GAP analiza

A1-Injection

- Injection napad predstavlja familiju napada koje se ispoljavaju kada se interpreteru proslijedi nevalidirani podaci pomću kojih se izvršavaju neautorizovane instrukcije i uz pomoc kojima se pristupa neautorizovanim podacima. Injection napadi obuhvataju sledeće napade: SQL, OS,XXE,LOG itd.
- Npr. SQL Injection se ekspolatišem direktnim proslijeđivanjem ulaznih parametara u SQL upite.
  UserID: 105 or 1=1
  SELECT * FROM Users where UserId=105 or 1=1;
- Ranjivosti detektovane u implementiranom sistemu:
  - SQL injection
    - Zaštita - upotreba Hibernate-ovog validatora i parametrizovanih upita;
  - LOG injection
    - Zaštita - upotreba klase Logger iz slf4j biblioteke i konfigurisanjem njegovog formata;
  - XXE injection
    - Zaštita - konfigurisanje JAX-B parsera da onemogući eksternih entiteta (IS_SUPPORTING_EXTERNAL_ENTITIES=false, default-na konfiguracija je takva);
  - HQL injection
    - Zaštita - upotreba parametrizovanih upita;

Reference :

https://www.w3schools.com/sql/sql_injection.asp

A2-Broken Authentification and Session Management

- Aplikacije koje ispoljavaju ovu ranjivost su podložne kompromitovanju lozinki i Session ID-a. Aplikacija može biti ranjiva ako:
  - Korisnički kredencijali nisu propisno zaštićeni
    - Zaštita - upotreba PBKDF2 (hash & salt) mehanizma za heširanje lozinki;
  - Upotreba slabih lozinki
    - Zaštita – inicijalna lozinka sadrži 12 karaktera (mala slova i cifre) i ističe 2 sata nakon kreiranja;
    - Zaštita – upotreba jakih lozinki pri čemu se izbjegavaju uobičajeni šabloni ;

- Session ID su prikazani u URL-u
  - Zaštita – ni sa jednim zahtjevom Session ID neće biti smješten u URL;
- Session fixation
  - Zaštita – pri svakom uspostavnljanju sesije generiše se novi Session ID;
- Nezaštićena prenos kredecijala
  - Zaštita – konfigurisanje HTTPS protokola;

Reference :

https://docs.spring.io/spring-boot/docs/current/reference/html/howto-embedded-servlet-containers.html

A3-Cross-Site Scripting

- Sadržaj koji server šalje kao saržaj html stranic je izvršiv od strane browser-a. Dva primjera XSS napada su Stored XSS and Reflected XSS. Stored XSS se realizuje u slučaju da se malciozni izvršivi podaci perzistuju na serveru. Reflected XSS napada podrazumeva direktno prikazivanje korisničkog unosa u browseru, pre nego što se na serveru unos validira.
- Zaštita - upotreba AngularJS v1.4.8. vrši zaštiti aplikacije od XSS napada eskejpovanjem specijalnih karaktera prije obrade od strane browser-a

Reference :

https://nvd.nist.gov/vuln/search/results?adv_search=false&form_type=basic&results_type=overview&search_type=all&query=angular

A4-Broken Access Control

- Aplikacija je izložena ovoj stavci ako nije spriječen:
  - Neautorizovan pristup podacima
    - Zaštita - podrazumjeva konfigurisanje kontrole pristupa na nivo sistema za upravljanje bazom podataka
  - Neautorizovan pristup metoda
    - Zaštita – sistem za kontrolu pristupa na nivou aplikacije (RBAC) upotreba interseptora i custom annotation-a.

A5-Security Misconfiguration

Loša konfiguracija bezbednosti aplikacije bi podrazumevala:

- Upotrebu neaužuriranih softvera, koji uključuju operativni sistem, sistem za upravljanje bazom podataka, komponentama i spoljašnjim bibliotekama
- Postojanje aktivnih portova, servisa, stranica i privilegija koji ne bi trebali postojati u finalnoj verziji aplikacije.
- Postojanje predefinisanih naloga sa visokim privilegijama (admin/admin).
- Otkrivanje previše informacija korisnicima prilikom rukovanja greškama.

