1. Uvod

Svrha dokumenta jeste da opiše i odgovori na iznesena pitanja i probleme ranjivosti softvera prateći savremene prakse pisanja bezbednog koda, kao i bezbednosnih konfiguracija softvera. Dokument se oslanja na *best practice* principe OWASP fondacije, kao *de facto* standard za prve korake ka minimizaciji softverskih, ali i poslovnih rizika, kao i edukaciju softversko-razvojne kulture. [1]

Dokument pokriva ranjivosti navedene u OWASP Top Ten standardu, primenjujući njihova rešenja i neophodne konfiguracije na rent-a-car aplikaciji

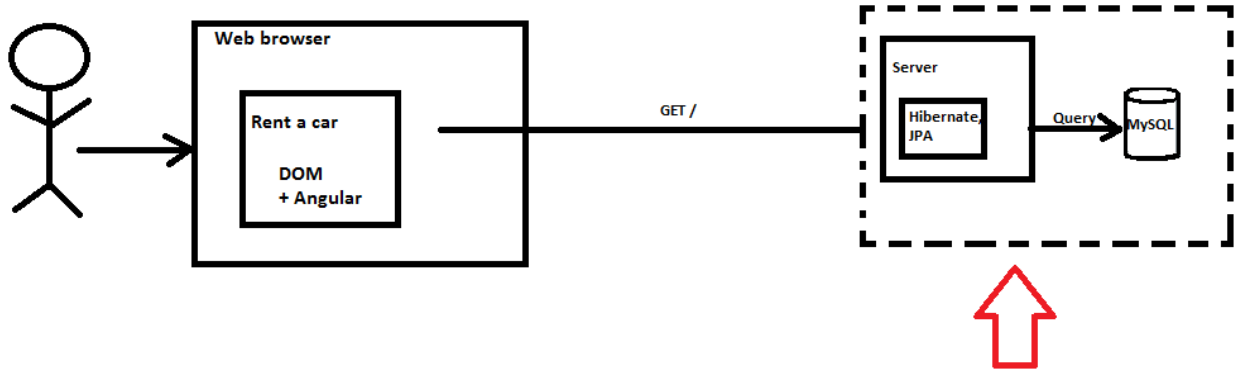
2. OWASP struktura, alati

Do aktuelnog datuma, prvih deset bezbednosnih rizika se ne razlikuju od deset detaljno opisanih rizika navedenih 2017-te [2] godine usled procesa prikupljanja podataka za novu top 10 listu fondacije.

U svrhu otkrivanja ranjivosti upotrebljena su dva alata iz grupe penetracionog testiranja – owasp-zap i dirbuster

2.1 Injection

Slanje malicioznih podataka ka interpreteru.



Razrešenje stoji u separaciji koda i podataka, i to:

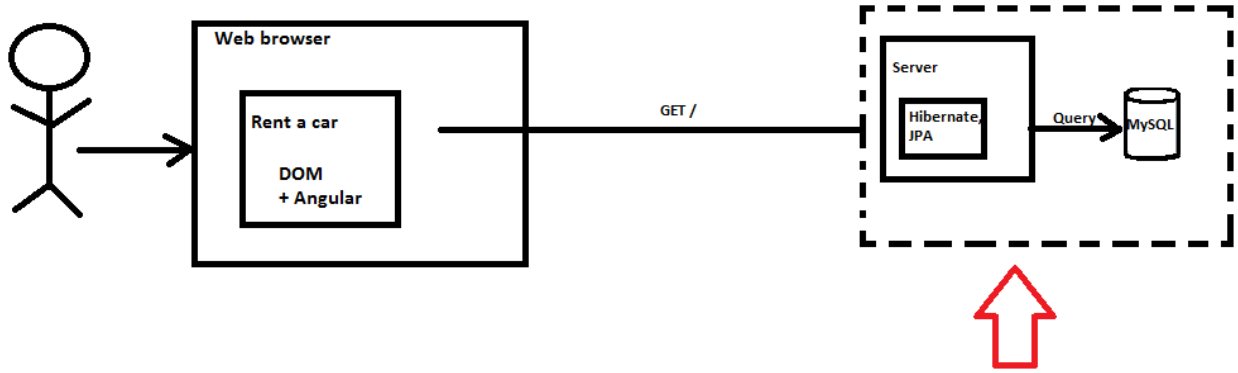
1. Parametrizovanjem upita,
2. Validaciji ručno unetih podataka pre dolaska ka interpreteru,
3. *Escape*-ovanjem posebnih karaktera.

Parametrizovanjem svih upita sprečavaju se SQL naredbe koje kombinuju kod i podatke.
Na nivou hibernate i JPA sloja, rešenje je primenjeno u upotrebi @Query anotacije sa @Param parametrima nad repozitorijum interfejsima tamo gde je potreba za ručnim upitima. U jednostavnijim scenarijima, upotrebljeni su JPA ugrađeni mehanizmi dobavljanja podataka.

Validacija svih podataka je, u okviru rent-a-car projekta, neophodna iz dva aspekta. Očuvanje *suštinske* konzistentnosti baze podataka i sprečavanje raznih familija napada ili pristupa podataka usled kao primer, *stored xss* napada.
Podatke je neophodno validirati na klijentskoj i serverskoj strani.
Klijentskoj strani, gde se korisniku pruža direktna informacija o validnom i nevalidnom unosu po određenom kriterijumu unosa.
Serverskoj strani, gde je upotrebom javax.validation *dependency*-a doveden niz anotacija (@Regex, @Size, @NotBlank, @NotNull, itd.) zaduženih za sprečavanje napada koji zaobilaze klijentsku stranu.

*Escape*ovanje posebnih karaktera je značajn segment sanitizacije podataka koji je postignut @Regex šablonima, ne dozvoljavajući specijalne karaktere tamo gde je slobodan unos od strane korisnika dozvoljen.

2.2 Broken Authentication



Prevencija na nivou rent-a-car sistema:

1. Otpornije korisničke lozinke,
2. Detekcija i prevencija brute-force i *stuffing* napada,
3. Sesija mora isteći.

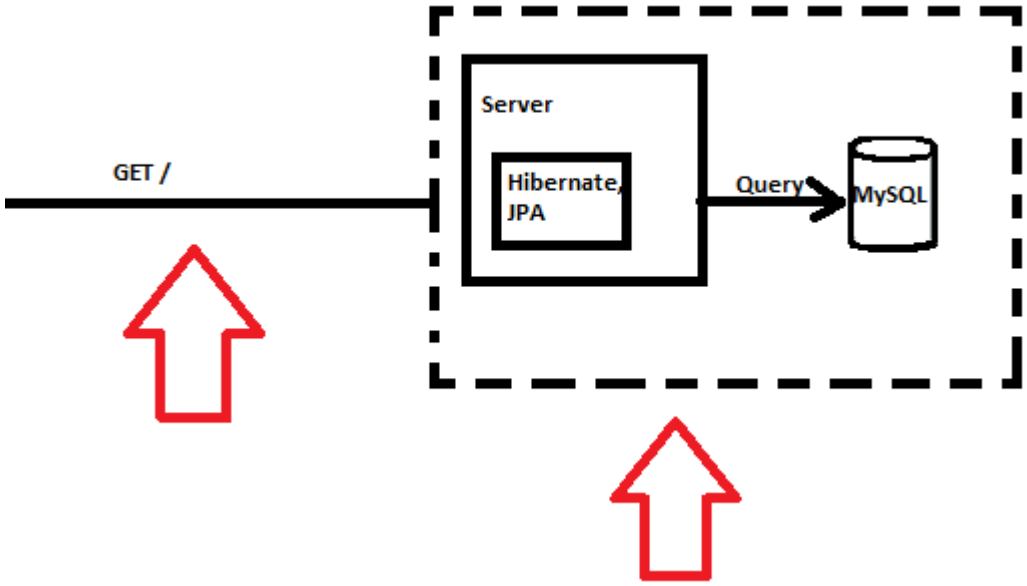
Prilikom kreiranja lozinke korisnik je, prema RFC kao i drugim [3] preporukama, prinuđen da sadrži minimalno 10 karaktera od kojih su neophodne kombinacije malih i velikih slova kao i neki od specijalnih znakova.

