Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/08BE14Dimitrijevic

ANDROID APLIKACIJA ZA NADGLEDANJE SISTEMA BAZIRANIH NA AMAZON AWS PLATFORMI

ANDROID APPLICATION FOR MONITORING AMAZON AWS BASED SYSTEMS

Danilo Dimitrijević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – RAČUNARSTVO I AUTOMATIKA

Kratak sadržaj – *U radu je opisan sistem za nadgledanje servera pomoću AWS servisa Amazon Inspector. Specificiran je model sistema kroz UML dijagrame. Opisana je implementacija sistema na Android platofrmi.*

Ključne reči: *Amazon Inspector, AWS servisi, Android*

Abstract – *This paper presents the server monitoring by the AWS service called Amazon Inspector. Detailed description of a model through UML diagrams is given as well as implementation of the system in Android platform.*

Keywords: *Amazon Inspector, AWS services, Android*

1. UVOD

Servisi za nadgledanje servera predstavljaju svakodnevnicu. Najčešći zahtev prilikom nadgledanja servera je da se primi upozorenje kada server nije u funkciji sa razlogom zbog kojeg je server prestao sa radom. Međutim, to nije dovoljno da se server održi bezbednim s obzirom da postoji mogućnost je da je server ranjiv na novu softversku grešku, da se nalazi pod nekom vrstom napada ili da je već kompromitovan. To su neke od stvari koje bi trebalo znati kako bi se preduzele akcije da se takve aktivnosti saniraju u najkraćem mogućem roku. Nadgledanje servera za bezbednosne događaje pruža mogućnost sprečavanja bezbednosnih problema, čak i ako se nešto loše desi može se kontrolisati kako se to odvija, a ne samo da se posmatra dok se sve „ruši" [1].

U okviru ovog rada prikazana je *Android* aplikacija za nadgledanje sistema baziranih na *Amazon AWS cloud* platformi, sa posebnim osvrtom na servis *Amazon Inspector*.

2. AMAZON INSPECTOR

Amazon Inspector predstavlja automatizovani servis procene bezbednosti koja pomaže u poboljšanju bezbednosti i usklađenosti aplikacija postavljenih na *AWS* servisu. *Amazon Inspector* automatski procenjuje: izloženost na napade, odstupanje od preporučenih praksi i ranjivosti aplikacija. Nakon obavljanja procene *Amazon Inspector* daje detaljnu listu bezbednosnih nalaza sa prioritetom prema stepenu ozbiljnosti [2].

Procena bezbednosti *Amazon Inspector*-a pomaže u proveri ranjivosti *Amazon Elastic Compute Cloud* (*Amazon EC2*) instanci. Procena *Amazon Inspector*-a nudi predefinisane pakete pravila koji se sastoje od bezbednosnih preporučenih praksi i definicija ranjivosti. Primeri ugrađenih pravila uključuju: proveru pristupa *EC2* instancama sa interneta, mogućnost daljinskog korišćenja servera ili proveru ranjivosti instalirane verzije softvera [2].

Amazon Elastic Compute Cloud (*Amazon EC2*) je servis koji pruža siguran i promenljiv kapacitet računara na *cloud*-u. *Amazon EC2* smanjuje vreme potrebno za pokretanje novog servera i omogućava brzu promenu kapaciteta servera u zavisnosti od potreba servera [3].

Pomoću *Amazon Inspector*-a može se automatizovati bezbednosna procena ranjivosti kroz razvoj *pipeline*-a. Ovo pruža mogućnost da bezbednosno testiranje postane redovan deo razvoja i IT operacija. *Amazon Inspector* nudi predefinisani softver koji se naziva agent i koji se može instalirati u operativni sistem podignut na *EC2* instanci koju je neophodno nadgledati. Agent nagleda ponašanje *EC2* instance uključujući mrežu, sistem datoteka i aktivnosti procesa, prikuplja širok skup o ponašanju i konfiguraciji [4].

