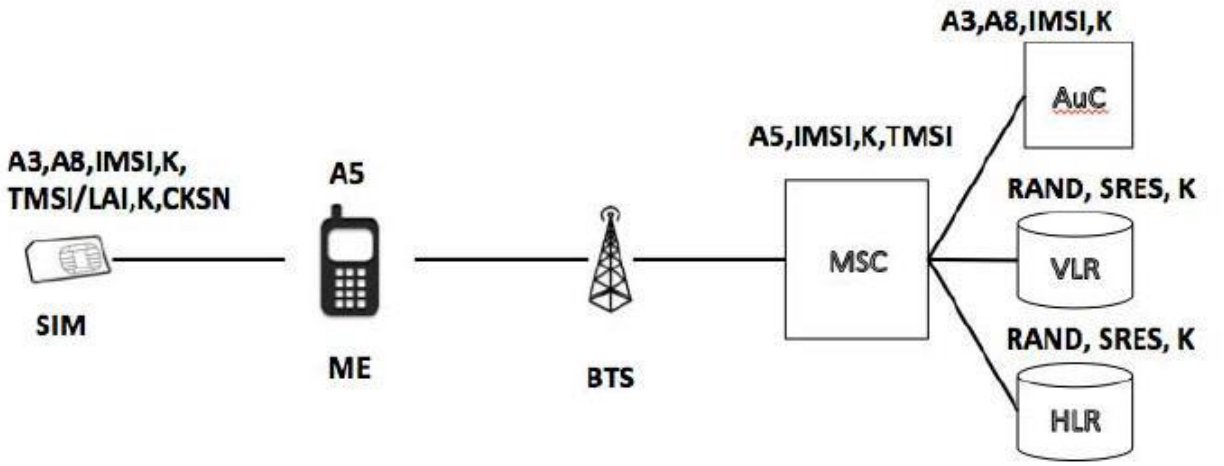


Fig. 2.2.2.a Conectarea la o stație de bază (13)

2.2.3. Tipuri de atacuri populare

În majoritatea tipurilor de atac cunoscute, atacatorul trebuie să pretindă că este ori o stație de bază pentru terminalul mobil, ori o stație mobilă pentru celula la care dorește a se conecta. Aceste tipuri de atac sunt cunoscute, așa cum am menționat și în capitolul anterior, sub numele de *man-in-the-middle.* Atacurile asupra unei rețele GSM pot fi atât active cât și pasive, dar un atac activ le implică pe amândouă.

Înainte de un atac activ, atacatorul trebuie să asculte și să învețe din ce este formată informația pe care stația mobilă o trimite stației de bază. În momentul în

[5] Temporary Mobile Subscriber Identity

care s-a infiltrat între cele două stații, atacatorul va avea posibilitatea să controleze toate mesajele care sunt trimise către telefonul mobil. După ce a furat identitatea utilizatorului, acesta poatre trimite mesaje false în numele acestuia.

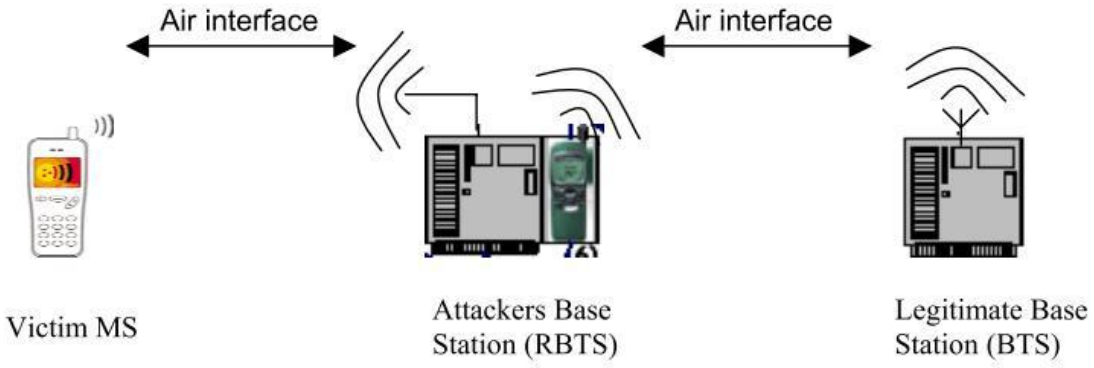


Fig. 2.2.3.a Mecanismul unui atac Man-in-the-Middle (12)

Mai jos, vor fi prezentate două din cele mai populare tipuri de atac asupra rețelelor GSM, după cum urmează (12):

Atacul asupra anonimității utilizatorilor GSM

Pentru a putea localiza un abonat GSM, atacatorul se poate folosi de momentul în care celula inițiază procedura de identificare a utilizatorului, care se inițializează numai în cazul în care rețeaua nu poate identifica stația mobilă folosind un TMSI. Dacă atacatorul deține tehnologiile necesare pentru a se da drept o stație de bază, este suficient de ușor să ceară codul IMSI unui telefon mobil.

În momentul în care un atacatorul deține codul IMSI al victimei, acesta poate abuza de procedura de identificare prin cererea unui cod TMSI. Terminalul mobil nefiind conștient de faptul că este conectat la o stație de bază falsă, îl va comunica și astfel locația îi poate fi dezvăluită deoarece antena falsă va putea face cereri țintite către terminalul ce deține codurile IMSI/ TMSI.

Atacul asupra algoritmului de autentificare

Majoritatea operatorilor GSM folosesc algoritmi de criptare ce sunt recomandați în GSM MoU [6] în loc să își creeze algoritmi proprii pentru autentificare și generare de chei. Acest lucru se întâmplă datoriă imposibilității

[6] MoU – Memorandum of Understanding, mai târziu devenit GSM Association GSMA

schimbării tuturor cartelelor SIM ce conțin algoritmii de criptare dar și a costurilor implicate de aceste schimbări.

Acest atac presupune clonarea unei cartele SIM originale, ce se poate face atât fizic cât și prin intermediul aerului. Astfel, din momentul în care atacatorul a reușit să copieze cheia secretă și codul IMSI într-o cartelă goală, se poate infiltra în rețea. În final, ținta atacului își va pierde intimitatea în rețeaua GSM întrucât îi vor fi decriptate toate apelurile.

2.2.4. Modul de funcționare al IMSI Catcher-urilor

În rețelele GSM, telefoanele mobile vor încerca să se conecteze la stația de bază care oferă cel mai puternic semnal. Odată ce telefonul a identificat stația de bază cu semnalul cel mai bun, poate începe conectarea la aceasta. Cum celulele GSM au posibilitea să ceară telefoanelor să oprească criptarea datelor, la fel și un dispozitiv IMSI Catcher poate face acest lucru.

Următorul pas în conectarea la o stație de bază este identificarea terminalului mobil. Telefonul reușește să se identifice în rețea printr-un cod IMSI care se află stocat pe cartela SIM primită de la furnizorul de servicii. În momentul în care un dispozitiv malițios s-a infilitrat în rețea, acesta preia codul primit de la telefon după care se retrage pentru a permite telefonului să se întoarcă în rețea. Acesta este modul de bază în care un IMSI Catcher preia codul IMSI al telefonului dar de aici se poate continua cu diferite atacuri mai sofisticate.

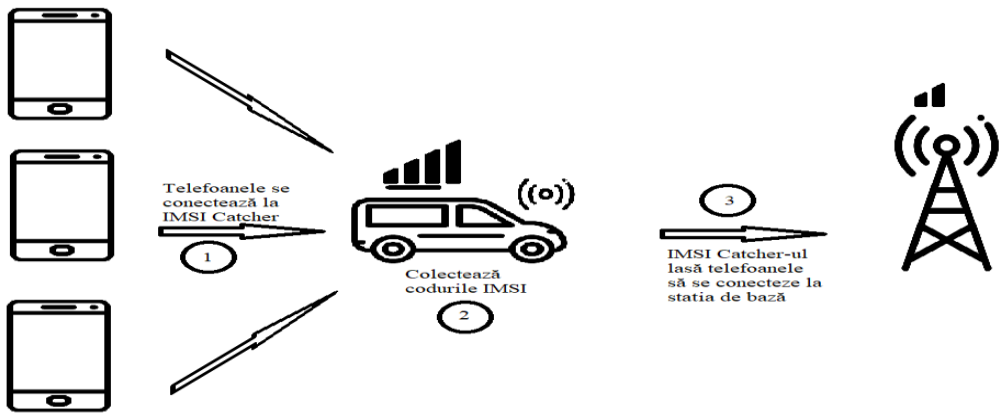


Fig. 2.2.4.a Modul de operare al unui IMSI Catcher

Odată obținute codurile IMSI, atacatorul poate porni un atac activ, și anume să fure identitatea în rețea a victimei. Pentru a realiza acest lucru, persoana rău intenționată trebuie să parcurgă doi pași principali (14):

Autentificarea în rețea

1. Atacatorul se conectează la celula de bază cu un răspuns la *Location Update Request*(cerere periodică făcută de antenă tuturor telefoanelor pentru a putea ruta apeluri și SMS-uri rapid).
2. Ca urmare a acestui răspuns, stația de bază îi cere dispozitivului malițios să se autentifice în rețea folosind o *Cerere de autentificare.* Bineînțeles, acesta va răspunde folosind codul IMSI furat anterior.
3. În acest moment, stația de bază va cere IMSI Catcher-ului să rezolve o problemă criptografică folosind o cheie privată. Cum nu are acces la cheia respectivă, ea fiind stocată pe cartela SIM a telefonului, acesta va pasa problema telefonului, care o va rezolva și va da înapoi răspunsul.
4. În final, este acceptată conexiunea între dispozitivul malițios și rețea.

Acest lucru este însă posibil numai la nivelul rețelelor 2G. Cu toate acestea, atacatorii cu experiență au posibilitatea de a degrada conexiunea telefonului de la 3G/ 4G la GSM. Posibilitatea se bazează pe faptul că stația de bază poate alege ce configurație dorește și să o impuna telefonului. Mai mult, telefonul poate decide singur să degradeze conexiunea în momentul în care benzile de conexiune 3G sau 4G sunt prea aglomerate sau există prea mult zgomot pentru a reuși să realizeaze o conectare sigură.

Rezolvarea problemei criptării

După cum am specificat anterior există mai mulți algoritmi folosiți în GSM, aceștia având nume precum: A5/1, A5/2 etc ... iar A5/0 înseamnă ca nu este folosită nici o modalitate de criptare.

Există două posibilități în acest caz. Rețeaua poate specifica telefonului să comunice folosind criptare, iar dispozitivul malițios să răspundă ca nu are capabilități de criptare. În al doilea rând, rețeaua poate stabili să folosească un anume algoritm dar acestea pot fi de obicei sparte în timp real. În ambele cazuri, în acest punct, atacul este complet iar atacatorul poate citi mesajele în clar dintre telefon și stația de bază.

Cele doi pași prezentați mai sus fac referire la momentul în care un dispozitiv *IMSI Catcher* este utilizat pentru a intercepta comunicarea dintre

telefonul mobil și stația de bază la care s-a conectat. Deși este o problemă gravă, adevărata problemă o reprezintă urmărirea locației în timp real. Astfel, chiar și acest tip de atac se împarte în două categorii (14):

Verificarea prezenței

Pentru acest tip de atac nu este nevoie ca atacatorul să pretindă că este o celulă de bază ci poate folosi echipamente precum un SDR și un laptop pentru a monitoriza semnalele din zona sa.

Acest lucru este posibil deoarece în cazul tehnologiilor fără fir se utilizează mesaje de tip RRC[7]. În momentul în care rețeaua are de trimis un mesaj și dorește să îl direcționeze către un telefon, trimite un mesaj RRC care este primit de toate telefoanele din zona de interes, cerându-i telefonului țintă să contacteze stația de bază pentru a realiza conexiunea și a primi apelul sau mesajul. Astfel, telefoanele monitorizează permanent aceste mesaje, le primesc și le dau la o parte pe cele care nu le sunt adresate.

În cazul în care numărul de telefon al victimei este cunoscut, acest procedeu prezentat mai sus poate fi forțat prin trimiterea unui SMS către aceasta. Rețeaua va putea fi monitorizată pentru a capta momentul de timp în care telefonul se conectează la stația de bază pentru a prelua mesajul.

Dobândirea locației exacte(coordonate GPS)

În acest scenariu, atacatorul a reușit să atragă telefonul să se conecteze la stația lui falsă, folosind metodele prezentate mai sus. Dispozitivul malițios are acum posibilitatea de a trimite o comandă telefonului prin care îi trimite id-urile a cel puțin 3 celule din apropiere și frecvențele acestora de conectare. Telefonul primește această comandă și îi întoarce dispozitivului malițios puterile semnalului oferit de fiecare dintre celulele cerute. Poziția dispozitivului țintă este apoi calculată prin triliterație, în cazul telefoanelor mai vechi, iar în cazul unora mai noi, răspunsul dispozitivului conține și coordonatele exacte ale acestuia.

Triliterația, diferită de triangulație, presupune calcularea intersecției cercurilor desenate în jurul fiecărei celule menționate în comanda pe care *IMSI Catcher-ul* o dă telefonului. Raza fiecărui cerc este o funție ce depinde de puterea semnalului, pe care telefonul o trimite înapoi, ca răspuns, dispozitivului malițios.

