Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 007:004.056
DOI: https://doi.org/10.24867/16BE08Zecevic

PRIMENA BEZBEDNOSNIH ELEMENATA AZURE PLATFORME U SISTEMIMA ZA ELEKTRONSKA PLAĆANJA

APPLICATION OF SECURITY CONTROLS OF AZURE PLATFORM IN ELECTRONIC PAYMENT SYSTEMS

Helena Zečević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U ovom radu predstavljeni su problemi i izazovi sistema koji se bave eletktronskim plaćanjima. Predstavljen je jedan koncept sistema kroz koji su istražene bezbednosne kontrole i zahtevi nekih od regulativa čije implementacije se smatraju obaveznim. Poseban akcenat stavljen je na mehanizme koje pruža i olakšava Microsoft Azure platforma. Istraženo je više servisa koje ova platforma nudi i opisani su prednosti ali i nedostaci istih.*

Ključne reči: *elektronsko plaćanje, informaciona bezbednost, Azure*

Abstract – *In this paper presented are the problems and challenges faced by the electronic payment systems. One concept of such system is presented here, through which security controls have been investigated along with requirements of several regulations which implementations are considered a must in these systems. In this paper highlighted are the mechanisms which are offered and made easier for implementation by Microsoft Azure platform. Multiple services offered by this platform have been explored and both advantages and downsides have been presented in this paper.*

Keywords: *electronic payment, information security, Azure*

1. UVOD

U današnje vreme sistemi bazirani na elektronskom poslovanju su postali značajan udeo tržišta i uveliko zamenjuju tradicionalne načine poslovanja. Sa razvojem internet bankarstva, odlazak u banku postao je izuzetak, a ne pravilo, a sa razvojem *online* trgovine postalo je moguće dobaviti i najosnovnije potrepštine za život iz udobnosti svog doma. Razvoj ovakvih elektronskih sistema, a naročito *online* trgovine, doveo je do toga da se značajan udeo novca razmenjuje na internetu.

Razvoj sistema za elektronsko poslovanje je povukao za sobom i razvoj sistema za elektronsko plaćanje. Elektronsko plaćanje danas je moguće obaviti na više načina: posredstvom sistema razvijenih od strane banaka putem sistema kao što je *PayPal* [1], ali i razmenom kriptovaluta poput *bitcoin*-a [1].

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

[1] https://www.paypal.com/

Ono što je zajedničko za sve *online* sisteme jeste pažnja koja se pridaje bezbednosti. Sistemi elektronskog plaćanja upravljaju osetljivim korisničkim podacima poput brojeva računa, brojeva kartica, bezbednosnih kodova na karticama, tajnih ključeva za pristup sistemima poput *PayPal*-a, itd.

S obzirom na prirodu ovih osetljivih podataka, rizik ovakvih sistema je visok, stoga je važno posvetiti posebnu pažnju informacionoj bezbednosti. Informaciona bezbednost se definiše kao zaštita implementirana u informacionom sistemu kako bi se očuvali poverljivost (eng. *Confidentiality*), dostupnost (eng. *Availability*) i integritet (eng. *Integrity*) resursa informacionog sistema [2]. Ova tri pojma poznatija su u informacionoj bezbednosti kao CIA trijada i predstavljaju srž računarske bezbednosti [2].

Zbog ekonomske isplativosti, skalabilnosti i strožijih potreba za dostupnosti servisa, veliki broj aplikacija danas se nalazi na *cloud*-u ili je u procesu integracije. Veliki *cloud* provideri, poput *Microsoft Azure*, takođe nude širok opseg već implementiranih bezbednosnih aspekata i jednostavnu integraciju sa njima, ili olakšavaju implementaciju istih.

Postoje brojni standardi koji se tiču bezbednosti u informacionim i *cloud* sistemima, među kojima su neki od poznatijih ISO-27001, GDPR, HIPAA, PCI DSS, itd. [3].