Povodom brute-force i dictionary napada, pokušaji neuspele autentifikacije se trajno beleže u odgovarajućoj log datoteci, gde se od informacija čuvaju broj pristupa endpointu kao i IP adresa sa koje se zahtev šalje.

Time se u produkciji otvara mogućnost implementacije IP blacklist funkcionalnosti, gde za zahteve čiji brojač prelazi određen *threshhold* IP adresa biva banovana.
Distribuirani brute force napadi su i dalje mogući, i detaljnijom analizom HTTP saobraćaja se mogu utvrditi šabloni IP adresa sa kojih pristižu zahtevi. U takvim slučajevima, zahtevi stakeholders, kao i poslovni model firme diktiraju najbolji pristup ka prevenciji.
Trajanje JWT-a je podešeno na nisku vrednost, pri čemu se odjavljivanjem kao i vremenskim istekom token invalidira.

2.3 Sensitive Data Exposure

Umesto nasilnog napadanja kriptovanih vrednosti, napadač se okreće ka krađi ključeva, *man-in-the-middle* napadima – *in transit*, ili uzimanju osetljivih podataka sa servera – *at rest*.
Osetljiv podatak predstavlja resurs koji je zakonski, ugovorno, ili na neki drugi način definisan, a istovremeno biva poželjan od strane napadača (primer – lični podaci).



Slaba zaštita osetljivih podataka**,** prema OWASP fondaciji je [4]**:**

1. Neenkriptovana (neheširana) perzistencija,
2. Upotreba slabih algoritama enkripcije i/ili slabih ključeva,
3. Pogrešna ili nepostojeća validacija sertifikata,
4. *Expose-ovanje* ličnih podataka, odnosno kreditnih kartica/šifri.

Prevencija**:**

- Ne čuvati osetljive podatke, ukoliko je to moguće,
- Enkripcija podataka tokom perzistencije kao i saobraćaja,
- Upotreba provereno “jakih” kriptografskih algoritama.

Na nivou **rent-a-car** rešenja, osetljivi podaci su kredencijali, .properties konfiguracija, log datoteke,potencijalno i *source code*. Šifre se heširaju upotrebom BCrypt-a, pokazane kao stabilno kriptografsko rešenje već dve decenije. Međutim, heširanje lozinki nije dovoljno jer bi napadač pažljivom kriptografskom analizom došao do željenih originalnih vrednosti heša. Zato se unete šifre uvek dodatno posole nasumičnim *SecureRandom* brojem. Na nivou BCrypt biblioteke, **salt** mehanizam se vrši automatski.

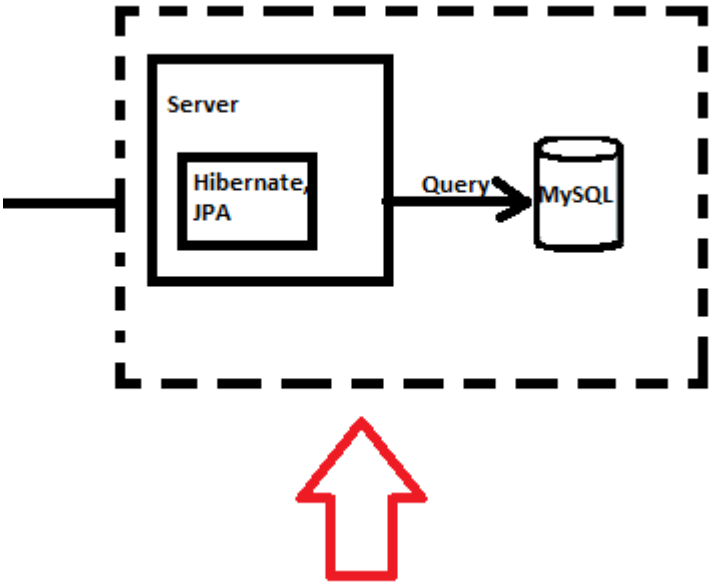
Saobraćaj između komponenti u sistemu bi trebao da se sprovodi kroz HTTPS protokol. Na nivou rent-a-car rešenja, HTTPS saobraćaj se uspostavlja na nivou komunikacije klijent – gateway. U produkciji, bilo bi diskutabilno da li prednosti uspostavljanja TLS-a između svih komponenti mikroservisne arhitekture vagaju u odnosu na HTTP. Konfiguracija stotine servisa da komuniciraju preko TLS sloja, gde svaki sertifikat ima svoje vreme isticanja bi sa aspekta bezbedne konfiguracije predstavljala više glavobolju nego sigurnost. Potencijalno rešenje je uspostavljanje TLS-a na onim komponentama koje imaju pristup zaista osetljivim podacima, kao primer sistemu za prijavljivanje.

Sertifikati upotrebljeni u okviru sistema se generišu na *infrastrukturi javnih ključeva*, u sklopu rent-a-car rešenja pomoću RSA algoritma sa 4096 key-size za *certificate authority*, odnosno 2048 key-size za *end-entity* generisane sertifikate.

Pristup osetljivim konfiguracionim datotekama je zaštićen na nivou ACL-a.
Podatke u sistemu je neophodno replicirati, čime bi se izbegli ransomware napadi.

U produkciji pristup .properties fajlovima je onemogućen jer aplikacija se pokreće iz izolovanog okruženja, gde osetljiva konfiguracija neće biti prisutna.

2.4 XML External Entities (XXE)



Nepodobno iskonfigurisan XML parser može prihvatiti XML dokument, smatrati ga bezbednim i dozvoliti izvršavanje - DoS familije napada, *espionage*, *port scanning of sensitive data*, *sensitive data exposure* i drugih.

Prevencija:
1. Izbegavati XML razmenu podataka,
2. Moderna konfiguracija parsera,
3. Validacija XML dokumenata

Prvi korak u prevenciji, u okviru rent-a-car rešenja je nažalost nemoguć usled razmene podataka preko SOAP protokola sa monolitnom agentskom aplikacijom.

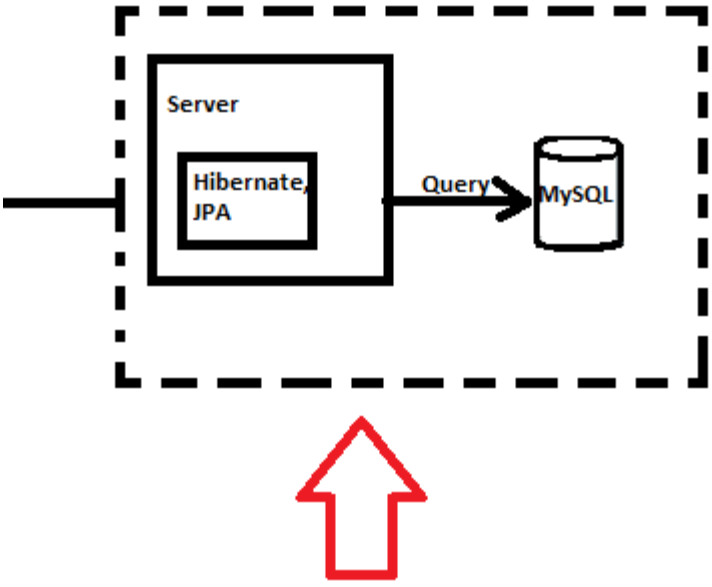
JAXB parser će biti neophodno iskonfigurisati tako da sprečava external entities, što je već unapred podešeno upotrebom savremenijih verzija.

- From 3.2.8 version of spring-web onwards Jaxb2RootElementHttpMessageConverter sets the processExternalEntities to false which in turn sets the XMLInputFactory property IS_SUPPORTING_EXTERNAL_ENTITIES to false. [5]

Prispele XML dokumente je neophodno validirati putem šeme.

2.5 Broken Access Control

Pristup neautorizovanim stranicama, uzdizanje na administratorske privilegije, pristup tuđim resursima ili podacima.



