UDC: 004.42:004.451 Info M str. 18-25

PRIJEDLOG POBOLJŠANOG SISTEMA DOZVOLA NA ANDROID PLATFORMI
PROPOSAL FOR AN IMPROVED SYSTEM PERMISSIONS ON THE ANDROID PLATFORM

ma Aleksandar Keleč, dipl. inž.

REZIME: U radu je detaljno analiziran sistem dozvola, kao jedan od najvažnijih sigurnosnih mehanizama Android platforme. Opisana je uloga ovog sistema na različitim slojevima Android arhitekture. Pored toga, opisani su izazovi sa kojima je suočen ovaj mehanizam, a posebno su analizirane posljedice nepravilne upotrebe sistema dozvola. Posebna pažnja posvećena je mehanizmima za verifikaciju aplikacija prilikom njihove objave, pri čemu je pokazana njihova nedovoljna efikasnost kada je u pitanju zloupotreba sistema dozvola. Kao praktični dio rada, razvijene su dvije Android aplikacije koje demonstriraju pokušaj zloupotrebe sistema dozvola u cilju izvođenja odgovarajućih malicioznih aktivnosti. Dodatno, izveden je pokušaj objavljivanja ovih aplikacija na Google Play Store-u. Na kraju, dati su prijedlozi za poboljšanje sistema dozvola, kao i sigurnosne preporuke za povećanje nivoa sigurnosti cjelokupne platforme i krajnjih korisnika.

KLJUČNE REČI: Android, sistem dozvola, sigurnost aplikacija

ABSTRACT: This paper provides an analysis of a System Permissions as one of the most important security mechanisms of the Android platform. The roles of this system in the different layers of Android architecture were described. Additionally, the challenges facing this mechanism are described and the consequences of improper use of the System Permissions are particularly analyzed. A special attention is given to the process of publishing applications and mechanisms for the verification of applications as well as their lack of efficiency when it comes to abuse of the System Permissions. As a practical work, the two Android applications, that demonstrate an attempt to abuse the System Permissions in order to perform the certain malicious activities, were developed. In addition, publication of these applications on the Google Play Store is attempted. At the end, the proposal for improving the System Permissions is given, as well as the security guidelines that should improve the security of the entire platform and its users.

KEY WORDS: Android, System Permissions, application security

1. UVOD

Sve veća upotreba pametnih telefona (eng. *smartphones*) i usluga koje su dostupne preko ovih uređaja uticala je i na enormno povećanje broja aplikacija koje nude ove usluge. Kada su aplikacije za pametne telefone u pitanju, ljudi obično ne obraćaju pažnju na moguće posljedice preuzimanja i korištenja tih aplikacija. Prilikom instalacije aplikacije na uređaj, najčešće se ne čitaju uslovi korištenja aplikacije (eng. *Terms and conditions*), što potencijalnim napadačima otvara prostor da različite maliciozne funkcionalnosti ugrade u aplikaciju. Ovo se posebno odnosi na uređaje sa Android operativnim sistemom (Android OS) koji je, u trećem kvartalu 2016. godine, imao globalni tržišni udio od 87.8% [1].

U ovom radu analizirana je sigurnosna arhitektura Android platforme, pri čemu je poseban osvrt dat na sistem dozvola (eng. *System Permissions*) kao jedan od najznačajnijih sigurnosnih mehanizama Android platforme. Pri tome, opisani su izazovi sa kojima je suočen ovaj mehanizam, a posebno su analizirane posljedice nepravilne upotrebe sistema dozvola. Pored toga, analiziran je postojeći, unaprijeđeni sistem dozvola kao i njegovi nedostaci i moguće zloupotrebe. Dodatno, razvijene su dvije Android aplikacije koje demonstriraju pokušaj zloupotrebe sistema dozvola u cilju izvođenja odgovarajućih malicioznih aktivnosti. Posebna pažnja u radu posvećena je i mehanizmima za verifikaciju aplikacija prilikom njihove objave na najvećoj i jedinoj legitimnoj *online* prodavnici sa Android aplikacijama − *Google Play Store*-u. U skladu s tim, izveden je i pokušaj objavljivanja navedenih aplikacija na *Google Play Store*-u, u cilju ilustracije neefikasnosti mehanizama za verifikaciju aplikacija.

U drugom poglavlju analizirani su sigurnosti aspekti Android platforme, pri čemu je fokus stavljen na sistem dozvola kao centralni mehanizam zaštite Android OS-a, aplikacija i njihovih resursa, kao i krajnjih korisnika i njihove privatnosti. Opisan je koncept dozvola, primjena sistema dozvola na nivou operativnog sistema i na aplikativnom nivou, dat je pregled tipova dozvola, opisan je proces upravljanja dozvolama i data analiza novog sistema dozvola koji se primjenjuje od verzije 6 Android OS-a. U trećem poglavlju opisani su nedostaci postojećeg sistema dozvola kao i mehanizama za verifikaciju aplikacija prilikom njihove objave na *Google Play Store*-u. Opisani su praktični primjeri u vidu dvije aplikacije koje imaju za cilj da prikažu slabe strane postojećeg sistema dozvola, pri čemu je izveden i pokušaj objavljivanja ovih aplikacija. U četvrtom poglavlju, kao rezultat ovog rada, dati su prijedlozi za poboljšanje sistema dozvola, kao i sigurnosne preporuke za povećanje nivoa sigurnosti cjelokupne platforme, koje bi posebno pozitivno trebale uticati na očuvanje privatnosti krajnjih korisnika. Dodatno, razmatrana su i potencijalna poboljšanja vezana za mehanizme verifikacije aplikacija prilikom njihove objave, koja treba da doprinesu smanjenju broja objavljenih malicioznih aplikacija. Na kraju rada data je kratka rekapitulacija istraživanja i ukazano je na dalje pravce mogućih istraživanja i budući rad.

2. SISTEM DOZVOLA NA ANDROIDU

Dozvole su sigurnosni mehanizam za kontrolu pristupa osjetljivim komponentama i funkcionalnostima Android platforme, kao što su funkcije kamere, GPS-a, *Bluetooth*-a, funkcije za uspostavljanje telefonskih poziva, SMS/MMS funkcije,

mrežne konekcije, itd. Osim što se koriste za zaštitu sistemskih resursa, dozvole štite i aplikacije i njihove komponente od neovlaštenog pristupa i zloupotreba.