A6-Sensitive Data Exposure

- Da bi se zaštitili od ove slabosti na osnovu domena posmatranog sistema neophodno je uočiti osjetljive podatke i načine za njihovu zaštitu. Za naš sistem to su brojevi računa, lozinke i mejlovi.
    - Čuvanje podataka kao otvorenog teksta
        - Zaštita – heširanje lozinki kao najkritičnijih podataka našeg sistema; Čuvanje broja racuna i mail adresa u otvorenom tekstu zarad performansi;
    - Repliciranje baze podataka
        - Zaštita-nije implementirana u nedostatku resursa;
    - Prenos podataka u otvorenom tekstu
        - Zaštita-konfigurisanje HTTPS protokola;
    - Upotreba zastarjelih slabih kriptografskih algoritama
        - Zaštita – za simetričnu kriptografiju je korišćen AES/CBC algoritam, a za asisimetričnu RSA
    - Propisana dužina ključeva za kriptovanje
        - Zaštita-za AES je korišćen ključ dužine 128 bita, dok je za RSA korišćen 2048 bitni;

Reference:

Materijali sa vežbi.

A7-Insufficient Attack Protection

- Po uvođenju sistema u produkciju, dolazi do otkrivanja novih ranjivosti. Njihove manifestacije prijavljuju korisnici ili se otkrivaju analiziranjem ponašanja sistema. Nadgledanje ponašanja sistema moguće je vršiti postavljanjem SIEM alata koji će

beležiti događaje u sistemu. Dodatna mera bezbednosti koja detektuje malicioznog korisnika sa potpunom sigurnošću je tzv. honeypot mehanizam.

- Reakcije na događaje koje sistem smatra neuobičajenim su logovi na nivou upozorenja ili greške, notifikacije ovlašćenih korisnika (često, administratora) u realnom vremenu, blokiranje zahteva, profila i IP adresa koji se smatraju malicioznim. S obzirom na raslojenu arhitekturu sistema, u slučaju detekcije napada u jednoj od zona, prekida se komunikacija sa ostalim zonama kako bi se sprečila propagacija napada u ostale zone i održala funkcionalnost ostatka aplikacije.

Reference:

https://www.owasp.org/index.php/Top_10_2017-A7-Insufficient_Attack_Protection

Materijali sa vežbi.

A8-Cross Site Request Forgery

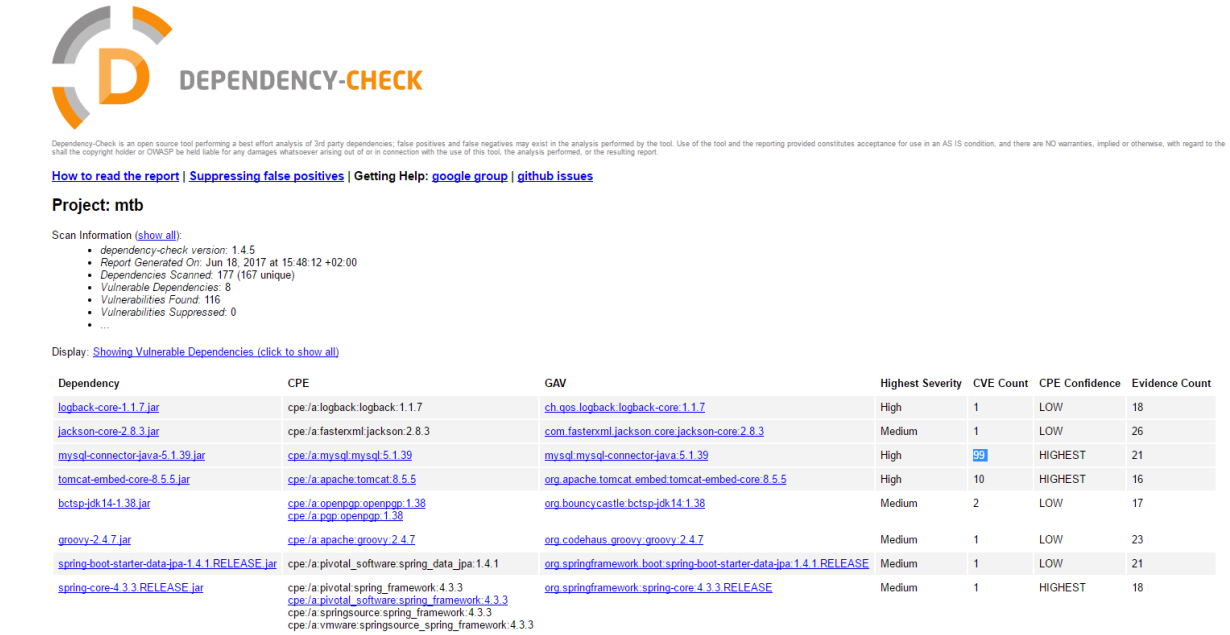
- U zaglavljima svakog zahtjeva koji stigne na server mora da postoji X-XSRF-TOKEN. Bez tih tokena napadači mogu da šalju maliciozne zahtjeve u ime ulogovanog korisnika.
- Zaštita-sa inicijalizacijom stranice šalje se zahtjev serveru za generisanje CSRF-TOKEN-a koji se potom smješta u sesiju, vraća i čuva na frontend-u. Angular-ov interceptor presreće svaki zahtjev poslat sa te stranice i u zaglavlja dodaje dobijeni token. Spring-ov interceptor presreće zahtjev presreće zahtjev prije izvršavanja metode na serveru i provjerava da li se token iz sesije i pristigli token poklapaju.

