Centar Informacijske Sigurnosti

# Ispitivanje sigurnosti mobilnih aplikacija


studeni 2011.



CIS-DOC-2011-11-030

# Upozorenje

Podaci, informacije, tvrdnje i stavovi navedeni u ovom dokumentu nastali su dobrom namjerom i dobrom voljom te profesionalnim radom CIS-ovih stručnjaka, a temelje se na njihovom znanju i petnaestak godina iskustva u radu u informacijskoj sigurnosti. Namjera je da budu točni, precizni, aktualni, potpuni i nepristrani.

Ipak, oni su dani samo kao izvor informacija i CIS ne snosi nikakvu izravnu ili posrednu odgovornost za bilo kakve posljedice nastale korištenjem podataka iz ovog dokumenta.

Ukoliko primijetite bilo kakve netočnosti, krive podatke ili pogreške u ovom dokumentu, ili imate potrebu komentirati sadržaj molimo Vas da to javite elektroničkom poštom na adresu info@CIS.hr.

# O CIS-u

**CIS izrađuje** pregledne dokumente (eng. white paper) na teme iz područja informacijske sigurnosti koji će biti korisni zainteresiranoj javnosti, a u svrhu **podizanje njezine svijesti o informacijskoj sigurnosti** i **sposobnosti za čuvanje i zaštitu informacija i informacijskih sustava**. Pored toga, CIS razvija i održava mrežni portal www.CIS.hr kao referalnu točku za informacijsku sigurnost za cjelokupnu javnost; izrađuje obrazovne materijale namijenjene javnosti; organizira događaje za podizanje svijesti o informacijskoj sigurnosti u javnosti i pojedinim skupinama te djeluje u suradnji sa svim medijima.

CIS **okuplja mlade** zainteresirane za informacijsku sigurnost i radi na njihovom pravilnom odgoju i obrazovanju u području informacijske sigurnosti te pripremu za **profesionalno bavljenje informacijskom sigurnošću**.

Centar informacijske sigurnosti [CIS] nastao je 2010. godine na poticaj Laboratorija za sustave i signale[LSS] Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu, a kao posljedica 15togodišnjeg rada na istraživanju, razvoju i primjeni informacijske sigurnosti. LSS je među ostalim potaknuo osnivanje CARNetovog CERTa i sudjelovao u izradi Nacionalnog programa informacijske sigurnosti RH.

**Smisao CISa** je da bude **referentno mjesto za informacijsku sigurnost** za javnost, informatičare i posebno za mlade te da sustavno podiže njihovu svijest i sposobnosti u području informacijske sigurnosti.

Rad CISa podržava Ministarstvo znanosti, obrazovanja i sporta Republike Hrvatske, a omogućuju sponzori.

# Prava korištenja

# Sadržaj

# 1. Uvod

U posljednje vrijeme značajno raste udio pametnih telefona na tržištu, a time i broj preuzetih mobilnih aplikacija. U 2010. godini broj preuzetih aplikacija dosegao je broj od 10,9 milijardi u svijetu. Rastom broja preuzetih mobilnih aplikacija stručnjaci predviđaju da će rasti i broj zlonamjernih programa i sigurnosnih prijetnji za mobilne uređaje.

Andrew Hoog iz tvrtke *viaForensics*, koja je specijalizirana za sigurnost mobilnih aplikacija, rekao je da mnoge aplikacije za pametne telefone imaju sigurnosne propuste. Ispitivali su oko pedesetak popularnih mobilnih aplikacija, a samo nekoliko njih je prošlo početna ispitivanja. Ta ispitivanja su se odnosila na sigurno spremanje korisničkih imena, lozinki i drugih osjetljivih podataka. Aplikacija *Facebook* je među onima koje nisu pozitivno ocijenjene na provjeri.

Nedavno je tvrtka *McAfee* objavila da je za 46% u odnosu na 2009. godinu porastao broj zlonamjernih programa za mobilne telefone, te da se procjenjuje kako oko 55 000 novih prijetnji nastaje svaki dan. Ovo se događa jer mobilni uređaji postaju sve moderniji i postoji nekoliko vodećih platformi na tržištu, pa ne treba raditi puno različitih zlonamjernih programa kako bi se nanijela šteta velikom broju korisnika. Dovoljno je napraviti zlonamjerni program za određenu platformu. Postaje sve lakše napraviti aplikacije za mobilne uređaje, pa tako postaje i lakše napraviti zlonamjerne programe. Više nije potrebno veliko znanje za izradu mobilnih aplikacija i zlonamjernih programa, jer se sve upute i potrebni programi nalaze na Internetu.

Problem postaje još i veći, samom činjenicom da se mobiteli koriste za osobne i poslovne potrebe te da se preko njih spaja na Internet na raznim mjestima i pomoću raznih tehnologija.

U daljnjem tekstu može se vidjeti razvoj mobilnih platformi na tržištu, popis platformi koje su postale najpopularnije te njihovi sigurnosni propusti. Najviše pažnje posvećeno je *Apple iOS* i *Android* platformi zbog njihovog brzog razvoja i zauzimanja velikog dijela tržišta mobilnih uređaja.

# 2. Razvoj mobilnih platformi i aplikacija

Tržište korisničkih mobilnih aplikacija doživjelo je veliki porast u zadnjih nekoliko godina. Ove aplikacije su, između ostalog, pružile odgovarajući pristup bankovnim računima, podacima o kreditnim karticama, osobnim identifikacijskim informacijama, informacijama o putovanjima i osobnim računima elektroničke pošte.

Sigurnosni rizik koji je povezan s ovim aplikacijama često se može identificirati i umanjiti tako da se aplikacije podvrgnu sigurnosnom ispitivanju. U usporedbi s aplikacijama za računala ili web aplikacijama, mobilnim aplikacijama je teže provjeravati sigurnost i upravo zbog toga one se manje ispituju. Ovo ne znači da su mobilne aplikacije sigurnije od aplikacija za računala i web aplikacija.

## *2.1. Platforme na tržištu*

Na tržištu se trenutno nalazi puno različitih platformi za mobilne uređaje, zbog toga što svaki proizvođač razvija platformu za svoje mobilne uređaje. Neke od popularnijih mobilnih platformi su:

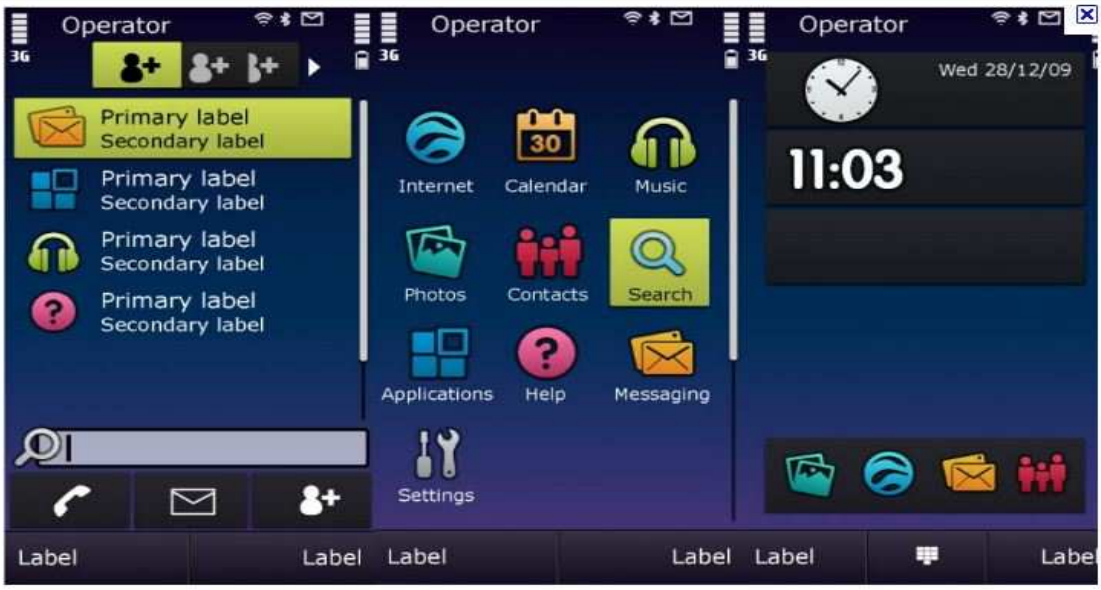
### 2.1.1. Java Platform ili Java ME

*Java Platform ili Java* ME (eng. *Micro Edition*) predstavlja reduciranu inačicu platforme *Java SE* (eng. *Standard Edition*) koja je napravljena za mobilne uređaje i ugradbene sustave. Ovu platformu je razvio *Sun Microsystems,* koji danas pripada kompaniji *Oracle Corporation*. Nakon izlaska na tržište zamijenila je prethodnu platformu *PersonalJava*. Postoje dodatni profili koji su namijenjeni za određeni skup uređaja kao što su: *MID Profil* (eng. *Mobile Information Device Profile*) koji je namijenjen klasičnim mobitelima i *Personal Profile* namijenjen ugradbenim i PDA[1] uređajima.