[7] RRC – Radio Resource Control, protocol folosit pentru comunicarea dintre telefon și stația de bază (14)

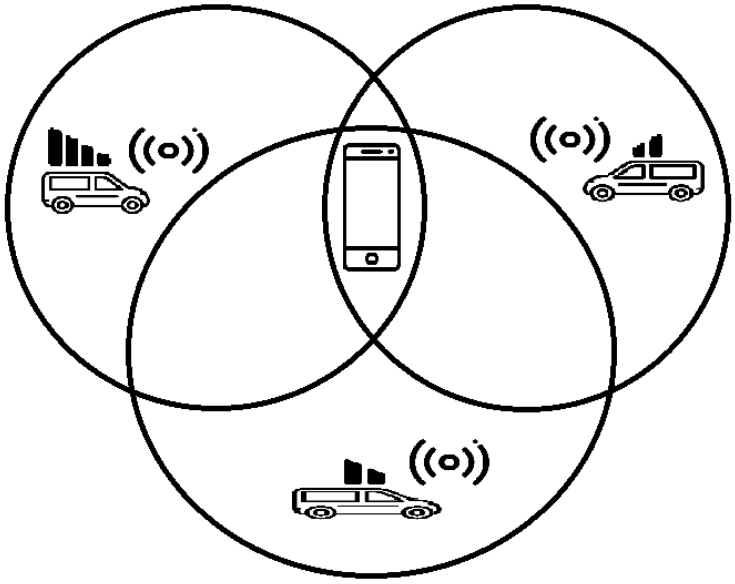


Fig. 2.2.4.b Triliterația folosită ca metodă de calcul ale coordonatelor GPS ale telefonului țintă

Un ultim tip de atac pe care un dispozitiv *Imsi Catcher* îl poate realiza asupra unei rețele este un atac de tipul *Denial of Service*[8]. Deși există metode simple de a realiza un astfel de atac, precum aglomerarea rețelei cu semnal alb, sunt folosite tehnici pentru atacuri DoS care țintesc telefoane individuale.

Un atac *denial of service* asupra unui singur dispozitiv prespune exact același procedeu ca în cazul degradării protocolului, acțiune ce a fost descrisă mai sus, cu excepția faptului că *IMSI Catcher-ul* va răspunde cu mesajul „serviciile LTE și non-LTE nu sunt permise”. Telefonul va intra într-o stare în care nu va mai realiza nici o conexiune cu rețeaua și nu va reveni la starea inițială decât în momentul în care va fi repornit (14).

[8] DoS – Denial of Service, tip de atac în care ținta nu mai poate accesa serviciul dorit. De obicei acest atac se realizează prin aglomerarea rețelelor.

3 SOLUȚII ȘI TEHNOLOGII FOLOSITE ÎN DETECTAREA UNEI STAȚII DE BAZĂ FALSE

Există numeroase soluții prin care se poate detecta existența unui dispozitiv malițios de tip IMSI Catcher însă nici una nu poate opri telefonul de la a se conecta la aceste antente false. Utilizatorul poate fi anunțat, poate primi rapoarte în legătură cu activitatea telefonului de la diferite aplicații dar din păcate nu poate fi ferit de atacatori. Acest tip de aplicații se bazează pe colecționarea de date de la mediul înconjurător și analizarea acestora pentru a lua o decizie în funcție de rezultatele obținute. Datele colecționate se obțin atât de la antenele ce aparțin furnizorilor de telefonie mobilă cât și de la presupuși atacator, acest lucru ajutând la compararea parametrilor între ei.

Așadar, în acest capitol se va discuta despre detectoare *IMSI Catcher*, despre tehnicile de detectare iar în final se va realiza și o evaluare a acestora pentru a stabili cât sunt de eficiente. Prima metodă, rudimentară, dar care poate fi folosită de oricine dispune de un telefon mobil pentru a detecta dacă s-a conectat sau nu la o antenă falsă este încercarea de a inița un apel. Deoarece antena nu este înregistrată într-o rețea de telefonie mobilă, această încercare nu va a avea succes. De asemenea, telefonul respectiv nu se va putea conecta la internet și nu va putea primi sau trimite mesaje de tip SMS.

Soluția prezentată poate fi încercată dar nu garantează rezultate corecte deoarece atacul nu durează pentru o perioadă lungă de timp. În cele mai multe cazuri telefonul se conectează la dispozitivul malițios, după care este respins de acesta în momentul în care a obținut datele dorite. Acest lucru înseamnă că verificarea trebuie facută atunci când telefonul își schimbă antena curentă pentru o antenă cu semnal mai bun.

După cum a fost menționat mai sus, prima metodă nu va oferi întodeauna rezultate clare dacă este utilizată individual și mai ales dacă este utilizată de persoane fără un bagaj de cunoștințe în acest domeniu al rețelelor de telefonie. Poate fi de folos în cadrul unor aplicații care utilizează și alte modalități, iar rezultatele fiecărui test în parte sunt analizate și vor rezulta într-un răspuns mai complex. Astfel, în subcapitolul ce urmează vor fi prezentate câteva metode existente de detecție ale dispozitivelor malițioase de tip *IMSI Catcher*.

3.1. Soluții existente de detectare

Acest subcapitol al prezentei lucrări se va axa pe diferite metode de detecție ale dispozitivelor *IMSI Catcher*, încercând să evalueze punctele tari și punctele slabe ale acestora. Astfel, detectoarele care urmează a fi descrise pot fi împărțite în trei mari categorii, după cum urmează (15):

- detectoare ce folosesc aplicații mobile
- detectoare ce folosesc senzori
- detectoare ce utilizează rețeaua de telefonie

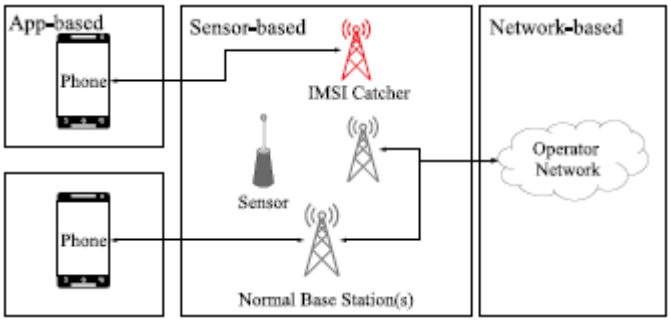


Fig. 3.1.a Cele trei metode de detecție prezentate (15)

În continuare, se va lua fiecare metodă în parte pentru a fi analizată (15).

Detectoare ce folosesc aplicații mobile

Acest tip de aplicații pot fi regăsite pe diferite platforme cum ar fi *Google Play*[9] sau *AppStore*[10] sub formă gratuită sau cu bani. Acestea utilizează și sunt limitate în acest sens de către bibliotecile, referitoare la rețeaua de telefonie mobilă, puse la dispoziție de sistemele de operare și se bazează doar pe informațiile pe care smartphone-ul le vede în rețea și le pune la dispoziția acestora. Putem afirma, de asemenea, că sunt limitate și de dispotivul fizic. Pe lângă aceste informații, pentru o detecție îmbunătățită, pot utiliza și baze de date publice ce conțin locațiile stațiilor de bază, cum ar fi *OpenCellId*.

Modul de funcționare al acestor aplicații se bazează pe faptul că deși dispozitivele *IMSI Catcher* pot fi configurate astfel încât să fie cât mai asemănătoare cu o stație de bază reală, anumiți parametri nu pot fi aceeași, tocmai pentru a induce telefonului că trebuie să se conecteze la acestea. Astfel, acest tip de

[9] Google Play – magazinul de aplicații utilizat de telefoanele ce folosesc sistemul de operare Android
[10] AppStore – magazinul de aplicații utilizat de telefoanele ce folosesc sistemul de operare iOS

detectoare salvează într-o bază de date proprie configurația fiecărei celule cu care interacționează telefonul mobil, pentru ca mai apoi să avertizeze utilizatorul dacă anumiți parametri nu se potrivesc sau dacă există prea multe diferențe între celula curentă și cele vecine. Utilizatorul dispozitivului mobil, va putea primi notificări în timp real, la scurt timp după ce telefonul acestuia s-a conectat la o stație de bază falsă.

Pentru utilizatorii ce doresc și mai multă intimitate în rețea, există un dispozitiv mobil, numit *Cryptophone*, care conform descrierii produsului garantează apeluri telefonice și mesaje securizate și mai mult, vine cu o aplicație de tip *Imsi Catcher Detector* deja instală. Potrivit unor evaluări (15), utilizarea unui astfel de telefon pentru detecție a dat aceleași rezultate, sau chiar mai bune decât în cazul utilizării unei aplicații gratuite. Acest lucru se datorează faptului că un *Cryptophone* utilizează anumiți parametri care nu sunt disponibili aplicațiilor obișnuite și mai mult în caz de detecții acesta poate trimite rezultatele către sistemul *GSKM Overwatch*[11] pentru o analiză amănunțită.

Așadar, luând în considerare cele prezentate mai sus, se poate afirma că unul din avantajele folosirii unei astfel de aplicații este acela că poate informa utilizatorul în cel mai scurt timp că a fost victima unui atac. Pe de altă parte, dezavantajele sunt reprezentate de limitările telefonului referitoare la detaliile pe care le obține de la antena la care se conectează, cum ar fi locația exactă a acesteia, pe care o poate afla numai conectându-se la surse externe, adică baza de date publică menționată anterior. Mai mult, nu este de neglijat nici faptul că o astfel de aplicație este o mare consumatoare de baterie, lucru care nu este tocmai plăcut pentru utilizatorul obișnuit.

Detectoare ce folosesc senzori

Detectoarele bazate pe senzori, sunt ori dispozitive mobile pasive, ori un senzor integrat într-o stație de bază care monitorizează în continuu stațiile de bază din împrejurimi. Detecția continuă, care acoperă o zonă mare, este de dată de faptul că aceste dispozitive sunt fixe și monitorizează în același timp rețelele, urmărind diverși factori cum ar fi apariții ale unor noi stații de bază, durata de viață a acestora sau diverse anomalii de configurare.

Unii senzori funcționează activ în rețea, devenind *honeypot-uri*[12]. Astfel, modul de funcționare al acestori senzori se bazează pe o conexiune cu stația de bază, pentru ca mai apoi să poată analiza toate mesajele pe care antena le schimbă

[11] GSMK Overwatch – sistem pentru detecția, localizarea și neutralizarea stațiilor mobile malițioase (24)
[12] Honeypot – mecanism de securitate a care încearcă să detecteze atacurile asupra sistemelor informatice

cu terminalul mobil. Acesta poate detecta lipsa serviciilor de telefonie mobilă, despre care am discutat la începutul capitolului, sau dacă schimbul de mesaje se realizează într-un mod criptat, lucru care nu se întâmplă în cazul *IMSI Catcher-urilor*. Rezultatele analizei sunt transmise mai apoi către operatorul de rețea.

Principalele avantaje ale aceste tehnici de detecție sunt:

- dimensiunea antenei, ce ajută la observarea unor zone mai largi decât în cazul telefoanelor mobile;
- se poate realiza o analiză pe o durată mai lungă de timp și astfel rezultatele vor fi mai concludente;
- se axează numai pe detecție, neavând procese ce rulează în același timp, care pot afecta capacitatea de analiză a mesajelor schimbate cu stația de bază.

Durata de timp îndelungată pe parcursul căreia se realizează analiza poate reprezenta de asemenea un dezavantaj deoarece nu poate împiedica atacurile unui *IMSI Catcher*, care își va fi terminat atacul cu mult timp înainte să fie detectat.

Detectoare ce utilizează rețeaua de telefonie

Acest tip de detectoare utilizează, după cum îi spune și numele, informațiile colectate de rețea, neavând nevoie să realizeze măsurători suplimentare pentru detecția *IMSI Catcher-urilor*.

Informațiile pe care rețeaua de telefonie mobilă se bazează pentru a detecta dispozitivele malițioase sunt numeroase, dar un exemplu ar fi puterea semnalului care poate fi preluată de la telefon și comparată cu cea pe care celula o emite, astfel identificând prezența și locația unui *IMSI Catcher*. Mai mult, având drepturi depline asupra rețelei, administratorii pot interzice telefoanelor să se conecteze la dispozitivele utilizate pentru atacuri.

Așadar, informațiile actualizate pe care operatorul de rețea le obține atât de la terminalele mobile , cât și de la antenele sale reprezintă un mare avantaj în detecția atacurilor. Cu toate acestea, unul dintre dezavantaje constă în faptul că identificarea dispozitivului se realizează prea târziu, la fel ca în cazul detectoarelor ce utilizează senzori, abia după ce telefonul mobil a fost deja ținta unui atac. Mai mult, operatorul de rețea nu poate stabili nu poate stabili cât de grave au fost consecințele atacului, adică dacă atacatorul a dezactivat criptarea datelor pentru interceptarea informațiilor.

O modalitate prin care aceste metode de decție ar da rezultate mult mai concludente, ar fi utilizarea unei combinații ale acestora (15). Un bun exemplu ar fi

utilizarea unui senzor împreună cu o aplicație deoarece primul poate da informații la care al doilea tip de detector nu are acces, cum ar fi momentul în care rețeaua de telefonie decide să declaseze nivelul rețelei. Dacă senzorul ar detecta o ca deși există o conexiune 4G într-un loc dar cu toate acestea celula la care este conectat declasează nivelul de la 4G la 2G, acesta este un semn ca un *IMSI Catcher* se află în preajmă, informație care ar putea fi trimisă catre aplicația mobilă. Problema care stă la baza acestei soluții este faptul că, momentan, senzorii utilizează rețeaua de internet pentru a putea transmite informațiile referitoare la dispozitivele malițioase. Aplicația mobilă nu poate primi nici un mesaj în timp real atunci când atacul se află în desfășurare. Soluția care poate rezolva problema este utilizarea și celei de a treia metode, cu ajutorul căreia se poate interzice telefonului să părăsească anumite celule pentru unele cu semnal mult mai puternic.

3.2. Prezentarea bazei de date OpenCellId

Deoarece aplicația dezvoltată în cadrul acestui proiect, se bazează efectiv pe datele obținute din telefon, a fost necesară utilizarea unei baze de date publice pentru obținerea informațiilor veridice despre celule.