Reference :

https://stormpath.com/blog/angular-xsrf

A9-Using components with known vulnerabilities

- U cilju analiziranja ranjivosti spoljašnjih zavisnosti, korišćen je OWASP-ov Dependency-Check plugin. Rezultati prvobitne analize su generisani u output.html-u (Slika 1.0).



DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: google group | github issues

Project: mtb

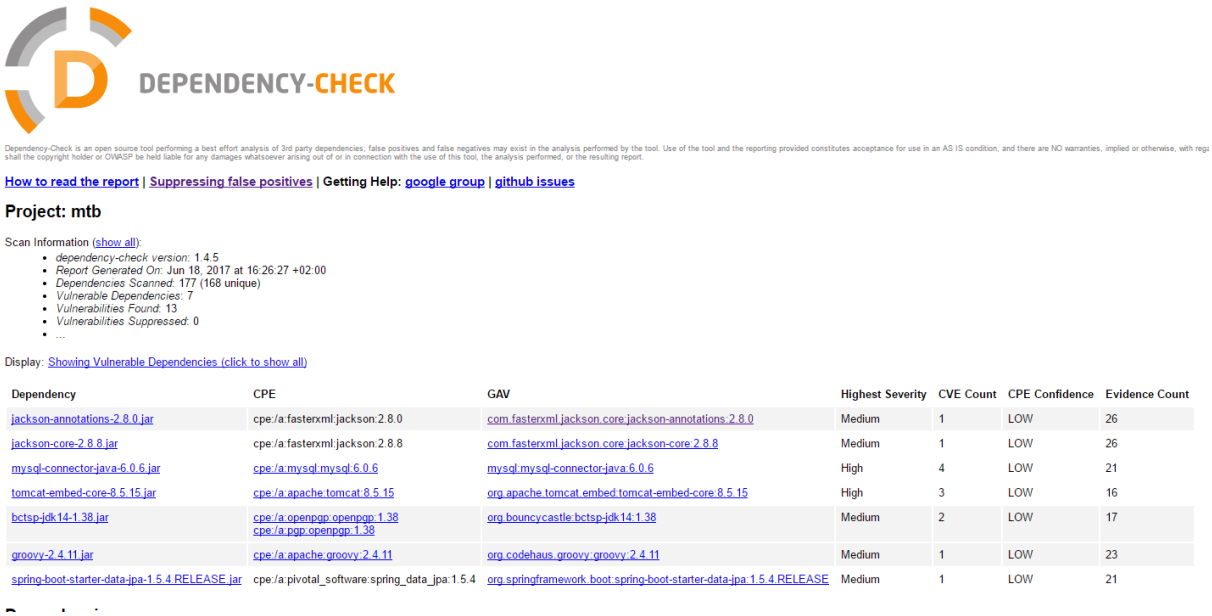
Scan Information (show all):