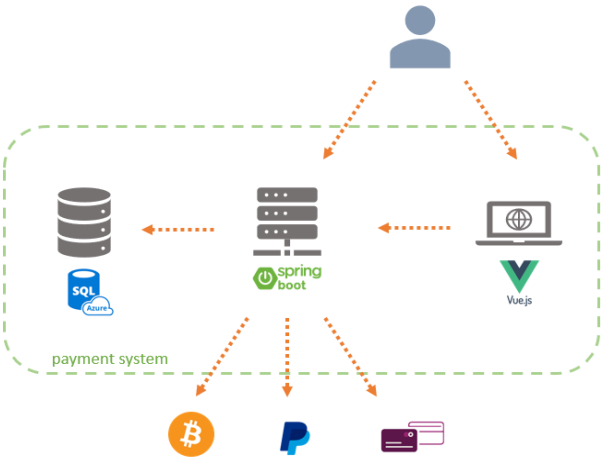
U ovom radu biće izloženi bezbednosni elementi nekih od servisa koje nudi *Azure* platforma, kao i detalji o njihovim prednostima i manama.

2. SPECIFIKACIJA SISTEMA ZA ONLINE PLAĆANJE

Koncept sistema za *online* plaćanje podrazumeva sistem koji korisnicima omogućuje plaćanje posredstvom nekoliko različitih načina plaćanja. Sistem je zamišljen tako da se može proširivati lako novim načinima plaćanja, a u inicijalnoj specifikaciji podrazumeva podršku za plaćanje platnim karticama, putem *PayPal* platforme i *bitcoin* valutom. Sam sistem sastoji se iz više komponenti.

Slika 1 prikazuje arhitekturu i osnovne komponente koncepta sistema za plaćanje koji se sastoji iz: relacione baze podataka, centra sistema za plaćanje, klijentske aplikacije i eksternih servisa za plaćanje.

Korisnici mogu sa sistemom da interaguju direktno koristeći centralnu aplikaciju putem *web* API-a ili posredstvom korisničke *web* aplikacije. Centralna aplikacija izvršenje plaćanja omogućuje putem interakcije sa eksternim servisima za plaćanje, poput *Bitcoin*-a i *PayPal*-a.



Slika 1. *Sistem za plaćanje (uokviren zelenom bojom): relaciona baza podataka (levo), poslovna logika (sredina) i klijentska web aplikacija (desno).*

2.1. Relaciona baza podataka

Podaci koji se koriste u sistemu za plaćanje su po svojoj prirodi strukturirani, stoga se relaciona baza činila kao logičan izbor za skladištenje podataka. Neki od podataka koji se skladište u sistemu su: podaci o prodavcima (eng. *merchants*), podaci o uplatama, detalji o podržanim metodama plaćanja za svakog prodavca, itd.

U finalnoj implementaciji sistema korišćena je *Azure SQL Database* [4] baza podataka, o kojoj će biti više reči kasnije.

2.2. Centar za plaćanje

Centralna aplikacija predstavlja srž sistema za plaćanje. Ona je zadužena za rukovanje svim zahtevima koji pristižu od klijenata koji koriste ovaj sistem za plaćanje.

Korisnici prvog reda ovog sistema za plaćanje zapravo predstavljaju druge sisteme, u najčešćem slučaju *online* trgovine.

Krajnji korisnici ovih trgovina koriste sistem za plaćanje prilikom kupovine proizvoda, međutim njihovi podaci se u ovom sistemu ne skladište i ne postoji potreba za bilo kakvom vrstom registracije ovakvih korisnika.

S druge strane trgovine koje žele da koriste ovaj sistem moraju na neki način da se prijave kako bi bile poznate sistemu, ali i da prilože podatke koji su potrebni za obavljanje transakcija, poput npr. broja bankovnog računa.