Acestă bază de date conținea (16) în 21 august 2017 în jur de 35.5 milioane de celule unice, ceea ce o face potrivită pentru verificarea informațiilor pe care antena la care se conectează terminalul mobil le face publice. Este de înțeles că unele celule pot lipsi și de aceea analiza realizată la nivelul aplicației nu se bazează doar pe aceste date, adică, în cazul în care datele unei stații de bază nu se găsesc, aplicația nu va eticheta acea stație ca fiind un *IMSI Catcher* ci o va cataloga ca activitate suspicioasă, până stabilirea unei decizii finale.

Conectarea și obținerea datelor necesare analizei se realizează cu ajutorul unui api pus la dispoziție de firma ce a creat baza de date. Astfel, pentru a trimite cereri se poate utiliza unul din endpointurile listate în documentația api-ului. Programatorul îl poate folosi pe oricare, însă recomandarea este să îl folosească pe acela care este cel mai apropiat din punct de vedere geografic pentru a minimiza latența. În cadrul proiectului am utilizat serverul de mai jos:

- https://eu1.unwiredlabs.com/v2/process.php

Tot în documentația api-ului este prezentată modalitatea în care trebuie realizată cererea și cum va arăta răspunsul primit de la server.

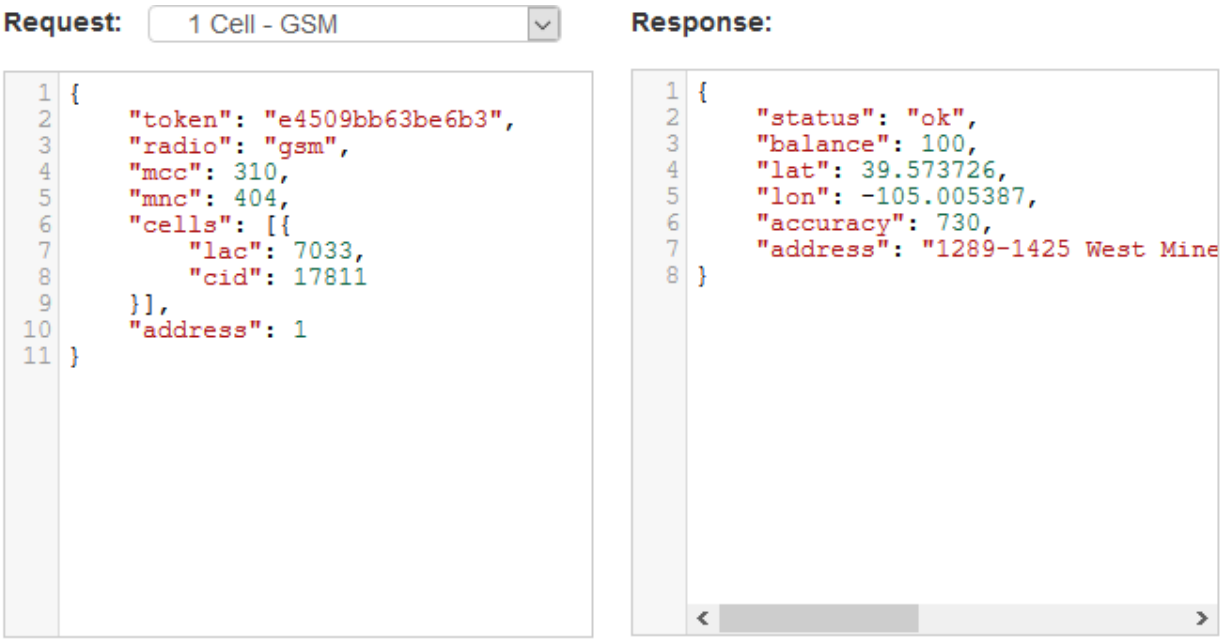


Fig. 3.2.a Cerere și răspuns api OpenCellId

În continuare vor fi detaliați fiecare parametri pe care aplicația trebuie să îi pună la dispoziție pentru a putea primi un răspuns pozitiv înapoi.

- token – cheia pentru a putea accesa api-ul ce se poate obține de pe site-ul producătorului
- radio – tipul celulei pe care dorim să o căutam. În cazul de față nu vom utiliza decât GSM
- mcc – codul țării în care este înregistrată cartela SIM
- mnc – codul operatorului de telefonie mobilă din țara în care este înregistrată cartela SIM
- lac – codul locației în care se află celula căutată
- cid – codul de identificare a celulei

Ca răspuns api-ul întoarce statusul cererii, *ok* însemnând că cererea a fost tratată cu succes. În caz contrar, mesajul *error* va apărea la câmpul *status*. Mai mult, răspunsul conține numărul de cereri care se mai pot trimite către api iar în cazul în care statusul este unul negativ, acestea nu se iau în considerare. În continuare, sunt furnizate clientului detalii despre locația celulei, cum ar fi latitudinea și longitudinea precum și adresa exactă.

Toate datele primite de la server sunt preluate de aplicația client și afișate utilizatorului, iar dacă celula nu există acesta va fi înștiințat de faptul că în zona sa

poate exista un dispozitiv malițios de tip *IMSI Catcher*, rezultatul final urmând să fie stabilit prin efectuarea tuturor testelor din cadrul aplicației.

În cadrul aplicației dezvoltate a fost utilizat un *HttpRequest*, metoda *Post*, care a primit ca parametru un *Json* în care au fost introduse toate datele cerute de către api. Răspunsul primit, a fost tot sub forma unui obiect Json și a fost parsat tot în cadrul clasei ce realizează cererea pentru a putea salva informațiile, ce urmează a fi folosite în aplicație pentru analiză sau pentru informarea utilizatorului.

```
URL urlToPost = new URL(url);
HttpURLConnection connection = (HttpURLConnection) urlToPost.openConnection();
connection.setRequestMethod("POST");
connection.setRequestProperty("Authorization", "Bearer " + token);
connection.setRequestProperty("content-type", "application/json");
connection.setDoOutput(true);
connection.setDoInput(true);

JsonObject cellObject = new JsonObject();
cellObject.addProperty( property: "lac", lac);
cellObject.addProperty( property: "cid", cid);

JsonObject jsonObject = new JsonObject();
jsonObject.addProperty( property: "token", token);
jsonObject.addProperty( property: "radio", radio);
jsonObject.addProperty( property: "mcc", mcc);
jsonObject.addProperty( property: "mnc", mnc);
jsonObject.add( property: "cells", cellObject);
jsonObject.addProperty( property: "address", address);
```

Fig. 3.2.b Pornirea conexiunii http și crearea obiectului Json

Toate apelurile metodelor de mai sus se realizează în mod asincron pentru a nu interfera cu restul operațiilor ce se execută în cadrul aplicației. Pe lângă faptul că mediul de lucru nu permite executarea sarcinilor de acest tip pe firul principal de execuție, a fost nevoie de o astfel de abordare deoarece se execută un număr mare de cereri de care utilizatorul nu este conștient, fapt care ar fi însemnat blocarea aplicației.

4 DESCRIEREA APLICAȚIEI DEMONSTRATIVE

Prezentul capitol al acestei lucrări își propune descrierea aplicației ce pune în practică o parte din noțiunile teoretice prezentate anterior în acest document. Astfel, se vor descrie mediul de lucru utilizat pentru dezvoltarea aplicației informatice, tehnicile de programare folosite, câteva principii care au fost respectate în ceea ce privește design-ul aplicației și pașii urmați în analiza datelor pe care celula la care telefonul se conectează le furnizează.

Capitolul se va încheia cu prezentarea modului de funcționare al aplicației, simulând detectarea unei stații de bază false. Rezultatele simulării vor fi expuse sub forma unor figuri și comentarii pe baza acestora care să ateste faptul că modul de funcționare al programului este unul corespunzător.

4.1. Prezentarea arhitecturii și descrierea mediului de lucru

4.1.1. Descrierea mediului de lucru

Aplicația informatică, numită sugestiv *Imsi Catcher Detector*, a fost dezvoltă cu ajutorul unui mediu de dezvoltare, numit Android Studio, pus la dispoție de Google. Acest *IDE*[13] este specializat pe dezvoltarea de aplicații pentru sistemul de operare Android și este utilizat încă din anul 2013, atunci când a fost anunțat de Google într-o conferință (17).

Android Studio îmbină editarea vizuală bazată pe fișiere xml cu limbajele de programare Java sau Kotlin. Mai mult, acesta pune la dispoziția programatorilor un emulator pe care aceștia pot rula sau depana aplicațiile, nefiind nevoie neapărat de un dispozitiv fizic.

Astfel, interfața grafică a programului dezvoltat în cadrul acestui proiect a fost realizată folosind editorul vizual menționat anterior, cu ajutorul cărora au fost adăgate componente grafice mai mult sau mai puțin complicate cum ar fi: texte, casete de text, tabele, hărți, liste etc. Parametrii fiecărei componentă în parte au

[13] IDE – integrated development environment – mediu de dezvoltare pentru diferite aplicații informatice.

fost modificați cu ajutorul editorului vizual, a fișierelor XML aferente fiecărui ecran din aplicație sau prin limbajul Java, întrucât fiecare componentă grafică are corespondent un obiect Java. Acest obiect, permite programatorului să modifice programatic felul în care componenta vizuală arată, locația acesteia din ecran precum și să definească comportamentul acesteia în funcție de anumite acțiuni ale utilizatorului. Mai mult, editorul vizual pune la dispoziție previzualizarea interfeței grafice a aplicației dar și modificarea dimensiunilor acesteia pentru a vedea cum se comportă pe ecrane diferite.

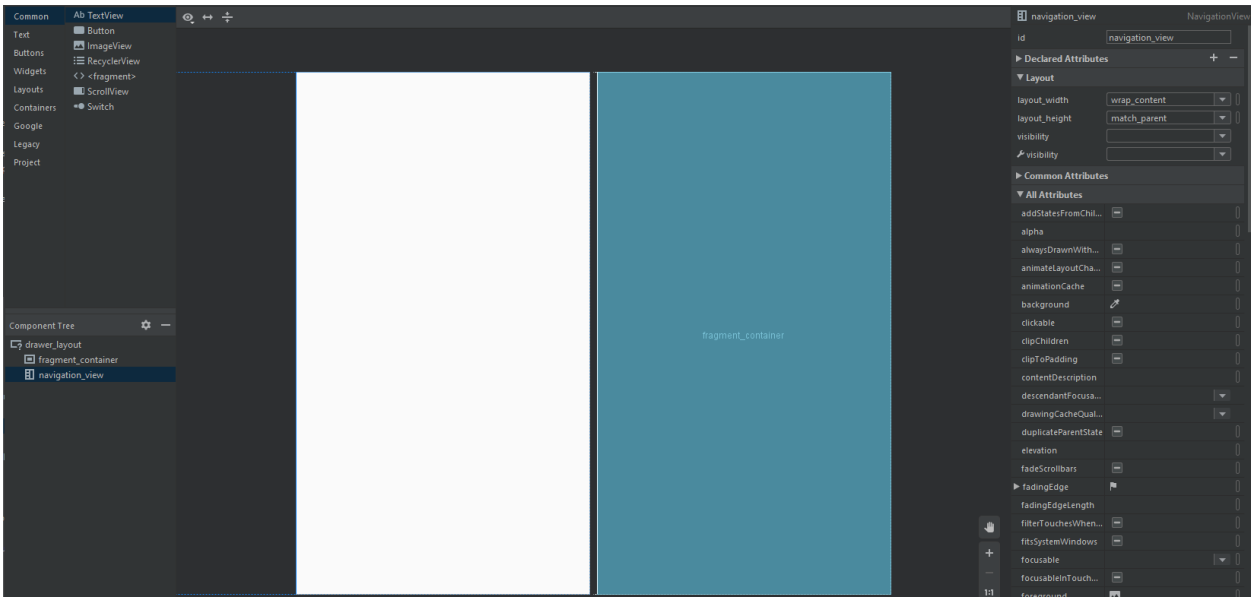


Fig. 4.1.1.a Editorul vizual din Android Studio

Pe lângă dezvoltarea aplicației în Android Studio a fost nevoie de un server ce conține o bază de date care să fie accesibilă tuturor dispozitivelor pe care aplicația rulează. Astfel, a fost utilizat serviciul celor de la Google numit Firebase ce pune la dispoziție un api cu ajutorul căruia datele pot fi sincronizate între toți utilizatorii aplicației. Modul de funcționare al acestei baze de date se bazează pe stocarea datelor sub forma unor colecții de date imbricate, pe care aplicațiile, fie ele mobile sau web, le pot accesa în mod asincron fără a bloca firul principal de execuție cât timp se realizează transferul de date.

Astfel, datele au fost organizate sub forma a 3 colecții denumite în funcție de tipul celulelor detectate:

- GoodCells – celule ce aparțin operatorilor de telefonie mobilă
- WarningCells – celule despre care încă nu se cunoaște dacă sunt stații de bază sau dispozitive malițioase
- AlertCells – dispozitive de tip *IMSI Catcher*

Fiecare intrare din colecție este denumită în funcție de id-ul celulei, asigurând așadar unicitatea intrărilor din baza de date. Fiecare document din colecție conține informații precum id, local area code, latitudine, longitudine etc.

