Zadanie 5 – WEBOVÁ BEZPEČNOSŤ

Vypracovali: Samuel Michalčík, Marek Štrba

1. Inštalácia a spustenie

Na spustenie Juice Shop aplikácie sme inštalovali Docker, v ktorom sme spustili následné príkazy:

docker pull bkimminich/juice-shop docker run–rm-p 127.0.0.1:3000:3000 bkimminich/juice-shop

2. Injection úlohy

- Login admin **
    - Úlohou bolo prejsť cez login formulár bez správneho emailu či hesla
    - Použili sme SQL injection, kde sme namiesto emailu vložili vstup ' OR TRUE-- vďaka čomu vyzeral SQL príkaz nasledovne:

```
SELECT * FROM users WHERE email = '' OR TRUE--' AND password = 'input_password';
```

    - Takýto SELECT je vyhodnotený vždy ako TRUE a umožní prihlásenie.

- Login Bender ***
    - Úlohou bolo prejsť cez login formulár a prihlásiť sa do konkrétneho účtu používateľa Bender
    - Email používateľa sme našli v časti „About Us“
    - Vstup do poľa email sme použili ako bender@juice-sh.op'-- vďaka čomu SQL príkaz vyzeral následovne:

```
SELECT * FROM Users
WHERE email = 'bender@juice-sh.op'--
-- AND password = 'input_password'
-- AND deletedAt IS NULL;
```

3. XSS úlohy

- DOM XSS *
    - Úlohou bolo vykonať XSS útok cez search bar

- Pri použití search baru sme zistili, že sa hľadaný výraz zobrazí (outputne) na stránke, takže sme ako string do vyhľadávacieho pola použili <iframe src="javascript:alert(`xss`)"> čím sme zobrazili alert okno

- Bonus Payload *
  - Úloha naväzovala na predošlú, v podstate išlo o rovnaký spôsob útoku, ale na vstupe sme použili <iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe> čím sme do stránky vložili html element iframe, ktorý automaticky prehráva audio

4. Broken Access Control úlohy

- Web3 Sandbox *
  - Úlohou bolo dostať sa na omylom deploynutú podstránku obsahujúcu code-sandbox. Pomocou F12 sme si v prehliadači zobrazili zobrazili main.js súbor, v ktorom sme pomocou Ctrl+F vyhľadali slovo sandbox a našli cestu k danej podstránke ako „localhost/#/web3-sandbox/.
- Admin Section **
  - Úlohou bolo dostať sa na zabezpečenú stránku administrácie. Ako prvé sme sa prihlásili ako admin (na základe prvej úlohy). Následne sme preskúmali web a zistili aké webové technológie využíva. Zistili sme, že frontend webovej aplikácie je vytvorený pomocou Angularu, ktorý má administratívnu sekciu pod url „/administration“, čím sa sa úspešne dostali na túto podstránku.

5. Sensitive Data Exposure

- Confidential document *
  - Úlohou bolo nájsť dokument s citlivými údajmi, ktorý je možné sprístupniť napríklad cez url pri nevhodnom zabezpečení.
  - Dokument sme našli cez http://localhost:3000/ftp/acquisitions.md

```
# Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.

Our shareholders will be excited. It's true. No fake news.
```