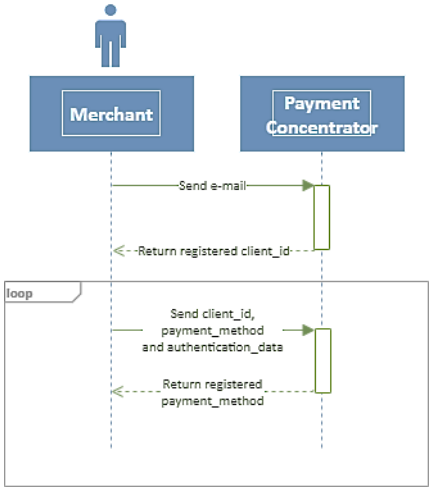
2.2.1 Tok registracije na različite načine plaćanja

Slika 2 prikazuje tok registracije jednog prodavca (eng. *merchant*) na sistem za plaćanje (eng. *payment concentrator*).

Prodavci se najpre registruju na sistem sa *e-mail* adresom, nakon čega dobijaju potvrdu o registraciji u vidu *client_id* oznake.

Nakon toga prodavac je u mogućnosti da se registruje na načine plaćanja koje želi da omogući svojim kupcima, a koji su podržani od strane sistema. Prilikom registracije na željeni način plaćanja potrebno je proslediti svoj *client_id*, odabrani metod, kao i podatke za autentikaciju na željeni metod (npr. API ključ u slučaju *PayPal*-a).

Nakon registracije prodavca, kupci su u mogućnosti da koriste isključivo načine plaćanja koji su podržali prodavci.



Slika 2. *Prikaz toka registracije prodavca na sistem za plaćanje.*

2.2.2 Tok plaćanja

Online plaćanje nije atomična operacija, već je predviđen određen sled koraka, gde u svakom može doći do greške. Sledi opisan primer toka plaćanja putem platnih kartica.

Kupac nakon odabira proizvoda bira način plaćanja od ponuđenih. Podaci se sa sistema prodavca prosleđuju na sistem za plaćanje, među kojima su i URL adrese na koje treba preusmeriti korisnika nakon uspešnog i neuspešnog plaćanja. Podaci se dalje prosleđuju sistemu banke prodavca, koja nakon provere podataka inicijalizuje plaćanje i sistemu vraća URL na koji treba preusmeriti korisnika kako bi se plaćanje realizovalo. Korisnik na novoj stranici unosi podatke iz kartice, koji se prosleđuju banci. Banka proverava podatke i u slučaju da je i kupac u istoj banci ona vrši transfer sredstava, a ukoliko je u drugoj banci, kontaktira asocijaciju koja prosleđuje podatke banci kupca. Nakon izvršenog transfera novca, banka javlja sistemu o rezultatu transakcije, a naš sistem dalje korisnika preusmerava na unapred zadatu stranicu.

2.3. Klijentska aplikacija

Klijentska aplikacija zamišljena je kao *web* aplikacija jednostavnog izgleda. Implementirana je u *Vue.js*[2] okviru za rad. Ovaj okvir za rad je lak za korišćenje i nema velike količine suvišnog koda (eng. *overhead*) kao kod nekih drugih okvira. Aplikacija je vrlo jednostavna i ima mali broj stranica.

3. BEZBEDNOSNI MEHANIZMI AZURE PLATFORME

Jedan od aspekata bezbednosti o kojima posebno treba voditi računa prilikom izrade svakog softverskog sistema jesu osetljivi podaci i njihovo skladištenje. Sistem za plaćanje po svojoj prirodi rukuje osetljivim podacima i gubitak podataka može da dovede do značajne materijalne štete. Stoga je važno posvetiti pažnju očuvanju poverljivosti i integriteta osetljivih podataka i mehanizama koji mogu dovesti do njihovog kompromitovanja.

3.1 GDPR i Azure

GDPR regulativa doneta je kako bi se regulisalo prikupljanje, skladištenje i upotreba ličnih korisničkih podataka,

[2] https://vuejs.org/

sa ciljem njihove zaštite, kao i povećanjem kontrole korisnika nad svojim ličnim podacima. GDPR obuhvata pravila koja predstavljaju dobre prakse u informacionoj bezbednosti kojih bi se trebalo držati bez obzira na primenljivost same regulative.

