Mérési útmutató a „Hálózatok és webes rendszerek biztonsági ellenőrzése (PENT)” című méréshez

2021. feb,

A mérést kidolgozta:
Bencsáth Boldizsár
Ács-Kurucz Gábor

BME, CrySyS Adat- és Rendszerbiztonság Laboratórium

Tartalomjegyzék

1. Bevezetés **3**
1.1. Metodológia . . . 3
1.2. Webes sérülékenységek . . . 4
1.2.1. SQL Injection . . . 4
1.2.2. XSS . . . 4
1.3. Port scan . . . 4

2. Mérési eszközök **4**

3. Feladatok **5**
3.1. Szolgáltatások felderítése . . . 5
3.1.1. Nyitott portok megkeresése . . . 5
3.1.2. Szolgáltatások és verzióik beazonosítása . . . 5
3.2. Lehetséges támadási felület keresése . . . 5
3.3. Sérülékenység keresése a weblapon . . . 5
3.3.1. Sérülékenység kihasználása kézzel . . . 5
3.3.2. Sérülékenység kihasználása automata program segítségével . . . 6
3.4. Adminisztrátor felhasználó szerzése . . . 6
3.5. További problémák keresése a jelenlegi lehetőségekkel . . . 6
3.6. Lokális problémák felderítése . . . 6
3.7. Helyi root jogosultság szerzése . . . 6
3.8. Jelszavak kinyerése a shadow file-ból . . . 6

4. További információk **7**
4.1. NMAP . . . 7
4.2. SQLMAP . . . 7
4.3. John The Ripper . . . 8
4.3.1. Telepítés . . . 8
4.4. A GCC használata . . . 8
4.5. Sérülékenységek / Exploitok keresése . . . 9

A. Jegyzőkönyv **9**

1. Bevezetés

A következőkben rövid áttekintést nyújtunk a pentest módszertanáról és fogalmairól. Először szót ejtünk a mérés szempontjából is fontos lehetőségekről, eszközökről majd néhány fontos támadási felületet mutatunk be.

A mérés célja egy a való élethez közeli, de egyszerűsített *black-box* sérülékenységvizsgálat lépéseit követve a célszámítógépen rendszergazdai jogosultságok szerzése, valamint a jelszavak megismerése, így a gyakorlati tapasztalat bővítése.

Fontos, hogy egy pentest során nem elég egyetlen hibát megtalálni és azon keresztül bejutni a rendszerbe, hanem arra kell törekedni, hogy minden hibát megtaláljunk. Ha csak egyetlen sérülékenység marad a rendszerben amit nem vettünk észre és a támadó igen, akkor ő nyert. Ha nem is találunk meg minden hibát, nem szabad az első után megállni, hogy minimálisra csökkentsük a támadó lehetőségeit.

1.1. Metodológia

Alapvetően kétféle megközelítést szoktak alkalmazni pentest során:

- **black-box tesztelés**: Nem rendelkezünk semmilyen belső információval a hálózatról illetve a szolgáltatásokról. Ilyenkor egy külső támadó szemével gondolkozunk és így próbálunk minél több hasznosítható információt szerezni.
- **white-box tesztelés**: Forráskód és architektúra ismeretében történik a tesztelés. Ebben az esetben a tesztet végző személy a belső információk alapján hamarabb felderíthet olyan problémákat is, melyek black-box teszt során jóval nehezebben megtalálhatóak. A hátránya, hogy kevésbé modellezi jól a külső támadó szemszögét, valamint a túl sok információban könnyebben el lehet veszni és a lényeg fölött átsiklani.