### 2.1.2. Symbian Platform

*Symbian Platform* je operacijski sustav otvorenog koda i platforma za razvoj programske podrške za pametne telefone. Ova platforma je u vlasništvu organizacije *Symbian Foundation*. Izgled jednog korisničkog sučelja može se vidjeti na slici u nastavku pogljavlja (Slika 1). Nokia je njezin većinski vlasnik uz Sony Ericsson, Motorolu, LG Electronics, Samsung, Sharp i druge. Jezgra *Symbian* platforme je operacijski sustav *Symbian OS,* koji je razvijen 1999. godine, a namijenjen je za mobitele i pametne telefone. Prednosti su mu veliki broj dostupnih aplikacija, velika postojeća baza arhitekata i programera te otvorenost sustava.

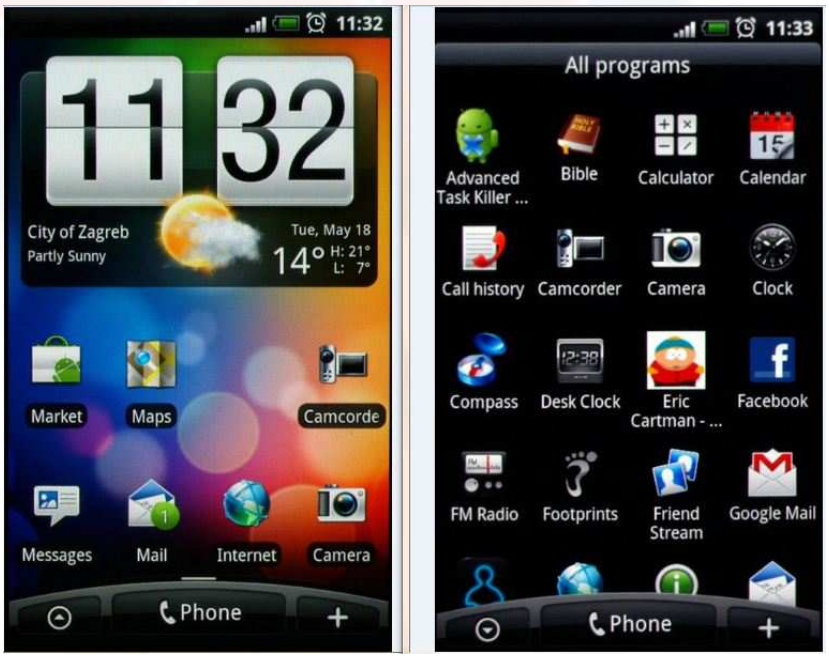
[1] PDA (eng. *Personal DigitalAssistant*) je digitalni prijenosni uređaj koji funkcionira kao osobni informacijski menadžer. Sadašnji PDA uređaji imaju mogućnost spajanja na Internet putem WiFi tehnologije ili mobilnog interneta. Osnovne funkcije PDA uređaja su rokovnik, adresar, podsjetnik i kalkulator, kao i mogućnost rada drugih aplikacija koje se učitavaju s memorijskih kartica.




***Slika 1. Primjer korisničkog sučelja za operacijski sustav Symbian***
***Izvor: Google***

### 2.1.3. Android OS

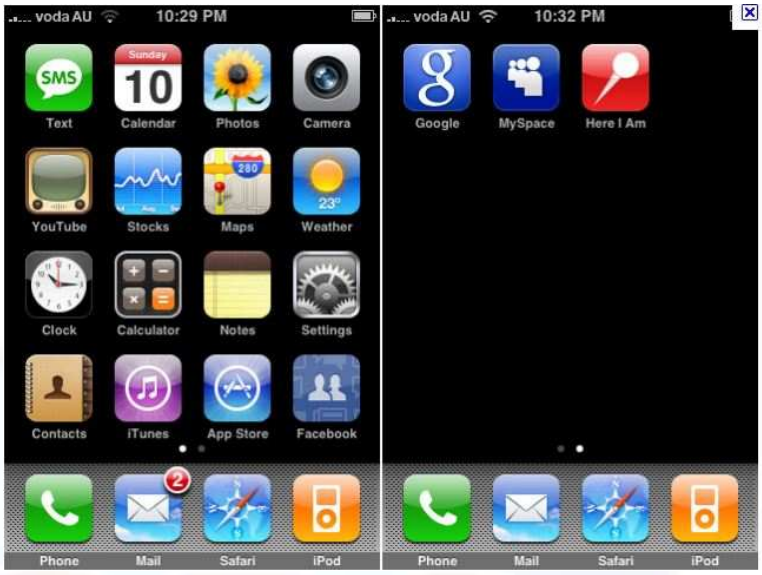
*Android Inc*., koji je od 2005. godine u većinskom vlasništvu Googlea, je razvio operacijski sustav *Android*. Temelji se na operacijskom sustavu *Linux* i otvorenog je koda. Godine 2007. prvi je put predstavljen na tržištu i od tada ga razvija organizacija *Open Handset Alliance* - konzorcij koji se sastoji od 78 kompanija, a najznačajniji predstavnici su: Google, HTC, Dell, Intel, Motorola, Samsung, Qualcomm, Texas Instruments, T-Mobile, Nvidia i LG Electronics. Prvi mobilni uređaj na tržištu s operacijskim sustavom *Android* bio je *HTC Dream*. Aplikacije se pišu u programskom jeziku *Java* i one koje se nalaze na *Android Marketu* imaju sigurnosno jamstvo. Slika 2 prikazuje izgled jednog korisničkog sučelja.




***Slika 2. Primjer korisničkog sučelja za operacijski sustav Android***
***Izvor: Google***

### 2.1.4. Apple iOS

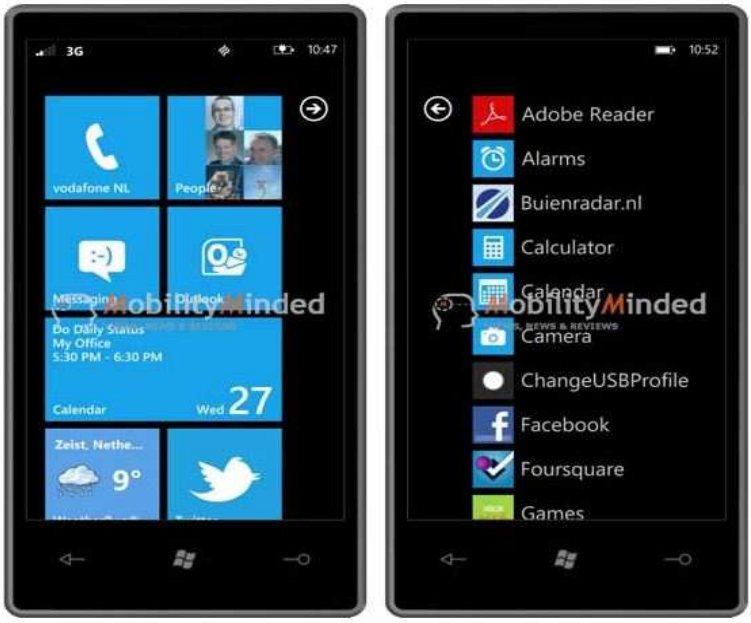
Operacijski sustav *Apple iOS* nalazi se na Appleovim uređajima kao što su *iPhone*, *iPod Touch*, *iPad* i *Apple TV*. Kad je ovaj operacijski sustav 2007. godine prestavljen, bio je namijenjen samo mobitelima *iPhone*. Razvijen je iz operacijskog sustava *Mac OS X,* koji je pak temeljen na Unix OS-u. Aplikacije se nalaze na *AppStoreu* i pišu se u programskom jeziku *Objective-C*. Na slici u nastavku (Slika 3) nalazi se primjer korisničkog sučelja za Apple OS.




***Slika 3. Primjer korisničkog sučelja za operacijski sustav Apple iOS***
***Izvor: Google***

### 2.1.5. Windows Mobile

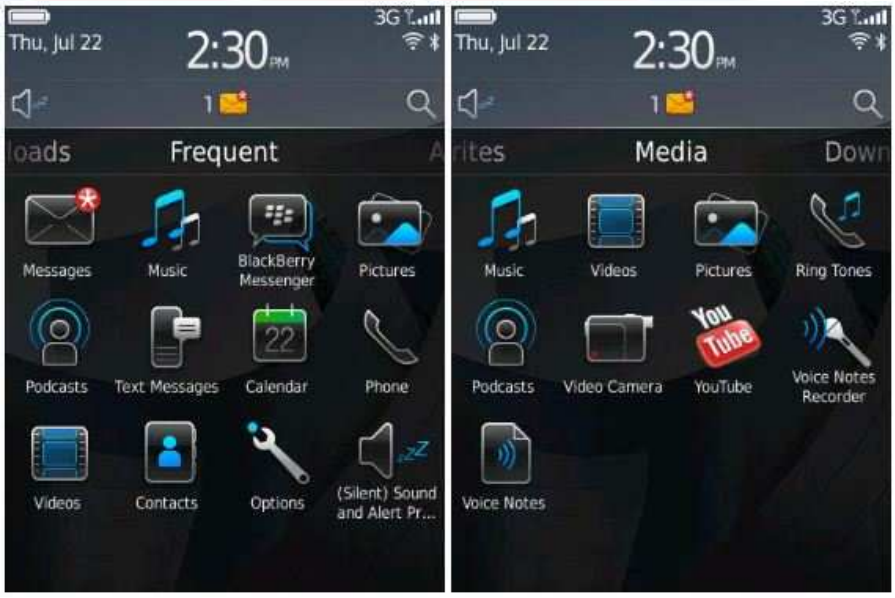
Platformu *Windows Mobile* razvila je tvrtka *Microsoft*, a zadnja inačica je Windows Mobile 6.5. Nakon ove inačice, Microsoft je krenuo razvoj nove inačice OS-a koja se zove *Windows Phone 7*. Njegovo korisničko sučelje može se vidjeti na slici u nastavku poglavlja (Slika 4). Prvi put se pojavljuje 2000. godine u Microsoftovom Pocket Pc-u. Neki od proizvođača koji su u svojim uređajima imali ovaj operacijski sustav su: HTC, Sony Ericsson, Samsung, LG Electronics, Motorola i Sagem. Aplikacije se pišu u programskim jezicima C#, Visual Basic i C++ te se nalaze na *Windows Marketplace*-u.