Najvažniji elementi servisa *Amazon Inspector* su:

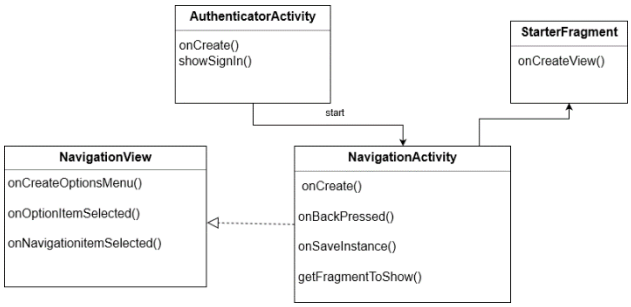
1. ***Assessment Target*** - predstavlja kolekciju *AWS* resursa koji deluju zajedno kao jedinica kako bi pomogli u ostvarivanju poslovnih ciljeva. *Amazon Inspector* procenjuje sigurnosti resursa koji predstavljaju *assessment target*. Trenutno, *Amazon Inspector assessment target*-i se mogu sastojati od *EC2* instanci [5].
2. ***Rules Packages and Rule*** - upoređuje ponašanje i bezbednosnu konfiguraciju *assessment target*-a u odnosu na izabrane pakete pravila (*rules packages*) za proveru. U kontekstu *Amazon Inspector*-a, pravilo (*rule*) je bezbednosna provera koju *Amazon Inspector* vrši prilikom izvršavanja *assessment run*-a [5].
3. ***Assessment Template*** - predstavlja konfiguraciju koja se koristi prilikom pokretanja *assessment run*-a [5].
4. ***Assessment Run*** – predstavlja proces otkrivanja potencijalnih bezbednosnih problema kroz analizu konfiguracije i ponašanja *assessment target*-a na osnovu specificiranih bezbednosnih paketa pravila. Tokom rada *assessment run*-a *Amazon Inspector* nadgleda, prikuplja i analizira konfiguraciju i ponašanje podataka iz resursa specificiranog *assessment target*-a. Nakon toga, *Amazon Inspector* analizira podatke i upoređuje ih sa skupom bezbednosnih paketa pravila koja su specificirana u *assessment template*-u tokom pokretanja *assessment run*-a. Kompletirani *assessment run* proizvodi listu *finding*-a koji predstavljaju potencijalne bezbednosne probleme različitih nivoa ozbiljnosti [5].

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio prof. dr Goran Sladić.

5. ***Finding*** - predstavlja potencijalne bezbednosne probleme koje je *Amazon Inspector* otkrio prilikom procene nadgledanja *assessment target*-a. *Finding* se može prikazati pomoću *Amazon Inspector* konzole ili pomoću API-ja [5].
6. ***Report*** - predstavlja dokument koji opisuje šta se testira u *assessment run*-u i daje rezultate procene. Dobijeni *report*-i se mogu skladištiti, deliti sa timom kako bi se sanirali problemi ili koristiti za povećanje podataka o reviziji usklađenosti. Nakon završetka *assessment run*-a može se generisati izveštaj [5].

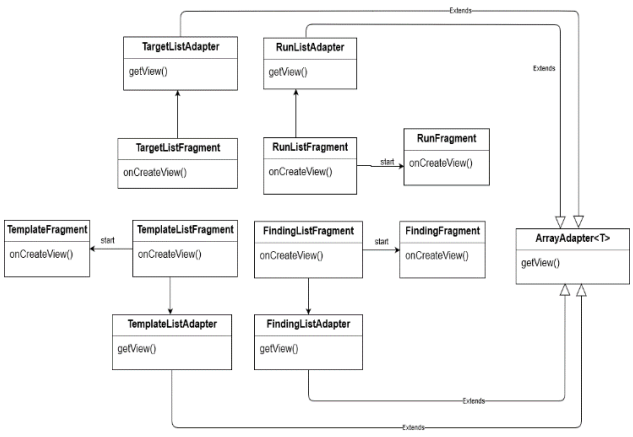
3. MODEL SISTEMA

Slika 1. prikazuje dijagram klasa glavnih komponenti *android* aplikacije. *AuthenticatorActiviy* je klasa koja se pokreće prilikom inicijalizacije. Zadatak te klase je da proveri da li je korisnik već prijavljen na sistem, ukoliko jeste automatski se prelazi na aktivnost *NavigationActivity*, ukoliko nije otvara se stranica za prijavljivanje na aplikaciju. *NavigationActivity* predstavlja glavnu aktivnost aplikacije koja se sastoji iz dva dela, prvi deo je *StarterFragment* klasa koji služi kao prikaz u glavnoj aktivnosti, dok je drugi deo *NavigationView* koji omogućava navigacioni prikaz i pokretanje ostalih fragmenata.



Slika 1. – *Dijagram klasa glavnih aktivnosti*

Na slici 2 prikazan je dijagram klasa fragmenata koji služe za listanje elemenata *Amazon Inspector*-a. Fragmenti u sebi sadrže adaptere koji nasleđuju ugrađenu *android* klasu *ArrayAdapter<T>*. Namena adaptera je da svaki element iz liste prikažu na identičan način. Fragmenti za listanje elemenata mogu pokrenuti fragment za detaljan prikaz elemenata (npr. fragment *RunListFragment* može pokrenuti fragment *RunFragment*).



