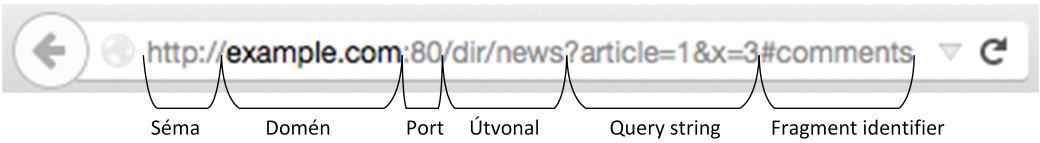
Kódolás és IT biztonság
(VIHIBB01)
Laboratóriumi gyakorlat

Webalkalmazások biztonsági tesztelése

Szerző:
Futóné Papp Dorottya

2020. október 2.

Tartalomjegyzék



1. ábra. Universal Resource Locator

1. Oktatási célok

Ezen a gyakorlaton egy sérülékeny webalkalmazás kezdeti biztonsági tesztelését fogja elvégezni. A laboratóriumi gyakorlatnak kettős célja van. Az első cél, hogy demonstrálja a manapság elérhető, web alkalmazásokra specializált biztonsági elemző szoftverek képességeit. A gyakorlat során az OWASP Zed Attack Proxy (ZAP) szoftverrel fog megismerkedni, beleértve az eszköz használatát és a grafikus felhasználói felületének főbb képességeit.

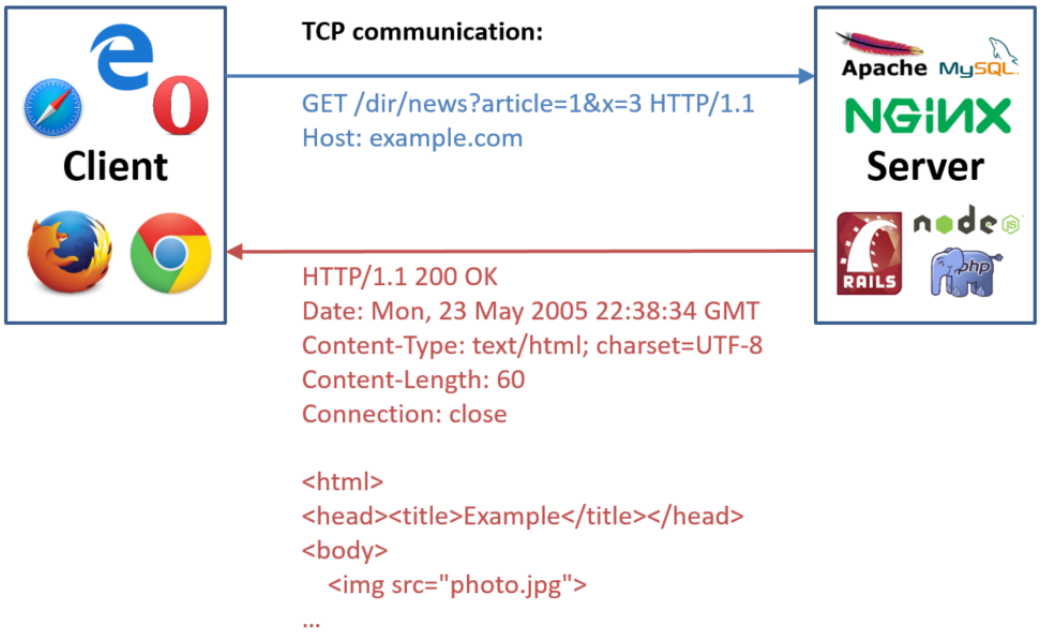
A második cél az, hogy megtegye az első lépéseket a weboldalak biztonsági tesztelése terén. A ZAP grafikus felületét felhasználva fog támadni egy sérülékeny webalkalmazást, az OWASP Juice Shopot. Ennek a webalkalmazásnak a támadása nem von maga után jogi következményeket, mivel kifejezetten azért fejlesztették, hogy webes biztonsági tréningeken támadják demonstrációj jelleggel.