2.1. *Primjena sistema dozvola na nivou Linux kernela*

Arhitektura Android platforme sastoji se iz četiri sloja: Linux kernel, izvršni sloj (eng. *Runtime layer*), aplikativni *framework* (eng. *Application framework*) i sloj aplikacija (eng. *Applications*) [2]. S obzirom na to da je Linux jedan od najpoznatijih *open source* projekata koji nosi epitet sigurnog, povjerljivog i stabilnog softvera, ne iznenađuje činjenica da je upravo Linux kernel izabran kao osnova Android operativnog sistema i cjelokupne Android platforme. Pored toga što je zadužen za upravljanje važnim komponentama Android platforme, kao što su procesi, memorija, fajl sistem itd, Linux je također veoma važna komponenta u sigurnosnoj arhitekturi platforme.

Sigurnosne karakteristike na nivou Linux kernela bazirane su na konceptu izolacije svake aplikacije u toku njenog izvršavanja poznatom kao *Application Sandbox* (AS). AS mehanizam podrazumijeva dodjelu jedinstvenog korisničkog identifikatora – UID (eng. *User Identifier*) i identifikatora grupe – GID (eng. *Group Identifier*) svakoj aplikaciji prilikom njene instalacije i pokretanja te aplikacije kao odvojenog procesa, kako bi se obezbijedio određeni nivo sigurnosti između aplikacija. Prilikom instalacije, svakoj aplikaciji dodjeljuju se jedinstveni UID i GID, koji se ne mijenjaju tokom cijelog životnog vijeka aplikacije na uređaju. Drugim riječima, svakoj instaliranoj aplikaciji se pridružuje odgovarajući Linux korisnik (eng. *Linux user*). Korisničko ime Linux korisnika ima format *app_x*, a odgovarajući UID se kreira prema formuli *Process.FIRST_APPLICATION_UID + x*, gdje je *Process.FIRST_APPLICATION_UID* konstanta koja definiše početnu vrijednost (koja iznosi 10000) u skupu mogućih vrijednosti koje se mogu dodijeliti instaliranim aplikacijama kao UID i GID. Na primjer, ako instalacioni fajl aplikacije nosi naziv *app1.apk*, onda će, nakon instalacije, aplikacija dobiti korisničko ime *app_1* i UID koji ima vrijednost 10001 [3].

Koncept Linux korisnika i grupa pogodan je da se ograniči pristup sistemskim resursima, tako što se komponenti, koja želi pristupiti, dodijeli odgovarajući UID ili GID. Ovakav tip restrikcija moguće je primijeniti na sistemske resurse kao što su fajlovi, drajveri uređaja, mrežni soketi, kamera, eksterne memorije i sl. Klasičan primjer primjene dozvola fajl sistema je u ograničavanju pristupa funkcionalnostima kamere, od strane aplikacija. Ograničenje je postignuto postavljanjem dozvole sa vrijednošću 0660 na drajver kamere */dev/cam*, za korisnika *root* i grupu *camera* [3]. To znači da samo procesi koji su pokrenuti sa *root* korisničkim privilegijama ili pripadaju grupi *camera*, imaju pristup funkcionalnostima kamere uređaja. Mapiranja između naziva dozvola i odgovarajućih grupa definisana su u fajlu *platform.xml*[1], čiji je izvod prikazan na sl. 2.1. Ukoliko u toku instalacije ili rada aplikacija zahtijeva dozvole za kameru i ukoliko ih korisnik odobri, onda se ta aplikacija pridružuje aplikacijama koje imaju GID *camera* i na taj način joj se omogućava da koristi funkcionalnosti kamere uređaja.

```
<permissions>
    <permission name="android.permission.BLUETOOTH" >
        <group gid="net_bt" />
    </permission>
    <permission name="android.permission.CAMERA" >
        <group gid="camera" />
    </permission>
    <permission name="android.permission.INTERNET" >
        <group gid="inet" />
    </permission>
    <permission name="android.permission.READ_LOGS" >
        <group gid="log" />
    </permission>
</permissions>
```

Slika 2.1 - *Mapiranje između naziva dozvola i Linux grupa*

Dakle, na nivou Linux kernela sprovođenje sistema dozvola obavlja se provjeravanjem da li je aplikacija uključena u odgovarajuću Linux grupu. Samo članovi ove grupe imaju pristup zaštićenim funkcionalnostima. Ukoliko korisnik omogući aplikaciji zahtijevane dozvole u toku instalacije ili u toku rada, aplikacija će dobiti odgovarajući GID i na taj način imati pristup zaštićenim resursima.

Pored sistemskih resursa, sistem dozvola koristi se i za zaštitu komponenata aplikacija. Da bi koristila zaštićenu komponentu aplikacije A, aplikacija B mora zahtijevati dozvolu za njeno korištenje, nakon čega aplikacija A mora odobriti zahtijevanu dozvolu. Više riječi o primjeni sistema dozvola na aplikativnom nivou biće u narednom odjeljku.

2.2. *Primjena sistema dozvola na aplikativnom nivou*

Kako bi mogao koristiti zaštićene komponente, programer aplikacije, tj. aplikacija, mora zahtijevati odgovarajuće dozvole, specifikovane u fajlu *AndroidManifest.xml*. Prilikom instaliranja aplikacije ili prilikom pokretanja i rada aplikacije (od verzije 6 Android OS-a), sistem prikazuje dijalog prozor krajnjem korisniku sa listom dozvola koje aplikacija zahtijeva, pri čemu korisnik mora da odobri tražene dozvole kako bi se aplikacija uspješno instalirala na uređaj, odnosno uspješno izvršavala.

Kao što je navedeno u prethodnom odjeljku, svaka aplikacija koja se izvršava na Android platformi podrazumijevano dobija svoj identitet u vidu pridruženih UID i GID vrijednosti. Pored toga, i komponente operativnog sistema imaju svoj UID i GID pomoću kojih operativni sistem razlikuje ove komponente, povećavajući na taj način sigurnost cjelokupne platforme. Jedan od tih identiteta vezan je za proces *System Server*, kao komponentu koja objedinjuje sve servise koje nudi Android operativni sistem. *System Server* ima privilegovan pristup resursima operativnog sistema i svaki servis koji se izvršava u okviru *System Server*-a omogućava kontrolisan pristup pojedinim funkcionalnostima od strane drugih komponenata OS-a i aplikacija. Ovaj kontrolisan pristup baziran je upravo na sistemu dozvola [3].

Svaki servis Android OS-a, odnosno svaka njegova metoda, zaštićena je posebnom labelom koja se zove dozvola.

