Megoldás

Feladat 9.

Biztonsági tesztelés

A biztonsági tesztelés feladatot is igyekeztünk úgy összeállítani, hogy az is képes legyen jó eredményt elérni, aki ezzel a témával még nem foglalkozott. Ugyanakkor a könnyebb problémák mellett elhelyeztünk néhány nagyobb tapasztalatot igénylő hibát is. A visszajelzések alapján sok versenyző az első biztonsági tesztelését hajtotta végre. Reméljük, hogy mindenkinek a fejlődését szolgálta egy ilyen példa elkészítése.

A pontozásnál több szempontot vettünk figyelembe, különböző súlyozással. Nagyon sok funkcionális hibát is kaptunk, ezekere a feladatkiírásnak megfelelően nem tudtunk pontot adni. Sokan hibának jelezték, hogy a kapcsolat nem biztonságos http (https), illetve hogy a szerverről túl sok információ érhető el, mely lehetőséget ad például verziófüggő biztonsági rések kihasználására. Ezek valóban hibák és biztonsági tesztelés során ezekre is figyelni kell, de ezekre sem tudtunk pontot adni, mert a feladatban leírtuk, hogy ezek meglétét feltételezzük.

Pontozott hibák (zárójelben a hibára kapható pontszám):

1. Belépéskor hibás jelszóra és hibás felhasználónévre más üzenetet ír ki, vagyis ki lehet deríteni, hogy van-e olyan felhasználónév (6 pont)
2. SQL injection lehetősége (pl. jelszó input mező belépésnél). Bármely érvényes azonosítóval és pl. a következő jelszóval be lehetett lépni: „jelszo' OR 1='1” (6 pont)
3. A jelszó mezőben a karakterek nem rejtettek belépésnél (6 pont)
4. Autocomplete=on (pl. belépésnél a jelszó mezőre is), így túl nagy szabadságot adunk a böngészőknek, hogy kliens oldalon megjegyezzék a jelszót (6 pont)
5. Felhasználónév és jelszó esetén sem tesz különbséget a kis- és nagybetűk között belépésnél (6 pont)
6. Nincs korlátozás a sikertelen bejelentkezés próbálkozások számára (brute-force támadásra ad lehetőséget) (6 pont)
7. Elfelejtett jelszó funkciónál bárkinek a jelszavát felül lehet írni, és a kiküldött jelszót nem kell megváltoztatni és le sem jár (6 pont)

8. PHP hibakódot ír ki fájlnévvel és minden infoval az Elfelejtett jelszó oldalon a program által hibásnak tekintett e-mail esetén (6 pont)
9. Regisztráció űrlapon ha van már olyan felhasználó, mint amilyet regisztrálni szeretnénk, akkor kiírja a nevét (6 pont)
10. Captcha kód nincs ellenőrizve a regisztrációkor (így robotok is könnyedén tudnak regisztrálni, brute-force támadást könnyebb megvalósítani) (6 pont)
11. Cross site scripting (XSS) lehetősége több mezőnél (pl. regisztrációkor keresztnév mezőbe a következő script helyezhető el: <script>alert('hello');</script> , belépéskor pedig megjelenik a Javascript popup) (6 pont)
12. Regisztrációkor a jelszó mezőben a speciális karaktereket duplán számolja, illetve a szóközt is számolja és "gyenge" jelszavakat is meg lehet adni (pl. az azonosítót) (6 pont)
13. Jelszóváltoztatáshoz nem kéri be a régi jelszót (6 pont)
14. Sikertelen jelszóváltoztatásnál kiírja a végrehajtott SQL-t (6 pont)
15. Az email alapján a régi rendszerből át lehet tölteni a jelszavakat az új rendszerbe, ez a régi rendszerrel kapcsolatos komoly hiányosság, hiszen ha egy rendszerben meg lehet nézni bárhogy a jelszót, az visszaélésekre ad lehetőséget (külön kértük a feladatkiírásban, hogy a régi rendszerrel kapcsolatos biztonsági észrevételek is fontosak) (5 pont)
16. Session problémák (pl. session cookie kliens oldalról is módosítható, a session nincs lezárva, túl hosszú timeout) (6 pont)
17. A jelszó nincs titkosítva az adatbázisban (ehhez egy sikertelen jelszóváltoztatást kellett megvalósítani, ahol kiírta az SQL-t) (5 pont)