2. Háttéranyag

2.1. HTTP áttekintő

Egy weboldal betöltése a böngészőben több lépésből áll. Az első lépés a Universal Resource Locator (URL) által azonosított erőforrás letöltése. Az URL-nek több része van, ahogy azt a 1. ábra mutatja. Egyes részekhez a böngészők alapértelmezett értékeket rendelnek: ha nincs külön kiírva, akkor a séma alapértelmezett értéke `http://`, a porté `:80` és az útvonalé `/`. A query sztring és a fragment identifier komponensek opcionálisak.

A böngésző ezután elküldi az útvonalat és a query stringet egy kérésben annak a webszernek, amelyiket a domén azonosít. Ez a lépés általában megköveteli a böngészőt futtató számítógépet, hogy egy vagy több DNS kéréssel megállapítsa a doménhez tartozó IP címet. Az elküldött kérés a HyperText Transfer Protocol (HTTP) nevű protokollt követi, ami a kliens-szerver architektúrára épülő kétirányú kommunikációs protokoll. Egy példa

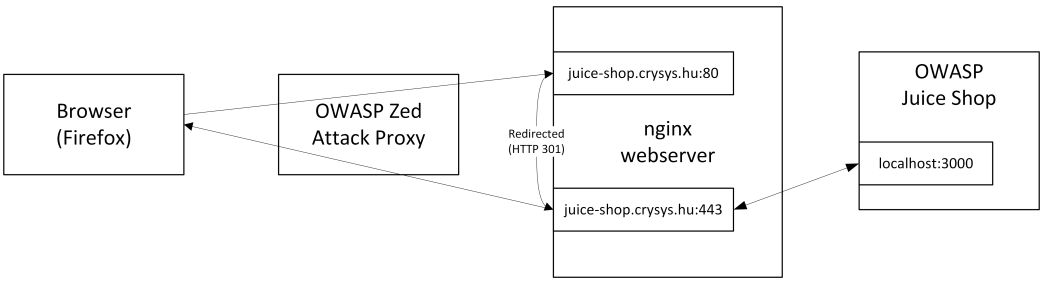


2. ábra. Example HTTP Request-Response Pair

kérés-válasz interakciót mutat be a 2. ábra. A kérésben szerepel egy HTTP `verb` (pl. `GET`, `POST` or `PUT`), a lekérdezett útvonal és a query string, valamint a kérés során használt protokoll verzió. A kérés ezen felül több fejlécet is tartalmaz, amivel többlet információt szolgáltat a szervernek. Néhány széleskörben használt fejléc mező:

- `Accept`: Válaszként elfogadott média típusok (pl. `text/html`, `application/xml`)
- `Host`: A szerver doménje és opcionálisan portja
- `User-Agent`: a kérést küldő klienst azonosító karaktersorozat (pl. `Mozilla/5.0 (X11; Linux x86_64; rv:12.0)`)
- `Referer`: A legutóbb meglátogatott weboldal címe, ami a böngészőt a most lekért oldalra irányította

A szerver válaszában szerepel a protokoll használt verziója, a HTTP *státusz kód*, további fejléc mezők és, amennyiben a lekérés sikeres, a kért erőforrás. A HTTP státusz kódokat öt csoportra osztjuk:



3. ábra. A gyakorlat során használt mérési elrendezés

- Információs (`1xx`), e.g. `100 Continue` and `101 Switching Protocols`
- Sikeres (`2xx`), e.g. `200 OK`, `201 Created`
- Átirányítás (`3xx`), e.g. `301 Moved Permanently`, `307 Temporary Redirect`
- Kliens hiba (`4xx`), e.g. `400 Bad Request`, `403 Forbidden`, `404 Not Found`
- Szerver hiba (`5xx`), e.g. `500 Internal Server Error`, `502 Bad Gateway`, `503 Service Unavailable`

A letöltött erőforrástól függ a böngészőbeli feldolgozás. Szkriptek esetén a böngésző futtatja a leírt parancsokat. Weboldal (HyperText Markup Language (HTML) formátumú adat) esetén a böngésző memóriába olvassa az adatot, ha szükséges, lekéri az alerőforrásokat (képek, szkriptek, frame-ek, stb.), majd ha minden erőforrás rendelkezésre áll, megjeleníti a tartalmaz a felhasználónak. Végezetük a böngésző egy esemény ciklust futtat, amiben egyes eseményekről (pl. űrlapok beküldése) értesíti a Javascript motort, amely lekezeli az eseményeket.