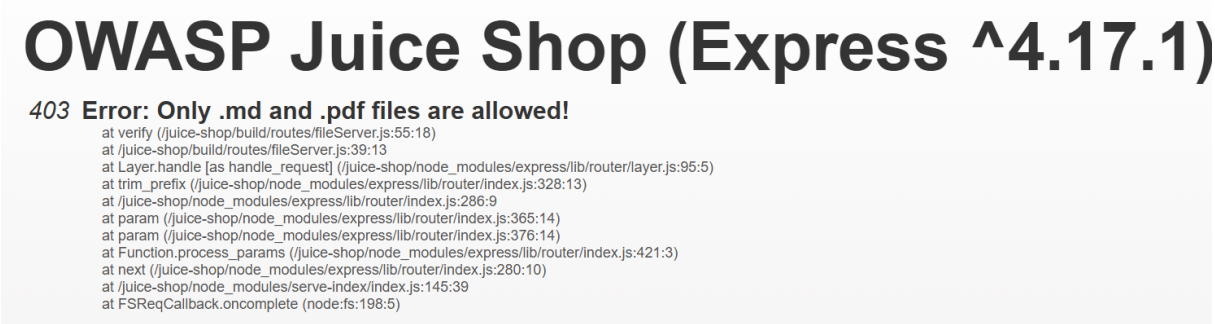
- Exposed Metrics *
    - Úlohou bolo nájsť endpoint, ktorý poskytuje usage dáta pre monitorovací systém prometheus
    - Po chvíli skúšaniach rôznych endpointov sme natrafili na ten správny - http://localhost:3000/metrics
    - krátka ukážka z dlhého výpisu:

```
# HELP file_uploads_count Total number of successful file uploads grouped by file type.
# TYPE file_uploads_count counter

# HELP file_upload_errors Total number of failed file uploads grouped by file type.
# TYPE file_upload_errors counter

# HELP juiceshop_startup_duration_seconds Duration juiceshop required to perform a certain task during startup
# TYPE juiceshop_startup_duration_seconds gauge
juiceshop_startup_duration_seconds{task="validateConfig",app="juiceshop"} 0.024596911
juiceshop_startup_duration_seconds{task="cleanupFtpFolder",app="juiceshop"} 0.041710053
juiceshop_startup_duration_seconds{task="validatePreconditions",app="juiceshop"} 0.055829171
juiceshop_startup_duration_seconds{task="datacreator",app="juiceshop"} 7.129136961
juiceshop_startup_duration_seconds{task="customizeApplication",app="juiceshop"} 0.006967843
juiceshop_startup_duration_seconds{task="customizeEasterEgg",app="juiceshop"} 0.00352212
juiceshop_startup_duration_seconds{task="ready",app="juiceshop"} 7.809
```

- Access Log ****
    - Úlohou bolo získať prístup k súboru s logmi, ktorý je po správnosti neprístupný, avšak v aplikácii juice shop je na simulovanie zranitelnosti zámerne prístupný.
    - Na linuxe sme sputili príkaz **ffuf -u http://127.0.0.1:3000/FUZZ -w common.txt**
    - ffuf je webový fuzzer určený na odhaľovanie skrytých endpointov, súborov a adresárov, pričom odosiela viacero požiadaviek HTTP s cieľom nájsť zraniteľnosti alebo skryté zdroje, vstupy ťahá z common.txt súboru, ktorý obsahuje typické názvy rôznych webových súborov a adresárov.

- Zistili sme, že vieme pristúpiť k /ftp, ale to už nám je známe z predošlej úlohy, preto sme skúšali príkaz **ffuf -u http://127.0.0.1:3000/support/FUZZ -fs 1925 -w common.txt** ktorý prehľadáva o jednu úroveň nižšie na url /support, ktorú sme zvolili na základe nápovedy z tutoriálu, kde bol tip na zistenie url podľa už predtým nájdených skrytých súborov.
- Príkaz našiel adresár s názvom logs, ktorý už následne vieme prístupniť cez url http://localhost:3000/support/logs, čím sme sa dostali k hľadanému skrytému súboru s logmi:

- Forgotten developer backup ****
  - Úlohou bolo získať prístup k zabudnutému súboru so zálohou, ktorý má častokrát príponu .bak
  - Prehľadali sme už nám známu skrytú časť /ftp, kde sme našli súbor http://localhost:3000/ftp/package.json.bak, avšak nie je možné pristúpiť/stiahnúť daný súbor, kvôli nevhodnej prípone
  - Webová aplikácia dovolí pristúpiť iba k súboru s príponou .md a .pdf

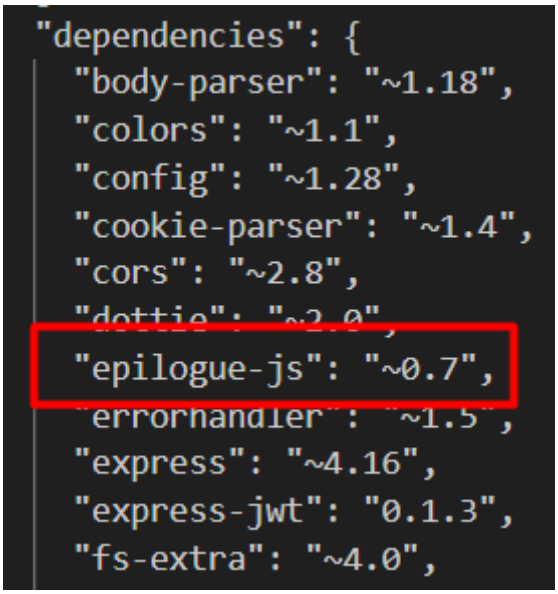


  - Pokúsili sme sa injectnúť null byte následovne: http://localhost:3000/ftp/package.json.bak%00.md, čím by sme boli schopný obísť overenie prípony, keďže aplikácia by videla iba .md súbor, ktorý je valídny
  - Dostali sme bad request error, tak sme zmenili kódovanie %00 na url kódovanie a do url dosadili: http://localhost:3000/ftp/package.json.bak%25%30%30.md
  - súbor sme úspešne stiahli