1 https://android.googlesource.com/platform/frameworks/base/+/master/data/etc/platform.xml

Prilikom svakog poziva jedne takve metode od strane nekog servisa, provjerava se da li je tom servisu pridružena odgovarajuća dozvola. Ukoliko jeste, metoda će se uspješno izvršiti, a ukoliko nije, biće bačen sigurnosni izuzetak (eng. *exception*) vezan za provjeru dozvola (najčešće *SecurityException*). Na primjer, ako aplikacija želi da koristi funkcionalnosti kamere, u fajl *AndroidManifest.xml* potrebno je dodati sljedeću liniju: *<uses-permission android:name=”android.permission.CAMERA”/>*. Ovo znači da, ako korisnik odobri zahtijevane dozvole, aplikacija dobija pristup API-ju kamere i može da koristi sve funkcionalnosti kamere, kao što su fotografisanje, snimanje video zapisa, skeniranje i sl.

2.3. *Tipovi dozvola*

Opisani model dozvola donosi značajan napredak u sprovođenju sigurnosti na Android platformi. Međutim, ovaj model ujedno je i neefikasan iz razloga što su sve dozvole ravnopravne sa stanovišta sigurnosti. Na primjer, servis koji omogućava slanje SMS poruka je dosta kritičniji sa sigurnosnog aspekta od servisa koji omogućava postavljanje alarma na uređaju, ali se dozvole za oba servisa tretiraju na isti način. Da bi ovaj problem bio riješen, na Androidu su uvedena četiri nivoa zaštite dozvola: *normal*, *dangerous*, *signature* i *signatureOrSystem* [4]. Oznaka za nivo zaštite dozvole može biti navedena unutar OS-a (kada su u pitanju sistemske dozvole) ili dodijeljena od strane programera aplikacije, unutar deklaracije o korisnički definisanim dozvolama (u fajlu *AndroidManifest.xml*). Nivo zaštite dozvole utiče na odluku da li aplikaciji treba odobriti zahtijevanu dozvolu ili ne. Da bi bile odobrene, dozvole sa oznakom *normal* moraju jednostavno biti navedene u okviru fajla *AndroidManifest.xml*. Dozvole sa oznakom *dangerous*, pored navođenja u manifestu, moraju biti odobrene od strane korisnika u toku instaliranja ili pokretanja aplikacije. Dozvolu sa oznakom *signature* odobrava sistem samo u slučaju da je aplikacija, koja je zahtijeva, potpisana istim digitalnim sertifikatom kao i aplikacija za koju je vezana data dozvola. Dozvola sa oznakom *signatureOrSystem* biće odobrena ili u slučaju da su obje aplikacije potpisane istim digitalnim sertifikatom ili ako je aplikacija koja zahtijeva dozvolu sistemska aplikacija. Na sl. 2.2 prikazani su nivoi zaštite za neke od standardnih dozvola na Androidu koji su definisani u okviru fajla *AndroidManifest.xml*[2], koji se nalazi u sklopu izvornog koda Android *framework*-a.

```
<manifest>
    ...
    <permission android:name="android.permission.SEND_SMS"
            android:protectionLevel="dangerous" />
    <permission android:name="com.android.alarm.permission.SET_ALARM"
            android:protectionLevel="normal" />
    <permission android:name="android.permission.CAMERA"
            android:protectionLevel="dangerous" />
    <permission android:name="android.permission.ACCESS_MOCK_LOCATION"
            android:protectionLevel="signature" />
    <permission android:name="android.permission.INSTALL_PACKAGES"
            android:protectionLevel="signature|privileged" />
    ...
</manifest>
```

Slika 2.2 - *Nivoi zaštite za neke od dozvola na Androidu*

Sada je jasno da odgovarajući servisi imaju različite nivoe zaštite u skladu sa njihovom ulogom, ali i u zavisnosti od mogućnosti koje bi potencijalni napadač imao u slučaju zloupotrebe odgovarajućeg servisa. Tako na primjer, servis koji omogućava postavljanje alarma na uređaju ima nivo zaštite *normal*, dok su servisi koji omogućavaju slanje SMS poruka i pristup funkcionalnostima kamere kritičniji sa aspekta sigurnosti, pa shodno tome imaju viši nivo zaštite − *dangerous*. Nivo zaštite *signature* ima npr. servis koji omogućava programeru da kreira testnog provajdera za testiranje funkcionalnosti navigacije. Ovaj nivo zaštite obično se dodjeljuje u slučaju kada postoje dvije aplikacije kreirane od strane istog programera (potpisane istim sertifikatom) od kojih jedna koristi neke funkcionalnosti druge aplikacije. Nivo zaštite *signatureOrSystem* ima npr. servis koji omogućava direktnu instalaciju APK fajla (eng. *Android Application Package*), tj. aplikacije, na uređaj. Podrazumijevano, programeri *third-party* aplikacija nemaju pristup funkcionalnostima koje su zaštićene sistemskim dozvolama nivoa zaštite *signature* i *signatureOrSystem*.

2.4. *Upravljanje dozvolama*

Sistemski servis *PackageManagerService* zadužen je za upravljanje instalacijom, deinstalacijom, kao i ažuriranjem aplikacija na Android OS-u. Pored toga, ovaj servis odgovoran je i za upravljanje dozvolama. On čuva informacije koje omogućavaju provjeru da li je nekom Android paketu pridružena odgovarajuća dozvola. Pored toga, tokom instalacije ili ažuriranja aplikacije, *PackageManagerService* provodi različite provjere, kako bi ustanovio da integritet sistema dozvola nije narušen u toku ovih procesa. Metode ovog servisa predstavljaju posljednju kariku u lancu provjera dozvola.

PackageManagerService skladišti sve podatke vezane za dozvole *third-party* aplikacija u fajl */data/system/packages.xml* [3]. Ovaj fajl se koristi kao perzistentno skladište između “dva pokretanja” sistema, dok se u toku rada uređaja sve informacije o dozvolama čuvaju u RAM memoriji, čime se postiže efikasniji i brži rad sistema.

2.5. *Sistem dozvola na Androidu 6*

Sa verzijom 6 Android OS-a, pod nazivom *Marshmallow*, sistem dozvola znatno je izmijenjen i stepen zaštite privatnosti korisnika bitno je povećan. Način na koji funkcioniše novi sistem dozvola sličan je onom koji već duži period postoji na iOS operativnom sistemu, a to su dozvole na zahtjev (eng. *Runtime Permissions*[3]). To znači da korisnik treba da odobri određene dozvole aplikaciji u trenutku njenog izvršavanja a ne prilikom instaliranja aplikacije, kao što je bio slučaj u ranijim verzijama Android OS-a. Na primjer, ako aplikacija želi u jednom trenutku da pokrene i koristi kameru uređaja, korisniku će biti prikazan dijalog prozor u kojem će on moći da odluči da li da odobri aplikaciji traženu dozvolu ili ne. Na ovaj način povećava se stepen upravljanja dozvolama od strane krajnjih korisnika. Takođe, korisniku je omogućeno selektivno dodjeljivanje dozvola, što ranije nije bio slučaj. Na primjer, aplikaciji za fotografisanje ko-

2 https://android.googlesource.com/platform/frameworks/AndroidManifest.xml

3 http://developer.android.com/training/permissions/requesting.html

risnik može odobriti dozvolu za kameru, ali ne mora dozvoliti pristup lokacijskim podacima.