2.2. Mérési elrendezés

A 3. ábra mutatja a laboratóriumi gyakorlat során használt mérési elrendezést. A virtuális gépben mindegyik komponenst előre telepítettük, azonban az nginx webszerver konfigurációja egy előző labor gyakorlat témája. Amennyiben azt a labort nem tudta teljesíteni, a sérülékeny webalkalmazást a `localhost:3000`-en keresztül éri el.

Az OWASP Zed Attack Proxy[1] (ZAP) egy nyílt forráskódú webes biztonsági tesztelő szofter. Képes HTTP forgalmat elkapni és lehetővé teszi a felhasználó számára, hogy tetszőlegesen módosítsa a fejlécek és a body tartalmát. Többféle automatizált tesztelési technikát is implementál, pl. fuzzing, de ezek a labor gyakorlatnak nem képezik részét. A működéshez a szoftver előkonfigurálja a böngészőt, hogy a HTTP és HTTPS kéréseket is a ZAP-on keresztül küldje a böngésző az elemzett webalkalmazásnak (ezt nevezzük proxy-zásnak).

Az nginx reverse proxy szerverként üzemel a virtuális gépben futó webalkalmazások számára. Az ilyen elrendezések előnye, hogy a webalkalmazások bármilyen motort használhatnak a bejövő kérések lekezelésére, pl. Apache, node.js, stb., a forgalmukat pedig a proxy webszerver menedzseli. A proxy feladata végpontként megjelenni HTTPS kapcsolatok számára, majd lokálisan továbbítani a kéréseket a megfelelő webalkalmazás számára. Ennek az elrendezésnek előnye, hogy az üzemeltetőknek nem kell külön infrastruktúrát fenntartani minden üzemeltetett webalkalmazás számára, ehelyett egy központi infrastruktúrával nyújhatnak biztonságos kommunikációs csatornát minden webalkalmazásnak. Az nginx-hez tartozik a virtuális gépben egy szolgáltatás, aminek a neve `nginx`. Az nginx beállítása reverse proxy szerverként egy másik laboratóriumi gyakorlat feladata. Amennyiben nem teljesítette azt a feladatot, közvetlenül kell kommunikálnia a sérülékeny webalkalmazással.

A laboratóriumi gyakorlat során a OWASP Juice Shop[2] webalkalmazást használjuk. Ez egy nyílt forráskódú webalkalmazás, amit kifejezetten azért fejlesztenek, hogy sérülékenységeket tartalmazzon. A webalkalmazást oktatási célokra lehet használni: webes biztonsági tréningeken lehet vele tipikus webes sérülékenységeket demonstrálni kontrollált környezetben. A virtuális gépben a Juice Shopot közvetlenül a `localhost:3000`-es címen találjuk és tartozik hozzá egy indító szolgáltatás, melynek neve `juice-shop`. A webalkalmazás angularjs-t használ, aminek eredményeként a weboldalakat egy `#/` kezdetű útvonallal érjük el, pl. `localhost:3000/#/search`. A webalkalmazásban 88 feladat található 6 nehézségi szintre osztva, némelyikhez tartoznak tippek a megoldás elkezdéséhez. A laboratóriumi gyakorlat során ebből a 88 feladatból kell néhányat megoldani.