Microsoft Azure cloud computing platforma posvećuje posebnu pažnju bezbednosti svojih servisa, kao i jednostavnoj integraciji bezbednosnih mehanizama u servise korisnika. Objavljen je dokument [5] koji prikazuje sažetak GDPR regulative i delova koje *Microsoft* izdvaja kao posebno značajne, kao i istaknute principe privatnosti i bezbednosti. U dokumentu se takođe mogu videti primeri *Azure* servisa i funkcionalnosti koje korisnici mogu da upotrebe kako bi na jednostavan način ispoštovali zahteve regulative. Neki od bezbednosnih principa istaknutih u dokumentu su: kontrola pristupa, logovanje i nadgledanje servisa, kriptografija, maskiranje podataka, itd.

3.2 Drugi osetljivi podaci

Pored osetljivih podataka koji spadaju u identifikujuće podatke (eng. PII – *personally identifiable information*) postoje i drugi osetljivi podaci koji nisu identifikujući za korisnike ali su po svojoj prirodi tajni, poput lozinki, API ključeva, brojeva platnih kartica, itd.

3.2.1 Azure SQL Database

Azure SQL Database, drugačije poznata kao SQL DB, omogućuje korisnicima da na jednostavan način, a često i jeftiniji, ispoštuju GDPR propise ali i druge bezbednosne principe. Zabrana suvišnih konekcija ka bazi predstavlja jedan od nivoa zaštite, i implementira se dodeljivanjem prava pristupa eksplicitnim IP adresama.

Autorizacija korisnika je kod SQL DB implementirana u vidu principa minimalnih privilegija. Svaki novi dodati korisnik podrazumevano nema nikakva prava, i ona mu se eksplicitno moraju dodeliti [5]. Preporučuje se kreiranje rola koja obuhvataju određene privilegije, i dodeljivanje rola korisnicima, za razliku od direktnog dodeljivanja privilegija svakom korisniku [5].

Enkripcija podataka je prisutna i u transportu i u skladištu. TLS (eng. *transport layer security*) protokolom zaštićen je sav saobraćaj koji ide ka i od baze podataka i nije potrebno ništa dodatno konfigurisati. Podaci u skladištu su enkriptovani zahvaljujući funkcionalnosti TDE (eng. *transparent data encryption*) [5] i obuhvata podatke iz baze, logove i podatke iz rezerve (eng. *backup*). Podrazumevano je uključena i nema uticaj na aplikacije koje koriste ove podatke, jer se oni enkriptuju i dekriptuju prilikom upisa i čitanja, a o kriptografskim ključevima i njihovoj rotaciji sam servis vodi računa [5].

Kao dodatni način zaštite podataka u upotrebi, SQL DB podržava maskiranje podataka. Maskiranje podrazumeva delimično sakrivanje osetljivih podataka od neprivilegovanih korisnika. Postoje dve vrste maskiranja i obe su podržane: statičko i dinamičko.

3.3 Azure Key Vault

Pored korisničkih podataka u sistemu, skladištenih u relacionoj bazi podataka, postoje i drugi osetljivi (nekorisnički) podaci koje je potrebno zaštititi. To su podaci poput konfiguracionih ključeva, tajni za pristup drugim servisima, konekcionih parametara za bazu, kriptografskih ključeva, itd. Kompromitovanje ovakvih tajni može dovesti do gubitka velike količine ličnih korisničkih podataka i može da ima ozbiljne posledice na sistem.

Azure Key Vault je servis namenjen upravo za ovakve svrhe. Kreiran je za svrhu skladištenja API ključeva, sertifikata, lozinki, itd. [6]. Dodatna prednost korišćenja *Key Vault*-a, u odnosu na druge načine poput skladištenja u sistemske varijable, je što su ključevi skladišteni na centralizovanom mestu. Sistemi koji se sastoje iz više komponenti često imaju potrebu da pristupaju istim ključevima iz različitih servisa. Kompromitovanje ključa u slučaju skladištenja u sistemskim varijablama servisa zahtevalo bi izmenu ključa na svakom mestu, dok je sa *Key Vault*-om dovoljno izmeniti kompromitovanu tajnu na jednom mestu, i promeniti ključeve za pristup samom *Vault*-u na onom mestu gde je došlo do napada. *Key Vault* koristi i softverske i hardverske mehanizme da zaštiti sve skladištene tajne [6].