Pentest illetve támadás során a lépések általában a következők:

- Információ gyűjtése a leendő áldozatról/cégről (nevek, elérhetőségek, kedvenc állatok, gyerekek neve, stb) - erre a mérés során nem térünk ki
- Támadási felület felderítése
- A lehetőségek felmérése
- Gyenge pontok keresése
- Gyenge pontok megtámadása
- Jogosultságok növelése
- Nyomok eltüntetése / Jelentés elkészítése

1.2. Webes sérülékenységek

Irányadónak az `https://www.owasp.org/index.php/Main_Page`OWASP TOP 10 sérülékenységét szokás vizsgálni, melyekről részletes információt az OWASP weboldalán lehet találni.

Néhány sérülékenységről bővebben is nyújtunk tájékoztatást.

1.2.1. SQL Injection

SQL Injection alatt azt a támadást értjük, amelynél módosítani tudjuk az adatabázis-lekérést a felhasználói inputokon keresztül. Ezekkel a módosításokkal kompromittálhatjuk a szolgáltatást.

A támadásról pontosabb leírás az OWASP (`https://www.owasp.org/index.php/SQL_Injection`) és a W3Schools oldalán (`http://www.w3schools.com/sql/sql_injection.asp`) is található, példákkal.

Az alábbi oldalon található egy kis összefoglaló a támadás miként jéről: `http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet`

Automatikus támadására a legismertebb program az `sqlmap` mely letölthető a `https://github.com/sqlmapproject/sqlmap` oldalról.

1.2.2. XSS

Az XSS (Cross Site Scripting) sérülékenységek a támadó javascript kódjának futtatását teszik lehetővé. Kétféle változata van:

- Permanens: Az oldalon megmarad a támadó js kódja és minden látogatónál külső beavatkozás nélkül le is fut.
- Temporális: Általában URL-be ágyazható javascript kód, ami csak akkor fut le, ha a felhasználó rákattint a külső forrásból származó javascriptet tartalmazó url-re.

További információk: `https://www.owasp.org/index.php/XSS`

1.3. Port scan

Port scan során egy adott számítógép nyitott portjait keressük. Megpróbálunk az összes porthoz kapcsolódni, és elkönyveljük a válaszokat. A legszélesebb körben használt port scan kivitelezésére alkalmas eszköz az `nmap` melyről bővebben írunk később.

2. Mérési eszközök

A mérés áldozata a egyedileg kiosztott ip című számítógép, a mérést végző felhasználó, vagy párosok sorszámos beosztása a moodle-be kerül elhelyezésre. A számozás alapján a vcenternben láthatja a mérő vagy mérőpáros, hogy melyik az ő gépe és melyik az ő szervere, a vcenter mutatja az

ip számot. Ha szükséges, a mérésvezetőt lehet kérni a gépek újraindítására. Más számítógép megtámadása nem engedélyezett. A virtuális gépek vezérlésére szolgáló felhasználói azonosító: itseclab@ib111.hit.bme.hu , a belépési felülete https://vcenter.ib111.hit.bme.hu . A VPN használati leírás: https://www.crysys.hu/vpn/ a vpn kliens név vpnstudent, a jelszavakat a mérés során közöljük.

Az ajánlott felhasználható programok:

- SQLMap: `https://github.com/sqlmapproject/sqlmap`
- John The Ripper: `http://www.openwall.com/john/`

3. Feladatok

3.1. Szolgáltatások felderítése

3.1.1. Nyitott portok megkeresése

Keresse meg az elérhető szolgáltatásokat nmap segítségével. Értelmezze a kimenetet.

3.1.2. Szolgáltatások és verzióik beazonosítása

Azonosítsa be a szolgáltatások pontos verzióját valamint az operációs rendszer típusát nmap segítségével. Értelmezze a kimenetet.

3.2. Lehetséges támadási felület keresése

A megtalált szolgáltatásokra ellenőrizze le verziószámuk alapján, hogy mennyire elavultak, milyen támadási lehetőségek vannak benne. Fontos, hogy a mérési környezet nagyjából 2018-ban került kialakításra, az azóta talált hibák kihasználhatóak, de nem voltak tervezettek 2018 környékén. Célszerű korábbi hibák keresése.