A következő bash parancsokat érdemes használnia a virtuális gép működtetéséhez és hibakezeléséhez:

[1] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
[2] https://www.owasp.org/index.php/OWASP_Juice_Shop_Project

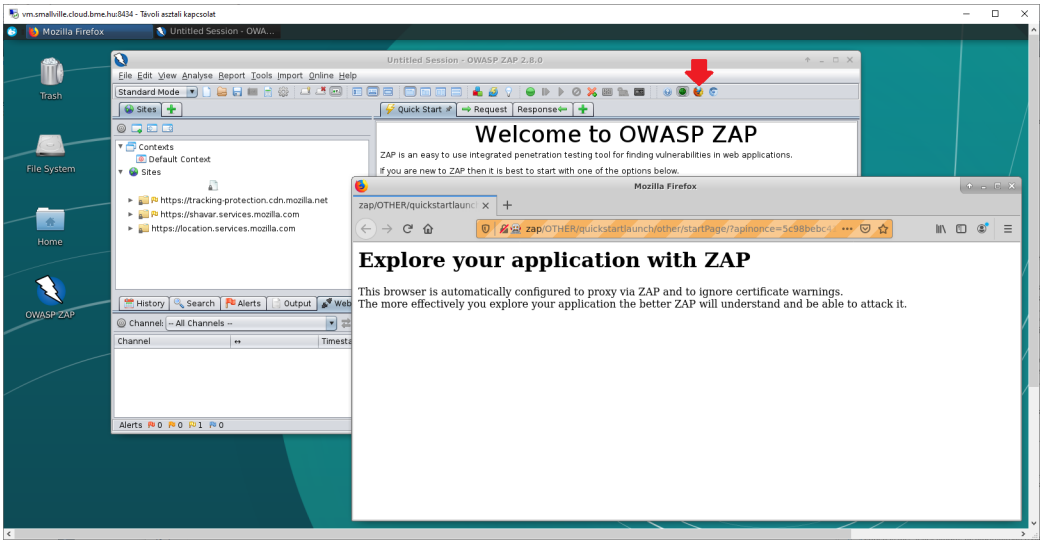
```
# szolgáltatás indítása
sudo service <service name> start

# szolgáltatás újraindítása (pl. konfiguráció változás miatt)
sudo service <service name> restart

# szolgáltatás állapotának lekérdezése
sudo service <service name> status

# szolgáltatás leállítása
sudo service <service name> stop

# naplóbejegyzések olvasása
sudo journalctl -xe

# nyitott portok és hozzájuk tartozó folyamatok listázása
sudo netstat -tulpn

# keresés egy parancs kimenetében
<command> | grep <search condition>

# futó folyamatok listázása
ps aux
top

# szövegszerkesztő megnyitása parancssorból
mousepad <file name>
nano <file name>
```

3. Feladatok

3.1. Vezetett rész

A laboratóriumi gyakorlat során az OWASP Zed Attack Proxy (ZAP) tesztelő eszközt használjuk. A szoftver megismeréséhez teljesítenie kell a beépített bevezetőt.