***Slika 4. Primjer korisničkog sučelja za operacijski sustav Windows Phone***
***Izvor: Google***

## 2.1.6. BlackBerry OS

Platformu *BlackBerry OS* razvila je kanadska firma RIM (eng. *Research In Motion*) 1999. godine. Najviše je namijenjen poslovnim korisnicima i zbog toga ima izvrsnu podršku za pregled elektroničke pošte. Ovaj operacijski sustav nalazi se samo na pametnim telefonima *BlackBerry* i njegovo korisničko prikazuje Slika 5. U usporedbi s dostupnim aplikacijama za ostale mobilne platforme, za *BlackBerry OS* dostupno je relativno malo aplikacija koje se pretežno pišu u programskom jeziku *Java*. Nadogradnje operacijskog sustava dostupne su preko BlackBerry OTASL (eng. *Over The Air Software Loading*) sustava.




***Slika 5. Primjer korisničkog sučelja za operacijski sustav BlackBerry OS***
***Izvor: Google***

## 2.2. Povijest razvoja

Android je zasnovan na operacijskom sustavu Linux, a razvili su ga Google i organizacija Open Handset Alliance. Razvoj aplikacija se radi isključivo u programskom jeziku Java. Programski stog operacijskog sustava Android sastoji se od aplikacija koje su izrađene u programskom jeziku Java i pokreću se na virtualnom računalu Dalvik (eng. *Dalvik Virtual Machine, DVK*). Desetog svibnja 2010. godine Google je objavio da Android Market ima 200 000 aplikacija.

Danas su mobilni telefoni „minijaturna računala“ i aplikacije koje se pokreću na njima slične su web aplikacijama. Obzirom na to, jednom kad se namjesti posrednik i prevede kod aplikacije, ispitivanje sigurnosti je svedeno na ispitivanje sigurnosti ili pregled koda kao i kod bilo koje druge aplikacije za osobna računala.

iPhone je objavljen u lipnju 2007. godine i od tada stječe 25% tržišta mobilnih telefona. To znači da je Apple prodao otprilike 60 milijuna iPhone mobitela od dana kada ga je objavio. Apple je zauzeo još veći udio u tržištu svojim iPad uređajem, kojih je do danas prodano 3 milijuna. Veliko zanimanje za iPad i iPhone javlja se dijelom zbog toga što ima jako puno raznolikih i dostupnih aplikacija koje se ne mogu pokrenuti na inačici verziji Apple iOS-a, nego je potrebno napraviti jailbreaking[2]. Ovo proširuje raspon aplikacija od produktivnih i financijskih pa sve do igrica i zabave. Trenutno Apple App Store sadrži preko 225 000 ovakvih aplikacija i one su preuzete 5 milijardi puta. Otprilike je na 10% uređaja koje pokreće operacijski sustav Apple iOS napravljeno uklanjanje ograničenja, odnosno jailbreaking. Programski jezik koji se koristi za ra razvoj iPhone i iPad aplikacija je Objective-C, koji sa sobom donosi i problem prelјeva međuspremnika koji nije bio problem za okoline J2ME i mobilni .NET. Bilo je nekoliko prijavljenih ranjivosti, zbog prelјeva međuspremnika, vezanih i za operacijski sustav iOS.

Brzi pregled vijesti vezanih uz sigurnost iOS-a otkrivju nekoliko kategorija incidenata s Apple iOS aplikacijama. Najvažnije su prikazane u nastavku poglavlja.

Incidenti sa skupljanjem podataka:

- **MogoRoad** – Korisnici iPhone aplikacije *ID Mobile's MogoRoad* žalili su se kako dobivaju pozive iz kompanije.
- **Storm8's iSpy** – Tvorac nekih od najpopularnijih igara za iPhone potajno je skupljao korisničke brojeve mobitela bez njihovog dopuštenja.
- **Aurora Feint** – Prva aplikacija koja je izbrisana zbog narušavanja privatnosti. Ova aplikacija je pretraživala po listi kontakata korisnika i slala ih na poslužitelje kako bi se usporedili s njihovim prijateljima koji su trenutno na vezi.

Crvi (eng. Worms):

- **Ikee** – iPhone vlasnicima u Australiji uređaji su bili na meti aplikacije s napadima koji se sami umnažaju i prikazuje sliku koju nije bilo lako ukloniti.
- **Dutch Ransom** – Napadač u ovom slučaju drži nizozemske iPhone uređaje zbog otkupnine. Uzrok ovog problema bila je zadana SSH lozinka na iPhone uređaju na kojem je napravljen jailbreaking.
- **iPhone/Privacy.A** – Ovaj crv krade osobne podatke kao što su elektronička pošta, SMS, kontakti, multimedijski podaci, kalendar i drugi.
- **Ikee.B (DUH)** – S ovim crvom napadači su željeli otkriti dva čimbenika autentifikacije ING Direct banke.

Ranjivosti:

- **Libtiff** – Omogućuje napadačima preuzimanje nadzora nad iPhone uređajem tako da se iskorištava problem prelјeva međuspremnika, koji se događa prilikom obrade biblioteke TIFF u Safari pregledniku.
- **SMS Fuzzing** – Napadačima omogućuje preuzimanje nadzora nad iPhone uređajem pomoću izrađenih zlonamjernih SMS poruka.
- **Jailbreakme** – Sigurnosni propust u svim iOS4 uređajima koji omogućuju napadačima potpuni pristup uređaju pregledom zlonamjernog PDF dokumenta u Safari pregledniku.

[2] Jailbreaking je proces uklanjanja ograničenja koje je postavio Apple na svoje uređaje koje pokreće operacijski sustav iOS. Takvi uređaji su: iPhone, iPod Touch i iPad. Jailbreaking korisnicima omogućava da dobiju administratorski pristup operacijskom sustavu, što im omogućava preuzimanje dodatnih aplikacija, proširenja i tema koje nisu dostupne na Apple App Store službenim stranicama.

Može se vidjeti raznolikost napada kao i zlonamjernih aplikacija. Vrlo je važno, ako se razvija aplikacija ili se razmišlja o razvoju aplikacije za iPhone na kojem je napravljen jailbreaking, ispitivati aplikacije kako bi se osiguralo pružanje potrebnog stupnja sigurnosti.

# 3. Razvoj mobilnih aplikacija

Kod razvoja mobilnih aplikacija prvo treba imati jasnu ideju kakva aplikacija se želi razviti. Aplikacije se mogu podijeliti u dvije kategorije, a to su:

1. aplikacije koje imaju bogatu funkcionalnost (nalaze se na uređaju i ne treba ima stalna povezanost s Internetom) i
2. aplikacije koje za rad moraju imati stalnu povezanost s Internetom preko Internet preglednika.

Ako se razvija aplikacija koja mora imati stalnu povezanost s Internetom za određeni uređaj, prvo se treba utvrditi na koje sve načine je uređaju omogućeno spajanje na Internet. Danas to nije problem, jer većina pametnih telefona ima mogućnost spajanja na WiFi mreže ili putem mobilnog Interneta. Većina pametnih telefona već dolazi s aplikacijom Internet preglednika koji imaju slične mogućnosti kao i preglednici na računalima. Preglednici na mobilnim uređajima često se nazivaju mini ili mikro preglednici (npr. Opera mini).

Ako se zna za koju platformu se radi aplikacija, onda je puno lakše, jer su određeni načini i alati definirani. Najčešće se može koristiti programski razvojni paket (eng. *Software Development Kit* - SDK ) od proizvođača platforme, razvojni alat i simulator uređaja.

Java, C, C++, C#, Objective-C, Phyton i .NET neki su od programskih jezika za razvoj mobilnih aplikacija. Prenosivost je vrlo poznata prednost programskog jezika Java i ona je zbog toga prisutna u većini mobilnih uređaja. Postoji nekoliko vrsta Java programskih okruženja, kako bi se korisnički program mogao prilagoditi za različite uređaje. Tako na primjer Java ME (eng. *Micro Edition*), koja je prije bila poznata pod nazivom J2ME (eng. *Java 2 Micro Edition*), nalazi se u mobilnim uređajima i nekim elektroničkim uređajima kao što su televizori i pisači. Tablica 1 prikazuje osnovne mogućnosti trenutno aktualnih mobilnih platformi.