3.3. Sérülékenység keresése a weblapon

A 2 weblap közül az egyikben súlyos hiba található. Keresse meg és dokumentálja!

3.3.1. Sérülékenység kihasználása kézzel

A megtalált hibát használja ki kézzel! Dokumentálja az oldalon megjelenő információt! Kicsi értelmezés: Egy rejtett porton bővebb webes szolgáltatások elérhetőek, ami több részből áll. Ennek egy részében könnyen tesztelhető sérülékenység van, bizonyítsa, hogy itt valami gond van és hogy érdemes ezt mélyebben megvizsgálni!

3.3.2. Sérülékenység kihasználása automata program segítségével

A megtalált hibát sqlmap segítségével is exploitálja, nyerjen ki minél több hasznos információt az alkalmazásból!
Lehetőleg ne szemetelje össze az oldalt mindenfélével, mert valós helyzetben ez könnyedén lebukással járhat. Próbáljon minél kevesebb zajt okozni és a releváns információkat megtalálni.
Dokumentálja a megtalált információt és a megoldás menetét!

3.4. Adminisztrátor felhasználó szerzése

Törje föl a megtalált adminisztrátor felhasználó jelszavát! Dokumentálja a megoldás menetét és a feltört jelszót!

3.5. További problémák keresése a jelenlegi lehetőségekkel

A weboldal adminisztrátori felhasználója adta extra lehetőségeket kihasználva keressen lehetőséget shell szerzésére!
Gondoljon az előadáson / gyakorlaton elhangzottakra a keresés közben!
Dokumentálja a lépéseket!

3.6. Lokális problémák felderítése

Keresse meg a rendszer gyenge pontjait amelyeket jelenlegi jogosultsággal kihasználhatónak vél.

Dokumentálja a problémákat. Itt a sima felhasználó root felhasználóvá való feltörése a cél. Erre korlátos az eszköztár: rossz jelszó, sudo, rossz verziójú root joggal futó szoftver, de a legfontosabb a kernelhiba, a kernel verziójának ellenőrizze.

3.7. Helyi root jogosultság szerzése

Használja ki az előző pontban megtalált sérülékenységet és szerezzen root shellt. Használja ehhez az exploit adatbázisokat pl. exploit-db.com, keressen rá mi lehet jó. Két szavas jó tanácsunk: „overlay fs”

3.8. Jelszavak kinyerése a shadow file-ból

Töltse le majd fordítsa le a John The Ripper alkalmazást, majd segítségével szerezze meg a számítógépen található `user` felhasználó jelszavát is. Ezt egy egyszerű grep paranccsal is megteheti, ha már root joga van. A jelszó feltörését ne a célgépen végezze, mert lebukik, hanem az elemző kliensen, a kali linuxon. Ott a john már telepítve van, az /usr/sbin-ben találja, ha alkalmasabb, sudo-val kérjen root jogot, akkor a path-ban lesz, de persze ez egy biztonsági kockázat is.

4. További információk

A mérés során felhasznált programokról részletesebb angol nyelvű leírást általában a `man <program neve>` illetve `<program neve> -help` paranccsal kaphatunk a mérés kliensgépein.

4.1. NMAP

Az nmap (Network Mapper) egy nyílt forráskódú program melynek segítségével feltérképezhetjük a célszámítógép elérhető szolgáltatásait.
Megtalálhatjuk a nyitott portokat, valamint az sok esetben az ezen portok mögött futó szolgáltatás típusát, a kiszolgáló alkalmazás nevét, verziószámát is meg tudjuk állapítani segítségével.