1. Parancssorból indítsa el az `nginx` és `juice-shop` szolgáltatásokat:



4. ábra. ZAP-paroxy-val konfigurált Firefox indítása

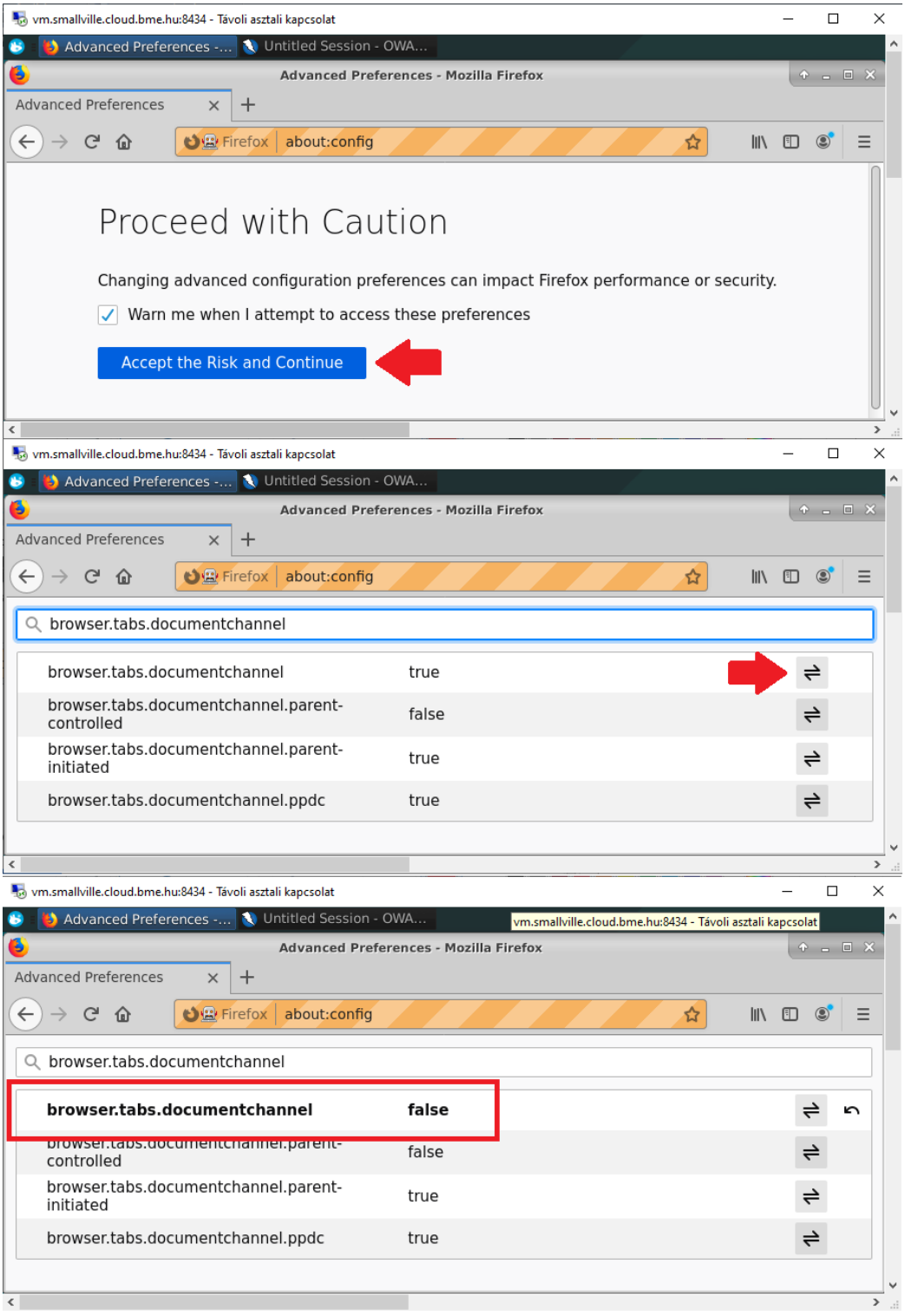
```
sudo service nginx start
sudo service juice-shop start
```

2. Indítsa el a ZAP-ot az asztalon található parancsikon segítségével! (Ez eltarthat egy ideig a háttérben futó konfigurációs lépések miatt, 5 percet mindenképp érdemes várni.)

3. A ZAP felületén teljes képernyős módban a menüsor alatt találhatóak apró ikonok. Az ikonsorozat jobb szélén két olyan ikon van, amelyik előre konfigurált böngészőt nyit meg ZAP-pal történő proxy-záshoz. Indítsa el a Firefox böngészőt az ikonon keresztül! Onnan tudja, hogy ZAP-on keresztül proxy-zik böngészés közben, hogy az URL beviteli mező háttere narancssárga, ahogy azt a 4. ábra is mutatja.