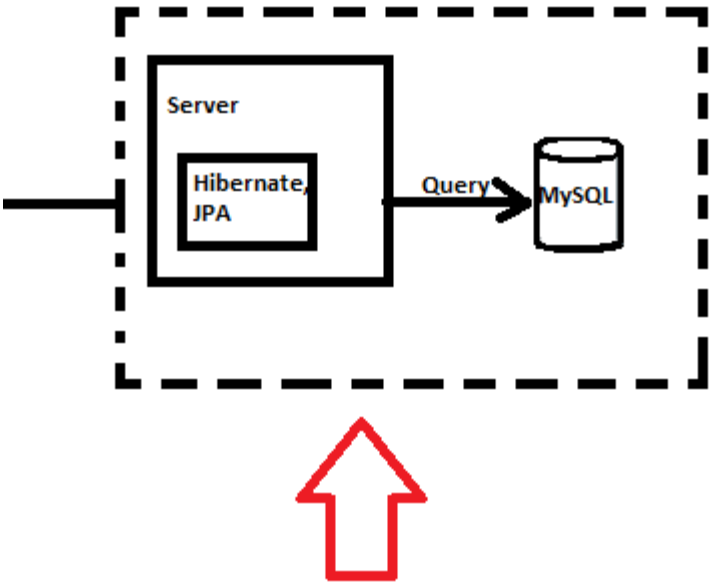
Neophodno je:
1. Ne dozvoliti pristup endpointima od strane bilo koga,
2. Ograničiti do koje mere se pristup može imati,
3. Trajno logovati upozorenja i greške

Na nivou rent-a-cara autorizovan pristup endpointima je definisan preko RBAC modela. Token koji korisnik šalje se validira i iz njega se izvlače korisnikove privilegije.

Trajno logovanje korisničkih akcija nad POST, PUT i DELETE endpointima, čime se neporecivost garantuje.
Neke od definisanih privilegija nisu trajne, i administrator ima prava oduzimanja privilegija korisniku ukoliko se iz logova primeti kreiranje nedozvoljenih zahteva ili pristup resursima koji nisu dozvoljeni.

Na nivou frontenda pristup elevated stranicama nije dozvoljen i definisan je pomoću angular guardova.

2.6 Security Misconfiguration



Pogrešna bezbednosna konfiguracija se može dogoditi na bilo kom nivou aplikacionog steka, mrežnog servisa, web servera, aplikacionog servera, baze podataka, konkretnog frameworka, pa i kontejnera u kojem se aplikacija pokreće.

Mane dobijene usled nepravilne konfiguracije bezbednosnih kontrola, ranjivosti biblioteka u upotrebi, *default*-nih registrovanih naloga sa privilegijama dovode napadače ka neželjenim sistemskim podacima ili funkcionalnostima.

Često, faktori koji dovode do ranjivosti su:

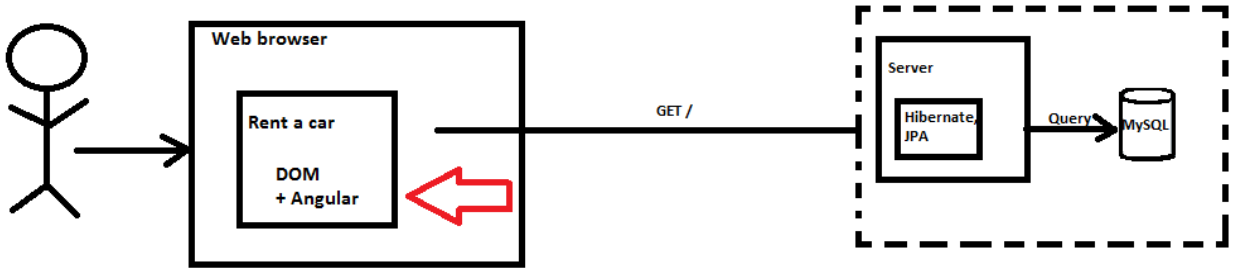
- Nepravilna konfiguracija bezbednosnog koda,
- Nepotrebne funkcionalnosti su još uvek “uključene”,
- *Default* korisnički nalozi su nalaze u produkciji,
- Greške otkrivaju osetljive podatke aplikacije

Neophodno je ugasiti sve servise i endpoint-e koji nisu više u upotrebi, kao i detaljnija analiza pomoću penetracionih alata.
Iz izveštaja penetracionih testova neke od spring security kontrola su se pokazale kao nepravilno implementirane i čitalac će u izveštaju pronaći više informacija, kao i o ostalim mehanizmima zaštite primenjene na rent-a-car-u.

Default korisničke naloge za potrebe razvijanja aplikacije je neophodno izbrisati iz sistema prilikom produkcije.

Greške prolaze kroz @ControllerAdvice *handler* koji poruke o greškama filtrira i prikazuje na način više prilagođen korisnicima.

2.7 Cross-Site Scripting (XSS)



HTML sadržaj upliće sadržaj (html), prezentaciju (css) i kod(js, ts) u jednu celinu. Ukoliko napadač može da izmeni DOM stablo, tehnički može učiniti bilo šta što može i korisnik.

Razrešenje:

1. Enkodovanju korisničkih podataka na bezbedne vrednosti, npr <script> se transformiše u <script>
2. Upotreba *Content Security Policy-a*

Angular cross-site scripting security model tretira sve vrednosti kao *nebezbedne* [6] po default-u. To podrazumeva da bilo koji korisnički unos biva tretiran od strane DOMSanitizer servisa, koji *escapuje* posebne karaktere pre prikazivanja, odnosno slanja podataka. Ovakav pristup u velikoj, ali ne i potpunoj meri uklanja DOM XSS oblike napada, no stored XSS napadi su i dalje mogući.

U tom pogledu, validacija podataka koji stižu na end-point-e servera je izvršena, i navedena je u segmentu 2.1 Injection.

Content Security Policy[7] kao *defense in depth* kontrola, je dodata na serverskoj strani kao neophodan sloj zaštite od mogućih XSS napada.
Polisa sadrži direktive koje omogućavaju definisanje poverljivih izvora izvršavanja bilo kojih skripti, i zauzima white-list pristup. Uz svaki HTTP zahtev se uključuje i bezbednosna polisa sadržaja.
Definisane su direktive **Content-Security-Policy: script-src 'self' https://localhost:4200 , <ostali poverljivi sajtovi> image-src ***

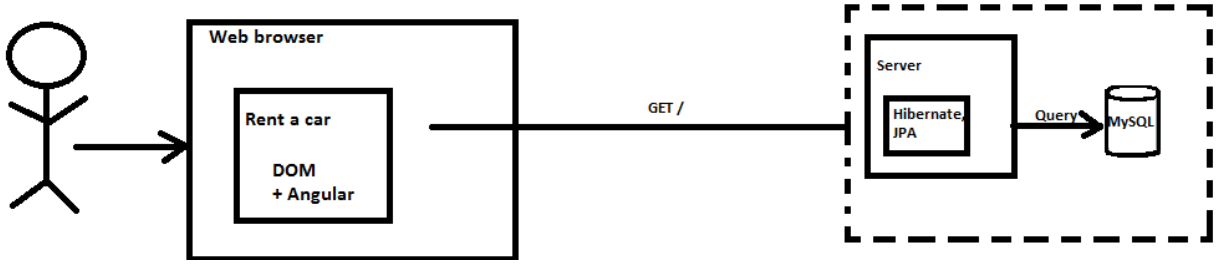
- **script-src 'self'** naznačava poverljive izvore sa kojih je dozvoljeno izvršavanje skripta na klijentskoj strani.
- **image-src** * naznačava izvore sa kojih je moguće učitavanje slika.

2.8 Insecure Deserialization

Potrebno je primati na endpointe samo podatke od poverljivih izvora, kao i proste tipove podataka.
Serializacione greške je potrebno trajno čuvati u log zapisima.

Neophodno je pratiti izveštaje alata koji detektuju serializacione greške, poput owasp zap-a.

2.9 Using Components with Known Vulnerabilities



Moderne aplikacije sadrže obimnu količinu (ranjivog) third-party koda.

Neophodno je:

1. Analizirati sistem u potrazi za starijim, neupotrebljenim bibliotekama,
2. Zaobići detektovane ranjivosti,
3. Redukovati aplikaciju na samo neophodne biblioteke.