6. Vulnerable Components

- Legacy Typosquatting ****
  - Úlohou bolo nájsť typosquatting zraniteľnosť, teda škodlivý package, ktorý by vývojár náhodne nainštaloval s vedomím, že sa jedná o známu dependency. Trik spočíva vo veľmi podobnom alebo významovo rovnakom

názve dependency. V už preskúmanom zabudnutom backup súbore sme našli škodlivú dependency epilogue-js, ktorá by mala mať po správnosti názov epilogue.



- Vulnerable library ****
    - Úloha bola podobná ako predošlá, dôležité bolo nájsť aplikáciou používanú knižnicu, ktorá obsahuje zraniteľnosti. Z jednej XSS úlohy vieme o probléme, ktorý sa týka html sanitize, preto sme preskúmali knižnicu "sanitize-html": "1.4.2", ktorá ako sa ukázalo, obsahuje zraniteľnosti a bola správnou odpoveďou.

7. Broken Authentification

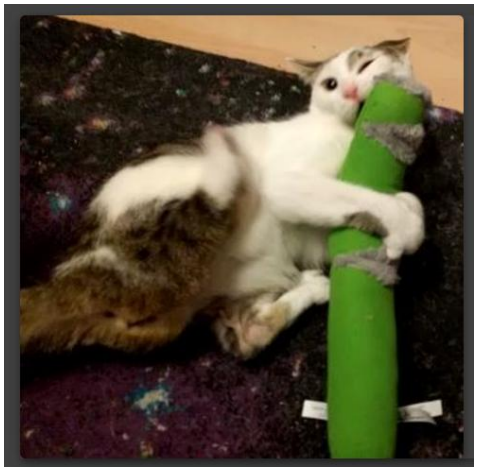
- Password Strength **
    - Úlohou bolo prihlásiť sa pod účtom administrátora bez SQL injection. V časti „about us“ sme zistili adminov email a následovlai interaktívny tutoriál, ktorý napovedal použiť primitívne heslo.
    - Po krátkom skúšaní sme zistili prihlasovacie údaje:
        - admin@juice-sh.op
        - Admin123
- Bjoern's Favorite Pet ***
    - Úlohou bolo resetovať heslo používateľa Bjoern cez prihlasovací formulár – „forgot password“
    - Na internete sme našli video, na ktorom sú odhalené informácie ako email a odpoveď na security otázku, čím sme sa dostali do účtu
        - bjoern@owasp.org
        - Zaya (pet’s name requested in security question)

8. Improper Input Validation

- Missing encoding *
    - Úlohou bolo zobraziť fotku mačky používateľa Bjoern v časi „Photo wall“
    - Image src url obsahovala #, ktorý slúži ako anchor a vo všeobecnosti rozdeľuje url
    - Bolo potrbné prepísať # tak, aby vyhovoval url encodingu
    - # zapíšeme ako %23

```
<img _ngcontent-wyg-c94 class="image" src="assets/public/images/uploads/😼-#zatschi-#whoneedsfourlegs-1572600969477.jpg" a
```

```
<img _ngcontent-wyg-c94 class="image" src="assets/public/images/uploads/😼-%23zatschi-%23whoneedsfourlegs-1572600969477.jpg"
```



- Zero stars *
    - Úlohou bolo odoslať recenziu s 0 hviezdičkami (by default je rozsah 1-5)
    - Zistili sme, že formulár nejde odoslať, kým nie je vyplnený rating (počet hviezdičiek)
    - Formulár nejde odoslať kvôli tlačidlu, ktoré je nastavené na disabled, toto vieme ale cez F12 obísť, tlačidlu status disabled vymazať a formulár odoslať.
    - Cez network si vieme pozrieť http request/response, kde vidíme rating nastavený na 0

```
▼ {captchaId: 23, captcha: "1", comment: "test (anonymous)", rating: 0}
    captcha: "1"
    captchaId: 23
    comment: "test (anonymous)"
    rating: 0
```