***Tablica 1. Prikaz mobilnih platformi i njihovih mogućnosti***
***Izvor: Wikipedia***

| | Programski jezik | Dostupnost alata za pronalazak pogrešaka | Dostupnost simulatora | Integrirano razvojno okruženje | Mogućnosti instalacijskog paketa | Trošak razvojnog alata |
|---|---|---|---|---|---|---|
| **Android** | Java, ali dijelovi koda mogu biti u programskom jeziku C i C++ | Pronalaženje pogrešaka integrirano u Eclipseu, dostupan i samostalno | DA | Eclipse, Project Kenai Android dodatak za NetBeans | apk | Besplatno |
| **BlackBerry OS** | Java | Pronalaženje pogrešaka integrirano s IDE-om | DA | Eclipse | alx, cod | Besplatno |
| **Apple iOS SDK** | Objective-C | Pronalaženje pogrešaka integrirano u Xcode IDE | U paketu s iPhone SDK, integriran s Xcode IDE | Xcode | Samo preko App Storea, treba pregled i dopuštenje Apple Inc. | Alati i ispitne simulacije su besplatni za Mac, ali za instalaciju treba platiti za pristupni ključ za programere |
| **Symbian** | C++ | DA | Besplatni simulator | Puno izbora | SIS razvoj | Komercijalni i besplatni alati su dostupni |
| **Windows Mobile** | C, C++ | DA | Besplatni simulator u paketu s IDE-om | Visual Studio 2010, 2008,2005 u paketu s VC++ | OTA razvoj, CAB dokumenti ActiveSync | Besplatni alati ili ugrađeni u VC++ ili u Visual Studio |
| **Windows Phone** | C# | DA | Besplatni simulator, također u paketu s IDE-om | Visual Studio 2010 | OTA razvoj, XAP dokumenti | |

# 4. Sigurnosno ispitivanje mobilnih aplikacija

Mobilne aplikacije koje se instaliraju u na mobilne uređaje slične su aplikacijama koje se instaliraju na računala. Nove aplikacije dodaju datoteke, mijenjaju unose registara i postavke konfiguracije, bilježe nove usluge i obavljaju druge sistemske aktivnosti tijekom instalacije. Za dobar sigurnosni ispit važno je da se sve ove komponente analiziraju i ispituju. Analiza datoteke sustava i primjena obrnutog inženjeringa dva su vrlo važna aspekta obavljanja analize na strani klijenta.

## *4.1. Analiza potpisa aplikacija*

Za aplikacije koje se instaliraju na mobilni uređaj, analiza potpisa aplikacije počinje i prije same instalacije. Kad se aplikacija instalira ona u datoteci ostavlja svoj potpis koji se mora nadzirati i analizirati. Programeri često pretpostave da je memorija mobitela sigurno mjesto pohrane i koriste ju da bi spremili važne informacije kao što su korisnička imena, lozinke i ostale podatke. Analiziranje datoteke sustava je mukotrpan proces i treba ga pažljivo napraviti. Kad se analiziraju datoteke sustava mobilnog uređaja, glavni ciljevi su:

1. Otkriti datoteke koje je napravila aplikacij na mobilnom uređaju tijekom instalacije. Ako postoji dostupna opcija, treba instalirati aplikaciju na vanjsku memoriju kao što je memorijska kartica (eng. *Compact Flash Card*). Ovo se treba napraviti zbog toga što neki modeli mobilnih uređaja ne dozvoljavaju pristup svom sadržaju unutarnje memorije i time čine težim posao analiziranja tih datoteka. Kad se datoteke identificiraju, daljnja analiza se može obaviti.
2. Identificirati promjene koje su napravljene u postojećim datotekama tijekom višestrukih operacija aplikacije.
3. Analizirati informacije koje su zapisane u datoteku sustava mobilnog uređaja tijekom različitih faza operacije.

## *4.2. Obrnuti inženjering*

Nakon uspješne instalacije, konfiguracije i temeljitog ispitivanja aplikacije koja se ispituje, gotova je temeljita analiza datoteke sustava. Sad su poznate sve datoteke koje su instalirane na mobilni telefon, sve promjene tih datoteka koje su nastale nakon primjene različitih radnji i sadržaj datoteka. Naoružani ovim znanjem, ispitivač ulazi u drugu fazu, obrnuti inženjering, sa sljedećim ciljevima:

1. Pokušava se otkriti logika i kod aplikacije što pomaže u promjeni koda aplikacije ako bude potrebno, te tako svladati bilo koje sigurnosne mjere. Često se događa da se u mnogim mobilnim aplikacijama vjeruje okolini klijenta. Ako se pronađe način da se zaobiđu sigurnosni mehanizmi, to često pruža pristup poslužitelju, ali i mnogo više.
2. Identificirati primjenu i/ili konstruirati nedostatke i pronaći metode kako bi ih se moglo iskoristiti.
3. U kodu aplikacije tražiti skrivene tajne kao što su lozinke, enkripcijski ključevi i ostale važne podatke.

## *4.3. Aplikacije zasnovane na Internet pregledniku*

Veliki broj mobilnih aplikacija koje susrećemo napravljene su tako da ovise o Internet pregledniku i ne trebaju nikakve komponente za instalaciju. Pokušaji pristupa ovim aplikacijama pomoću poznatih preglednika za Internet često preusmjeravaju korisnika na pravu aplikaciju (za razliku od mobilne inačice) ili vraćaju stranicu pogreške. Web poslužitelji analiziraju zaglavlja zahtjeva kako bi otkrili tip uređaja i Internet preglednik kako bi donijeli odluku o vraćanju sadržaja.

Dva zahtjeva u zaglavljima koja se često koriste za odluke o vraćanju sadržaja su: *User-Agent* i *Accept request header*.

### 4.3.1. *User-Agent* zahtjev u zaglavlju

Svi web preglednici uključuju zaglavlje *User-Agent* u svojim zahtjevima. Ovo zaglavlje se koristi kako bi se identificirao web preglednik (npr. Internet Explorer, Mozilla Firefox, Opera, Google Chrome ili neki drugi) i uređaj na kojem je web preglednik pokrenut. Gotovo svi popularni preglednici na mobilnim telefonima uključuju i vrstu uređaja u zaglavlju *User-Agent*. Veliki broj web poslužitelja i aplikacija u ovisnosti o sadržaju ovog zaglavlja odlučuju koji će sadržaj vratiti te pružiti funkcionalnost. Primjer toga je slučaj kad se stranici http://m.google.com pristupi s računala koristeći Internet Explorer ili Mozilla Firefox. U ovisnosti o vrijednosti zaglavlja *User-agent* web poslužitelj preusmjerava Internet preglednik na http://www.google.com/mobile/ odnosno http://m.google.com kad se pristupi preko preglednika mobilnog Internet. Tada se korisniku otvara stranica gdje može preuzeti različite mobilne Google aplikacije. Promjena zaglavlja *User-Agent* na računalu s onim mobilnog uređaja kojeg podržava web poslužitelj, uzrokuje da prikazane web stranice budu upravo one koje su namijenjene za taj uređaj. Ovakvo ponašanje se može iskoristiti u korist ispitivanja. Aktivni upravljanjem zaglavlja *User-Agent* nastaju tehnike za ispitivanje:

- **Korištenje web posrednika** – Većina web posrednika (kao Paros[3] ili Fiddler[4]) pruža korisnicima mogućnost mijenjanja zaglavlja zahtjeva s poznatim izrazima ili kroz izravne zamijene. U većini slučajeva ova promjena konfiguracije mora se jednom napraviti, a kako bi se napravila mora se poznavati zaglavlje *User-Agent* uređaja koji se ispituje. Konfiguracija web posrednika kako bi zamijenio postojeće zaglavlje *User-Agent* s korištenjem web posrednika može dozvoliti potencijalni pristup aplikacijama s računala.
- **Dodatak pregledniku Mozilla Firefox za zamjenu zaglavlja *User-Agent*** – S ovim odgovarajućim dodatkom mogu se stvoriti i spremiti višestruka proizvoljna zaglavlja *User-Agent*. Mobilni profil *User-Agent* može se pokrenuti tako da pretražuje aplikaciju na računalu na isti način kako bi ju pretraživao da je na mobilnom uređaju, bez potrebe da web posrednik mijenja zaglavlje.

### 4.3.2. *Accept* zahtjev u zaglavlju

Svi web preglednici imaju uključeno zaglavlje *Accept* u svoje zahtjeve. Ovo zaglavlje se koristi za obavještavanje poslužitelja o vrsti podataka koje preglednik može prihvatiti. Primjer toga je kada web poslužitelj koristi zaglavlje *Accept* kako bi odredio može li preglednik čitati programski jezik za izradu web stranica (eng. *Wireless Markup Language - WML*) stranice i tako dati odgovarajući sadržaj. *WMLbrowser,* dodatak za Firefox, daje mogućnost pregleda WML stranica i mijenja zaglavlje *Accept* kako bi isto odražavao. Prednost korištenja ovakvog alata je taj da jednom kad se preglednik na računalu podesi, on pretražuje aplikaciju, a sva daljnja ispitivanja se mogu obavljati kao i kod sigurnosnog ispitivanja regularnih web aplikacija. Ponekad se dogodi situacija u kojoj poslužitelj, unatoč promjeni zaglavlja, ne tretira preglednik na računalu kao mobilni uređaj. U tom slučaju, vanjski web posrednik se može podesiti za mobilni uređaj i tako sav promet aplikacije preusmjeriti kroz njega. Važno je uočiti kako pretraživanje aplikacije s mobilnog uređaja ima nedostatke kao što su smanjenje brzine kojom se ispitivanje provodi kao i alati dostupni za uporabu.