Dependency izveštaj

Biblioteke na koje se rent-a-car rešenje oslanja su u segmentu ispod navedene u formatu Spring: "*groupId:artifactId" – "verzija*".
Analiza ranjivosti sprovedena je pomoću CVE Details platforme i baze podataka, prikazana ispod *dependency-a*.

Dependabot [8] github alat skenira biblioteke u sistemu, gde unutar *pull requesta* svakog od zahteva je naveden detaljan changelog biblioteke. Ovim alatom je moguće detektovanje poznatih ranjivosti onih biblioteka koje dependabot prepozna.

Confidentiality Impact	Partial (There is considerable informational disclosure.)
Integrity Impact	Partial (Modification of some system files or information is possible, but the attacker does not have control over what can be modified, or the scope of what the attacker can affect is limited.)
Availability Impact	Partial (There is reduced performance or interruptions in resource availability.)
Access Complexity	Low (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit.)
Authentication	Not required (Authentication is not required to exploit the vulnerability.
Gained Access	None

- primer CVE izveštaja.

Spring dependencies:

- org.springframework.boot:spring-boot-starter-parent – 2.3.0.RELEASE.

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2018-1196	59			2018-03-19	2018-04-20	4.3	None	Remote	Medium	Not required	None	Partial	None
Spring Boot supports an embedded launch script that can be used to easily run the application as a systemd or init.d linux service. The script included with Spring Boot 1.5.9 and earlier and 2.0.0.M1 through 2.0.0.M7 is susceptible to a symlink attack which allows the "run_user" to overwrite and take ownership of any file on the same system. In order to instigate the attack, the application must be installed as a service and the "run_user" requires shell access to the server. Spring Boot application that are not installed as a service, or are not using the embedded launch script are not susceptible.														
2	CVE-2017-8046	20			2018-01-04	2018-08-15	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Malicious PATCH requests submitted to servers using Spring Data REST versions prior to 2.6.9 (Ingalls SR9), versions prior to 3.0.1 (Kay SR1) and Spring Boot versions prior to 1.5.9, 2.0 M6 can use specially crafted JSON data to run arbitrary Java code.														

- **Pronađene ranjivosti definisane u tabeli su od verzije 2.x.x tretirane**.
- **Napomena, enkapsulirane biblioteke unutar starter-parent-a su*: spring-boot-starter-security, spring-boot-starter-data-jpa, spring-boot-starter-web, spring-boot-starter-log4j2, spring-boot-starter-amqp, spring-boot-starter-mail, spring-boot-starter-data-rest, spring-boot-starter-validation***

- io.jsonwebtoken:jjwt – 0.9.1

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.

- **Nije pronađena dokumentovana ranjivost prethodnih verzija na CVE, no analizom github izvornog koda autori nailaze na napomenu verzije 0.8:** "[...] The following dependencies were updated to the latest release version: maven compiler, maven enforcer, maven failsafe, maven release, maven scm provider, maven bundle, maven gpg, maven source, maven javadoc, jackson, bouncy castle, groovy, logback and powermock. Of significance, is the upgrade for jackson as a security issue was addressed in its latest release.." [9]

- mysql:mysql-connector-java – 8.0.20

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2019-2692	20			2019-04-23	2019-09-30	3.5	None	Local	High	Single system	Partial	Partial	Partial
Vulnerability in the MySQL Connectors component of Oracle MySQL (subcomponent: Connector/J). Supported versions that are affected are 8.0.15 and prior. Difficult to exploit vulnerability allows high privileged attacker with logon to the infrastructure where MySQL Connectors executes to compromise MySQL Connectors. Successful attacks require human interaction from a person other than the attacker. Successful attacks of this vulnerability can result in takeover of MySQL Connectors. CVSS 3.0 Base Score 6.3 (Confidentiality, Integrity and Availability impacts). CVSS Vector: (CVSS:3.0/AV:L/AC:H/PR:H/UI:R/S:U/C:H/I:H/A:H).													

- **Ranjivost pronađena 2019-te se zaobilazi upotrebom verzija od 8.0.16**

- org.projectlombok:lombok – 1.18.12

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.

- **Lombok projekat je build-time only dependency i nije neophodan prilikom upotrebe aplikacije u produkciji. Kao takav, veoma malo je verovatno da poseduje ranjivosti.**

- com.fasterxml.jackson.core:jackson-core – 2.11.0

ž	CVE-2019-17267	20			2019-10-06	2019-10-10	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A Polymorphic Typing issue was discovered in FasterXML jackson-databind before 2.9.10. It is related to net.sf.ehcache.hibernate.EhcacheJtaTransactionManagerLookup.														
2	CVE-2019-16943	20			2019-10-01	2019-10-11	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A Polymorphic Typing issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.10. When Default Typing is enabled (either globally or for a specific property) for an externally exposed JSON endpoint and the service has the p6spy (3.8.6) jar in the classpath, and an attacker can find an RMI service endpoint to access, it is possible to make the service execute a malicious payload. This issue exists because of com.p6spy.engine.spy.P6DataSource mishandling.														
3	CVE-2019-16942	20			2019-10-01	2019-10-08	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A Polymorphic Typing issue was discovered in FasterXML jackson-databind 2.0.0 through 2.9.10. When Default Typing is enabled (either globally or for a specific property) for an externally exposed JSON endpoint and the service has the commons-dbcp (1.4) jar in the classpath, and an attacker can find an RMI service endpoint to access, it is possible to make the service execute a malicious payload. This issue exists because of org.apache.commons.dbcp.datasources.SharedPoolDataSource and org.apache.commons.dbcp.datasources.PerUserPoolDataSource mishandling.														
4	CVE-2019-16335	20			2019-09-15	2019-09-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A Polymorphic Typing issue was discovered in FasterXML jackson-databind before 2.9.10. It is related to com.zaxxer.hikari.HikariDataSource. This is a different vulnerability than CVE-2019-14540.														
5	CVE-2019-14540	20			2019-09-15	2019-09-24	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A Polymorphic Typing issue was discovered in FasterXML jackson-databind before 2.9.10. It is related to com.zaxxer.hikari.HikariConfig.														
6	CVE-2019-14439	200	+Info		2019-07-30	2019-09-05	5.0	None	Remote	Low	Not required	Partial	None	None
A Polymorphic Typing issue was discovered in FasterXML jackson-databind 2.x before 2.9.9.2. This occurs when Default Typing is enabled (either globally or for a specific property) for an externally exposed JSON endpoint and the service has the logback jar in the classpath.														
7	CVE-2019-14379	20	Exec Code		2019-07-29	2019-10-06	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
SubTypeValidator.java in FasterXML jackson-databind before 2.9.9.2 mishandles default typing when ehcache is used (because of net.sf.ehcache.transaction.manager.DefaultTransactionManagerLookup), leading to remote code execution.														
8	CVE-2019-12814	200	+Info		2019-06-19	2019-09-05	4.3	None	Remote	Medium	Not required	Partial	None	None
A Polymorphic Typing issue was discovered in FasterXML jackson-databind 2.x through 2.9.9. When Default Typing is enabled (either globally or for a specific property) for an externally exposed JSON endpoint and the service has JDOM 1.x or 2.x jar in the classpath, an attacker can send a specifically crafted JSON message that allows them to read arbitrary local files on the server.														
9	CVE-2019-12384	502	Exec Code		2019-06-24	2019-09-05	4.3	None	Remote	Medium	Not required	Partial	None	None
FasterXML jackson-databind 2.x before 2.9.9.1 might allow attackers to have a variety of impacts by leveraging failure to block the logback-core class from polymorphic deserialization. Depending on the classpath content, remote code execution may be possible.														
10	CVE-2019-12086	200	+Info		2019-05-17	2019-09-17	5.0	None	Remote	Low	Not required	Partial	None	None
A Polymorphic Typing issue was discovered in FasterXML jackson-databind 2.x before 2.9.9. When Default Typing is enabled (either globally or for a specific property) for an externally exposed JSON endpoint, the service has the mysql-connector-java jar (8.0.14 or earlier) in the classpath, and an attacker can host a crafted MySQL server reachable by the victim, an attacker can send a crafted JSON message that allows them to read arbitrary local files on the server. This occurs because of missing com.mysql.cj.jdbc.admin.MiniAdmin validation.														