Novi sistem dozvola donio je promjene i u samoj implementaciji Android aplikacija. Android *Marshmallow* uveo je API nivoa 23, koji je potrebno koristiti kako bi aplikacije uspješno funkcionisale sa dozvolama na zahtjev. Sve Android aplikacije moraju biti ažurirane tako da podržavaju novi API.

Korisnici mogu dozvolama da upravljaju i kroz menadžer dozvola (eng. *Permission Manager*). Ovaj menadžer se nalazi u sekciji *Apps* u meniju *Settings*. Na listi dozvola nalaze se različite opcije, kao što su: *body sensors*, *calendar*, *camera*, *contacts*, *location*, *microphone*, *phone*, *SMS*, *storage* i tri nove kategorije koje donosi *Marshmallow*, a to su: *car information, read instant messages* i *write instant messages.* Odabirom neke od dozvola, korisniku je vidljiva informacija o tome kojim aplikacijama je omogućena ta dozvola. Upravljanje dozvolama je pojednostavljeno – sve što korisnik treba da uradi jeste da, korištenjem *switch* dugmića, odobri ili zabrani dozvolu. Na ovaj način dozvole za aplikacije su transparentnije i korisnicima je razumljivije na koji način odobravaju dozvole i šta svaka od instaliranih aplikacija, u stvari, zahtijeva. Ukoliko im neka dozvola nije logična, kao npr. da neka igra zahtijeva korištenje kamere, lako mogu da posumnjaju da je aplikacija potencijalno maliciozna.

Mišljenja su podijeljena o tome da li novi sistem dozvola donosi značajno poboljšanje, jer često korisnici nisu svjesni činjenice šta treba da dozvole aplikaciji, a šta ne. U slučaju da korisnik pojedine dozvole ne odobri aplikaciji, aplikacija će se nastaviti izvršavati sa ograničenim mogućnostima. Ovo pruža prostor novim potencijalnim problemima u smislu da može doći do neuobičajenog i nekonzistentnog ponašanja aplikacije. Takođe, ovakav pristup otvara i nove mogućnosti za potencijalne napadače, da iskoriste ova ograničenja nastala usljed neodobravanja određenih dozvola aplikaciji. U svakom slučaju, vrijeme će pokazati da li je novi sistem dozvola bolji od starog, a s obzirom na to kojim tempom Android uređaji prelaze na najnoviju verziju OS-a, činjenica je da će još dugo vremena stari sistem dozvola ostati dominantan. Raspodjela verzija Android OS-a, prema podacima iz februara 2017. godine[4], pokazuje da je verzija 6 prisutna na 30.7% uređaja, a najnovija, verzija 7 (*Nougat*), na 1.2% uređaja. Činjenica da se Android *Marshmallow* pojavio prije skoro dvije godine, govori da je prelazak uređaja na najnoviju verziju Androida spor i kompleksan proces.

Kada je u pitanju najnovija verzija Android OS-a, *Nougat*, osim manjih izmjena vezanih za povećane restrikcije kod pristupa osjetljivim direktorijumima i privatnim podacima, nisu uvedene značajnije izmjene u okviru sistema dozvola [5].

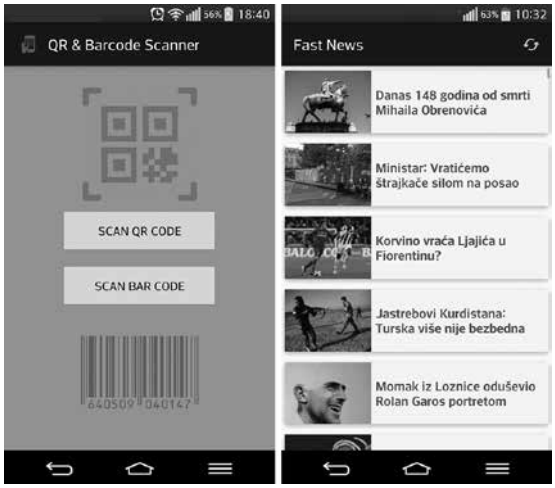
3. NEDOSTACI POSTOJEĆEG SISTEMA DOZVOLA

Iako unaprijeđen, postojeći sistem dozvola ima dosta manjkavosti i nedorečenosti. I pored toga što je sistem prilično jasan programerima aplikacija, kao i korisnicima koji imaju određen nivo IT znanja, za prosječnog korisnika je i dalje nedovoljno razumljiv i slabo dokumentovan.

4 https://developer.android.com/about/dashboards/index.html

3.1. *Praktični primjeri zloupotrebe sistema dozvola*

Da bi se pokazale slabosti postojećeg sistema dozvola, kao i neefikasnost mehanizama za verifikaciju aplikacija prilikom njihove objave, u okviru ovog rada razvijene su dvije aplikacije, *QR & Barcode Scanner* [6] i *Fast News* [7]. Zadatak aplikacija je da demonstriraju pokušaj zloupotrebe sistema dozvola u cilju izvođenja odgovarajuće maliciozne aktivnosti. Na sl. 3.1 prikazani su početni ekrani aplikacija *QR & Barcode Scanner* i *Fast News*, respektivno.



Slika 3.1 - *Početni ekrani aplikacija QR & Barcode Scanner i Fast News*

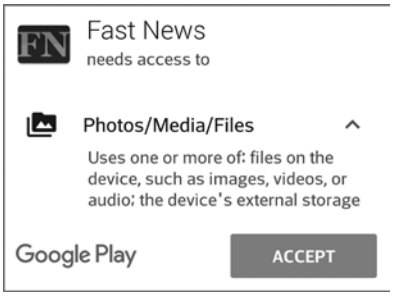
QR & Barcode Scanner je Android aplikacija razvijena sa ciljem da ilustruje kako je moguće zaobići sigurnosne mehanizme Android platforme i izvršiti napade kojima se narušava privatnost korisnika. Sama aplikacija razvijena je tako da korisnik nije svjestan činjenice da je žrtva malicioznog napada. Osnovna svrha aplikacije je skeniranje QR koda i barkoda. Kako bi aplikacija mogla da obezbijedi funkcionalnost skeniranja, prilikom instaliranja, odnosno pokretanja aplikacije, potrebno je omogućiti joj dozvolu za korištenje kamere. Pored svoje osnovne funkcije, aplikacija obavlja i jednu malicioznu aktivnost, a to je fotografisanje korisnika i slanje fotografija na specifikovanu e-mail adresu. Kako bi spriječila korisnika da posumnja u rad aplikacije, *QR & Barcode Scanner* aplikacija mora ukloniti bilo kakvu naznaku da se izvodi napad kojim će se narušiti privatnost korisnika. U skladu s tim, aplikacija će, prije izvođenja napada, isključiti zvuk kamere koji se reprodukuje u trenutku slikanja, kao i vibriranje uređaja, te sakriti prikaz ekrana kamere (eng. *camera preview*).