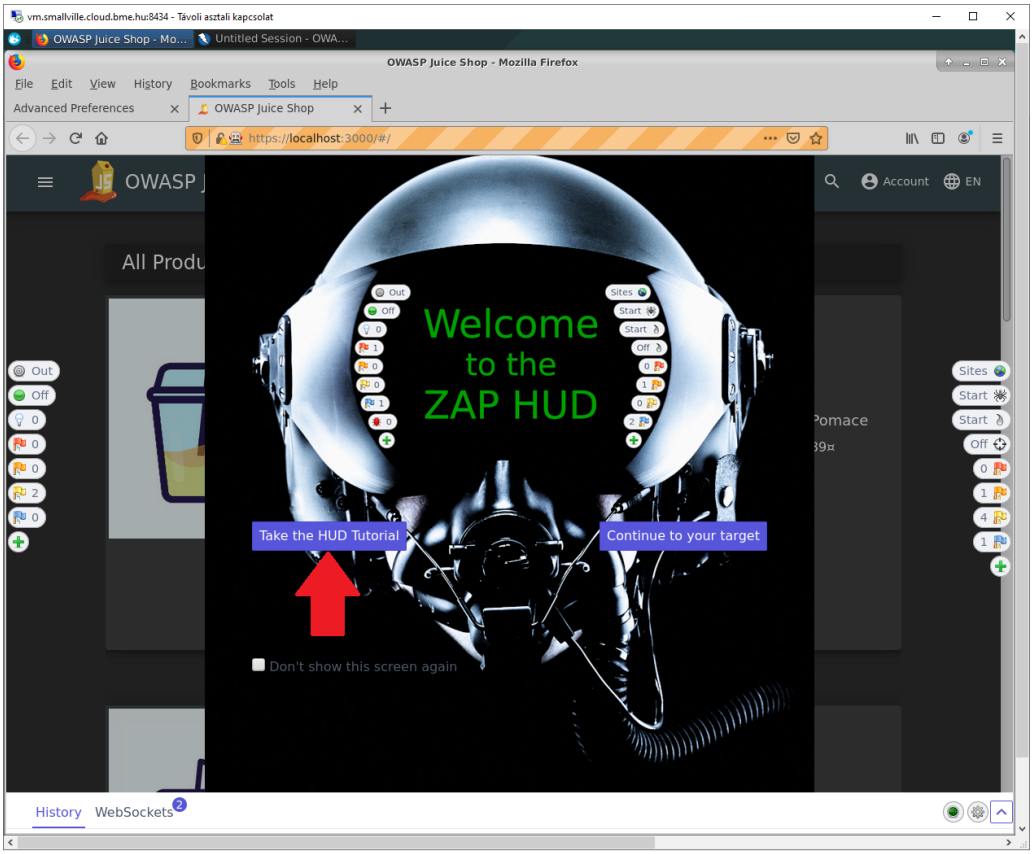
4. A legújabb Firefox böngészőben szükséges egy konfigurációs lépés a ZAP zavartalan működéséhez. Ehhez egy új tabon látogassa meg az `about:config` oldalt, keressen rá a `browser.tabs.documentchannel` preferenciára és állítsa értékét `false`-ra! A konfiguráció lépéseit 5. ábra szemlélteti.

5. Látogassa meg a `juice-shop.crysys.hu`-t és várjon, amíg megjelenik



8

5. ábra. Firefox konfigurálása



6. ábra. ZAP HUD

a ZAP HUD.[3] A HUD-ot a 6. ábra mutatja. A HUD-on keresztül indítsa el és teljesítse a bevezetőt!

Moodle-ben adja meg, hogy melyik HTTP headert kellett módosítani a bevezető "Resend" fejezetében!

3.2. Önálló rész

Az önálló feladatokat többféleképpen is meg lehet oldani, az alábbiakban csak egy lehetőséges megoldási útvonalat részletezünk. A feladatokat az OWASP Juice Shop fejlesztői készítették, így a sérülékeny webalkalmazásban olyan ellenőrzések találhatóak, amelyek figyelik az érkező kéréseket és jeleznek a böngészőben, ha valamelyik feladatot megoldja.