- **Jackson biblioteka predstavlja i najranjiviju pristupnu tačku celokupnog sistema. U produkciji, neophodno je sa dosta odgovornosti pristupiti analizi upravo onih biblioteka koje su se istorijski pokazale kao ranjive, ili ovaj dependency ukloniti kompletno. Izložene ranjivosti (čiji je celokupan spisak evidentiranih 26. stavki) se tretiraju upotrebom najažurnije verzije 2.11.x databind dependency-a.**

- org.springframework.cloud:spring-cloud-starter-netflix-eureka-client–2.2.3

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.

- **Nije pronađena dokumentovana ranjivost prethodnih verzija na CVE, ili changelog podacima eureka client git stranice.**

- org.springframework.cloud:spring-cloud-starter-netflix-zuul – 2.2.3

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.

- **Nije pronađena dokumentovana ranjivost prethodnih verzija na CVE, no bitan update naveden pod sekcijom 3.10.2 changelogs** https://zuul-ci.org/docs/zuul/reference/releasenotes.html **je izbegnut upotrebom najažurnije verzije.**

- org.springframework.cloud:spring-cloud-starter-openfeign–2.2.3

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.

- **Praćenjem changelog-a** https://github.com/OpenFeign/feign/blob/master/CHANGELOG.md**, najažurnija verzija se takođe pokazuje kao najproverenija, gde od verzije 10.7 su ranjivosti tretirane.**

Angular dependencies:

```
"devDependencies": {
  ”@angular-devkit/build-angular” –”^0.803.24”,
  ”@angular/cli” –”~9.0.5”,
  ”@angular/compiler-cli” –” ~9.0.5”,
  ”@angular/language-service” –”~9.0.5”,
  ”@types/jasmine” –”~3.5.0”,
  ”@types/jasminewd2” –”~2.0.3”,
  ”@types/node” –”^12.11.1”,
  ”codelyzer” –”^5.1.2”,
  ”jasmine-core” –”3.5.0”,
  ”jasmine-spec-reporter” – ”~4.2.1”,
  ”karma” – ”~4.3.0”,
  ”karma-chrome-launcher” – ”~3.1.0”,
  ”karma-coverage-instanbul-reporter” – ”~2.1.0",
  ”karma-jasmine” – ”2.0.1”,
  ”karma-jasmine-html-reporter” – ”^1.4.2”,
  ”protractor” –”~5.4.3”,
  ”ts-node” –”~8.3.0”,
  ”tslint” –”~5.18.0”,
  ”typescript” –”~3.7.5”
 },

"dependencies": {
  ”@angular/animations” – ”^9.0.7”,
  ”@angular/cdk” – ”^9.2.1",
  ”@angular/common” – ”~9.0.5”,
  ”@angular/compiler” – ”~.9.0.5”,
  ”@angular/core” – ”~9.0.5”,
  ”@angular/flex-layout” – ”^9.0.0-beta.29”,
  ”@angular/forms” – ”~9.0.5”,
  ”@angular/material” – ”~9.2.1”,
  ”@angular/platform-browser” – ”~9.0.5”,
  ”@angular/platform-browser-dynamic” – ”~9.0.5”,
  ”@angular/router” – ”~9.0.5”,
  ”angular8-yandex-maps” – ”^1.11.4”,
  ”ngx-toastr” – ”^12.0.1”,
  ”rxjs” – ”~6.5.4”,
  ”tslib” – ”^1.10.0”,
  ”zone.js"– ”~0.10.2”
 }
```

Na spisku svih third party pronađene su sledeće ranjivosti pomoću *npm audit i dependabot*.

npm audit

1

Low	Prototype Pollution
Package	minimist
Dependency of	karma [dev]
Path	karma > optimist > minimist
More info	https://npmjs.com/advisories/1179

rešenje: `npm install --save-dev karma@5.0.9`

2

Low	Prototype Pollution
Package	yargs-parser
Dependency of	@angular-devkit/build-angular [dev]
Path	@angular-devkit/build-angular > webpack-dev-server > yargs > yargs-parser
More info	https://npmjs.com/advisories/1500

rešenje: `npm install --save-dev @angular-devkit/build-angular@0.901.7`

3

Low	Prototype Pollution
Package	yargs-parser
Dependency of	protractor [dev]
Path	protractor > yargs > yargs-parser
More info	https://npmjs.com/advisories/1500

rešenje: `npm install --save-dev protractor@7.0.0`

4

```
High            Denial of Service
Package         http-proxy
Dependency of   @angular-devkit/build-angular [dev]
Path            @angular-devkit/build-angular > webpack-dev-server >
                http-proxy-middleware > http-proxy
More info       https://npmjs.com/advisories/1486

High            Denial of Service
Package         http-proxy
Dependency of   karma [dev]
Path            karma > http-proxy
More info       https://npmjs.com/advisories/1486
```

rešenje: `npm update http-proxy --depth 4`

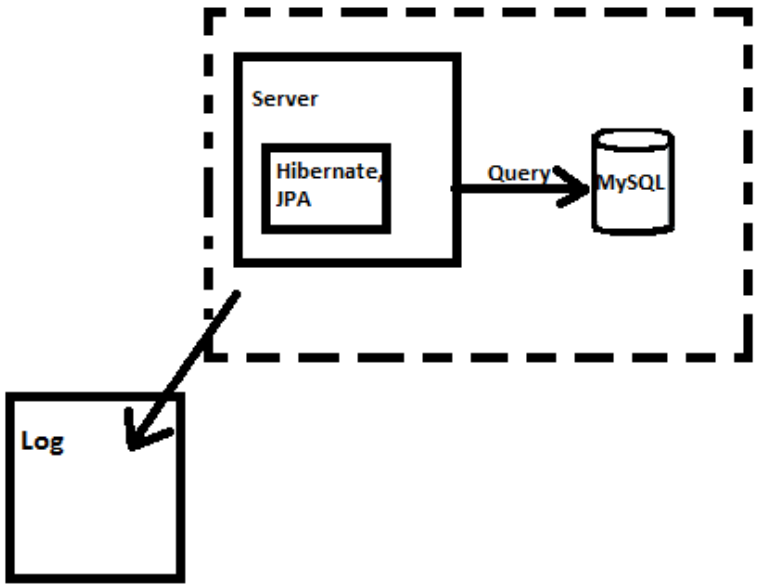
dependabot:

Biblioteke su skenirane i ažurirane su primarno front-end biblioteke pomoću dependabot alata.

Non-security related updates:
- ngx-toastr from 12.0.1 to 12.1.0 - https://github.com/n-dusan/wroom/pull/32
- ts-node from 8.3.0 to 8.10.2 - https://github.com/n-dusan/wroom/pull/33
- karma-jasmine from 2.0.1 to 3.3.1 - https://github.com/n-dusan/wroom/pull/35

Security related updates:
- websocket-extensions from 0.1.3 to 0.1.4 - https://github.com/n-dusan/wroom/pull/27

2.10 Insufficient Logging & Monitoring



Log zapisi garantuju evidenciju onih napada koji su uspešno zabeleženi i skladišteni.

Važnost log zapisana je trojaka:
- Detektovanje incidenata,
- Razumevanje spleta događaja u sistemu,
- Neporecivost

Svaka komponenta u mikroservisnoj rent-a-car aplikaciji sadrži sopstveno skladište apache log4j2 log datoteka, prilagođena nivou akcije ili greške koja se može desiti prilikom rada aplikacija.
Log datoteke su podeljene na **error** (*detektovanje incidenata)*, **info** (*razumevanje spleta događaja)* i **total** log zapise, time olakšavajući pretraživanje željenih prošlih akcija u sistemu.
Log zapisi se regenerišu na svakih 10MB zauzeća, i svaki zapis se trajno skladišti u odgovarajuće direktorijume na osnovu datuma kreiranja (ISO 8061 standard).
Svaki log zapis mora biti zaštićen od strane neautorizovanog pristupa, i to je postignuto ***ACL*** konfiguracijom nad log direktorijumom.