Fast News je Android aplikacija čiji je cilj da ilustruje moguće zloupotrebe mehanizama za izvršavanje udaljenog koda na Android uređajima, kao i sistem dozvola. Osnovna svrha aplikacije je da omogući korisniku pregled i čitanje najnovijih vijesti iz zemlje i svijeta, pri čemu se mogu naći sve kategorije vijesti. Kako bi aplikacija mogla da obezbijedi osnovnu funkcionalnost, potrebno joj je omogućiti korištenje Interneta. Pored svoje osnovne funkcionalnosti, aplikacija obavlja i jednu malicioznu aktivnost: preuzima i izvršava maliciozan kôd, kojim se narušava privatnost korisnika.

3.2. *Nedostaci sistema dozvola*

Prvi i veliki nedostatak sistema dozvola leži u činjenici da nije moguće utvrditi da li aplikacija zahtijevanu dozvolu koristi samo u legitimne svrhe, tj. za primjenu za koju joj treba data dozvola, ili je zloupotrebljava, odnosno koristi i za odgovarajuće maliciozne namjene. Ovaj problem sasvim efikasno demonstrira aplikacija *QR & Barcode Scanner*. S obzirom na to da je za skeniranje QR koda ili barkoda neophodna upotreba kamere, aplikacija *QR & Barcode Scanner* potpuno opravdano i legitimno zahtijeva dozvolu za upotrebu kamere i ista joj mora biti dodijeljena kako bi aplikacija uspješno funkcionisala. Međutim, u ovom slučaju to se pokazalo kao „mač sa dvije oštrice“ jer se, odobravanjem pristupa kameri, aplikaciji omogućilo da iskoristi sve mogućnosti koje kamera nudi, poput fotografisanja, snimanja video sadržaja, i sl. Dakle, radi se o nimalo bezazlenom problemu i velikom izazovu koji je stavljen pred sistem dozvola.

Drugi nedostatak sistema dozvola leži u činjenici da sistem, između ostalog, dodjeljuje aplikacijama i dozvole koje im nisu potrebne. Određena istraživanja [8] pokazala su da oko trećina objavljenih aplikacija zahtijeva više dozvola nego što im je potrebno. Iako princip rada mehanizama koji verifikuju aplikacije prilikom objave na *Google Play Store*-u nije poznat, veoma je jasno da ovi mehanizmi nisu dovoljno efikasni kada je u pitanju procjena dozvola koje aplikacije zahtijevaju. Kao dokaz ovoj tvrdnji, može se uzeti primjer *Fast News* aplikacije. Iako je aplikacija prilično jednostavna i ima samo jednu namjenu: da prikaže korisniku listu sa vijestima i da prikaže detalje izabrane vijesti, ipak je bez problema dobila dozvolu za čitanje i upis u eksternu memoriju uređaja. Na sl. 3.2 prikazan je prozor sa dozvolama koje zahtijeva aplikacija *Fast News* prilikom instalacije. Prošireni prikaz daje detaljan opis zahtijevane dozvole, uključujući primjere tipova fajlova kojim će aplikacija imati pristup. Međutim, postavlja se pitanje koliko prosječnom korisniku znači ovakav opis i da li on uopšte razumije sve što je napisano. U konkretnom primjeru, spominje se eksterna memorija uređaja, što za prosječnog korisnika ne predstavlja naročito korisnu informaciju.



Slika 3.2 - *Prozor sa dozvolama pri instalaciji aplikacije Fast News*

Sljedeći nedostatak sistema dozvola, koji može prouzrokovati značajne probleme korisnicima, vezan je za podjelu dozvola na grupe. S obzirom da postoji preko stotinu različitih dozvola koje aplikacija može zahtijevati, *Google* je sve dozvole grupisao po sličnosti, tako da prozor sa dozvolama bude što pregledniji i jasniji za korisnika [9]. Tako na primjer, u grupi „SMS“ nalaze se sve dozvole vezane za SMS i MMS funkcionalnosti, poput čitanja i slanja SMS i MMS poruka i sl. Problem koji se veže za raspodjelu dozvola u grupe, leži u činjenici da se, odobravanjem bilo koje dozvole, automatski odobravaju i sve ostale dozvole iz pripadajuće grupe [10]. To znači da, ukoliko korisnik odobri aplikaciji dozvolu za čitanje SMS poruka, aplikaciji će automatski biti dozvoljeno i slanje SMS poruka bez korisnikovog znanja. Ovakav pristup, iako izgleda donekle logičan i razumljiv, otvara značajne mogućnosti za potencijalne napadače, u smislu lakšeg dolaska do povjerljivih korisničkih informacija.

3.3. *Nedostaci mehanizama za verifikaciju aplikacija*

Google Play Store predstavlja jedini pravi izvor Android aplikacija, preporučen od strane kompanije *Google* i cijele Android zajednice. Prema određenim istraživanjima [11], sprovedenim u decembru 2016. godine, u ovoj *online* prodavnici nalazi se preko dva i po miliona aplikacija. S obzirom na to da broj korisnika Android uređaja, kao i broj programera Android aplikacija, raste iz dana u dan, jasno je da se i broj aplikacija povećava enormnom brzinom.

Da bi zaštitio krajnje korisnike od malicioznih aplikacija, Android razvojni tim provodi različite vidove provjera i testiranja aplikacija kao i programera tih aplikacija, prije njihove objave na *Google Play Store*-u. *Google Play Review* je sigurnosni mehanizam koji ima zadatak detekcije potencijalno opasnih aplikacija za krajnje korisnike i onemogućavanja takvim aplikacijama da se nađu na *Google Play Store*-u i samim tim da dođu do korisnika. To je jedan sveobuhvatan proces koji koristi različite metode za analizu aplikacija, kao što su: statička analiza koda aplikacija, dinamička analiza aplikacija, korištenje heuristika za pronalaženje sličnosti sa poznatim malicioznim aplikacijama, provjera potpisa aplikacije sa bazom prikupljenih potpisa malicioznih apllikacija, utvrđivanje povezanosti programera sa nekim prethodnim malicioznim aplikacijama, korištenje podataka drugih kompanija i istraživača koji se bave sigurnosću, itd [12].