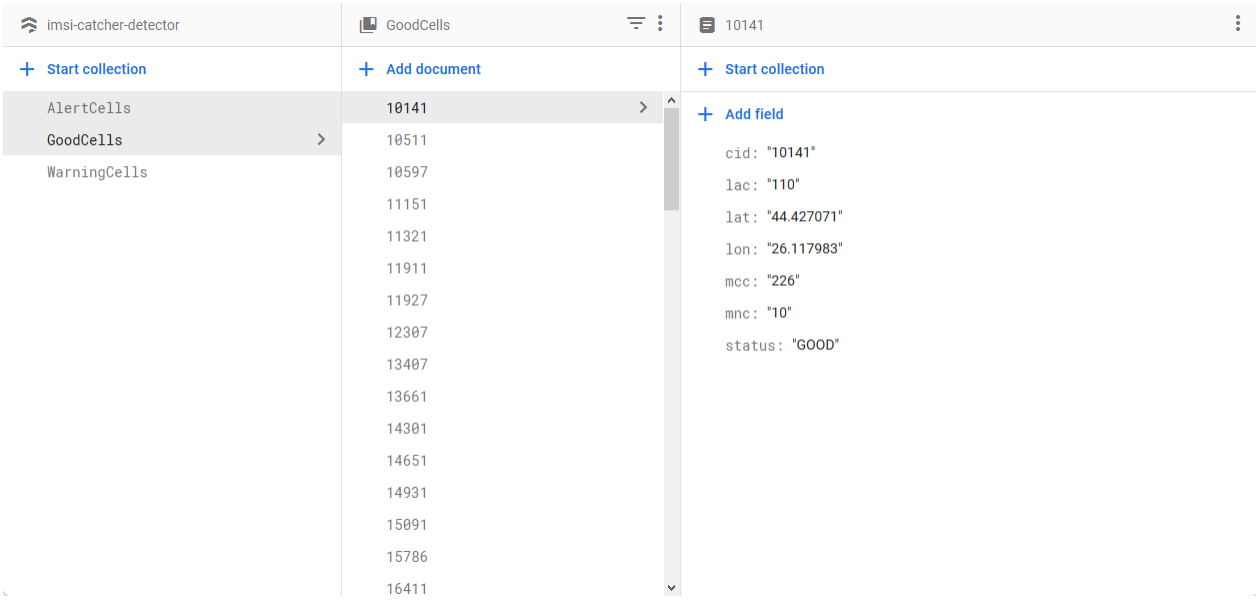


Fig. 4.1.1.b Intrările din baza de date

Baza de date este actualizată numai în cadrul aplicației în diferite momente de timp. Un prim caz în care se modifică este atunci când telefonul schimbă celula la care este conectat, moment în care stația de bază împreună cu toate celulele vecine declarate de aceasta sunt introduse în colecția corespunzătoare. Dacă aplicația are o suspiciune că antena ar putea fi un *IMSI Catcher*, datorată de faptul că nu există în baza de date publică, înainte de a face modificări în baza de date, aceasta verifică existența celulei respective în colecția *GoodCells*. Un alt caz de actualizare a datelor este atunci când după efectuarea analizei a rezultat că celula la care este conectat telefonul este un dispozitiv malițios, moment în care datele acelei celule vor fi introduse în colecția *AlertCells*.

Toate apelurile către această bază de date se realizează asincron pentru a nu interfera cu firul principal al aplicației. Se utilizează referințe pentru fiecare colecție și document din baza de date iar pentru fiecare apel există un *listener* care să specifice dacă execuția s-a terminat cu succes sau nu. Întrucât nu este o bază de date obișnuită, în care se pot realiza diferite operații, cum ar fi operații de tip *join*, toate verificările trebuie să devină imbricate. Acest lucru derivă tot din faptul că operațiile se realizează asincron, întrucât trebuie așteptate rezultatele primite de la un apel, pentru a-l putea porni pe următorul.

Spre exemplu, pentru a introduce o celulă în baza de date trebuie verificate mai întâi anumite colecții. Dacă celula a fost marcată ca fiind suspicioasă dar în urma testelor s-a stabilit că este reală, atunci trebuie verificată colecția *WarningCells*. Dacă a fost găsită în acea colecție trebuie mai întâi ștearsă de acolo și abia apoi introdusă în colecția *GoodCells*. La fel și în cazul introducerii unei celule suspecte în baza de date, trebuie verificat dacă celula nu există deja în *GoodCells* respectiv *AlertCells* pentru a nu exista duplicate. Toate aceste operații se realizează imbricat, după cum a fost prezentat anterior.

```
DocumentReference cellReference = mDatabaseReference.collection( collectionPath: "GoodCells").document(cell.GetCid());
cellReference.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
    @Override
    public void onComplete(@NonNull Task<DocumentSnapshot> task) {
        if(task.getResult().exists() && task.isSuccessful()) {
            mDatabaseResponse = MConstants.FirebaseHelper.EXISTS_IN_GOOD;
            internalDatabaseCallBack.OnReturnResponseCallback(mDatabaseResponse);
        } else {
```

Fig. 4.1.1.c Exemplu de verificare a existenței unei celule în baza de date

4.1.2. Prezentarea arhitecturii proiectului

Arhitectura proiectului se bazează pe modelul MVC, Model-View-Controller, tradus în mod sugestiv model-vizualizare-controlor. Acesta a fost utilizat pentru a separa partea vizuală de ceea ce se întâmplă în spate, ceea ce utilizatorul nu vede în mod direct. În continuare, se va detalia fiecare componentă în parte, pentru a oferi o înțelegere mai bună asupra modului de funcționanre al acestui model (18):

Modelul – încapsulează datele specifice aplicației și definește logica de calcul pentru acele date.

Vizualizare – reprezintă tot ceea ce utilizatorul vede. De asemenea definește modul în care se comportă atunci când acesta interacționează cu aplicația.

Controlorul – reprezintă un obiect intermediar între partea vizuală a aplicației și între modelele utilizate în cadrul acesteia.

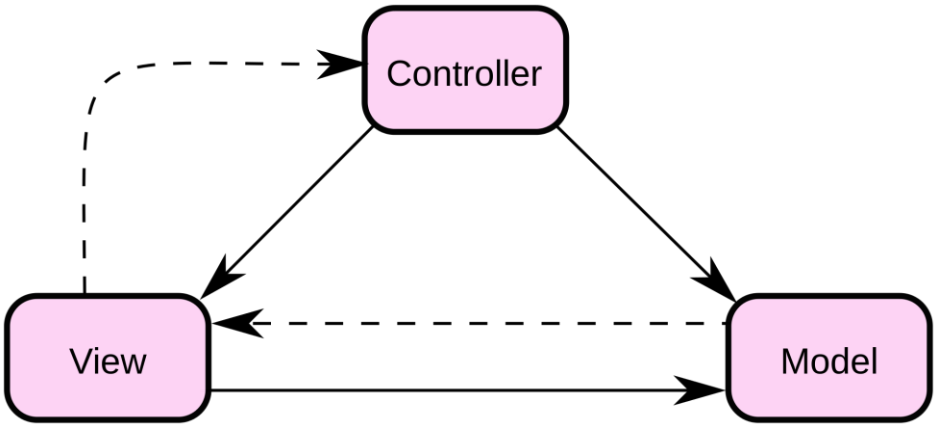


Fig. 4.1.2.a Modul în care interacționează componentele între ele (19)

Așadar, acest subcapitol își propune prezentarea modalității cu ajutorul căreia au fost structurate obiectele utilizate în cadrul aplicației. Va fi descrisă, de asemenea, interacțiunea dintre acestea urmând ca în final să se expună, atât textual cât și cu ajutorul figurilor, modul în care se realizează interceptarea dispozitivelor malițioase de tip *IMSI Catcher*.

Modelele principale utilizate în cadrul aplicației sunt de tipul *Cell, Device, Network* și SIM. Acestea au fost folosite pentru a putea stoca date care mai târziu au ajutat în cadrul analizelor efectuate, pentru expunerea anumitor detalii către utilizator sau pentru stocarea lor mai departe în baza de date corespunzătoare aplicației. Pentru a putea obține informațiile de care a fost nevoie, fiecare model are câte un *controller* aferent care se ocupă de trimiterea de cereri către api-ul corespunzător bazei de date publice și accesarea bibliotecilor necesare pentru a descărca informațiile pe care celula/ rețeaua/ telefonul le face publice programatorului.

Astfel, în controller-ul aferent modelului *Cell* se inițializează obiectul *TelephonyManager* care se ocupă de toate evenimentele legate de partea de rețea. Pentru a avea acces la datele stocate de telefonul mobil, se apelează metoda *getAllCellInfo,* din cadrul obiectului de mai sus, care întoarce o listă cu detalii despre celula curentă. Ne interesează din această listă de celule să extragem doar informații despre rețeaua GSM, așa că se va face un cast la *CellInfoGsm*. Odată obținut obiectul ce conține doar detalii despre 2G, se extrag alte 2 obiecte, unul referitor la identitatea celulei și unul referitor la puterea semnalului. Acestea prezentate mai sus se inițializează în cadrul constructorului controller-ului, urmând să fie întoarse către model, toate datele necesare. Clasa *TelephonyManager* se utilizează și în cazul controllere-lor pentru dispozitiv, rețea respectiv cartelă sim, însă sunt extrase date direct din obiectul *TelephonyManager*, fără a fi nevoie de

crearea unor alte obiecte intermediare, cum ar fi cele pentru celulă, prezentate mai sus.

Plecând de la datele de bază ale aplicației, fără de care nu ar putea funcționa corespunzător, descrierea elementelor continuă cu *listener-urile* utilizate în cadrul acesteia. Aceste obiecte conțin metode asincrone care „ascultă" schimbările apărute, în funcție de specializarea fiecăruia. Cel mai important din aplicație este cel care detectează schimbarea celulei telefonului, deoarece acesta este un moment prielnic pentru începerea verificării celulei. Pentru acest *listener* este utilizată tot clasa *TelephonyManger,* însă trebuie suprascrisă metoda ce se apelează în momentul în care se detectează schimbarea celulei curente. În cazul acestei aplicații, în această metodă s-a realizat introducerea celulelor în baza de date și verificarea propriu-zisă a celulei. Un al doilea astfel de obiect verifică locația telefonului folosindu-se de date primite de la senzorul de GPS al telefonului mobil pe care este instalată aplicația. Importanța obiectului ce detectează locația constă în faptul că în momentul în care programul are suspiciuni în ceea ce privește stația de bază la care este conectat dizpotivul mobil, atunci acesta va indica pe hartă utilizatorului locația actuală, anunțându-l că în apropierea sa poate exista un *IMSI Catcher*. Pentru a realiza această detecție, listener-ul pentru locație a trebuit să implementeze interfața *LocationListener* și să utilizeze un obiect de tipul *LocationManager* care întoarce locația dispozitivului. Pentru a putea accesa locația, a fost nevoie ca aplicația să ceară permisiunea utilizatorului în momentul în care aceasta se deschide pentru prima oară.

Obiectul ce se ocupă de pornirea testelor este denumit sugestiv *CheckerManager*. Acesta va porni fiecare verificator în parte și va centraliza răspunsurile venite pentru a putea executa testul final în care se calculează un scor pe baza căruia se va decide autenticitatea celulei. Există două metode în care manager-ul de verificatoare poate funcționa, și anume verificarea celulei curente sau verificarea unei anumite celule. Dacă prima se referă strict la momentul în care utilizatorul pornește verificările, cea de a doua variantă a fost implementată pentru momentul în care telefonul detectează o celulă falsă iar utilizatorul dorește să vadă în continuare ce rezultate au dus la această decizie. Testele realizate în cadrul aplicației sunt în număr de șase, după cum urmează:

- **Testarea consistenței celulei –** acest verificator va primi ca parametru obiectul de tip *Cell,* ce reprezintă antena la care telefonul este conectat în momentul pornirii testelor și va verifica dacă datele pe care aceasta le furnizează, corespund cu datele pe care aplicația le deține în baza de date online, despre care am discutat în subcapitolul anterior. Rezultatele primite în cadrul unui callback din baza de date sunt parcurse până în momentul în

care este găsită celula căutată. În acest moment, se compară toți parametrii celulei curente cu cei primiți de la baza de date, cum ar fi LAC, MCC sau MNC. În cazul în care nu este găsită celula, atunci răspunsul se va baza strict pe răspunsul primit de la baza de date publică și, în mod evident, dacă nu este găsită nici acolo, rezultatul testului va fi unul negativ. Codul utilizat pentru realizarea acestui test este prezentat mai jos:

```
public void CheckCellConsistency(Cell currentCell,final
InternalDatabaseCallBack internalDatabaseCallBack) {
    this.mFirebaseHelper.getAllCells(new DatabaseReaderCallBack() {
        @Override
        public void OnCallBack(List<Cell> databaseCellList) {
            for(Cell cell : databaseCellList) {
                if(cell.GetCid().equals(currentCell.GetCid())){

if(cell.GetLac().equals(currentCell.GetLac())&&cell.GetMcc().equals(currentCel
l.GetMcc())&&cell.GetMnc().equals(currentCell.GetMnc())) {

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_PASSED_RO);
                        return;
                    } else {

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_FAILED_RO);
                        return;
                    }

                }
            }
            if(currentCell.getmCellStatus().equals(MConstants.Cell.WARNING)) {

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_FAILED_RO);
                return;
            } else
if(currentCell.getmCellStatus().equals(MConstants.Cell.GOOD)){

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_PASSED_RO);
            }
        }
    },"GoodCells");
}
```