Slika 2. - *Dijagram klasa elemenata*

4. IMPLEMENTACIJA SISTEMA

U ovom odeljku će biti opisana implementacija *android* aplikacije za nadgledanje koja predstavlja sistem za nadgledanje server pomoću *AWS* servisa *Amazon Inspector*.

Amazon Web Services (*AWS*) je jedna od najrasprostranjenijih *cloud* platformi koja nudi preko 165 servisa iz *data* centara širom sveta [6].

Implementacija sistema se sastoji od dve aplikacije:

- Jedna aplikacija je urađena u programskom jeziku *Java* i predstavlja *AWS Lambda* izraze, od kojih jedan *Lambda* izraz služi za dodavanje *Amazon Inspector* elemenata u bazu, dok drugi služi za obradu *Amazon SNS* događaja i slanje notifikacija.
- Druga aplikacija je mobilna aplikacija urađena u okruženju *Android Studio* koji predstavlja zvanično okruženje za razvoj *android* aplikacija. Mobilna aplikacija služi za prikaz *Amazon Inspector* elemenata i primanje notifikacija.

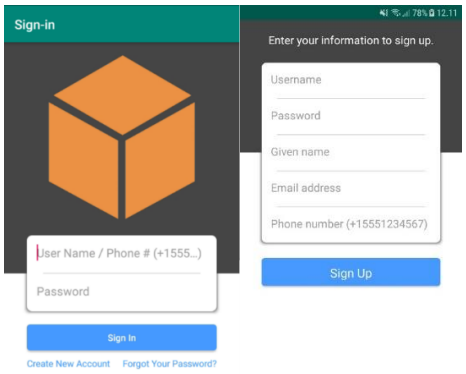
Amazon Lambda je servis koji omogućava pokretanje koda bez potrebe za obezbeđivanjem ili upravljanjem serverima [7]. *Amazon Simple Notification System* (*Amazon SNS*) je servis koji upravlja isporukom ili slanjem poruka pretplatničkim krajnjim tačkama ili klijentima. Ovaj servis se može koristiti za slanje mobilnih notifikacija, SMS poruka, mail-ova kao i za slanje događaja na druge *AWS* servise [8].

Kako bi korisnik mogao koristiti aplikaciju neophodno je da poseduje *AWS* nalog, kao i da ima kreiran bar jedan *assessment target* na *Amazon Inspector* servisu kako bi mogao da učita neophodne podatke u aplikaciju.

4.1. Prijava i registracija na sistem

Aplikacija zahteva od korisnika prijavu na sistem kako bi mogao da koristi aplikaciju. Za implementaciju stranice za prijavljivanje i registraciju je korišćena predefinisana stranica koju pruža *AWS Mobile Hub* (servis za mobile aplikacije) servis u kombinaciji sa servisom *Amazon Cognito* (servis koji predstavlja korisničku bazu) (slika 3.).

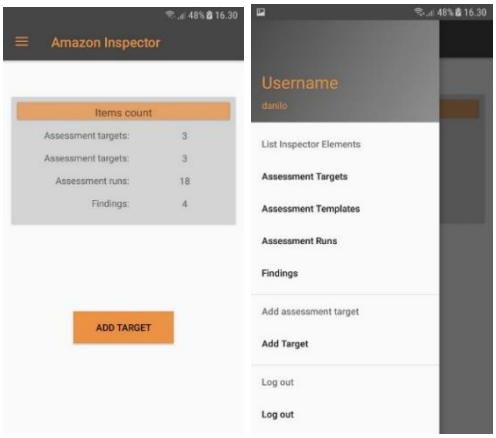
Ukoliko korisnik nema nalog, može ga napraviti odabirom opcije za pravljenje novog naloga (slika 3. opcija *Create New Account*) i otvoriće mu se prozor za registraciju. Kada se registruje njegovi kredencijali se automatski čuvaju u korisničkoj bazi.



Slika 3. - *Predefinisane stranice za prijavljivanje i registrovanje na aplikaciju*

4.2. Početna stranica

Početna stranica prikazuje listing svih elemenata sa ukupnim brojem tih elemenata u prvoj kartici i omogućava korisniku da napravi novi target (slika 4.). Početna stranica nudi i opciju za navigaciju kroz projekat (slika 4.).