Neporecivost akcije u log zapisu je postignuta evidentiranjem jedinstvenog identifikatora korisnika u sistemu koji je izvršio datu akciju nad datim resursom.

primer loga za prijavljivanje na sistem:
2020-06-05 01:56:27,133 INFO x.w.c.AuthController [https-jsse-nio-8081-exec-8] action=login user=mila@maildrop.cc ip_address=0:0:0:0:0:0:0:1 times=1

Ovakvim oblikom sažetog ključ=vrednost zapisa bi se znatno optimizovala analiza od strane **SIEM** alata, koji bi bio neophodan u detaljnoj analizi aplikacije u produkciji.

Evidentiranjem user segmenta log zapisa, kao IP adrese sa koje pristiže zahtev, može se na tačno identifikovati korisnik u sistemu, i otvara mogućnost implementacije *blacklist-ovanja* korisnika, ukoliko bi broj zahteva ka određenoj pristupnoj tački u sistemu prevazišao neki unapred zadat kriterijum. Konkretno, brute force familija napada.

3. Pentest izveštaj

Za potrebe penetracionog testiranja, upotrebljeni su OWASP ZAP[10] i dirbuster[11].

U okviru kandidatskih alata koji su spomenuti i mogli biti upotrebljeni, autori su se odlučili za komplementarnu kombinaciju ZAP – dirbuster, gde proxy scanning obavlja ZAP, dok proveru direktorijuma i skrivenih datoteka vrši dirbuster.
sqlmap je takođe bila mogućnost, no prilikom testiranja SQLi napada, autori su utvrdili da ZAP izveštaj ne pruža nikakve SQLi ranjivosti, te je sqlmap ostavljen sa strane.

Aplikacije su testirane unutar localhost https domena, windows 7 i 10 operativnim sistemima.

Izveštaj poseduje prikaz ranjivosti i njihovih razrešenja, u *proof of concept** ili implementacionom obliku.
Takođe, prilikom testiranja su dozvoljeni napadi na aplikaciju od strane neprivilegovanih korisnika. U produkciji i lokalu, operacije i napadi nad resursima će biti znatno otežani činjenicom da endpointi sadrže @PreAuthorize proveru privilegije ulogovanog korisnika. Autorizacijom, napadač za određene endpointe mora imati elevated pristup, što se vraća na segment Broken Authentication.

***Neke od ranjivosti navedene unutar alata se neće razrešavati za potrebe pokretanja aplikacije u lokalu, već u produkciji. Za potrebe ovog dokumenta, produkcija će se smatrati docker kontejner u kojem sistem izolovano funkcioniše.*

3.1 Zed attack proxy

Nakon otkrivanja endpoint-a aplikacije prolaženjem kroz ajax spider crawler, izvršen je *active scanning*, otkrivajući potencijalne ranjivosti primenom poznatih vrsta napada.

Terminologija:

Alert – naziv ranjivosti registrovan od strane *zap-a.*
Risk – nivo opasnosti koju zap alat interpretira, a vrednosti su:
- High,
- Medium,
- Low

Evidence – HTTPS događaj na osnovu kojeg je ranjivost pronađena.
Description – opis vrste napada.
Solution – PoC ili implementacioni.

Izveštaj

- Path Traversal **(*High)***
  - **Evidence*:*** *unspecified*
  - **Description: “**The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory.”
  - **Solution:**
    - Upotreba pravilne sanitizacije korisničkog unosa. U okviru projekta, korišćen je regexp obrazac hibernate validator biblioteke.
    - Pokretanje projekta u okviru kontejnera gde je pristup file systemu od strane malicioznog unosa nemoguć.

- Buffer overflow **(*Medium)***
  - **Evidence:** Content-Type: application/json, Origin: https://localhost:4200, Fetch-Site: same-site, Sec-Fetch-Mode: cors, Sec-Fetch-Dest: empty, Referer: https://localhost:4200/ Accept-Language: en-US,en;q=0.9, Host: localhost:8081
  - **Description**: Buffer overflow errors are characterized by the overwriting of memory spaces of the background web process, which should have never been modified intentionally or unintentionally. Overwriting values of the IP (Instruction Pointer), BP (Base Pointer) and other registers causes exceptions, segmentation faults, and other process errors to occur. Usually these errors end execution of the application in an unexpected way.
  - **Solution:** Refaktorisana validacija za korisnički unos gde se vrši provera dužine sekvence karaktera koji se unose od strane korisnika.

- Cross-site Scriping Weakness (Persistent in JSON Response) **(*Low*)**
    - **Evidence:** <script>alert(1);</script>
    - **Description:** A XSS attack was found in a JSON response, this might leave content consumers vulnerable to attack if they don't appropriately handle the data (response). Raised with LOW confidence as the Content-Type is not HTML.
    - **Solution:** Upotreba @Pattern(regexp="^[A-Za-z0-9]+$") validatora na svim korisničkim unosima. Takođe, Angular vrši defaultnu sanitizaciju svih prispelih karaktera, kao i CSP konfiguracija onemogućava efikasnost persisted XSS skripti.

- Information Disclosure - Debug Error Messages (***Low)***
    - **Evidence:** *unspecified*
    - **Description:** The response appeared to contain common error messages returned by platforms such as ASP.NET, and Web-servers such as IIS and Apache. You can configure the list of common debug messages.
    - **Solution:** Prepravljena provera null vrednosti na određenim endpointima. Bitno, jer bilo kakve default error poruke poslate klijentskoj strani pružaju napadaču dodatne informacije o aplikaciji, koje bi mogle biti korisne za dalje oblike napada.

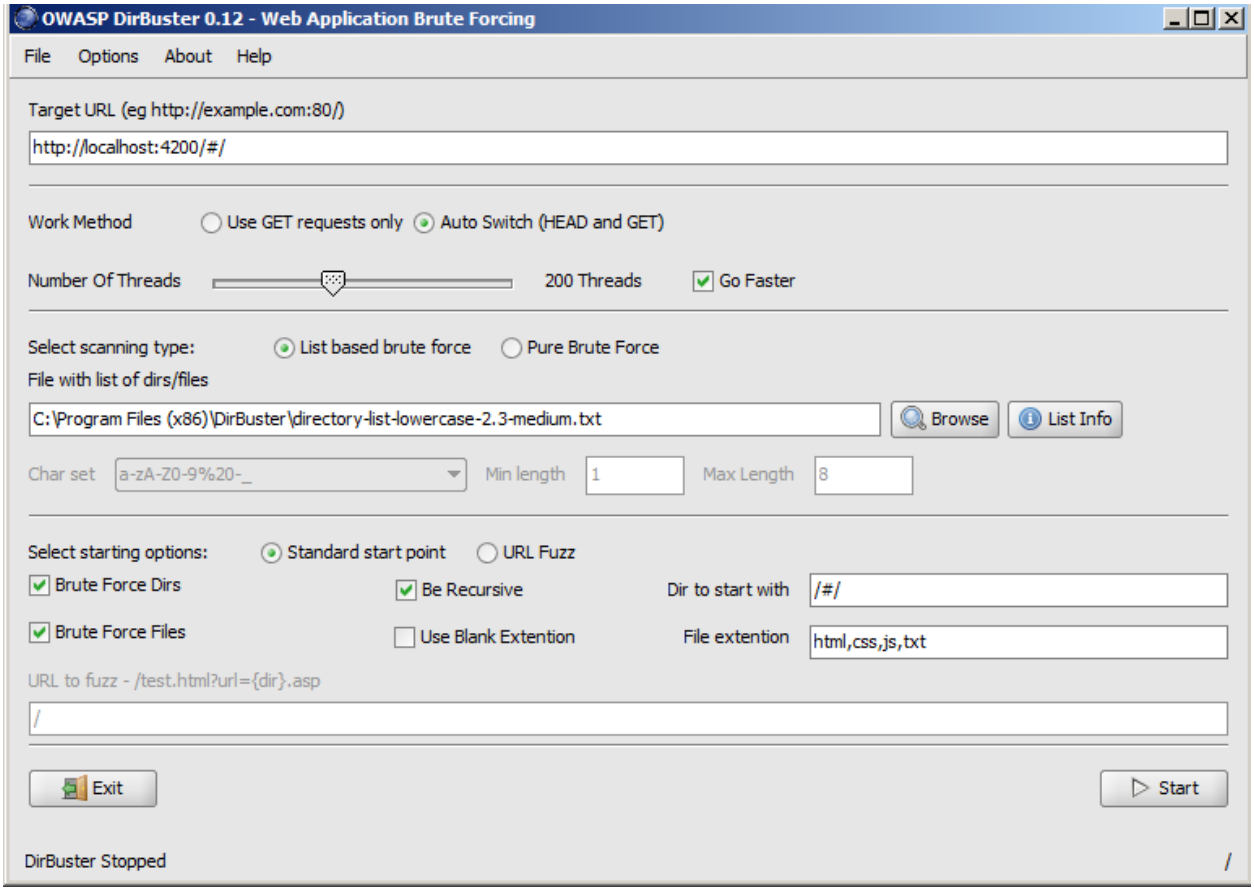
- X-Content-Type-Options Header Missing (***Low)***
    - **Evidence:** *unspecified*
    - **Description:** The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type.
    - **Solution:** Prepravljena .antMatchers konfiguracija unutar Spring Security konfiguracije, kao i omogućena default cache control Spring Security konfiguracija, koja se smatra best practice.