- **Testarea puterii semnalului** – în momentul în care telefonul se conectează la o nouă celulă deoarece deține o putere a semnalului mai mare, aplicația înregistrează in baza de date locală de pe dispozitiv valoarea puterii, dar numai în cazul în care celula se găsește în baza de date publică. Odată cu începerea testului, toate aceste valori sunt aduse din baza de date, se realizează o medie a acestora și se verifică dacă puterea semnalului celulei curente se află între anumite intervale – media plus/ minus o marjă de eroare. Codul aferent se poate vedea mai jos:

```
public CheckerResponse checkSignalStrength(Cell currentCell) {
    float currentTimeMilis = System.currentTimeMillis();


    mLastMovementTime = mSharedPreferences.getLong("lastMovementTime",
mLastMovementTime);
    if (currentTimeMilis - mLastMovementTime >= MAXIMUM_SAFE_PERIOD) {
        mDatabase.OpenConnection();
        ArrayList<Integer> signalValues =
this.mDatabase.getSignalValues(Integer.parseInt(currentCell.GetCid()));
        mDatabase.CloseConnection();
        Double signalAverage = signalValues.stream().mapToInt(val ->
val).average().orElse(0.0);
        int currentSignalStrenght =
Integer.parseInt(currentCell.GetSignalDbm());
        if(currentSignalStrenght > signalAverage + 8)
            return new CheckerResponse(MConstants.TEST_FAILED_RO);
        if(currentSignalStrenght < signalAverage - 8)
            return new CheckerResponse(MConstants.TEST_FAILED_RO);
    }
    return new CheckerResponse(MConstants.TEST_PASSED_RO);
}
```

- **Testarea conectivității** – odată cu începerea acestui test, aplicația pornește datele dispozitivului pentru a trimite un ping către serverul Google. Dacă este primit un răspuns înseamnă că telefonul are conexiune la internet și testul a trecut. Cazul contrar, poate însemna că telefonul nu se poate conecta din cauze ce țin de operatorul de servicii sau că telefonul s-a conectat la un dispozitiv malițios. În ambele cazuri, picarea acestui test aduce de la sine semne de întrebare și astfel un scor negativ în cadrul testării finale. Codul sursă este prezentat mai jos:

```
public boolean isOnline() {
    try {
        Log.i("CONNECTIVITY:","DEVICE IS ONLINE");
        return (Runtime.getRuntime().exec("ping -c 1 -w 1
google.com").waitFor() == 0);
    } catch (IOException e) {
        Log.i("CONNECTIVITY:","DEVICE IS OFFLINE");
        System.out.println(e);
        return false;
    } catch (InterruptedException e) {
        Log.i("CONNECTIVITY:","DEVICE IS OFFLINE");
        e.printStackTrace();
        return false;
    }
}
```

- **Verificarea bazei de date Firebase** – se verifică dacă telefonul a mai interacționat cu această celulă, ce oferă aceste date și sub ce formă a fost

salvată. Practic, căutarea în acest caz se face numai după id-ul celulei urmând ca restul datelor să fi verificate de către testul de consitență. Va fi un rezultat bun pentru testare faptul că celula a fost găsită în baza de date, însă acesta poate fi oricând anulat de rezultatul primit de la verificarea consistenței. Mai jos este prezentat codul sursă:

```
public void checkInternalDatabase(Cell currentCell,final
InternalDatabaseCallBack internalDatabaseCallBack) {
    mFirebaseHelper.checkDatabaseEntry(currentCell, new
InternalDatabaseCallBack() {
        @Override
        public void OnReturnResponseCallback(String response) {
            switch (response) {
                case MConstants.FirebaseHelper.EXISTS_IN_GOOD:

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_PASSED_RO);
                    break;
                case MConstants.FirebaseHelper.EXISTS_IN_WARNING:

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_NEUTRAL_RO);
                    break;
                case MConstants.FirebaseHelper.EXISTS_IN_ALERT:

internalDatabaseCallBack.OnReturnResponseCallback(MConstants.TEST_FAILED_RO);
                    break;
            }
        }
    });
}
```

- **Verificarea bazei de date publice –** este verificată baza de date publică prin trimiterea datelor pe care aceasta le cere. În cazul unui *IMSI Catcher* datele vor fi diferite față de cele salvate în mod public și astfel răspunsul întors va fi unul negativ. Codul sursă este cel de mai jos:

```
public PublicDBCheckerResponse checkPublicDB(Cell currentCell){
    String cellStatus = currentCell.getmCellStatus();
    if(cellStatus.equals(MConstants.Cell.GOOD))
        return new PublicDBCheckerResponse(MConstants.TEST_PASSED_RO);
    return new PublicDBCheckerResponse(MConstants.TEST_FAILED_RO);
}
```

- **Verificarea numărului de vecini –** aplicația folosește bibliotecile puse la dispoziție programatorilor pentru a verifica dacă celula la care este conectat telefonul pune la dispoziție lista cu vecini. Dacă aceasta este nulă, atunci testul va primi un scor negativ în cazul testării finale. Se verifică mai întâi dacă modelul telefonului nu este Samsung, deoarece aceste telefoane nu oferă întotdeauna acces la aceste informații, fapt care ar influența într-un

mod negativ rezultatele testării. Mai jos este prezentat codul sursă al acestei testări:

```
public NeighbourListCheckerResponse CheckNeighbourList(Cell currentCell){
    String manufacturer = this.mCurrentDevice.GetManufacturer();
    if(manufacturer.equals(MConstants.SAMSUNG_PHONE_MODEL)){
        return new NeighbourListCheckerResponse(MConstants.TEST_NEUTRAL_RO);
    }
    List<Cell> neighbourList = currentCell.getmNeighbouringCells();
    if(neighbourList.size() != 0){
        return new NeighbourListCheckerResponse(MConstants.TEST_PASSED_RO);
    }
    return new NeighbourListCheckerResponse(MConstants.TEST_FAILED_RO);
}
```

După cum se poate observa în secvențele ce conțin codul sursă al fiecărui test în parte, prezentate mai sus, verificările se bazează și pe alți parametri extrași înainte de începererea verifării în sine. Spre exemplu, request-ul către baza de date publică este trimis în momentul în care este creată celula și de aceea în cazul acestui test nu este verificat decât statusul celulei respective. În cazul testării conectivității, metoda expusă este utilizată în cadrul altei metode care doar verifică ceea ce funcția *isOnline* întoarce și creează un răspuns pentru această testare.

Testarea finală va prelua toate răspunsurile de la *CheckerManager* ce vor fi centralizate cu ajutorul unui obiect de tip *CheckerResponseManager*.Acesta din urmă, va reține fiecare răspuns în parte pe care le va transmite verificatorului final, în momentul începerii testării. Trebuie avut în vedere faptul că verificatorul final trebuie să aștepte încetarea tuturor testelor efectuate pentru a putea începe verificarea. Odată cu primirea fiecărui răspuns, acestea se vor salva în variabile serparate iar testul final va pargurge fiecare variabilă în parte și va crea un scor pe baza căruia va decide în finalul testării dacă celula la care s-a conectat telefonul este un dispozitiv malițios. La ficare test trecut, se aduna 10 puncte la testul final pe când la fiecare test picat se scad cele 10 puncte. În cazul în care testul pică din motive ce nu țin de celula în sine ci de alți factori, atunci la nu se va adăuga sau scădea nimic. Scorul calculat poate fi pozitiv, ceea ce înseamnă ca testele au trecut, sau negativ, fapt din care rezultă o prevalență a testelor picate.

În continuare vor fi prezentate două diagrame, a cazurilor de utilizare și o schemă logică, ce fac referire strict la modul de operare al aplicației în cazul testelor. Acestea vor descrie grafic fiecare obiect despre care s-a discutat mai sus, prezentând atât legătura dintre ele cât și momentul de timp în care acestea vor fi utilizate în cadrul aplicației. Diagramele au fost realizate manual cu ajutorul unui site web specializat în crearea de astfel de scheme.

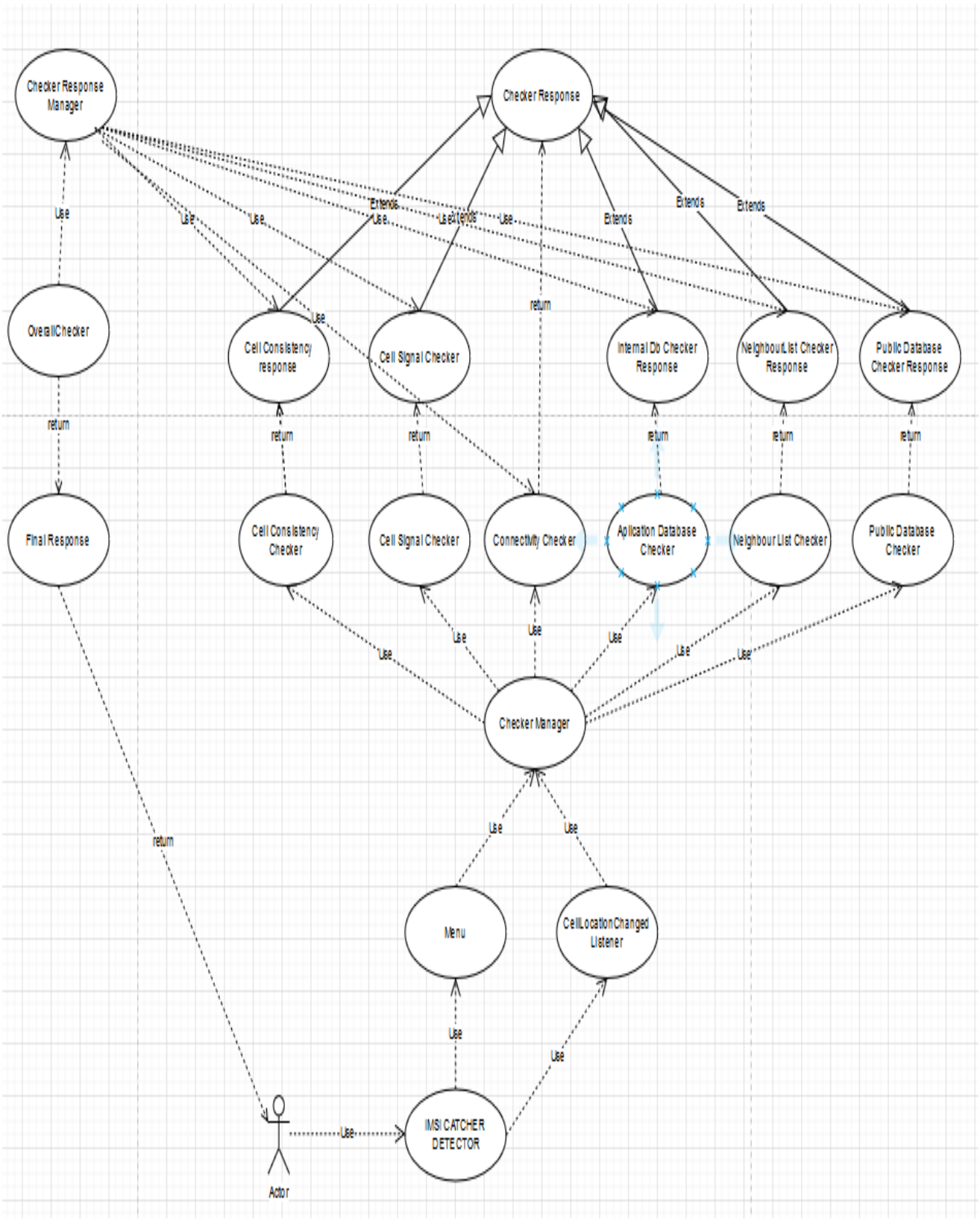


Fig. 4.1.2.b Diagrama cazurilor de utilizare

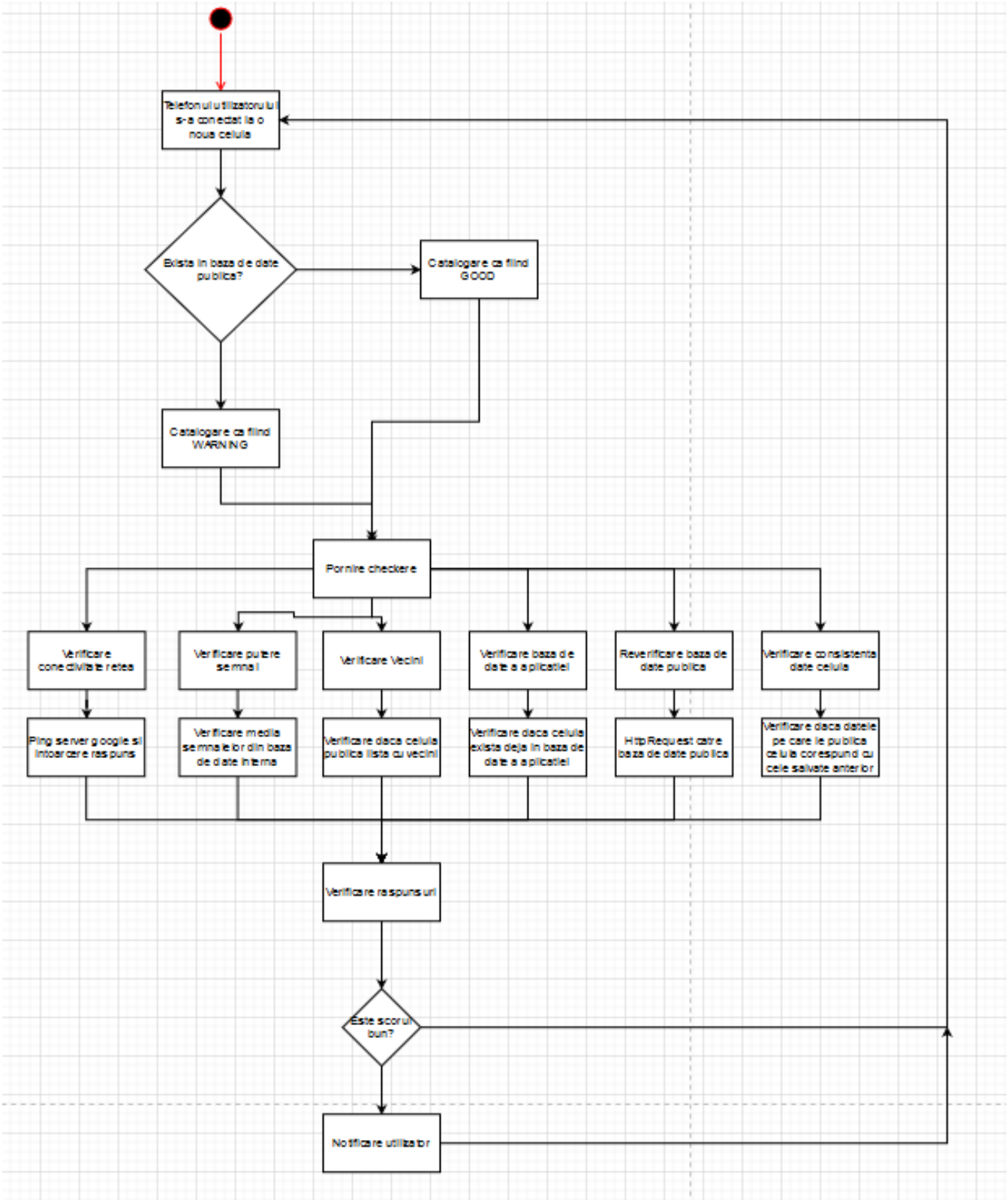


Fig. 4.1.2.c Diagramă ce prezintă efectuarea testelor

După cum se poate observa mai sus, în momentul în care utilizatorul deschide aplicația acesta are opțiunea de a porni testul prin intermediul unui meniu, pentru a realiza un test oricând dorește sau opțiunea de testare automată ce este realizată pe un fir secundar de execuție fără să îl deranjeze în utilizarea celorlalte opțiuni pe care aplicația i le pune la dispoziție. Mai mult, ciclul de testare se reia de fiecare dată când telefonul detectează schimbarea celulei pentru a putea verifica orice antenă care intră în contact cu dispozitivul mobil.