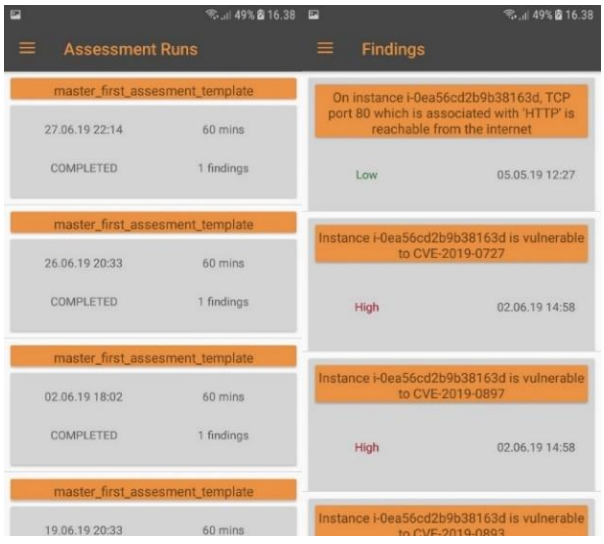


Slika 4. – *Početna stranica i navigacioni prikaz*

4.3. – Prikaz liste elemenata *Amazon Inspector*-a

Slike 5. prikazuje listing *Amazon Inspector* elemenata pod nazivom *assessment run* i *finding*. Prikazane su samo najbitnije informacije kako elementi liste ne bi zauzimali previše mesta na ekranu, za detaljnije informacije korisnik treba da klikne na element.

Ostali elementi *Amazon Inspector*-a su prikazani na sličan način kao i elementi prikazani na slici 5., jedina razlika je kod prikaza *assessment target*-a jer taj element nema mogućnost detaljnog prikaza.

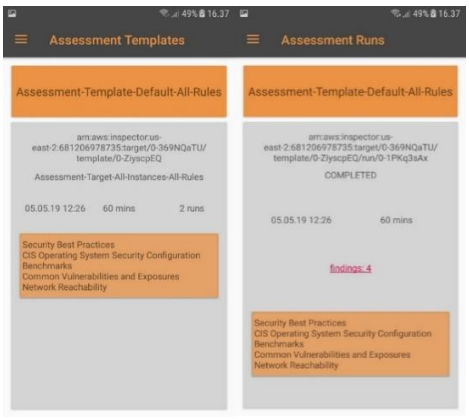


Slika 5. – *Prikaz listanja assessment run-ova i finding-a*

4.4. Detaljan prikaz elemenata *Amazon Inspector*-a

Slika 6. predstavlja prozor za detaljan prikaz *assessment template*-a i *assessment run*-a sa naslovima *template*-a, vremena pravljenja, paketima pravila koji su korišćeni prilikom pokretanja *run*-a.

Ostali elementi *Amazon Inspector*-a prikazani su na sličan način kao i dva prikazana na slici 6.



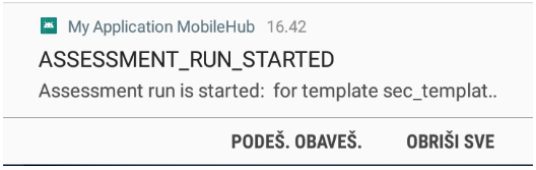
Slika 6. - *Detaljan prikaz assessment template-a i assessment run-a*

4.5. - Notifikacije

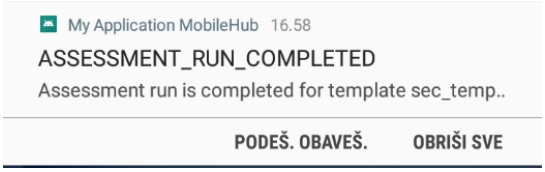
Za slanje notifikacija mobilnoj aplikaciji, korišćen je *Firebase Cloud Messaging* (*FCM*) sistem. *FCM* predstavlja *Firebase* platformu za slanje i primanje poruka i notifikacija [9].

Aplikacija prima notifikacije na dva događaja: početak i kraj *assessment run*-a.

Slike 6. i 7. predstavljaju prikaz notifikacija u mobilnoj aplikaciji. Korisnik ima mogućnost da klikne na notifikaciju i ode na aktivnost koja predstavlja detaljan prikaz *assessment run*-a.