## 4.4. Presretanje prometa aplikacije s web posrednikom

Do ovog dijela ispitivanja uspješno bi trebalo biti napravljeno praćenje potpisa aplikacije i pristup aplikaciji pomoću preglednika na radnoj površini računala. Potreba za presretanje prometa web aplikacije web posrednikom se koristi kada se malo toga zna o vrstama zahtjeva koji su poslani

[3] Paros je sigurnosni alat za procjenu ranjivosti web aplikacije. Napisan je za ljude koji trebaju procijeniti sigurnost svojih web aplikacija. Besplatan je i u potpunosti napisan u programskom jeziku Java.

[4] Fiddler je besplatni alat i može ispraviti pogreške prometa od gotovo bilo koje aplikacije koja podržava posrednika, uključujući Internet Explorer, Google Chrome, Safari Apple, Mozilla Firefox, Opera i mnoge druge. Također može ispraviti pogreške prometa uređaja poput Windows Phone, iPod / iPad i drugih.

poslužitelju ili o sredstvima koja su se tamo koristila. Iako se uspješno napravi obrnuti inženjering aplikacije, stvaranje „novog klijenta“ pomaže kod preusmjeravanja prometa aplikacije preko web posrednika. To se radi kako bi ga se moglo presresti, pogledati ponašanje aplikacije u detalje i promijeniti podatke za provjeru valjanosti podataka, autorizaciju i druga područja ispitivanja. Presretanje prometa aplikacije ispitivaču daje potpun nadzor nad interakcijom klijenta i poslužitelja i tako omogućuju izvođenje temeljitog ispitivanja. Nadalje, nakon uspješnog usmjeravanja prometa aplikacije na web posrednik, sva ispitivanja mogu biti obavljena na sličan način kao kod regularnih web aplikacija i korištenjem poznatih alata i tehnika. Jedan takav primjer je sustav za paketni prijenos podataka (eng. *General Packet Radio Service* - GPRS) podatkovnom mrežom i bit će prikazan u nastavku.

U GPRS mrežama, podatkovna komunikacija mobilnog telefona usmjerena je na GPRS pristupne točke (eng. *Access Point*). Za pokretanje veze s nekim odredišnim servisom na Internetu preko mobilnog uređaja, prvo je klijentsku aplikaciju potrebno usmjeriti na ispravno ime pristupnog čvora (eng. *Access Point Name - APN*). Telekom operatori definiraju APN različito za različite vrste usluga kao što su pretraživanje web-a, emaila i razmjena trenutnih poruka. Svaki APN ima povezani *IP Gateway* koji se koristi za usmjeravanje prometa. Za probu, ako se može promijeniti IP Gateway neke pristupne točke, onda se može i usmjeriti sav promet aplikacije kroz vlastiti posrednik. Ova tehnika koja mijenja konfiguraciju varira od uređaja do uređaja i na nekima može biti jako složena.

Važno je napomenuti da će sve navedeno u gornjem primjeru raditi samo ako mobilna aplikacija koja se ispituje ne koristi glavnu i najčešću metodu prijenosa informacija na Webu (eng. *Hypertext Transfer Protocol Secure* - HTTPS). U stvarnosti ovo često nije slučaj. Mobilne aplikacije se u pravilu oslanjaju na pohranu certifikata uređaja kako bi se utvrdili pouzdani davatelji certifikata. Ako traženi certifikat (koji pripada servisu na Internetu kojem korisnik pristupa) nije pronađen, u ovim pouzdanim pohranjenim certifikatima, komunikacija nije uspostavljena.

„Fiddler“ je primjer posrednika koji ima svoj certifikat, koji nije dio certifikata koji su pouzdano pohranjenih na uređaju. Pomoću malo podešavanja, moguće je koristiti takve posrednike za slučajeve kad je uključen HTTPS. Da bi mogli koristiti te posrednike za presretanje prometa, mora se prvo uvesti njihov obilježeni certifikat u pouzdanu pohranu certifikata uređaja.

## 4.5. Različite konfiguracije posrednika

Ovisno o dostupnosti sklopovlja, različite konfiguracije mogu se iskoristiti za usmjeravanje, presretanje i mijenjanje prometa aplikacije. Ovaj dio sažima neke od konfiguracija posrednika koji mogu biti korisni za ispitivanje:

- **Posrednik preko statičke javne IP adrese** – Ako na raspolaganju postoji statička javna IP adresa ovo je konfiguracija koja je za to potrebna. Treba se postaviti vatrozid tako da dopušta dolaženje prometa do priključnica na kojima web posrednik nadgleda promet. Ovo uređenje se lako konfigurira na mobilnim telefonima.
- **Posrednik preko bežične lokalne mreže (eng. *Wireless Local Area Network - WLAN*)** – Ako na raspolaganju ne postoji statička javna IP adresa, a mobitel koji se koristi za ispitivanje podržava bežičnu lokalnu mrežu (eng. *Local Area Network - LAN*), jednostavno je postaviti posrednik na unutarnju mrežu. Ako WLAN mreža upotrebljava statičku IP adresu ili protokol koji koriste klijenti, da bi dobili IP adresu od poslužitelja (eng. *Dynamic Host Configuration Protocol - DHCP*), s dugim periodom najma, za neke mobilne uređaje potreban je samo jedan privremeni dokument kako bi se završilo ispitivanje.
- **Posrednik s jednim mobilnim telefonom** – Ako nijedna od gornje dvije opcije nije dostupna, ispitivanje se ipak može obaviti. Ova konfiguracija je korisna kad se poslužitelj za aplikacije koje se ispituju nalazi u prostoru davatelja usluga i kad poslužitelj kojeg aplikacija koristi ne prima nikakav vanjski promet. Ove postavke pretpostavljaju da mobilni telefon koji se koristi za ispitivanje ima modem koji se može koristiti za spajanje računala (na kojem je web posrednik) na Internet. Jednom kad je računalo spojeno na Internet, dodjeljuje mu se IP adresa. Ova adresa mora biti konfigurirana kao adresa za web posrednik uređaja. Važno je

napomenuti da će se ovaj postupak trebati ponoviti više puta ako se mijenja dodijeljena IP adresa računalu.

- **Posrednik s vanjskom vezom na Internet** - Posljednja konfiguracija koja se može primijeniti u nekim slučajevima podrazumijeva da uređaj ima podatkovnu karticu, USB modem ili mobilni telefon koji se može koristiti za spajanje web posrednika na računalu s Internetom. Jednom kad je računalo spojeno na Internet, dodjeljuje mu se vanjska IP adresa, kao i u prethodnom slučaju. Ova adresa se onda mora konfigurirati kao adresa za web posrednik uređaja.

# 5. Praktični primjer ispitivanja aplikacija

Za postavljanje ispitnog okruženja postoji nekoliko načina ispitivanja mobilnih aplikacija kao što su:

1. Korištenje već postojećih web aplikacija za provjeru sigurnosti
2. Korištenje WinWAP-a s posrednikom
3. Korištenje simulatora mobitela s posrednikom
4. Korištenje mobilnog telefona za provjeru sigurnosti i kao posrednik za odlazne podatke s mobitela na računalo

U ovom dokumentu opisat će se simulator s posrednikom zbog toga što je to najlakša i najjeftinija opcija koja postoji za ispitivanje mobilnih aplikacija. Za neke druge platforme ovaj način može biti težak, ali za aplikacije za operacijske sustave Android i Apple iOS, korištenje simulatora je jednostavno i učinkovito.

## *5.1. Ispitivanje sigurnosti Android aplikacija*

Zahtjevi koje je potrebno ispuniti kako bi se ispitala sigurnost Android aplikacija su: računalo s operacijskim sustavom Microsoft Windows, Java inačice 5 ili 6, Eclipse[5] 3.5, Android SDK i Fiddler.