3.2 dirbuster

Dirbuster je aplikacija koja omogućava indeksiranje stranica i endpointa na sajtu koji nisu normalno prisutni prilikom *google* indeks pretraživača. Na primer, *www.mojsajt.com/admin.* Normalno, Google crawler *indexer* skenira klijentski source kod i traži linkove ka drugim stranicama unutar sajta. Algoritam posećuje pronađene stranice i radi rekurzivno dok ne formira mapu stranica.
Kao developer ili napadač, potrebno je skenirati stranice u potrazi za ranjivostima ili korisnim informacijama. Pošto Google crawler ne pronalazi sve stranice u okviru sajta jer prosto neke stranice ne sadrže direktan link u html kodu, koriste se third-party alati za skeniranje „skrivenih putanja" u aplikaciji.

Skrivene putanje su od posebnog interesa jer možda sadrže više ranjivosti od glavnih stranica sajta.
Dirbuster će pročešljati aplikaciju u potrazi za neindeksiranim resursima koristeći wordlist čestih naziva stranica i brute-force slati dictionary upite ka sajtu.

konfiguracija



- konfiguracioni prozor

Za potrebe testiranja skrivenih putanja, koristiće se list-based brute force skeniranje.
Pure brute force sken bi zahtevao duži rad alata, no ne bi doneo značajne rezultate u odnosu na postojeće rečnike.
Rečnik u upotrebi će biti lowercase *medium*, datoteka veličine 2MB u kojoj se nalazi rečnik najčešćih naziva direktorijuma i putanja.
Ukoliko bi se koristio lowercase *big,* datoteka za red veličine veća od *medium*, vreme izvršavanja brute-force napada bi iznosio par sati, i suštinski ne bi napravio veliku razliku u pronalasku.

Number of threads – 200. Veća količina zahteva od ovoga bi, u zavisnosti od hardverskih karakteristika servera, mogla predstavljati oblik *denial of service* napada.

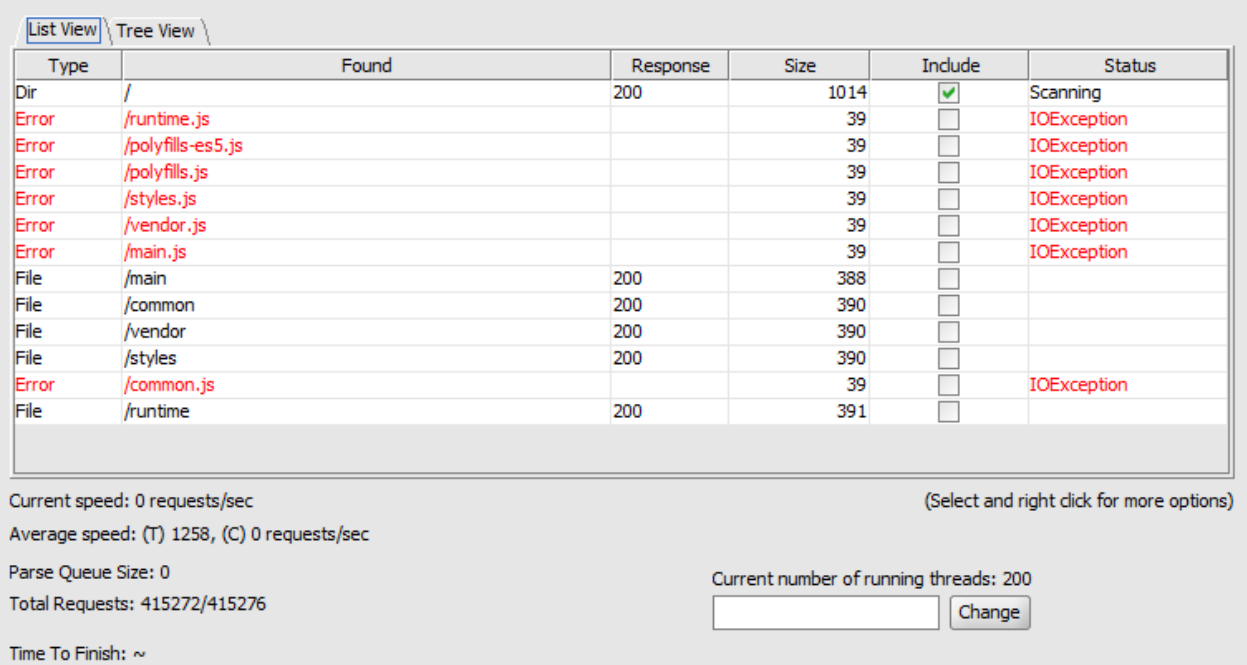
Rečnik će se rekurzivno primenjivati na otkrivene poddirektorijume.

pokušaj 1:

target url: http://localhost:4200/#/

- brute-force dirs, brute-force files, recursive, blank extension
- rečnik: directory-list-lowercase-medium

rezultati:



Pronađene su samo source code klijentske datoteke.

pokušaj 2:

target url: http://localhost:4200/#/

- brute-force dirs, brute-force files, recursive, blank extension
- rečnik: directory-list-lowercase-medium
- Podešen “Authorization” header i dobavljen JWT sa beka (simulacija polomljene autentifikacije) u dirbusteru

rezultati:

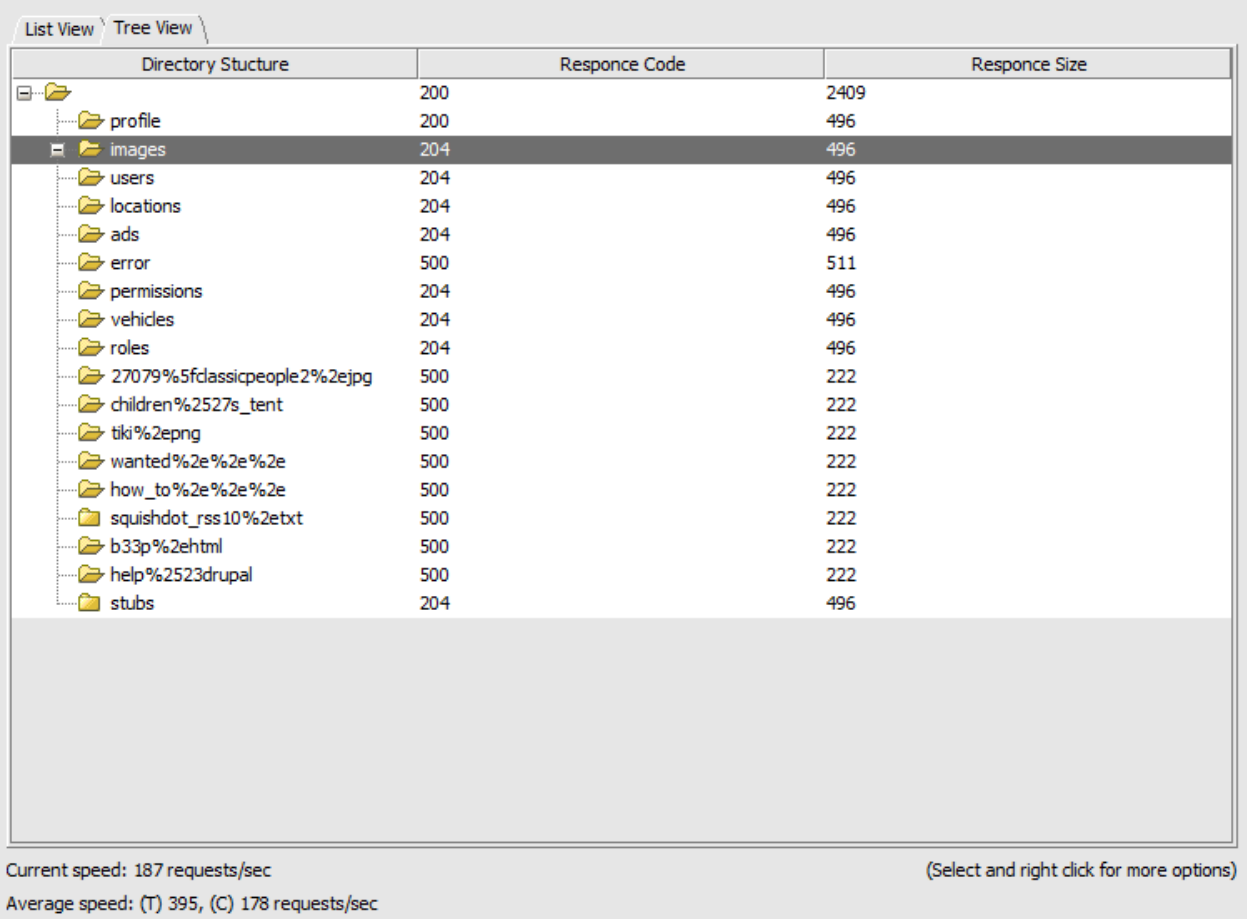
Identičan rezultat kao u slučaju 1.

pokušaj 3:

target url: http://localhost:8081

- brute-force dirs, brute-force files, recursive, blank extension
- rečnik: directory-list-lowercase-medium

rezultati:



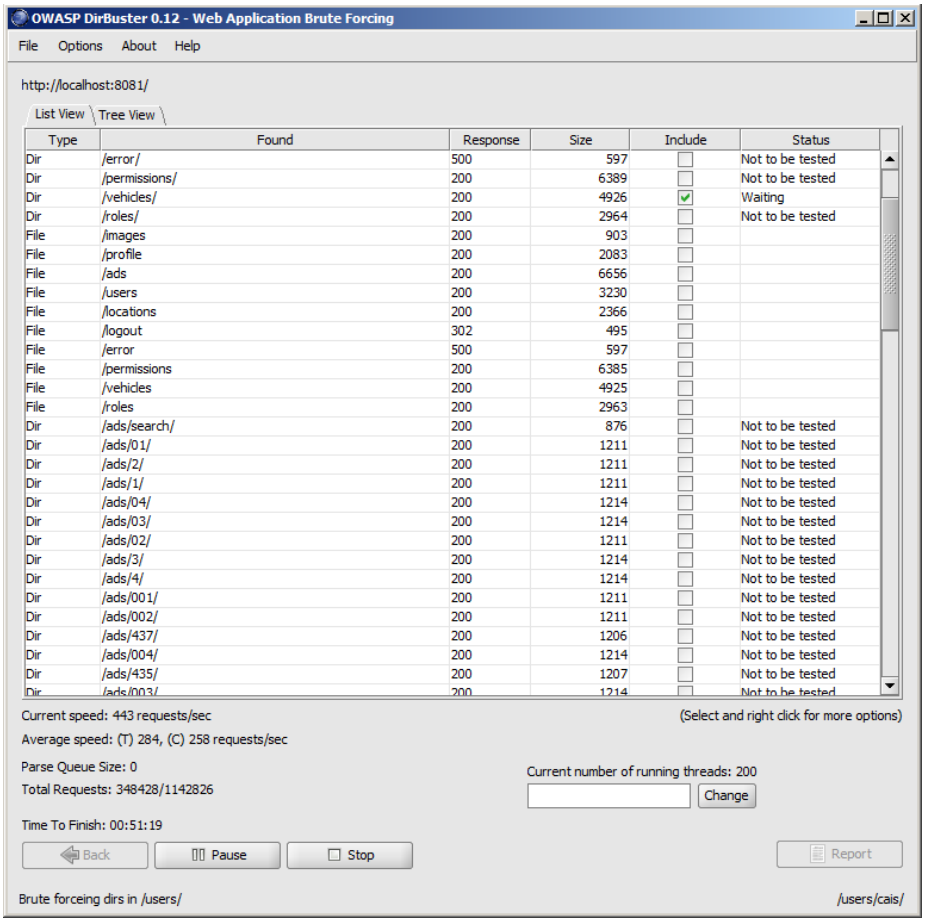
Brute force tehnikom, oko pola sata izvršavanja pronađeni su skoro svi endpointi bekenda koji postoje. Takođe, čitalac primećuje i false-positive rezultate koje nisu stvarne datoteke, već evidentiran odgovor 500 od strane servera na dati zahtev.

pokušaj 4:

target url: http://localhost:8081

- brute-force dirs, brute-force files, recursive, blank extension
- rečnik: directory-list-lowercase-small

rezultati:



Nad manjim rečnikom i pušten da radi oko 2h, pronalazi endpointe za pojedinačan oglas (ads/xxx) što značajno usporava brute-force sistem, jer algoritam se primenjuje rekurzivno. Pronalazi sve endpointe kao i nad medium rečnikom, ali sa generalno kraćim vremenom izvršavanja. Alat je prinudno zaustavljen.

zaključak

Dirbuster se pokazao kao zanimljiv, no ne i korisan alat za spring boot/angular aplikacije. Naime, problem je u uri šemi koju RESTful aplikacije nameću, koja ukoliko je od strane razvojnog tima ispraćena ne bi sadržala specifične sakrivene putanje.

Moderne web aplikacije prate single page paradigmu, i brute-force mehanizmi dirbuster alata se nisu pokazali kao efikasni.
Otkrivene datoteke i *tree view* omogućavaju napadačima da usmere svoju pažnju ka pronađenim čvorovima bilo koje ranjive web aplikacije.

Dirbuster alat bi takođe bio obeshrabren i invalidiran ukoliko u produkciji se server nalazi u sandbox izolovanom okruženju, bez dodatno osetljivih datoteka.
No, wordpress i php klasične web aplikacije bi bile sasvim validna meta ovog alata, i autori veruju da njegova efikasnost ukoliko je uperena ka pravilnoj aplikaciji, može biti delotvorna.

4. Reference

[1] – OWASP fondacija https://owasp.org/
[2] – 2017-ta najvažnijih 10 ranjivosti https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/
[3] – password strength best practices https://en.wikipedia.org/wiki/Password_strength#cite_note-24
[4] -_SDE https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure
[5] – jaxb https://stackoverflow.com/questions/28310617/prevent-xxe-external-entity-processing-attack-with-jaxb-spring-restful-web-s
[6] – Preventing cross-site scripting (XSS) segment https://angular.io/guide/security
[7] – uvod u CSP https://www.html5rocks.com/en/tutorials/security/content-security-policy/
[8] – dependabot https://github.com/marketplace/dependabot-preview
[9] – jjwt github 0.8 changelog https://github.com/jwtk/jjwt/blob/master/CHANGELOG.md
[10] – zap https://owasp.org/www-project-zap/
[11] – dirbuster https://pentestingtr.blogspot.com/2016/06/dirbuster.html

5. Autori

- Ana Svitlica RA 176-2016
- Olivera Sekulić RA 186-2016
- Dušan Nikolić RA 187-2016