- *dependency-check version:* 1.4.5
- *Report Generated On:* Jun 18, 2017 at 15:48:12 +02:00
- *Dependencies Scanned:* 177 (167 unique)
- *Vulnerable Dependencies:* 8
- *Vulnerabilities Found:* 116
- *Vulnerabilities Suppressed:* 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
logback-core-1.1.7.jar	cpe:/a:logback:logback:1.1.7	ch.qos.logback:logback-core:1.1.7	High	1	LOW	18
jackson-core-2.8.3.jar	cpe:/a:fasterxml:jackson:2.8.3	com.fasterxml.jackson.core:jackson-core:2.8.3	Medium	1	LOW	26
mysql-connector-java-5.1.39.jar	cpe:/a:mysql:mysql:5.1.39	mysql:mysql-connector-java:5.1.39	High	99	HIGHEST	21
tomcat-embed-core-8.5.5.jar	cpe:/a:apache:tomcat:8.5.5	org.apache.tomcat.embed:tomcat-embed-core:8.5.5	High	10	HIGHEST	16
bctsp-jdk14-1.38.jar	cpe:/a:openpgp:openpgp:1.38 cpe:/a:pgp:openpgp:1.38	org.bouncycastle:bctsp-jdk14:1.38	Medium	2	LOW	17
groovy-2.4.7.jar	cpe:/a:apache:groovy:2.4.7	org.codehaus.groovy:groovy:2.4.7	Medium	1	LOW	23
spring-boot-starter-data-jpa-1.4.1.RELEASE.jar	cpe:/a:pivotal_software:spring_data_jpa:1.4.1	org.springframework.boot:spring-boot-starter-data-jpa:1.4.1.RELEASE	Medium	1	LOW	21
spring-core-4.3.3.RELEASE.jar	cpe:/a:pivotal:spring_framework:4.3.3 cpe:/a:pivotal_software:spring_framework:4.3.3 cpe:/a:springsource:spring_framework:4.3.3 cpe:/a:vmware:springsource_spring_framework:4.3.3	org.springframework:spring-core:4.3.3.RELEASE	Medium	1	HIGHEST	18

- Posle promena verzija zavisnosti sa najvećim nivoima opasnosti, dobijeni su sledeći rezulti(Slika 1.1):



DEPENDENCY-CHECK

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and there are NO warranties, implied or otherwise, with regard to the shall the copyright holder or OWASP be held liable for any damages whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | Suppressing false positives | Getting Help: google group | github issues

Project: mtb

Scan Information (show all):

- *dependency-check version:* 1.4.5
- *Report Generated On:* Jun 18, 2017 at 16:26:27 +02:00
- *Dependencies Scanned:* 177 (168 unique)
- *Vulnerable Dependencies:* 7
- *Vulnerabilities Found:* 13
- *Vulnerabilities Suppressed:* 0
- ...

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	CPE	GAV	Highest Severity	CVE Count	CPE Confidence	Evidence Count
jackson-annotations-2.8.0.jar	cpe:/a:fasterxml:jackson:2.8.0	com.fasterxml.jackson.core:jackson-annotations:2.8.0	Medium	1	LOW	26
jackson-core-2.8.8.jar	cpe:/a:fasterxml:jackson:2.8.8	com.fasterxml.jackson.core:jackson-core:2.8.8	Medium	1	LOW	26
mysql-connector-java-6.0.6.jar	cpe:/a:mysql:mysql:6.0.6	mysql:mysql-connector-java:6.0.6	High	4	LOW	21
tomcat-embed-core-8.5.15.jar	cpe:/a:apache:tomcat:8.5.15	org.apache.tomcat.embed:tomcat-embed-core:8.5.15	High	3	LOW	16
bctsp-jdk14-1.38.jar	cpe:/a:openpgp:openpgp:1.38 cpe:/a:pgp:openpgp:1.38	org.bouncycastle:bctsp-jdk14:1.38	Medium	2	LOW	17
groovy-2.4.11.jar	cpe:/a:apache:groovy:2.4.11	org.codehaus.groovy:groovy:2.4.11	Medium	1	LOW	23
spring-boot-starter-data-jpa-1.5.4.RELEASE.jar	cpe:/a:pivotal_software:spring_data_jpa:1.5.4	org.springframework.boot:spring-boot-starter-data-jpa:1.5.4.RELEASE	Medium	1	LOW	21

- Gorenavedene korekcije nisu implementirane u krajnjem rešenju, kako bi se izbeglo narušavenje postojećih funkcionalnosti.

A10-Unprotected APIs

Potrebno je obezbediti:

- Sigurnu komunikaciju između klijenta i aplikacije.
- Jaku šemu za autentifikaciju i da svi kredencijali, ključevi i tokeni budu obezbeđeni.

- Zaštitu parsera protiv napada koji zloupotrebljavaju formate podataka.
- Zaštitu svih formi od injection napada.

Konfigurisana je sigurna komunikacija između klijenta i aplikacije. Sistem za autentifikaciju se zasniva na korisničkom imenu i lozinki, pri čemu se lozinka hešira zajedno sa salt-om i tako čuva u bazi. Ključevi se čuvaju u posebnim fajlovima (keystore-ovima). Validacija unosa iz formi je implementirana sa ciljem da korisniku olakša interakciju sa sistemom. Sam sistem se ne oslanja na validaciju koja je implementirana na frontend-u, već se vrše dodante validacije pre bilo kakve obrade.

Reference :

https://www.owasp.org/index.php/Top_10_2017-A10-Underprotected_APIs