3.2.1. Bizalmas fájlok elérése

Az első feladatban érzékeny adatokat tartalmazó fájlt kell találnia a webalkalmazásban. Automatizált megoldás:

1. Nyissa a ZAP által előre konfigurált Firefox böngészőt, látogassa meg az `about:config` oldalt, keressen rá a `browser.tabs.documentchannel` preferenciára és állítsa értékét `false`-ra!
2. Látogassa meg a `juice-shop.crysys.hu`[4]-t!
3. Vegye fel a scope-ba a Juice Shop oldalát a HUD-on keresztül!
4. Indítson egy `Spider`t az alkalmazás oldalainak felderítésére! Figyelem: csak feltérképezni szeretnénk az alkalmazás oldalait, nem aktívan támadni azokat! Ne használja az `Active Scan` opciót!
5. A Spider automatikus megtalálja a `/ftp` mappát, amiben több bizalmas fájl is található. A mappa tartalmát a `Sites` HUD elemen keresztül tudja megnézni.

Moodle-ben válassza ki, hogy a felsoroltak közül melyik fájl NEM található meg az `ftp` mappában!

[3] Amennyiben az nginx nincs reverse proxy-ként beállítva, az OWASP Juice Shopot a `localhost:3000`-en keresztül éri el.

[4] Amennyiben az nginx nincs reverse proxy-ként beállítva, az OWASP Juice Shopot a `localhost:3000`-en keresztül éri el.

3.2.2. A nulla csillagos visszajelzés

Az előző feladatok során kiderült, hogy sok probléma van az OWASP Juice Shoppal. Erről természetesen az üzemeltetőknek is tudniuk kell, ezért visszajelzést kell küldenie a webalkalmazáson keresztül. Hogy az üzemeltetők számára is világos legyen a helyzet súlyossága, olyan visszajelzést küldjön, ami egyetlen csillagot sem ad a weboldalnak!

- Nyissa a ZAP által előre konfigurált Firefox böngészőt, látogassa meg az `about:config` oldalt, keressen rá a `browser.tabs.documentchannel` preferenciára és állítsa értékét `false`-ra!
- Látogassa meg a `juice-shop.crysys.hu`[5]-t!
- Az oldalsó menőpontok közül válassza a `Customer Feedback` menőpontot és küldjön egy érvényes visszajelzést a böngészőn keresztül!
- Figyelje a `History` fület és nézze meg, milyen URL-re küldi a böngésző a jól formázott visszajelzést!
- Nyissa meg ezt az üzenetet, szerkessze a kérés (`Request`) body részét: írja át az értékelésre vontakozó kulcs-értékpár értékét `0`-ra, majd játssza vissza a kérést a parancssoron (`Replay in Console`)!

Moodle-ben adja meg azt az URL-t, amire a visszajelzést küldeni kell!

3.2.3. Eredményjelző

Ebben a feladatban meg kell találni az OWASP Juice Shopban elrejtett eredményjelző oldalt. Ez az oldal rejtett, vagyis csak akkor látogatható meg, ha ismerjük a rá mutató URL-t és azt közvetlenül látogatjuk meg. Az eredményjelzőn több hackelős feladat is fel van sorolva, a későbbi laborokon ezek közül fogunk néhányat megcsinálni.

1. Nyissa a ZAP által előre konfigurált Firefox böngészőt, látogassa meg az `about:config` oldalt, keressen rá a `browser.tabs.documentchannel` preferenciára és állítsa értékét `false`-ra!

[5] Amennyiben az nginx nincs reverse proxy-ként beállítva, az OWASP Juice Shopot a `localhost:3000`-en keresztül éri el.

2. Látogassa meg a `juice-shop.crysys.hu`[6]-t!

3. Miközben töltődik az oldal, figyelje a `History` fület és a `main-es2015.js` betöltődését

4. Keresse ki a Javascript fájlt a `Sites` HUD elemen keresztül!

5. Keresse meg a HTTP válaszban a rejtett eredményjelzőre mutató URL-t (Segítség: az eredményjelző szót angolra scoreboardként fordítjuk)!

Moodle-ben adja meg az URL-t, amellyel meglátogatható az eredményjelző! (Segítség: mivel a webldal angularjs-t használ, az URL-ek `<domain név>/#/<oldal name>` formában érhetőek el)

[6] Amennyiben az nginx nincs reverse proxy-ként beállítva, az OWASP Juice Shopot a `localhost:3000`-en keresztül éri el.