Pe lângă obiectele principale ale aplicației care iau parte la testarea propriu-zisă, au fost create și obiecte/ metode suplimentare ce ajută la o împărțire a sarcinilor corespunzătoare. În primul rând au fost utilizate interfețe de tip *CallBack* pentru extragerea datelor din metodele ce se executau asincron. În cazul în care nu se utilizează acest tip de obiect, folosit numai pentru preluarea datelor din funcțiile asincrone, a fost folosit un *Handler*, al cărui scop este de a produce o execuție întârziată a metodelor ce depind de datele obținute asincron, fără a afecta însă firul principal de execuție.

Pentru ambele baze de date ale aplicației, cea de pe Firebase și cea locală de pe dispozitiv, au fost utilizate obiecte de tip Adapter, ce se ocupă strict de interacțiunea cu baza de date, urmând să livreze datele obținute către obiectul care realizează cererea. Baza de date locală, de pe fiecare dispozitiv este structurată ca în figurile de mai jos.

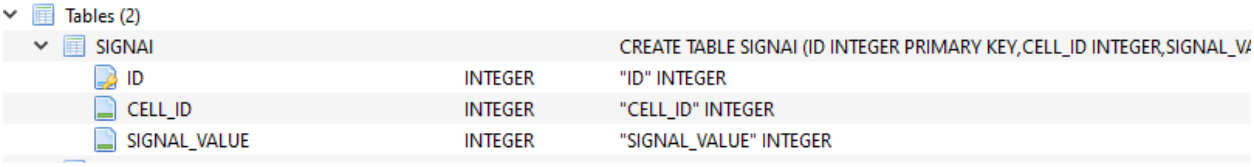


Fig. 4.1.2.d Structura bazei de date interne

ID	CELL_ID	SIGNAL_VALUE
Filter	Filter	Filter
1	31911	-57
2	31911	-57
3	31911	-57
4	31911	-51
5	31911	-53

Fig. 4.1.2.e Structura tabelului SIGNAL

După cum se poate observa datorită faptului că există două baze de date, cea de a doua, de pe dispozitivul mobil, nu conține decât valorile semnalelor pentru celulele cu care a interacționat telefonul mobil. Alegerea ca cea de a doua să nu fie găzduită pe un server online a fost făcută pe baza faptului că utilizatorii nu își folosesc telefoanele mobile din aceleași locații ceea ce duce la comportamente diferite ale celulelor, în ceea ce privește puterea semnalului pe care îl oferă.

Astfel, baza de date locală a fost creată cu ajutorul SQLite, o librărie care configurează un motor de baze date tranzacționale, ce nu au nevoie de nici o configurare și nici de un server pe care să fie găzduite (20). Scrierea se realizează direct pe disc, sub forma unor fișiere, care mai apoi pot fi extrase și vizualizate cu ajutorul altor programe. Codul sursă pentru crearea și utilizarea acestei baze de date este prezentat mai jos. După cum veți observa, se pot utiliza query-uri din limbajul SQL(fig. 4.1.2.f) sau există metode în cadrul limbajului Java care vor implementa query-urile(fig. 4.1.2.g), programatorul fiind nevoit să ofere doar valorile necesare.

```
public class IMSICatcherDetectorDatabase extends SQLiteOpenHelper {
    //region Constructor
    public IMSICatcherDetectorDatabase(Context context) {
        super(context, MConstants.Database.DATABASE_NAME, factory: null,MConstants.Database.VERSION);
    }
    //endregion


    //region Public Methods
    @Override
    public void onCreate(SQLiteDatabase db) {
        String CREATE_SIGNAL_TABLE = "CREATE TABLE " + MConstants.Database.SIGNAL_TABLE.TABLE_NAME + " ("
                + MConstants.Database.SIGNAL_TABLE.KEY_ID + " INTEGER PRIMARY KEY,"
                + MConstants.Database.SIGNAL_TABLE.CELL_ID + " INTEGER,"
                + MConstants.Database.SIGNAL_TABLE.KEY_SIGNAL_VALUE + " INTEGER)";
        db.execSQL(CREATE_SIGNAL_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + MConstants.Database.SIGNAL_TABLE.TABLE_NAME);
        onCreate(db);
    }

}
```

Fig. 4.1.2.f Crearea bazei de date folosind query SQL

```
public void AddSignalStrength(int cellId, int signalStrength) {
    ContentValues values = new ContentValues();
    values.put(MConstants.Database.SIGNAL_TABLE.CELL_ID,cellId);
    values.put(MConstants.Database.SIGNAL_TABLE.KEY_SIGNAL_VALUE,signalStrength);

    this.mDatabase.insert(MConstants.Database.SIGNAL_TABLE.TABLE_NAME, nullColumnHack: null,values);
}
```

Fig. 4.1.2.g Inserarea în baza de date folsind o metodă Java

4.1.2.1. Arhitectura vizuală a aplicației

Dacă în capitolul anterior am discutat despre ceea ce se întâmplă fără ca cel care folosește telefonul să vadă, capitolul curent va aborda partea de arhitectură vizuală a aplicației, mai exact modul în care datele sunt afișate utilizatorului. Astfel, aplicația este formată dintr-un număr de opt ecrane, fără a lua în calcul ferestrele de tip pop-up ce apar în timpul utilizării, după cum urmează:

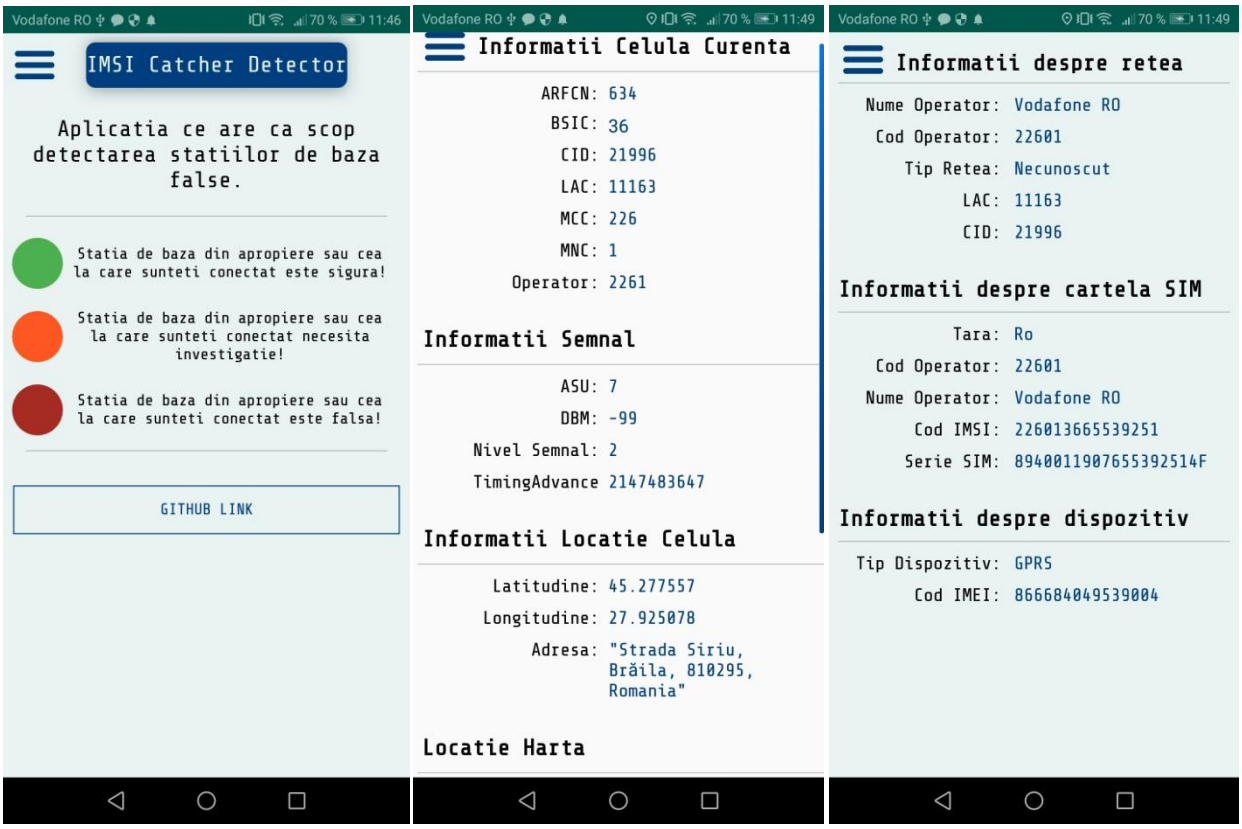
- ecranul de acasă
- ecranul de verificare celulă
- ecranul de detalii despre telefon/ sim
- ecranul de detalii despre celula curentă
- ecranul ce conține harta celulelor
- ecranul pentru vizualizarea elementelor din baza de date
- ecranul pentru vizualizarea statisticilor despre numărul de celule
- ecranul de opțiuni

Acestea de mai sus au fost utilizate pentru a împărți informațiile într-un mod cât mai structurat. Fiecare poate fi accesat cu ajutorul unui meniu lateral, vizibil din fiecare ecran, astfel încât utilizatorul să nu fie nevoit să parcurgă prea mulți pași pentru realizarea diferitor acțiuni. Mai mult, rezultatul verificăriii celulei curente, în momentul în care este negativ, va fi afișat printr-o fereastră de tip pop-up și nu depinde de ecranul curent. De asemenea, utilizatorul este înștiințat de rezultatul negativ și prin vibrarea telefonului timp de o secundă.

Fig. 4.1.2.1.a Fereastra pop-up pentru înștiințarea utilizatorului

Ecranul de acasă conține o scurtă descriere despre fiecare dintre celulele cu care aplicația poate interacționa, oferind utilizatorului o privire de ansamblu asupra culorilor folosite pentru a marca fiecare tip de celulă, acest aspect urmând să se reia în momentul în care este deschisă harta. De asemenea acesta conține și un buton care va redirecta utilizatorul către codul sursă al aplicației, postat pe GitHub. Ecranul ce conține informații despre telefon, cartela SIM sau și cel cu informații despre celula curentă, au fost create atât cu scop informativ dar și cu scopul de a oferi anumite detalii unui utilizator cu experiență în acest domeniu, care ar putea efectua propriile teste.

Harta celulelor, respectă principiul „*Overview first, zoom and filter, then details on demand”* (21) astfel încât ecranul oferă o privire de ansamblu a tuturor celulelor din zona în care se află dispozitivul mobil, utilizatorul are posibilitatea de a facem *zoom* pe o anumită zonă sau să filtreze celulele în funcție de tipul acestora iar în final, la apăsarea pe oricare din antene va apărea o fereastră de tip pop-up ce conține detalii despre fiecare celulă în parte.



b. c. d.

Fig. 4.1.2.1.b/c/d Ferestrele de acasă/ prezentare informații telefon/prezentare informații celulă

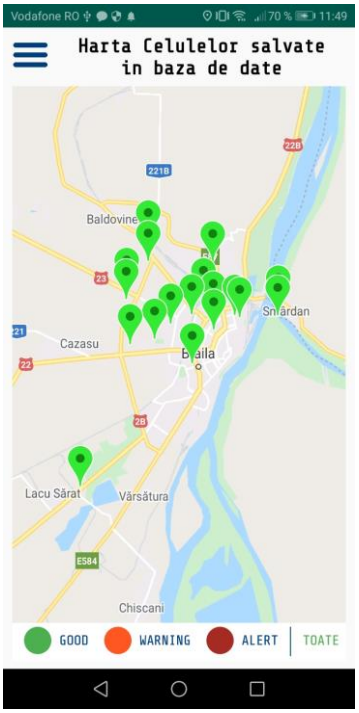


Fig. 4.1.2.1.e Fereastra ce conține harta celulelor

După cum se poate observa, partea de filtrare a fost așezată în partea de jos a ecranului pentru a putea fi la îndemâna utilizatorului.