Néhány fontosabb kapcsoló:

- **-sT**: TCP Scan, felépíti majd lebontja a TCP csatornát
- **-sS**: Syn Stealth Scan, csak TCP SYN csomagot küld, az ACK-ra nem válaszol
- **-Pn**: Nem végzi el a host ping segítségével történő tesztelését. Hasznos, ha az ICMP echo-reply (ping) le van tiltva a tűzfalban.
- **-sV**: Version detection, verziódetekciót hajt végre a megtalált szolgáltatásokra
- **-p**: Port kézi megadása
- **-O**: Operációs rendszer detekció

Példák:

- `nmap -v -A scanme.nmap.org`
- `nmap -sS -p22,80,443 192.168.0.0/16 10.0.0.0/8`
- `nmap -Pn -sV scanme.nmap.org`

4.2. SQLMAP

Az sqlmap egy nyílt forráskódú penetration teszteléshez fejlesztett eszköz, melynek célja az SQL injection hibák felderítése és kihasználása. A program letölthető az alábbi címről: `https://github.com/sqlmapproject/sqlmap`

Néhány fontosabb kapcsolója:

- **-u, –url**: Url megadása
- **-b, –banner**: DBMS banner kinyerése

- –**level**=**LEVEL**: Tesztek szintjének beállítása (1-5, default 1)
- –**risk**=**RISK**: Tesztek veszélyességének beállítása (1-3, default 1)
- –**dbs**: DBMS adatbázisok listázása
- –**tables**: DBMS táblák listázása
- –**columns**: DBMS táblák oszlopainak listázása
- –**schema**: DBMS schema kinyerése
- –**sql-shell**: interaktív SQL shell

4.3. John The Ripper

A John The Ripper egy jelszótörő alkalmazás ami brute-force alapú támadást hajt végre szótár illetve karakterhalmaz alapján.

4.3.1. Telepítés

A forrás letölthető a honlapról: `http://www.openwall.com/john/`

Itt a John the Ripper 1.8.0 (sources, tar.xz, 4.3 MB) választása szükséges.

Kitömörítés után az src mappában a következő parancs segítségével lehet fordítani:
`make clean linux-x86-64`
32 bites számítógép esetén: linux-x86-any használata javasolt.

Ezek után a `../run` mappában megtalálható a lefordult bináris és használatra kész. A mostani verzióban kapott KALI linux kliensen a john elő van telepítve és nem szükséges telepíteni!

4.4. A GCC használata

A GCC az egyik leggyakoribb unix alapú gépeken megtalálható C/C++ fordító program.

További információ a `man gcc` parancs használatával kérhető.

Néhány fontosabb kapcsolója:

- **-o**: A kimeneti (output) file neve
- **-O**: Optimalizáció (O0-tól O3-ig lehet állítani az optimalizálás mértékét, illetve egyéb beállításai is vannak. pl.: -Os méretre -Og debuggolásra optimalizál)
- **-g**: Debug szimbólumok meghagyása

- **-l**: Library hozzáadása linkelésnél

Példa:

- `gcc -o myprog -O2 a.c b.c -lfoo`

4.5. Sérülékenységek / Exploitok keresése

Egy program verziószámának ismerete esetén utána tudunk nézni, hogy az adott verzióhoz ismert-e a publikum számára sérülékenység illetve exploit.

Ismertebb nyilvántartások és exploit gyűjtemények:

- `https://cve.mitre.org/` - A hivatalos gyűjtemény
- `https://web.nvd.nist.gov/view/vuln/search`
- `http://www.securityfocus.com/`
- `https://www.exploit-db.com/`
- `https://google.com`

A. Jegyzőkönyv

A jegyzőkönyvet a mérés után egy héten fel kell tölteni a moodle rendszerre pdf formátumban. A jegyzőkönyvnek az alábbiakat kell tartalmaznia:

- Hallgató(k) neve és Neptun kódja
- Mérés neve
- Mérés időpontja
- Feladatok megoldása

A megoldások leírásánál törekedni kell a tömör, de érthető válaszra. A leírásból a megoldásnak reprodukálhatónak kell lennie (hosszabb kód mellékelhető a pdf-hez, nem feltétlenül kell belerakni)!