- Pokretanje simulatora – Android simulator dolazi zajedno s programskim razvojnim paketom (SDK). Ovo je alat za simulaciju uređaja koji je zasnovan na QEMU[6] i može se koristiti za stvaranje, pronalaženje pogrešaka i provjeru sigurnosti aplikacija u stvarnoj okolini operacijskog sustava Android. Prije pokretanja simulatora potrebno je stvoriti virtualni Android uređaj (eng. *Android Virtual Device* - AVD). Za pokretanje simulatora, treba unijeti naredbu: *emulator –avd testavd*. Sljedeće što treba napraviti je preuzeti neku Android aplikaciju (ili ju stvoriti), te pokrenuti provjere koji se nalaze sa simulatorom.
- Postavljanje alata posrednika - Ako aplikacija koristi HTTPS ili je web stranica koja se ispituje na Android pregledniku, sljedeći korak je postaviti alat posrednika kao što je Fiddler ili Paros. Postoje četiri glavna načina postavljanja ovakvog posrednika.
  1. Odrediti detalje posrednika kad se pokrene simulator koji koristi naredbu: *emulator –avd test avd –http-proxy http://localhost:8888*. Navedena naredba omogućuje korištenje posrednika na priključnici 8888 (zadana konfiguracija za Fiddler).
  2. Druga opcija je odrediti detalje posrednika u APN postavkama simulatora. Treba promijeniti sljedeće postavke: Name: *Internet*, APN: *Internet*, Proxy: *IP adresa računala*. Username i Password polja ne treba podešavati.
  3. Treća opcija je odrediti proxy poslužitelja pomoću adb shell programaa koristeći naredbu: *export HTTP_PROXY=http://localhost:8888 (*postavljanje varijable okoline).

[5] Eclipse je višejezično programsko razvojno okruženje i sadrži integrirano razvojno okruženje (eng. *Integrated Development Environment* – IDE). Napisan je uglavnom u programskom jeziku Java i može se koristiti za razvoj aplikacija u Javi, a putem različitih dodataka i u drugim programskim jezicima kao što su: Ada, C, C++, COBOL, Perl, PHP, Python, R, Ruby, Scala, Clojure, Groovy and Scheme.

[6] QEMU je simulator koji omogućava pokretanje kompletnog operacijskog sustava kao samo još jedan zadatak na računalu. Može biti jako koristan za isprobavanje različitih operacijskih sustava, ispitivanje programa te pokretanje programa koji se ne mogu pokrenuti na operacijskom sustavu računala.

4. Zadnja mogućnost je mijenjanje postavki posrednika u postavkama baze podataka iz dijela od kuda web pretraživač Android čita podatke. Postavke baze podataka koriste relacijsku bazu podataka temeljenu na C programskoj biblioteci SQLite (eng. *Structured Query Language Lite* – SQLite) koja se slaže s osnovnim SQL naredbama koje su preporučene ako se planira koristiti ova metoda. Treba promijeniti ime računala i informacije priključnice, a sve ostalo ostaviti kako je inicijalno podešeno. Naredba za izmjenu u bazi je:

```
>adb shell
#sqlite3
/data/dana/com.google.android.providers.settings/databases/settings.db
sqlite> INSERT INTO system VALUES (99,'http_proxy','localhost:8888');
sqlite>.exit
```

Nakon korištenja jedne od ovih opcija posrednik treba moći vidjeti zahtjeve i odgovore. Ključ svega je imati web posrednik koji presreće zahtjeve. Od ovog koraka pa nadalje, ispitivanje sigurnosti mobilnih aplikacija slično je provjeri ranjivosti regularnih web aplikacija.

- Alat za provjeru sigurnosti mobilnih Android aplikacija – *Android SDK* dolazi s nekoliko alata koji nisu napravljeni posebno za sigurnosno ispitivanje, ali mogu biti korisni u praksi za sigurnosnu provjeru. Neki od tih alata su: *Manifest explorer*, *Intent Sniffer* i *Intent Fuzzer*. Napomena: pri ispitivanju Android platforme treba iskoristiti skriveni izbornik za pronalaženje pogrešaka (eng. *Debug Menu*).
- Alat koji omogućuju komuniciranje sa simulatorom (eng. *Android Debug Bridge* - ADB) – Ovaj alat je dio Android SDK i s njim se mogu izvoditi Linux naredbe (kao što su *ls, cd, mv* i sl.) . U vodiču za razvoj Android aplikacija (eng. *Android Developer's Guide*) postoji cijeli niz naredbi koje se mogu koristi, a neke od njih su opisane u nastavku.
  1. ADB se može koristiti kako bi pronašli sve simulatore i Android uređaje koji su spojeni na računalo koristeći naredbu: *adb devices*.
  2. Još jedna važna naredba koju omogućuje ADB daje mogućnost preuzimanja/učitavanja podataka na/sa simulatora/uređaja. Ovo može biti korisno ako se žele preuzeti podaci sa simulatora/uređaja na računalo kako bi se pregledali ili obradili.
  3. Naredbe *dumpsys* ili *dumpstate* mogu se koristiti za prikazivanje podataka sustava na ekranu. Ovi podaci mogu sadržavati važne sigurnosne informacije.
  4. Naredbom *mksdcard* može se stvoriti virtualna kartica (eng. Secure Digital Card, SD) za simulator, jer je moguće da aplikacija koja se ispituje zahtjeva SD karticu kako bi se na nju instalirala baza podataka ili neki drugi podaci. Korisna je kad se ispituje aplikacija i kad se koristi simulator umjesto fizičkog uređaja. Mogu se pronaći skrivene informacije prolaskom kroz dokumente koje je spremila aplikacija na SD karticu. Uvijek treba tražiti lozinke, PIN-ove i ostale osobne informacije.
  5. Iz ADB-a može se pokrenuti naredba *sqlite3* za ispitivanje baze podataka koju je stvorila Android aplikacija i spremila u memoriju uređaja. Ovaj način, također, može otkriti važne informacije kao što su lozinke ili PIN-ovi spremljeni u čitljivom obliku (eng. plaintext). Takve baze podataka spremljene su s podatkovnim nastavkom *„.db“.*
- Manifest explorer – Svaka aplikacija koja radi na operacijskom sustavu Android ima *AndroidManifest.xml* datoteku. Ova je datoteka jako važna, iz sigurnosne perspektive, zbog toga što definira dopuštenja za zahtjeve aplikacije. Alat *Manifest Explorer* dopušta jednostavno pregledavanje XML datoteka. Kod sigurnosne provjere važno je provjeriti da li aplikacija slijedi princip „minimalnih ovlasti“ i da ne koristiti dopuštenja koja nisu potrebna za njezino funkcioniranje.
- *Intent Sniffer* – Ovo je mehanizam u operacijskom sustavu Android koji premješta podatke između procesa. Formira jezgru Android unutarnjeg procesa komunikacije (eng. *Inter Process Communication* - IPC).

- *Strace* – Alat za otkrivanje pogrešaka koji prati pozive i signale sustava. Ovaj alat dolazi instaliran s Android SDK-om. Jako je koristan kad se ispituje aplikacija koju nije lako presresti pomoću posrednika *Fiddler* ili drugih HTTP posredničkih alata.
- Prevođenje Android aplikacija (eng. *Decompiling Android Applications*) – Android paketi (*„.apk“* datoteke) su ustvari jednostavne ZIP datoteke. One sadrže između ostalih komponenti i AndroidManifest.xml, classes.dex te resources.arsc. Na računalu je moguće preimenovati nastavak i otvoriti ih sa ZIP alatom, kao što je WinZip, te provjeriti njihov sadržaj.
- Dozvole za datoteku u operacijskom sustavu Android – Android dozvole za datoteke koriste isti model kao i Linux operacijski sustavi. Za provjeru dozvola, treba u ADB napisati *ls –l*. Svaka *„.apk“* datoteka instalirana u simulatoru ima svoj jedinstveni korisnički ID. Ovo sprječava aplikaciju da pristupi podacima druge aplikacije. Bilo kojoj datoteci koju je stvorila aplikacija bit će dodijeljen taj korisnički ID i neće biti dostupan drugim aplikacijama. Ako je stvorena nova datoteka pomoću getSharedPreferences(), openFileOutput() ili createDatabase() mogu se odrediti MODE_WORLD_WRITEABLE i MODE_WORLD_READABLE zastavice koje onda dopuštaju drugim procesima (programima) da čitaju/pišu globalno u ove datoteke. Treba razmotriti traženje ovih zastavica u kodu i ispitati jesu li zaista potrebne. Treba napomenuti da je takva provjera jedino moguća ako ispitivač ima pristup izvornom kodu aplikacije.

## 5.2. Ispitivanje sigurnosti aplikacija za iPhone/iPad

Za ispitivanje aplikacija na iPhone/iPad uređajima potrebno je: Mac Book na kojem se nalazi operacijski sustav Snow Leopard 10.6.2 ili noviji, Apple iOS 4.0.1 (za provjeru sigurnosti iPhone aplikacija) i iOS 3.2 (za provjeru sigurnosti iPad aplikacija), Charles posrednik i SQLite paket za rad s bazama podataka.