În ecranul de verificare a celulei curente sunt prezentate în avans regulile pe care utilizatorul ar trebui să le respecte pentru o funcționare cât mai bună a aplicației, însă acesta are posibilitarea de a alege dacă dorește să le vadă de fiecare dată sau nu. Odată deschis ecranul, testarea începe automat iar pe măsură ce testele sunt realizate răspunsul apare în dreptul fiecărei verificări sub forma unui X sau au unui V, având culori sugestive. Din momentul în care testările au început, utilizatorul are posibilitatea de a opri testele și de a le reporni ulterior. Dacă unul din teste a picat, atunci când se face click pe el, apare un mesaj care prezintă pe scurt cauza/ cauzele pentru care rezultatul a fost unul negativ. De asemenea, se poate vedea când a fost pornit ultima dată verificatorul de către utilizator dar și stadiul în care se află testarea cu ajutorul unui *progressbar* și al unui text.

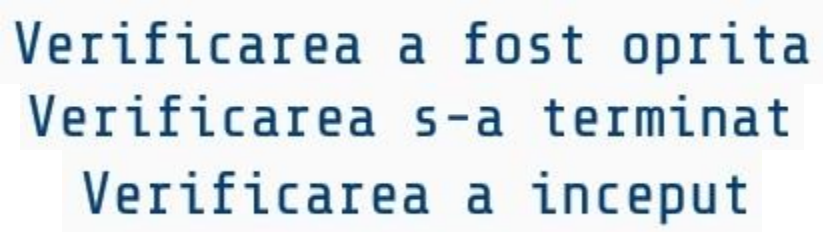


Fig. 4.1.2.1.f Stadiile în care se poate afla testarea

Fig. 4.1.2.1.g Rezultatele testelor pe măsură ce au fost terminate

În figura de mai sus au fost efectuate numai două teste deoarece verificarea a fost oprită de către utilizator. Primul test nu a trecut deoarece în telefonul de test nu a fost introdusă o cartelă SIM care să aibă o conexiune activă la rețeaua de internet. Cu toate că verificările au fost oprite, utilizatorul încă are posibilitatea să vadă motivul pentru care testul respectiv nu a trecut.

Fig. 4.1.2.1.h Prezentarea motivului pentru care testul a picat

Utilizatorul aplicației va avea întotdeauna acces la datele celulelor cu care a interacționat dispozitivul mobil de-a lungul timpului. Acest lucru este posibil prin accesarea meniului de vizualizare a bazei de date. Odată ajuns în acest ecran, va fi expusă întreaga bază de date sub forma unei liste ce conține itemi formați dintr-o imagine sugestivă pentru tipul celulei, id-ul acesteia precum și coordonatele GPS. Această listă poate fi filtrată de utilizator după CID, urmând să apară toate elementele ce conțin numerele introduse de utilizator. Mai mult, în momentul în care se apasă pe oricare dintre itemi, va fi deschisă fereastra de prezentare

informații despre celulă, care va conține numai informații precum CID/ LAC/ MCC/MNC precum și coordonatele gps și locația exactă pe hartă. Practic, vor fi afișate utilizatorului toate informațiile stocate în baza de date Firebase. Toate aceste informații sunt prezentate în figurile 4.1.2.1.i și 4.1.2.1.j.

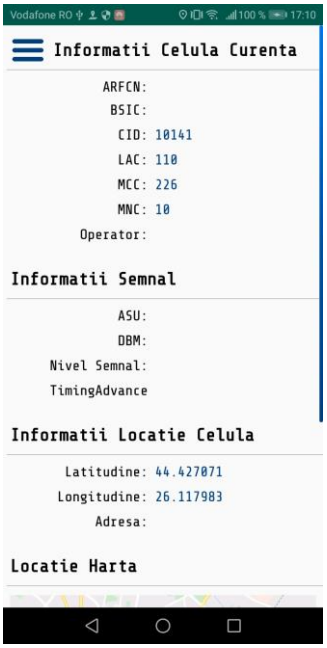


Fig. 4.1.2.1.i Lista cu celule **Fig. 4.1.2.2.j Ecran informații celulă**

Nu în ultimul rând, aplicația este disponibilă utilizatorilor în două limbi, aceștia având posibilitatea de a schimba limba aplicației în ecranul cu opțiuni. Limbile disponibile sunt română și engleză.

Fig. 4.1.2.1.k Limbile disponibile ale aplicației

4.2. Simularea detectării unei stații de bază false

Acest capitol își propune prezentarea tuturor componentelor verificate pe parcursul testării aplicației și de asemenea analizarea rezultatelor obținute.

Încă de la stadiile incipiente ale dezvoltării programului ce detectează prezența stațiilor de bază false, au fost verificate un număr de 79 de antene GSM din locații diferite ale Bucureștiului, una din ele fiind Academia Tehnică Militară „Ferdinand I" dar și din Brăila. Din rapoartele extrase din baza de date, s-a constatat că din acest număr total de celule, una singură a ridicat probleme, în sensul că a fost catalogată de către aplicație ca fiind *Warning*, având parametri ce nu conțineau valori normale. Prezența acestei celule a fost detectată o singură dată în București, însă necunoscând locația exactă a acesteia, a fost salvată locația dispozitivului mobil de la acea perioadă. Nu se poate spune cu exactitate dacă aceasta a fost un *IMSI Catcher* deoarece nu s-a realizat o testare a respectivei celule, întrucât nu era implementat modulul de verificare automată a antenelor la care telefonul se conectează, la acel moment acest lucru fiind posibil doar manual de către utilizator. Ulterior, s-a implementat acest modul de verificare automată însă telefonul nu s-a mai conectat la o astfel de celulă.

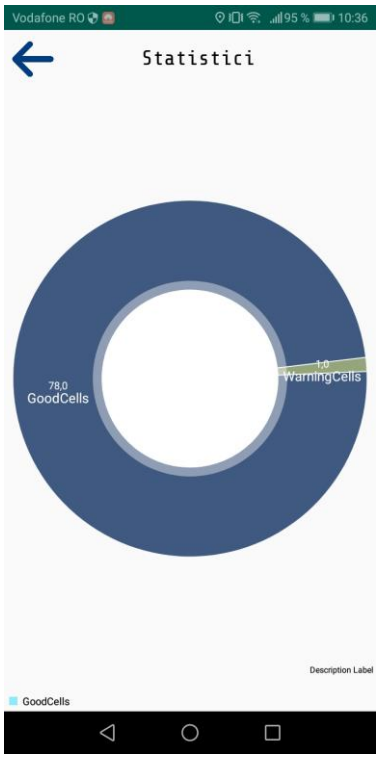
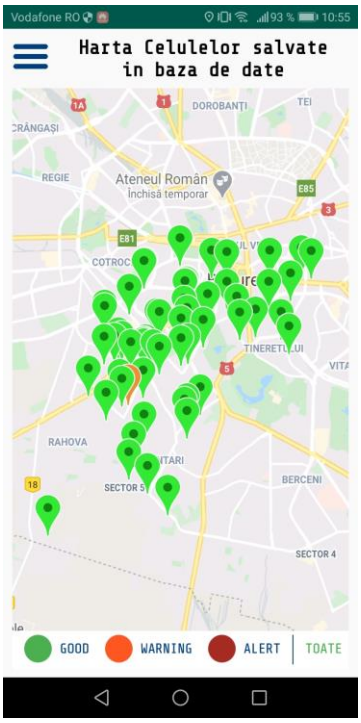


Fig. 4.2.a Statistică privind numărul celulelor verificate



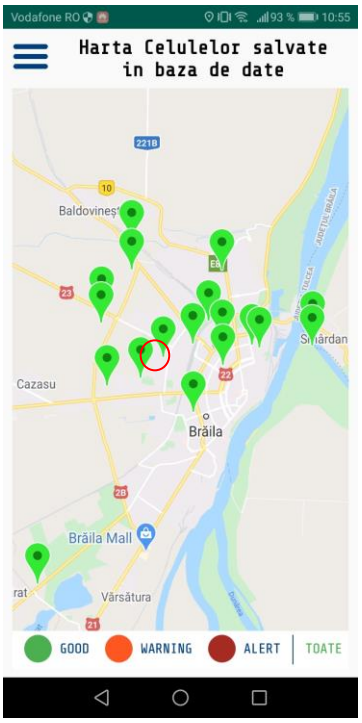


Fig. 4.2.b Celule verificate în București **Fig. 4.2.c Celule verificate în Brăila**

Atunci când dispozitivul mobil se afla în Brăila, locația acestuia se poate vedea în figura 4.2.c, aceasta fiind încercuită. După cum se observă, distanța față de celula la care se poate conecta telefonul variază, iar în teorie distanța maximă poate varia de la 5 kilometri până la 8 kilometri, acest lucru depinzând de obiectele ce se află în apropierea dispozitivului, ce pot influența puterea semnalului (22).

În ceea ce privește celula detectată de aplicație, având parametri în neregulă, locația aproximativă a acesteia este expusă în figura de mai jos, fig. 4.2.d, fiind înconjurată zona în care aceasta se află. A fost utilizată distanța minimă teoretică la care se poate afla o stație de bază, dar în practică aceasta poate fi chiar mai mică atunci când vine vorba de un *IMSI Catcher* deoarece aceste dispozitive nu au o putere de transmisie la fel de mare ca a unei antene ce aparține unui furnizor de servicii de telefonie mobilă. Cu toate acestea, distanța depinde foarte mult de tipul de dispozitiv folosit de către atacator iar în Statele Unite ale Americii există aparate ce au o rază de acțiune declarată de 30 de kilometri în zonele rurale și 5 kilometri în zonele urbane (23).

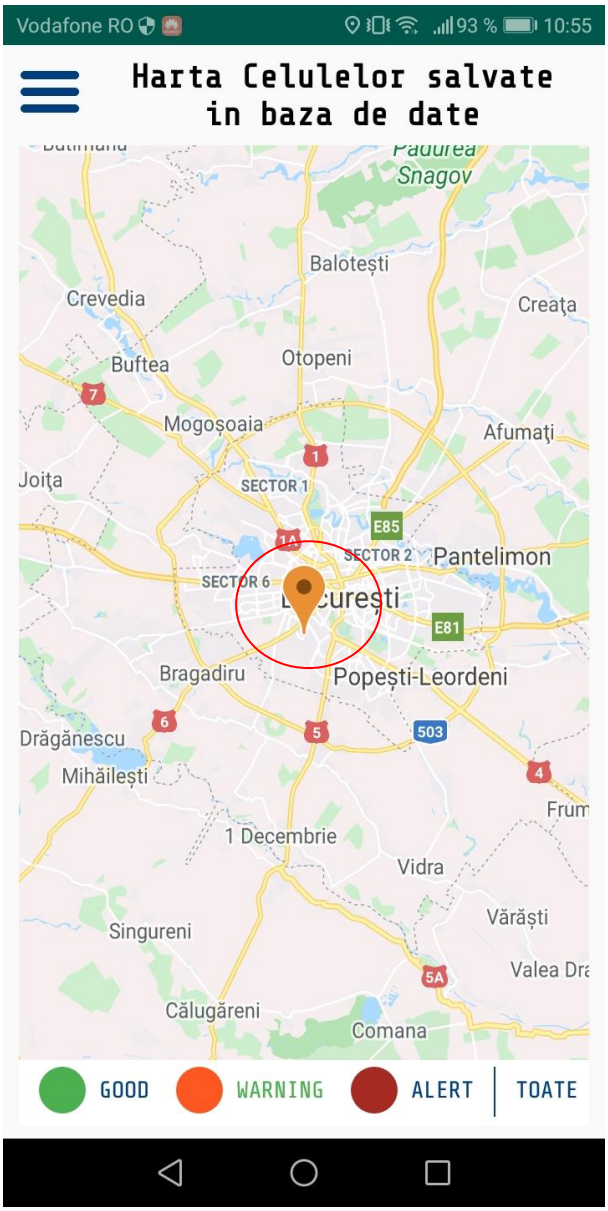


Fig. 4.2.d Celulă detectată posibil malițioasă de aplicație

Luând în considerare celula la care era conectat telefonul în momentul verificării, în figurile de mai jos se pot observa detalii despre aceasta, precum și rezultatele testelor efectuate. După cum se poate vedea, aplicația a constatat că celula este în regulă, aceasta având un rezultat pozitiv pentru toate verificările.

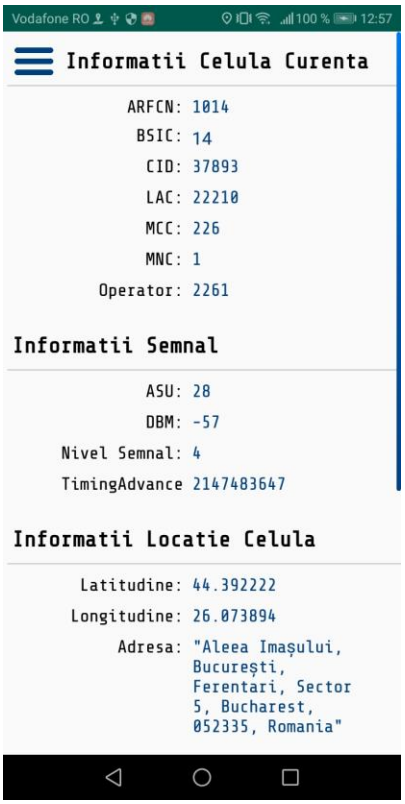


Fig. 4.2.e Datele celulei

Fig. 4.2.f Rezultatele testelor

Pentru a verifica dacă datele obținute de aplicație sunt corecte, am utilizat o altă aplicație, numită BTS Tracker, ce preia informații de la celula la care este conectat dispozitivul mobil, rezultatele fiind asemănatoare.