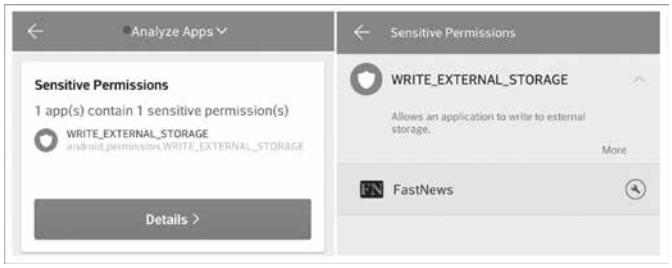
Na osnovu navedenog, može se zaključiti da svaku aplikaciju čeka potencijalno ozbiljan proces revizije prije njenog objavljivanja na *Google Play Store*-u. Dodatno, periodičnim skeniranjem postojećih aplikacija, sprovodi se konstantno uklanjanje malicioznih aplikacija sa *Google Play Store*-a. Prema [12], manje od 1% Android uređaja „pogođeno“ je malicioznim aplikacijama koje se nalaze na *Google Play Store*-u.

Ipak, u ovom radu pokazano da je i ovako detaljan proces revizije moguće zaobići i na *Google Play Store*-u objaviti aplikacije koje obavljaju različite maliciozne aktivnosti i narušavaju privatnost krajnjih korisnika. Aplikacije razvijene u okviru ovog rada uspješno su objavljene na *Google Play Store*-u i, za manje od godinu dana, dostigle su nekoliko hiljada instalacija na korisničke uređaje. Dakle, ni nakon skoro godinu dana, mehanizmi za verifikaciju aplikacija nisu detektovali maliciozne aktivnosti koje sprovode aplikacije *QR & Barcode Scanner* i *Fast News*. Na taj način, identifikovan je ozbiljan problem na relaciji sistem dozvola - mehanizmi za verifikaciju aplikacija i u nastavku su dati prijedlozi za poboljšanje sistema dozvola i pokušaj prevazilaženja ovog problema.

4. POBOLJŠANI SISTEM DOZVOLA

Kao što je ranije rečeno, novi sistem dozvola koji funkcioniše na principu „dozvola na zahtjev“, donio je unapređenje u smislu prenosa veće odgovornosti na krajnje korisnike i prepuštanja im odluke o odobravanju pojedinih dozvola aplikacijama. Pored toga, sistem je sigurniji jer upozorava korisnika svaki put kad aplikacija pokuša da pristupi nekoj funkcionalnosti uređaja za koju je potrebna odgovarajuća dozvola. Međutim, ovakav pristup stalnog zahtijevanja od korisnika da odobri neku dozvolu u toku rada aplikacije, može biti naporan za korisnika, pa se često korisnici odlučuju da isključe i zanemare ovu opciju. Naime, prilikom drugog pokretanja aplikacije, korisniku će se na dijalogu za dozvole pojaviti opcija „Ne pitaj me više“ (eng. *Never ask again*) u vidu *checkbox* dugmića. Ukoliko korisnik selektuje ovaj dugmić, aplikacija će smatrati da je korisnik odobrio datu dozvolu i za ubuduće i više se neće prikazivati ovakav dijalog prozor. Dodatno, korisniku se može ponuditi i opcija da promijeni podešavanja dozvola vezanih za ovu aplikaciju, tako što će ga odvesti u podešavanja dozvola na uređaju, gdje on može dozvoliti ili zabraniti sve, ili pojedine, dozvole za bilo koju aplikaciju. Praksa je pokazala da korisnici često koriste ovu mogućnost kako bi isključili, za njih naporan, zahtjev za dozvole koji im se prikazuje pri svakom korištenju aplikacije. Na taj način praktično se dolazi do starog sistema dozvola i ne vidi se značajniji pomak u približavanju sistema dozvola krajnjim korisnicima.

Preporuke za poboljšanje sistema dozvola, koje su opisane u nastavku ovog rada, baziraju se upravo na ideji približavanja sistema krajnjim korisnicima i njegovog prilagođavanja korisnikovim očekivanjima. Prvi korak u tim adaptacijama bio bi da, nakon instalacije aplikacije, sistem prikaže korisniku koje su to dozvole, kritične sa aspekta sigurnosti, koje aplikacija zahtijeva. Pored toga, uz svaku dozvolu bili bi prikazani i primjeri mogućih zloupotreba date dozvole kao i odgovarajući savjeti koji bi podstakli korisnika na razmišljanje i pokušaj uspostavljanja veze između date dozvole i aplikacije kojoj je odobrena. Na taj način korisnik bi još jednom, nakon instalacije aplikacije, bio upozoren o ozbiljnosti situacije i o mogućim posljedicama sa kojima bi se mogao suočiti. Danas postoji nekoliko aplikacija koje koriste slične ideje da približe korisnicima sistem dozvola. Jedna od njih je aplikacija *ES File Explorer*[5], koja vrši skeniranje svake aplikacije nakon instalacije i prikazuje korisniku podatke o osjetljivim dozvolama, koje su odobrene aplikaciji. Primjer ovakvog skeniranja proveden nakon instalacije aplikacije *Fast News* prikazan je na sl. 4.1.

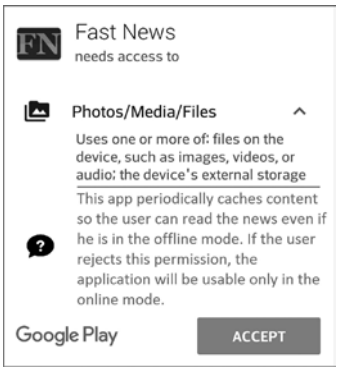


Slika 4.1 - *Prikaz osjetljivih dozvola u okviru aplikacije ES File Explorer*

5 https://play.google.com/store/apps/details?id=com.estrongs.android.pop

Na osnovu prikazanog primjera, vrlo je vjerovatno da će korisnik više pažnje obratiti nakon ponovnog i detaljnijeg prikaza dozvola koje je aplikacija dobila nakon instalacije, nego u toku instalacije.