- Instalacija iOS SDK – iPhone/iPad simulator nije dostupan za preuzimanje, kao zasebna aplikacija. Kako bi se koristio stimulator, potrebno je instalirati iOS SDK u kojem se nalazi i simulator. Samo registrirani Apple programeri mogu preuzeti SDK za za provjeru sigurnosti aplikacija zbog toga što je to jedini SDK koji daje mogućnost razvoja i ispitivanja aplikacija za iPad. Apple centar za razvoj – ADC (eng. *Apple Developer Center*) ne dopušta preuzimanje arhiviranih inačica iOS-a. SDK sadrži Xcode IDE, iPhone simulator, iPad simulator i ostale alate za razvoj i provjeru sigurnosti.
- Namještanje aplikacija za pokretanje u simulatoru – Kad programeri uspješno naprave aplikaciju pomoću alata Xcode, pokreću aplikaciju s ispravnim simulatorom za ispitivanje.
- Postavljanje alata posrednika – Prvi korak u postavljanju okoline za ispitivanje trebao bi biti namještanje posrednika. Jednom kad se ovaj dio namjesti, ispitivanje se svodi na tehniku ispitivanja sigurnosti standardnih web aplikacija. Postoji nekoliko alata za posrednike koji su dostupni za Mac OS X. Uobičajeni posrednički alati su: *WebScarb*, *Burp* i *Charles*. *Charles* posrednik je prvi izbor iz nekoliko razloga. Prvi razlog zbog kojeg ga se bira je taj što pruža opciju presretanja podataka za svaku aplikaciju koja se pokreće na Mac OS X-u bez potrebe ručnog mijenjanja postavki posrednika za svaku aplikaciju. Treba samo u *Charles* posredniku u izborniku *Proxy* omogućiti Mac OS X *Proxy* opciju. Nakon ovih postavki, Charles posrednik će presretati sve HTTPS zahtjeve iz Safari preglednika, simulatora i drugih programa. Druga velika prednost je da ga je lako postaviti i radi neprimjetno s iPhone/iPad simulatorima, pogotovo ako aplikacija izvodi provjeru valjanosti certifikata poslužitelja. On isto omogućuje korištenje posebne skripte kako bi se zaobišla provjera certifikata. Skripta podržava bazu podataka *TrustStore.sqlite3* i instalira *Charles SSL* certifikat za iPhone/iPad simulator. Ove postavke mogu se postići i ručno, bez uporabe skripte. Ako se baza podataka *TrustStore.sqlite3* otvori sa SQLite manager-om, može se primijetiti da sprema sažetak poruke *SHA1* certifikata poslužitelja u *tsettings* tablicu. Moguće je tablicu *tsettings* urediti ručno kako bi se zamijenio sažetak poruke SHA1 sa sažetkom poruke Charles certifikata. Za pronalazak sažetka poruke certifikata Charles posrednika, treba instalirati certifikat za Mac pomoću Safari ili Firefox preglednika. Treba otvoriti certifikat i pronaći vrijednosti sažete poruke koje mogu biti u *tsettings* tablici.

- Prevođenje iPhone/iPad aplikacija – Nekoliko je prednosti koje se dobivaju prevođenjem aplikacija pri sigurnosnom ispitivanju. Pomaže izvođenju cjelokupne sigurnosne procjene pregledavanjem koda aplikacije. Preporučeno je izvoditi statičku analizu koda na kako bi se identificirali problemi kao što su preljevi međuspremnika. Aplikacije za iPhone/iPad napisane su programskim jezikom *objective-C*, u kojem je prilično lako napraviti prevođenje. Prvo treba dobaviti binarnu aplikaciju tako da je se preuzme s *App Storea* i prenese na Mac pomoću *iTunesa*. Nakon toga treba s dostupnim alatima napraviti prevođenje koda. Za prevođenje postoji više alata, a najpoznatiju su *otool* (koji dolazi u kompletu s Xcodeom) i *class-dump-x,* alat koji binarni kod pretvara u ljudima razumjiv oblik.
- Statička analiza izvornog koda – Statička analiza koda je tehnika analiziranja koda kod koje nije potrebno izvesti kod. U većini slučajeva, analiza se obavlja na izvornom ili objektnom kodu. Tehnika proučavanja tijekom izvođenja aplikacije zove se dinamička analiza. Trivijalno je prevesti kod iPhone/iPad aplikacije i zbog toga napadači mogu koristiti potrebne alate za pronalazak nedostataka u aplikacijama. Upravo je zbog toga od iznimne važnosti napraviti isto tijekom faze sigurnosnog ispitivanja. Statička analiza aplikacije može se obaviti korištenjem besplatnih alata kao što su *Flawfinder* ili *Clan*. *Flawfinder* je koristan samo ako aplikacija koristi izvorne C biblioteke, a ako ih ne koristi onda treba koristiti *Clang*. Tehnika statičke analize može pomoći pri otkrivanju problema kao što su: gubitak memorije, neinicijalizirane varijable, dio programskog koda koji ne radi ništa, pogreške u pisanju, neslaganje tipova podataka te prepisivanje međuspremnika. Sve se to može napraviti pomoću Xcode-a ako postoji dostupan izvorni kod aplikacije. Statička analiza pregledava sve moguće scenarije izvođenja te identificira logičke pogreške kao što je na primjer alociranje sve raspoložive memorije.
- Dinamička analiza – Odnosi se na tehniku procjenjivanja aplikacija tijekom njezina rada. Postoji nekoliko alata za ovu svrhu, koje je pružio Apple. Dva glavna su: *Instruments* i *Shark*.
  1. *Instruments* alat izašao je s Mac OS X v10.5. Pruža skup moćnih alata za procjenu ponašanja aplikacije tijekom rada. Tipovi podatak koje nadzire ovaj alat su:
     - Nadziranje aktivnosti dokumenta (eng. *File Activity Monitor*) – omogućuje identifikaciju datoteka koje je aplikacija proizvela i obradila.
     - Nadziranje memorije – pomaže u otkrivanju gubitka memorije.
     - Nadziranje procesa – prikazuje procese u stvarnom vremenu.
     - Nadziranje mreže – zapisuje mrežne aktivnosti.
  2. *Shark* se najviše koristi za nadziranje izvođenja. Može raditi sljedeće: statičko uzorkovanje aplikacije tijekom određenog perioda vremena, praćenje razine sustava, statičku analizu, analizu Java koda i druge aktivnosti.
- Zaštita podataka – je vrlo važna kategorija kod ispitivanja mobilnih aplikacija zbog činjenice da su više podložne gubicima i krađi u usporedbi s aplikacijama koje su na računalu. Spremljeni podaci mogu biti pohranjeni u uređaje koji se koriste za sinkronizaciju i mogu se od tamo ukrasti. Istraživanje je pokazalo da iPhone pohranjuje važne informacije, kao što su tipke i snimke ekrana, često na duže vrijeme. Aplikacija i sama može pohranjivati važne informacije u privremene datoteke oblika „*.plist*“ ili u SQLite baze podataka na strani klijenta. Tijekom sigurnosnog ispitivanja važno je odrediti takve rizike i pružiti preporuku kako bi ih se moglo ublažiti.
- SQLite baza podataka – iOS aplikacije pohranjuju podatke na strani klijenta u SQLite baze podataka na uređaju. Informacije u bazama podataka često su kriptirane i zbog toga mogu sadržavati osjetljive informacije kao što su brojevi računa i druge. Mogu sadržavati i informacije o stanju aplikacije koje se mogu iskoristiti kako bi se zaobišla logika aplikacije. Za čitanje ili uređivanje SQLite baze podataka mogu se koristiti bilo koji dostupni klijenti, kao što je *SQLite Manager* dodatak za *Firefox*. Iz dosadašnjeg iskustva, ako je to moguće, važne informacije nikad se ne bi trebale spremati na strani klijenta. Enkripcija podataka u *SQLite* bazama podataka trebala bi se koristiti kao zadnji izbor, jer primjena može postati složena i zbog toga trebati pažljivo upravljati ključem. Popis svojstava datoteka, također, nije dobro mjesto za spremanje važnih informacija.

# 6. Budućnost

Tehnologija neprekidno napreduje. Mobilni telefoni su uređaji najnovije generacije, a mobilne aplikacije su najnoviji tehnološki napredak u tom području. Razvoj mobilnih aplikacija ovisi najviše o zahtjevima samih korisnika i o popularnosti pojedinih aplikacija. Sasvim je očekivano da ima sve više Apple iOS i Android korisnika, jer pružaju velike mogućnosti i veliki broj aplikacija pa svaki korisnik može imati svoj personalizirani uređaj.

U daljnjoj budućnosti mobilne aplikacije će nastaviti pratiti razvoj tehnologije. Razvijati će se i zlonamjerne aplikacije koje će i dalje pokušavati iskoristiti propuste u mobilnim platformama. Potrebno se je više usredotočiti na ispitivanje mobilnih aplikacija, kako ne bi došlo do većih sigurnosnih propusta, jer mobilni uređaji sve više zamjenjuju osobna računala te se na njih pohranjuju osobni osjetljivi podaci.

# 7. Zaključak

Razvojem mobilnih aplikacija, razvijaju se i zlonamjerne aplikacije. Zlonamjerne aplikacije iskorištavaju sigurnosne propuste u platformama i aplikacijama. Propusti se događaju zbog toga što je teže ispitivati mobilne aplikacije te zbog toga što većina programera mobilnih aplikacija ne provjerava sigurnost gotovih proizvoda. Neovisna istraživanja su pokazala da puno aplikacija koje se nalaze na poznatim servisima za preuzimanje mobilnih aplikacija ne prolaze sigurnosnu provjeru.

Potrebno je ispitivati mobilne aplikacije jer se na mobilne uređaje pohranjuju osjetljivi podaci koji ne bi smjeli doći do provalnika koji bi ih iskoristili (brojevi bankarskih kartica, osobni podaci, poslovne tajne i dr).