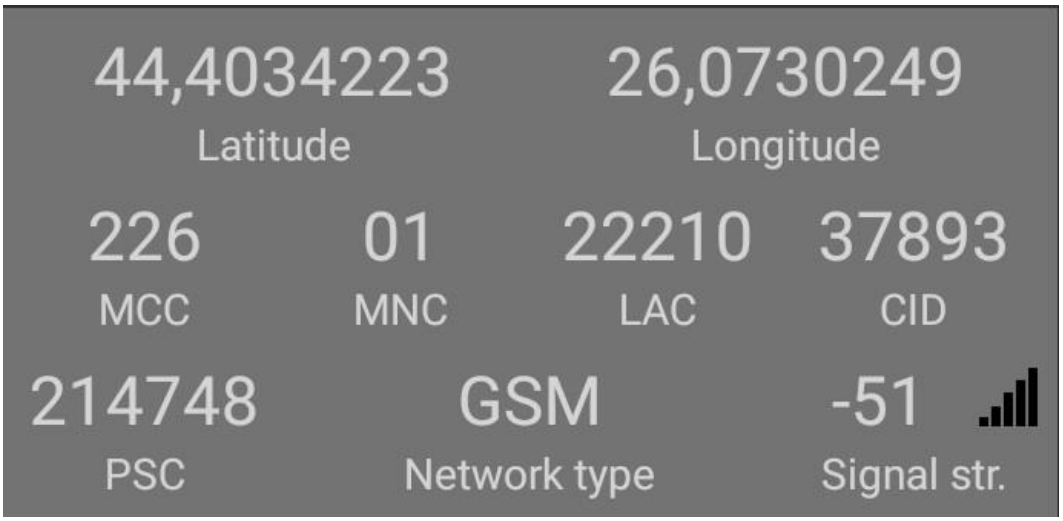


Fig. 4.2.g Date extrase de BTS Tracker

Pe parcursul dezvoltării aplicației s-au creat diferite module de verificare sau de obținere a datelor de la celula la care telefonul este conectat. Înainte de a fi integrate în module mai complexe care să le cuprindă pe toate, au fost efectuate diferite teste pentru a verifica validitatea datelor întoarse de către acestea.

Spre exemplu, în cazul utilizării datelor de la baza de date externă, informațiile primite în cadrul aplicației au fost comparate cu datele care pot fi obținute și accesând direct site-ul api-ului. După ce s-a constatat că acestea sunt similare, atunci s-a trecut la dezvoltarea unui modul, întrucât acesta trimitea datele corect și primea ce trebuie înapoi. În mod similar, s-a verificat și conexiunea cu baza de date Firebase pentru a verifica dacă informațiile trimise au fost stocate în formatul corect. La descărcarea datelor din aplicație, s-a verificat ca acestea să corespundă cu cele dorite.

Asemănător testărilor realizate mai sus, a fost verificat fiecare modul al aplicației ce realizază verificarea celulelor, întrucât acestea sunt cele mai importante și pot duce la erori de decizie. Așadar, în continuare, s-a trecut la testarea modulelor ce se ocupă de pornirea testelor și strângerea datelor de la fiecare în parte pentru a putea calcula un scor final.

După ce s-a constat că aplicația este capabilă de realizarea verificărilor atât automat cât și manual de către utilizator, prin efectuarea testelor precizate mai sus, au fost aduse îmbunătățiri în ceea ce privește experiența utilizatorului cu aplicația. Mai exact, au fost efectuate modificări asupra interfeței grafice, după consultarea unui număr de persoane care și-au exprimat opiniile asupra modalităților de perfecționare a acesteia.

5 CONCLUZII

5.1. Sinteza principalelor idei din lucrare

În concluzie, rețeaua GSM nu este sută la sută sigură, iar cum marea majoritate a populației utilizează telefonul mobil în fiecare zi, expunerea la atacuri asupra rețelei este crescută. Din păcate, utilizarea dispozitivelor de tip *IMSI Catcher* nu este reglementată atât de bine pe cât ar trebui, lucru care duce la abuzuri atât din partea autorităților, care nu recunosc de fiecare dată când folosesc acest tip de atac folosit pentru interceptare, cât și din partea altor actori rău intenționați. Aceste atacuri invadează intimitatea utilizatorului de telefonie mobilă, care nu poate fi urmărit de fiecare dată când poartă telefonul la el și, mai mult, convorbirile sale pot fi interceptate cu ușurință.

Astfel, o soluție pentru detecția acestor atacuri ar fi utilizarea unor aplicații ca cea prezentată în cadrul proiectului, cu toate că nici acestea nu îl pot feri pe utilizator, ci doar îl pot anunța, ca acesta să poată lua măsuri, cum ar fi eliminarea cartelei SIM din telefonul mobil, iar dacă atacurile continuă, să anunțe autoritățile competente să ia măsuri în acest sens, prin analizarea activităților de la nivelul rețelei GSM.

Aplicația prezentată în acest proiect, a fost creată în așa fel încât utilizatorul să aibă întotdeauna o evidență a celulelor cu care dispozitivul mobil interacționează. Acesta poate accesa oricând datele antenelor, cum ar fi CID/ LAC/ MCC/ MNC precum și locația acestora prin coordonatele GPS, având posibilitatea să vizualizeze pe hartă atât individual, cât și colectiv toate stațiile de bază. Astfel, va putea cunoaște care au fost zonele de risc prin care a trecut, dar și locația aproape exactă la care se afla în momentul în care a fost interceptat de un dispozitiv *IMSI Catcher*. Pentru o bună funcționare, aplicația trebuie să fie pornită permanent, singurul dezavantaj fiind faptul că este o mare consumatoare de energie electrică din cauza senzorilor pe care îi utilizează pentru a realiza detecția și verificarea celulelor.

În ceea ce privește partea de design, s-a încercat să fie cât mai prietenoasă cu utilizatorul, fără a fi nevoie ca acesta să țină minte acțiuni complicate. Butoanele și metodele de filtrare au fost așezate în partea de jos a ecranului pentru a fi cât mai accesibile utilizatorului. În ceea ce privește culorile, s-a luat în calcul faptul că aplicația poate fi folosită în exterior, la un grad de luminozitate crescut și, astfel, au

fost utilizate culori închise pentru a fi cât mai vizibile persoanei ce folosește aplicația.

5.2. Direcții pentru continuarea cercetării

În unul din capitolele anterioare, au fost prezentate metode existente de detecție ale dispozitivelor mailițioase *IMSI Catcher* și s-a expus faptul că separat fiecare din ele au diferite vulnerabilități. Astfel, ar putea fi luată în calcul crearea unui sistem complex care să cuprindă toate cele 3 metode existente de detecție prezentate anterior. Acestea ar urma să își împartă atribuțiile în rețea iar în momentul detectării unui atacator să trimită informații între ele pentru a putea realiza o detecție rapidă și precisă a atacului.

Mai mult, se poate realiza o aplicație care să poată fi rulată numai pe telefoane ale căror sistem de operare nu este tocmai oficial, întrucât acestea pot da acces la mai multe date decât o face telefonul în mod normal. Dezavantajul unei astfel de metode este că aplicația nu va mai fi accesibilă publicului larg care nu este întotodeauna dispus sau nu știe să facă modificări la nivelul sistemului de operare.

În ceea ce privește prezenta aplicație, poate primi îmbunătățiri atât din punctul de vedere al testelor pe care le efectuează, cât și din punctul de vedere al interfeței grafice. Aceasta ar putea implementa și detecția SMS-urilor silențioase primite de telefon, ce reprezintă mesaje trimise către dispozitivul mobil dar care nu atenționează și utilizatorul, în schimb trimițând un semnal către cel care a lasat mesajul. Mai mult, într-o versiune ulterioară a aplicației poate fi implementată și detecția femtocelulelor, ce reprezintă dispozitive pentru îmbunătățirea semnalului de telefonie mobilă ce pot fi utilizate într-o afacere sau sau chiar de acasă (24). În cazul în care femtocelulele sunt utilizate de o persoană rău intenționată, atacatorul poate clona un telefon mobil și îl poate utiliza pentru a efectua apeluri telefonice în numele victimei, pentru a realiza trafic de date etc (25).

În ceea ce privește interfața grafică, se pot adăuga diferite elemente, cum ar fi încercuirea zonei în care dispozitivul maliții os-ar putea afla pentru a oferi utilizatorului o imagine de ansamblu. Mai mult se poate crea un fișier care să salveze toate verificările realizate de telefon și care să poată fi accesat din cadrul aplicației oricând.

6 BIBLIOGRAFIE

1. *IMSI-Catch Me If You Can: IMSI-Catcher-Catchers.* **Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, Edgar Weippl.** 2014, ACSAC '14: Proceedings of the 30th Annual Computer Security Applications Conference, p. 10.

2. **Ashkan Soltani, Craig Timberg.** Tech firm tries to pull back curtain on surveillance efforts in Washington. *The Washington Post.* [Online] The Washington Post, 09 17, 2014. [Cited: 04 09, 2020.] https://www.washingtonpost.com/world/national-security/researchers-try-to-pull-back-curtain-on-surveillance-efforts-in-washington/2014/09/17/f8c1f590-3e81-11e4-b03f-de718edeb92f_story.html.

3. **Wikipedia.** Imsi-catcher. *Wikipedia.* [Online] 03 11, 2020. [Cited: 04 09, 2020.] https://en.wikipedia.org/wiki/IMSI-catcher.

4. **Naarttijärvi, Markus.** *Swedish police implementation of IMSI-catchers in a European law perspective.* Umeå : Computer Law & Security Review, 2016.

5. **International, Privacy.** *Privacy International's contribution to the half-day general discussion on Article 21 of.* s.l. : Privacy International, 2019.

6. **Tamir Israel, Christopher Parsons.** *Gone Opaque? An Analysis of Hypothetical IMSI Catcher Overuse in Canada.* Toronto : Telecom Transparency, 2016.

7. **Wikipedia.** Stingray phone tracker. *Wikipedia.* [Online] Wikipedia, 04 10, 202. [Cited: 04 13, 2020.] Stingray phone tracker.

8. **Lee, Timothy B.** The police are secretly using fake cellphone towers to spy on people. *Vox.* [Online] Vox, 04 22, 2015. [Cited: 04 13, 2020.] https://www.vox.com/2015/4/22/8463239/stingray-fbi-secret.

9. **Catherine Cullen, Brigitte Bureau.** Someone is spying on cellphones in the nation's capital. *CBC News.* [Online] CBC, 04 03, 2018. [Cited: 04 21, 2020.] https://www.cbc.ca/news/politics/imsi-cellphones-spying-ottawa-1.4050049.

10. **Wikipedia.** GSM. *Wikipedia.* [Interactiv] 13 09 2019. [Citat: 09 04 2020.] https://ro.wikipedia.org/wiki/GSM.

11. **xlxmarketing.** The Cell Phone Mysteries, What Are Dual Band, Tri Band And Quad Band Cell Phones And Where Will They Work? *Chinavasion.* [Online] Chinaviasion, 11 21, 2008. [Cited: 04 21, 2020.] https://blog.chinavasion.com/1099/the-cell-phone-mysteries-what-are-dual-band-tri-band-and-quad-band-cell-phones-and-where-will-they-work-part-2/.

12. **Korkusuz, Ammar Yasir.** *Security in the GSM Network.* Bogazici University, Electrical-Electronics Engineering Department : s.n., 2012.

13. **Umberto Ferraro Petrillo, Giancarlo De Maio,Giuseppe Cattaneo.** Security Issues and Attacks on the GSM Standard: a Review. *JOURNAL OF UNIVERSAL COMPUTER SCIENCE.* 16, 2013, Vol. 19.

14. **Lab, Threat.** *GottaCatch'EmAll:UnderstandingHowIMSI-CatchersExploitCellNetworks(Probably.* s.l. : Electronic Frontier Foundation, 2019.

15. **Shinjo Park, Ravishankar Borgaonkar,Altaf Shaik,Jean-Pierre Seifert.** *Anatomy of Commercial IMSI Catchers and Detectors.* London : RIGHTSLINK, 2019.

16. **Wikipedia.** OpenCellID. *Wikipedia.* [Online] Wikipedia, 04 06, 2020. [Cited: 05 04, 2020.] https://en.wikipedia.org/wiki/OpenCellID.

17. —. Android Studio. *Wikipedia.* [Online] Wikipedia, 04 27, 2020. [Cited: 05 04, 2020.] https://en.wikipedia.org/wiki/Android_Studio.

18. **Apple.** Model-View-Controller. *Apple Deveoper.* [Online] Apple Inc., 04 06, 2018. [Cited: 05 05, 2020.] https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html.

19. **Wikipedia.** Model-view-controller. *Wikipedia.* [Online] Wikipedia, 01 31, 2018. [Cited: 05 05, 2020.] https://ro.wikipedia.org/wiki/Model-view-controller.

20. **SQLite.** About SQLite. *SQLite.* [Online] SQLite. https://www.sqlite.org/about.html.

21. **Mihai-Lica Pura.** Representations and information visualization. *ATM-Wiki.* [Online] 03 18, 2020. [Cited: 05 07, 2020.] https://wiki.fsisc.ro/dokuwiki/_media/ioc/curs/11_representations_and_information_visualization.pdf.

22. **Wikipedia.** Cell site. *Wikipedia.* [Online] Wikipedia, 04 29, 2020. [Cited: 05 14, 2020.] https://en.wikipedia.org/wiki/Cell_site#Range.

23. **Paganini, Pierluigi.** Cellphone Surveillance: The Secret Arsenal. *INFOSEC.* [Online] InfoSec Institute, 01 08, 2016. [Cited: 05 14, 2020.] https://resources.infosecinstitute.com/cellphone-surveillance-the-secret-arsenal/#gref.

24. **GSMK.** GSMK. *NETWORK SECURITY.* [Interactiv] GSMK, 2017. https://www.gsmk.de/products/network-security/#overwatch.