3.4 Bezbednost servisa

Pored zaštite poverljivosti i integriteta podataka spomenutih u prethodnom poglavlju, drugi aspekt bezbednosti sistema ogleda se u bezbednosti samog servisa. Bezbednost servisa se sastoji iz više segmenata, poput zaštite komunikacije servisa sa ostalim komponentama (bazom podataka, klijentskim aplikacijama, drugim servisima, itd.), sprečavanja pristupa neovlašćenim korisnicima, kao i nadgledanja rada servisa i alarmiranja u slučaju sumnjivih radnji. Pored toga, i dostupnost (eng. *availability*) servisa je jedan od značajnih aspekata bezbednosti.

3.4.1 Deployment servisa

Kako bi aplikacija postala dostupna korisnicima, potrebno je prebaciti aplikaciju na neki od *cloud* servisa. Za one koji ne žele da imaju veliku kontrolu nad samom virtuelnom mašinom na kojoj se izvršava kod aplikacije, preporučuje se *Azure App Service* [7]. Ovaj servis je HTTP bazirani servis za hostovanje *web* aplikacija. Visoka dostupnost servisa je jedna od značajnih karakteristika *App Service*-a. Jedna aplikacija ovog servisa može biti pokrenuta na više instanci (do 30). Povećavanje broja instanci aplikacije povećava nivo dostupnosti, a ove instance opslužuju zahteve upućene servisu u *round-robin* maniru.

Slično kao i kod baze podataka, i *App Service*-u može se ograničiti pristup sa određenog skupa IP adresa, međutim u slučaju platnog sistema ovo nije moguće, imajući u vidu da je u pitanju javni servis.

Saobraćaj svih servisa obezbeđen je TLS protokolom, i za svaku aplikaciju moguće je i preporučljivo podesiti opciju odbijanja svih konekcija putem nebezbednog (HTTP) protokola. Sam *Azure* aplikacijama dodeljuje podrazumevani TLS sertifikat i brine o njegovoj rotaciji, bez potrebe za dodatnim konfigurisanjem ili kupovinom sopstvenog sertifikata.

3.4.2 Čuvanje logova

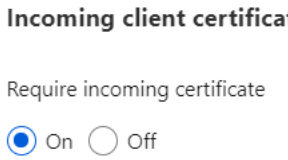
U *cloud* svetu se situacija sa otkrivanjem aplikativnih problema otežava jer nije moguće na isti način pristupati servisu, podacima i greškama kao na ličnom računaru. Iz ovog razloga treba voditi računa o tome kojim se sve informacijama (logovima) može pristupiti, gde se mogu

pronaći greške, itd. Pored toga, logovanje može doprineti boljem razumevanju korisničkih akcija i grešaka, ali i ukazati na sumnjive radnje koje mogu biti posledica potencijalnih napada na servis.

Kako bi se nivo detaljnosti logova *cloud* servisa, pored podrazumevanih logova *App Service*-a, među kojima su npr. logovi o *deployment* procesu i logovi *web* servera, poistovetio sa nivoom detaljnosti logova koje imamo prilikom lokalnog razvoja, *Azure Application Insights* [8] omogućuje integraciju sa *log4j*[3] logovima. *Application Insights* je servis koji omogućuje nadgledanje aplikacija, detekciju anomalija u performansama, agregiranje informacija o zahtevima, brzini odgovora, itd.

3.4.3 Dvosmerna autentifikacija

App Service podržava dvosmernu autentifikaciju i sve što je potrebno da bi se ona uključila jeste odabrati opciju za proveru klijentskog sertifikata (slika 3).