Posebnu ulogu u novom sistemu dozvola imali bi i programeri aplikacija, čiji zadatak bi bio da obezbijede detaljno obrazloženje za svaku od dozvola koje njihova aplikacija zahtijeva. Dakle, predloženi sistem zahtijevao bi i minimalne izmjene na formi za dodavanje nove aplikacije u okviru *web* stranice *Google Play Store*-a, gdje bi se za svaku zahtijevanu dozvolu nalazilo polje u koje bi programer unosio detaljan opis date dozvole. Opis bi morao da sadrži obrazloženje za zahtijevanje odgovarajuće dozvole od strane aplikacije. Zahvaljujući ovakvom pristupu, tim za verifikaciju aplikacija imao bi bolji uvid u zahtijevane dozvole i lakše bi mogao da testira funkcionalnosti aplikacije za koje su tražene dozvole potrebne. Na ovaj način u velikoj mjeri bi se eliminisao nedostatak postojećeg sistema dozvola, tj. minimizovalo bi se dodjeljivanje viška dozvola aplikacijama. Svaka dozvola za koju se ne bi pronašlo opravdanje u okviru aplikacije, bila bi automatski obrisana iz fajla *AndroidManifest.xml* i na taj način bi se spriječile mnoge zloupotrebe. Dodatno, opis koji programeri dostave uz svoje aplikacije, bio bi prikazan i krajnjim korisnicima u trenutku instaliranja aplikacije. Na taj način, korisnicima bi dodatno bili približeni razlozi za zahtijevanje pojedinih dozvola i samim tim bi se oni aktivnije uključili u cjelokupni sistem. Pored toga, postojala bi i opcija pomoću koje bi korisnici mogli postavljati pitanja timu iz *Google Play Store*-a ali i samim programerima, u slučaju nekih nejasnoća ili zahtijevanja detaljnijeg obrazloženja za korištenje pojedinih dozvola. Na sl. 4.2 prikazan je prozor sa dozvolama za aplikaciju *Fast News*, proširen za obrazloženje koje dolazi od programera aplikacije. Crvenim bojom označeno je obrazloženje programera, a pored njega se nalazi dugmić pomoću kojeg korisnik može postavljati pitanja i davati sugestije ili svoje sumnje u vezi pojedinih dozvola.



Slika 4.2 - *Prošireni prozor sa detaljnim objašnjenjem dozvola*

Pored detaljnog objašnjenja svake pojedinačne dozvole, predloženi sistem bi mogao uključiti i grafičke prikaze koji bi trebalo da doprinesu većem interesu korisnika i boljem razumijevanju sistema dozvola. Jedan prijedlog ovakvog pristupa opisan je u [13], gdje je ideja da se dozvole skaliraju prema njihovoj kritičnosti sa aspekta sigurnosti, i u skladu s tim da

se korisniku prikaže npr. skala sa maksimalno pet zvjezdica, gdje bi broj zvjezdica označavao nivo ozbiljnosti dozvole. Psihološki aspekt ovakvog pristupa bazira se na činjenici da korisnik brže primjećuje podatke koji su vizuelno predstavljeni, u odnosu na one koji su opisani tekstualno. Na taj način postiže se još jedan korak ka boljem animiranju korisnika i podizanju njihove svijesti o načinu funkcionisanja sigurnosnog sistema na njihovom uređaju.

Da bi se korišćenje dozvola potpuno opravdalo i da bi se minimizovale zloupotrebe, pored sistema dozvola, potrebno je razmotriti i odgovarajuće izmjene u okviru procesa objavljivanja aplikacija na *Google Play Store*-u. Praktično je pokazano da mehanizmi za verifikaciju aplikacija prilikom njihove objave ne funkcionišu na najbolji način i da se na *Google Play Store*-u mogu naći maliciozne aplikacije, kao i aplikacije koje imaju namjeru da to postanu. Osim toga, pokazano je da ovi mehanizmi ne obraćaju previše pažnju na dozvole koje aplikacija zahtijeva i tako svu odgovornost prebacuju na krajnje korisnike.

Jedan od razloga koji ide u prilog ovoj tvrdnji jeste činjenica da *Google* tim za verifikaciju aplikacija nema pristup izvornom kodu aplikacije koja se verifikuje. Umjesto toga, progamer, prilikom objave aplikacije, šalje samo njen APK fajl, iz kojeg je moguće dobiti izvorni kôd pomoću neke od tehnika inverznog inženjeringa, ali samo u slučaju da kôd prethodno nije obfuskovan. Međutim, u velikoj većini slučajeva, dostavlja se obfuskovan kôd, koji je znatno teži za razumijevanje i analizu. Obfuskacija koda predstavlja mijenjanje izvornog koda brisanjem tzv. *debugging* informacija i zamjenom svih smislenih naziva sa odgovarajućim besmislenim riječima, s ciljem da kôd postane teško čitljiv i da se onemogući rekonstrukcija originalnog koda dekompajliranjem obfuskovanog koda. Iako princip funkcionisanja mehanizama za verifikaciju nije poznat, može se zaključiti da pomenuta činjenica ima dosta uticaja na efikasnost ovih mehanizama. Statičkom analizom izvornog koda koji nije obfuskovan, dosta lakše i efikasnije bi se detektovale sumnjive radnje u tom kodu. Tako na primjer, ukoliko neka klasa implementira *Camera.PictureCallback* interfejs i metodu *onPictureTaken*, lako bi se došlo do zaključka da takav kôd koristi kameru za fotografisanje i za određeno procesiranje fotografije, kao što je to slučaj sa aplikacijom *QR & Barcode Scanner*. Međutim, u obfuskovanom kodu, ova klasa mogla bi da ima drugačiji naziv i na taj način bi se dosta teže detektovalo njeno prisustvo.

Iz navedenog razmatranja, postavlja se pitanje zašto *Google* ne zahtijeva originalan izvorni kôd od progamera. Odgovor na ovo pitanje nije u potpunosti poznat, ali se zna da on ima veze sa zaštitom autorskih prava programera, koja je sastavni dio *Google*-ove politike. Dobro definisana *Google*-ova politika, mogla bi isto tako da garantuje programerima da će se njihov izvorni kôd koristiti samo u svrhu verifikacije aplikacije i da neće biti korišten u bilo koje druge svrhe, niti zloupotrijebljen. Kada bi politika bila definisana na ovaj način, programeri bi, pored APK fajla, slali i komprimovan izvorni kôd, prilikom objave aplikacije. U tom slučaju, dosta teže bi bilo opravdati pojedine aktivnosti koje kôd namjerava da izvrši i na taj način bi *Google Play Review* mehanizam radio sa znatno većim procentom uspješnosti. Sve dok *Google* ne predstavi politiku sličnu pomenutoj, potencijalni napadači će lakše nalaziti način da zavaraju trag malicioznim akcijama.