Za svaku mobilnu platformu potrebni su drugi alati i uređaji za ispitivanje sigurnosti, te je potrebno znati koje sve dijelove treba podvrgnuti sigurnosnim provjerama. Postoji više načina ispitivanja ali je najbolja praksa koristiti dostupne, provjerene i jednostavne za korištenje alate. Na taj način ispitivač se fokusira na predmet koji ispituje (aplikaciju za pametne uređaje), a ne za alate s kojima će tu provjeru napraviti. Na primjeru sigurnosnog ispitivanja Android i Apple iOS platformi može se vidjeti da je potrebno puno znanja i vremena za sigurnosno ispitivanje mobilnih aplikacija, ali je i nužno zbog sve većeg korištenja mobilnih uređaja kao osobnih računala. No to ne smije biti opravdanje da se sigurnosne provjere ne provode. Štoviše, kako se radi o jako osjetljivim osobnim/poslovnim podacima ovakve provjere treba prepustiti (nezavisnim) stručnjacima.

Potrebno je još puno rada kako bi se mobilne aplikacije učinile sigurnima, jer se sve više koriste i postoji sve više zlonamjernih programa. Ovo je još uvijek nova tehnologija i nije se previše razmišljalo o sigurnosti dok se nisu pojavili zlonamjerni programi, te se onda uvidjelo koliko zapravo ima sigurnosnih propusta. Ali, kao i sa svim novim tehnologijama, jedino obrazovanjem kako krajnjih korisnika tako i programera koji razvijaju aplikacije za pametne telefone, moguće se kvalitetno (o)braniti od novih prijetnji koji dolaze.

# 8. Leksikon pojmova

**SQL injection napad** - Napad injekcijom SQL naredbe

Napadačka tehnika koja koristi sigurnosnu ranjivost kod pristupa web programa bazi podataka. Na taj način moguće je ugroziti sigurnost web programa koji konstruira SQL upite iz podataka koje su unijeli korisnici.

https://www.owasp.org/index.php/SQL_Injection

**IP protokol** - Internet Protocol

IP je jedan od glavnih protokola u Internetu, a koristi se za usmjeravanja paketa kroz Internet. U tu namjenu, dodjeljuje IP adrese izvora paketa i njegovog odredišta na temelju kojih će se paketi usmjeravati kroz nekoliko računalnih mreža.

http://compnetworking.about.com/od/networkprotocolsip/g/ip_protocol.htm

**HTTP protokol** - HyperText Transfer Protocol

Osnovna i najčešća metoda prijenosa informacija na Webu. Predstavlja protokol na aplikacijskom sloju OSI modela, a osnovna namjena je prijenos HTML dokumenata (tj. web stranica). HTTP je request/response protokol za komunikaciju između poslužitelja i klijenta. HTTP klijent, kao što je web preglednik najčešće inicira prijenos podataka nakon što uspostavi TCP vezu s udaljenim web poslužiteljem na određenom priključku. Poslužitelj trajno osluškuje zahtjeve na određenom mrežnom komunikacijskom priključku (tipično priključak 80), čekajući da klijent inicira komunikaciju.

http://hr.wikipedia.org/wiki/HTTP, http://www.w3.org/Protocols/

**MAC protokol** - Komunikacijski protokol za pristup mediju

Media Access Control (MAC) je protokol za komunikaciju podacima, također poznat kao Medium Access Control protokol (protokol upravljanja pristupom mediju). On omogućuje mehanizme adresiranja i nadzora pristupa kanalima koji služe za komunikaciju terminala, odnosno čvorišta, s mrežom koja ima više pristupnih točaka.

http://ahyco.ffri.hr/ritehmreze/teme/mac.htm

**Bežična pristupna točka** - Bežična pristupna točka

Bežična pristupna točka (engl. Wireless access point) je uređaj koji omogućuje bežičnim korisnicima (uređajima) pristup računalnoj mreži pomoću Wi-Fi, Bluetooth ili sličnih standarda. WAP se obično spaja na usmjerivač i može prenositi podatke između bežičnih uređaja i žičanih uređaja na mreži.

http://compnetworking.about.com/cs/wireless/g/bldef_ap.htm

**Reverzni inženjering** - Otkrivanje tehnoloških principa i načina rada određenog entiteta

Proces reverznog inženjerstva podrazumijeva otkrivanje tehnoloških principa i načina rada određenog uređaja, objekta ili sustava analizom njegove unutrašnje strukture. Često uključuje fizičko otkrivanje unutrašnjih dijelova (npr., mehanički uređaj, elektronička komponente, računalni program) i detaljno analiziranje. Ovisno o primjeni ciljevi mogu biti različiti. Moguće je otkriti određenu poslovnu tajnu rada uređaja, otkrivanje tajnog algoritma koji se primjenjuje i drugo. Prilikom analize programske potpore najčešće se žali zaobići određen dio koda koji primjenjuje određenu sigurnosnu politiku.

http://searchcio-midmarket.techtarget.com/definition/reverse-engineering

**XML** - EXtensible Markup Language

XML je kratica za EXtensible Markup Language, odnosno jezik za označavanje podataka. Ideja je bila stvoriti jedan jezik koji će biti jednostavno čitljiv i ljudima i računalnim programima. U XML-u se sadržaj uokviruje odgovarajućim oznakama koje ga opisuju i imaju poznato, ili lako shvatljivo značenje.

http://webdesign.about.com/od/xml/a/aa091500a.htm

**Crv** - Računalni crv

Računalni crv je samo-replicirajući zloćudni program koji koristi mrežu računala kako bi poslao vlastite kopije na druge čvorove mreže bez pomoći korisnika. Ovakvo širenje računalnom mrežom je obično posljedica ranjivosti računala.

http://virusall.com/computer%20worms/worms.php

**IA-32** - Intelova 32-bitna procesorska arhitektura

Intelova 32-bitna procesorska arhitektura predstavlja skup naredbi za najraširenij mikroprocesor organizacije Intel. To je 32-bitno proširenje x86 procesorske arhitekture a prvi mikroprocesor koji je se zasnivao na ovoj arhitekturi je Intel 80386.

http://www.pctechguide.com/ia-32-intel-architecture-32-base-instruction-set-for-32-bit-processors

**XMLDsig** - XML digitalni potpis

XMLDsig (također se nazivaju XML Signature, XML-DSig, XML-Sig) definira XML sintaksu za digitalne potpise, a definira ga W3C preporuka XML Signature Syntax and Processing(Sintaksa i obrada XML potpisa).

http://en.wikipedia.org/wiki/XML_Signature

**Prepisivanje memorije** - Napad prepisivanjem memorije

U programskom i sigurnosnom inženjerstvu označava anomaliju u kojoj program prepisuje određeni dio memorije kojemu inače ne bi trebao pristupiti. Prepisivanje memorije se može pokrenuti sa posebno stvorenim korisničkim unosom koji je stvoren za izvođenje programskog koda ili promjenu toka izvođenja programa. Iz tog razloga se smatra jednim od osnovnih izvora ranjivosti računalnih programa.

http://os2.zemris.fer.hr/ns/malware/2007_klaric/buffer_overflow.html

**BCP** - Business continuity management

Proces izrade i dorade logističkog plana koji daje smjernice kako izbjeći, ublažiti, te u slučaju najgoreg, oporaviti se, odnosno ponovno pokrenuti poslovanje, nakon kraha uzrokovanog nezgodom.

http://en.wikipedia.org/wiki/Business_continuity_planning

**WLAN** - Wireless Local Area Network

WLAN služi za bežično povezivanje dva ili više računala u lokalnu mrežu, a omogućuje i pristup Internetu preko bežične pristupne točke. Najraširenij standard u WLAN mrežama je standard 802.11 ili Wi-Fi.

http://searchmobilecomputing.techtarget.com/definition/wireless-LAN

**Payload** - Koristan teret

Na području informacijske sigurnosti, koristan teret označava odsječak koda pomoću kojeg se iskorištava određeni propust računala mete. Na primjer, koristan teret računalnog crva može sadržati modul za širenje vlastite kopije putem globalne mreže Internet.

http://searchsecurity.techtarget.com/definition/payload

**SQL** - Structured Query Language

SQL je programski jezik za pohranu, upravljanje i dohvat podataka pohranjenih u relacijskoj bazi podataka. SQL je najraširenij programskih jezik za upravljanje bazama podataka.

http://www.1keydata.com/sql/sql.html

# 9. Reference


[1] Gursev Kalra, Mobile application security testing, http://www.mcafee.com/us/resources/white-papers/foundstone/wp-mobile-app-security-testing.pdf

[2] Kunjan Shah, Penetration testing android aplications, http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pen-testing-android-apps.pdf

[3] Kunjan Shah, Penetration testing for iPhone/iPad applications, http://www.mcafee.com/us/resources/white-papers/foundstone/wp-pen-testing-iphone-ipad-apps.pdf

[4] Wikipedia, Mobile application development, http://en.wikipedia.org/wiki/Mobile_application_development

[5] Informacijska sigurnost, Zlonamjerni programi za mobilne platforme, http://sigurnost.lss.hr/images/dokumenti/lss-pubdoc-2010-10-001.pdf