Slika 3. *Opcija za uključenje dvosmerne TLS autentifikacije.*

Dvosmerna autentifikacija kod ovog servisa većinu posla ostavlja samoj aplikaciji, koja sama mora proveriti integritet sertifikata. Jedini zadatak *App Service*-a jeste da proveri da li je klijentski sertifikat zapravo i poslat. Ukoliko nije, *App Service* će vratiti korisniku grešku i zahtev neće ni stići do same aplikacije.

3.4.4 Aplikativni firewall

Azure Web Application Firewall predstavlja centralizovani vid zaštite za *web* aplikacije. Implementira se uz pomoć aplikativnog *gateway*-a (eng. *application gateway*) na regionalnom nivou i jedan isti *firewall* može da opslužuje više aplikacija [9]. Ideja aplikativnog *firewall*-a je da se zaštita prebaci ispred aplikacija, i da se reši problem svih poznatih ranjivosti na jednom mestu, umesto da se bezbednost implementira u kodu svake od aplikacija.

Azure firewall baziran je na OWASP *Core Rule Set* [10] pravilima za detekciju najpoznatijih ranjivost, a između ostalog i ranjivosti sa OWASP *Top Ten* liste. Ovaj projekat kontinuirano radi na otkrivanju novih ranjivosti, i na unapređenju pravila, a *Azure* prati ovaj razvoj i konstantno ažurira svoja pravila kako bi *firewall* bio uvek u toku sa trenutnim ranjivostima.

4. ZAKLJUČAK

U ovom radu predstavljen je koncept sistema za plaćanje koji objedinjuje različite vrste plaćanja. Prikazana je arhitektura sistema koja obuhvata relacionu bazu podataka, klijentsku aplikaciju, i centralni deo sistema koji komunicira sa eksternim servisima za plaćanje poput *PayPal*-a i bankarskih sistema. Glavna motivacija za ovaj rad bila je istraživanje bezbednosnih mehanizama na *Azure* platformi. Stoga je veći deo rada posvećen opisivanju bezbednosnih aspekata ovakvog sistema.

Azure platforma se pokazala kao vrlo pogodna za integrisanje ovakvog platnog sistema, s obzirom da pruža podršku i usaglašenost sa GDPR regulativom, kao i široki spektar servisa i bezbednosnih mehanizama. Takođe, *Azure* nudi odličnu podršku za rad sa *SpringBoot* okvirom za rad u kom je ovaj sistem i implementiran, i naizgled nema velikih razlika i poteškoća u odnosu na podršku za *Microsoft* tehnologije.

Predlog za dalje unapređenje servisa bio bi istraživanje i integracija *Azure DDoS Protection Standard* servisa. Pored toga, dalje istraživanje bi moglo biti posvećeno mogućnostima drugih metoda autentifikacije, poput *Azure Active Directory*.

5. LITERATURA

[1] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Manubot, 2019.

[2] W. Stallings, Cryptography and network security, Pearson Education India, 2006.

[3] https://itoc.com.au/blog/top-10-cloud-security-standards-and-control-frameworks.

[4] https://docs.microsoft.com/en-us/azure/sql-database/. [Poslednji pristup februar 2020].

[5] https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-overview. [Poslednji pristup februar 2020].

[6] https://docs.microsoft.com/en-us/azure/key-vault/key-vault-overview. [Poslednji pristup februar 2020].

[7] https://docs.microsoft.com/en-us/azure/app-service/overview. [Poslednji pristup mart 2020].

[8] https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview.

[9] https://docs.microsoft.com/en-us/azure/web-application-firewall/ag/ag-overview. [Poslednji pristup mart 2020].

[10] https://owasp.org/www-project-modsecurity-core-rule-set/. [Poslednji pristup mart 2020].

Kratka biografija:

Helena Zečević rođena je u Novom Sadu 1995. god. Osnovne akademske studije završila je na Fakultetu tehničkih nauka 2018. godine. Master rad iz oblasti Elektrotehnike i računarstva – Softversko inženjerstvo i informacione tehnologije odbranila je 2021. god.

kontakt: helena.zecevic@gmail.com

[3] https://logging.apache.org/log4j/2.x/