5. BUDUĆI RAD I ZAKLJUČAK

U ovom radu dat je detaljan opis sistema dozvola, kao jednog od najvažnijih sigurnosnih mehanizama Android platforme. Detaljna analiza ovog sistema neophodna je kako bi se uočili potencijalni nedostaci i slabosti sistema, koje bi se mogle eksploatisati u maliciozne svrhe. U skladu s tim, prikazani su neki nedostaci postojećeg sistema dozvola i na praktičnim primjerima pokazano kako je moguće zloupotrijebiti ovaj sistem u cilju narušavanja privatnosti i nanošenja štete kranjim korisnicima.

Posebna pažnja posvećena je sprezi između sistema dozvola i mehanizama za verifikaciju aplikacija prilikom njihove objave na *Google Play Store*-u. Pokazana je neefikasnost ovih mehanizama kada je u pitanju zloupotreba sistema dozvola, pri čemu su objavljene dvije maliciozne aplikacije koje demonstriraju neke od zloupotreba. Aplikacije su za manje od godinu dana dostigle nekoliko hiljada instalacija, dovodeći u opasnost nekoliko hiljada korisnika Android uređaja. Aplikacije su objavljene samo u naučno-istraživačke svrhe, bez namjere autora da na bilo koji način nanese štetu krajnjim korisnicima.

U skladu sa identifikovanim problemima, u radu su dati prijedlozi za poboljšanje sistema dozvola koji bi trebalo da doprinesu boljem razumijevanju samog sistema od strane korisnika i na taj način povećaju ulogu krajnjih korisnika u čitavom procesu odlučivanja, šta dozvoliti nekoj aplikaciji a šta ne dozvoliti. Kada je sistem dozvola u pitanju, posebno važnu ulogu imaju krajnji korisnici, jer od njihovih odluka, na kraju, zavisi koliki procenat uspješnosti u ostvarivanju malicioznih namjera će imati potencijalni napadači. Zato je veoma važno da korisnici razumiju na kom principu funkcionišu dozvole i da ih nauče koristiti na pravi način.

Kada je budući rad u pitanju, u planu je razvoj analizatora aplikacija koji će imati za cilj da dodatno analizira aplikaciju i sve dozvole koje ona zahtijeva, prilikom objave. Alat bi funkcionisao na principu provjere usklađenosti odgovarajuće dozvole sa potrebama aplikacije, tj. za svaku zahtijevanu dozvolu moralo bi se naći opravdanje u okviru funkcionalnosti koje aplikacija nudi.

Rezultati do kojih se došlo u ovom radu mogu da posluže, kako razvojnom timu Android OS-a, tako i krajnjim korisnicima, u minimizaciji sigurnosnih propusta i napada. Pored toga, primjenom predloženih sugestija, broj malicioznih aplikacija objavljenih na *Google Play Store*-u trebao bi se svesti na minimum.

LITERATURA

[1] „Statista's Global market share", dostupno na: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/, posjećeno: 10.02.2017. godine.

[2] A. Keleč, „Detekcija i eksploatacija sigurnosnih propusta na Android platformi“, Master rad, Elektrotehnički fakultet, Banja Luka, 2016.
[3] Y. Zhauniarovich, „Android Security (and Not) Internals“, ASANI Book, 2014.
[4] „Android Permissions“, dostupno na: https://developer.android.com/guide/topics/manifest/permission-element.html, posjećeno: 10.02.2017. godine.
[5] „Android 7.0 Behavior Changes“, dostupno na: https://developer.android.com/about/versions/nougat/android-7.0-changes.html#perm, posjećeno: 10.02.2017. god.
[6] A. Keleč, „QR & Barcode Scanner“, Google Play Store, 30.03.2016, https://play.google.com/store/apps/details?id=com.akelec.qrbarcodescanner, posjećeno: 10.02.2017. godine.
[7] A. Keleč, “Fast News“, Google Play Store, 10.06.2016, https://play.google.com/store/apps/details?id=com.akelec.fastnews, posjećeno: 10.02.2017. godine.
[8] A. P. Felt, E. Chin, S. Hanna, D. Song and D. Wagner, „Android Permissions Demystified“, in Proceedings of the *18th ACM conference on Computer and communications security*, pp. 627-638 2011.
[9] „Review app permissions thru Android 5.9“, dostupno na: https://support.google.com/googleplay/answer/6014972, posjećeno: 10.02.2017. godine.
[10] „System Permissions“, dostupno na: https://developer.android.com/guide/topics/security/permissions.html, posjećeno: 10.02.2017. godine.
[11] „Number of available applications in the Google Play Store“, dostupno na: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/, posjećeno: 10.02.2017. godine.
[12] „How we keep harmful apps out of Google Play and keep your Android device safe“, 2016, dostupno na: https://static.googleusercontent.com/media/source.android.com/en//security/reports/Android_WhitePaper_Final_02092016.pdf, posjećeno: 10.02.2017. godine.
[13] M. Derks, „Fair Privacy: Improving Usability of the Android Permission System“, *Master thesis,* Radboud University, Nijmegen, 2015.

ma Aleksandar Keleč, dipl. inž., Elektrotehnički fakultet, Univezitet u Banjoj Luci
Kontakt: aleksandar.kelec@etfbl.net
Oblasti interesovanja: objektno-orijentisano projektovanje i programiranje, razvoj mobilnih aplikacija, sigurnost, kriptografija i kompjuterska zaštita

infoM

UPUTSTVO ZA PRIPREMU RADA

1. Tekst pripremiti kao Word dokument, A4, u kodnom rasporedu 1250 latinica ili 1251 ćirilica, na srpskom jeziku, bez slika. Preporučen obim – oko 10 strana, single prored, font 11.

2. Naslov, abstakt (100-250 reči) i ključne reči (3-10) dati na srpskom i engleskom jeziku.

3. Jedino formatiranje teksta je normal, bold, italic i bolditalic, VELIKA i mala slova (tekst se naknadno prelama).

4. Mesta gde treba ubaciti slike, naglasiti u tekstu (Slika1...)

5. Slike pripremiti odvojeno, VAN teksta, imenovati ih kao u tekstu, radi identifikacije, u sledećim formatima: rasterske slike: jpg, tif, psd, u rezoluciji 300 dpi 1:1 (fotografije, ekranski prikazi i sl.), vektorske slike – cdr, ai, fh,eps (šeme i grafikoni).

6. Autor(i) treba da obavezno priloži svoju fotografiju (jpg oko 50 Kb), navede instituciju u kojoj radi, kontakt i 2-4 oblasti kojima se bavi.

7. Maksimalni broj autora po jednom radu je 5.

Redakcija časopisa Info M