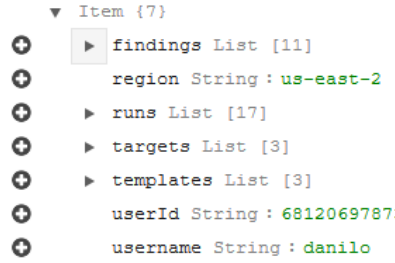
Slika 6. – *Notifikacija za pokretanje assessment run-a*



Slika 7. – *Notifikacija za završetak assessment run-a*

4.6 - Baza podataka

Za ovaj projekat kao baza podataka je korišćen *AWS* servis *Amazon DynamoDB* (*NoSQL* baza podataka). Na slici 8. predstavljen je jedan red u bazi podataka na kojem se vide elementi vezani za korisnika kao što su korisničko ime, region, identifikacioni broj korisnika na *AWS*-u i *Amazon Inspector* elementi vezani za tog korisnika.



Slika 8. – *Primer reda u bazi podataka*

Listing 1. predstavlja inicijalizaciju servisa *Amazon DynamoDB* u *android* aplikaciji i čitanje podataka pomoću metode *load*.

```
DynamoDBMapper dynamoDBMapper;
AmazonDynamoDBClient dynamoDBClient = new
AmazonDynamoDBClient(AWSMobileClient.getInstanc
e().getCredentials());
dynamoDBMapper = DynamoDBMapper.builder()
        .dynamoDBClient(dynamoDBClient)

.awsConfiguration(AWSMobileClient.getInstance()
.getConfiguration())
        .build();
InspectorModel inspectorModel =
dynamoDBMapper.load(InspectorModel.class,
targetDTOS[0].getUserId());
```

Listing 1. *Inicijalizacija servisa Amazon DynamoDB*

5. ZAKLJUČAK

Nadgledanje sistema je važno pogotovu u poslednjem periodu kada se pojavljuje sve više i više sofisticiranih napada na različite aplikacije i servise. Gotovo sve veće *cloud* platforme poseduju proizvod koji nadgleda te platforme kako bi sprečile ranjivosti i ažurirali sistem sa najnovijim izmenama. *Amazon* predstavlja jednu od najsigurnijih i najčešće korišćenih *cloud* platformi sa konstantnim razvijanjem novih i ažuriranjem starih servisa.

U okviru ovog rada opisana je *AWS Amazon Inspector* platforma za nadgledanje servera sa svojim karakteristikama i elementima. Dat je prikaz specifikacije i implementacije *android* aplikacije za nadgledanje AWS sistema. Takođe, prikazani su osnovni elementi *Amazon Inspector*-a sa svojim osnovnim informacijama u vidu liste, kao i detaljni prikazi elemenata u posebnom prozoru. Opisano je na koje događaje aplikacija prima notifikacije.

Neki od daljih smerova nadogradnje i razvoja sistema su:

- prijavljivanje na aplikaciju pomoću provajdera *Google* i *Facebook*,
- pravljenje novih *Amazon Inspector* elemenata pomoću android aplikacije i
- nadograditi *android* aplikaciju da radi sa najnovijim *AWS* servisom koji je predviđen za rad sa mobilnim aplikacijama pod nazivom *AWS Amplify*.

6. LITERATURA

[1] Server Monitoring Overview https://bobcares.com/-blog/server-security-monitoring/ (pristupljeno u oktobru 2019.)

[2] Amazon Inspector https://aws.amazon.com/inspector/ (pristupljeno u oktobru 2019.)

[3] Amazon EC2 https://aws.amazon.com/ec2/

[4] Amazon Inspector Introduction https://docs.aws.amazon.com/inspector/latest/userguide/inspector_introduction.html (pristupljeno u oktobru 2019.)

[5] Amazon Inspector Concepts https://docs.aws.amazon.com/inspector/latest/userguide/inspector_concepts.html (pristupljeno u oktobru 2019.)

[6] AWS https://aws.amazon.com/what-is-aws/ (pristupljeno u oktobru 2019.)

[7] AWS Lambda https://aws.amazon.com/lambda/ (pristupljeno u oktobru 2019.)

[8] Amazon SNS https://docs.aws.amazon.com/sns/latest/dg/welcome.html (pristupljeno u oktobru 2019.)

[9] Firebase Cloud Messaging https://firebase.google.com/docs/cloud-messaging (pristupljeno u oktobru 2019.)

Kratka biografija:

Danilo Dimitrijević rođen je 28.04.1994. godine u Kragujevcu. Osnovnu školu „Stanislav Sremčević“ je završio 2009. godine, srednju školu „Prva tehnička škola“ u Kragujevcu, završio je 2013. godine. Iste te godine upisao je „Fakultet tehničkih nauka“ u Novom Sadu, odsek računarstvo i automatika. Diplomirao je u roku 2017. godine i upisao master studije iste godine. Od 2018. godine radi u kompaniji „Synechron“.

kontakt: choda.94@gmail.com