Feladatkiírás

a mérnökinformatikus mesterképzés
Hálózatbiztonság
című tárgyának
1. házi feladatához

Ács-Kurucz Gábor és Dr. Bencsáth Boldizsár
2016-os munkája alapján a feladatot átdolgozta:
Dr. Holczer Tamás és Ládi Gergő

Utoljára frissült: 2022. március 3.

1. Kapcsolat

Probléma esetén (például nem elérhető a szerver vagy nem fut a szolgáltatás) értesítendő:

- Holczer Tamás <holczer@crysys.hu>
- Ládi Gergő <gergo.ladi@crysys.hu>

2. Eszközök

A házi feladat végrehajtása során a 152.66.249.144 IP címen futó szerverrel kell kommunikációt végrehajtani.

A hallgató tetszőleges programnyelvben végezheti el a feladatot. Javasoljuk a Python nyelvet, amelyben igen könnyen implementálható a feladat.

A hallgató a kliens oldali programot tetszőleges helyen futtathatja.

3. Feladatok

A házi feladat célja a hálózati kommunikációval kapcsolatos programozási képességek javítása, a hálózati biztonság egyes feladatainak mélyebb megértését segítő, gyakorlati feladatok elvégzése, hogy a hallgató ne csak elméletben, de saját munkáján keresztül is lássa egyes megoldások működését.

A feladat a szerveren tárolt "flag" kinyerése, ehhez szoftvereszköz fejlesztése, az eredmények rövid dokumentálása, az eredmények és a felhasznált szoftvereszköz benyújtása Moodle rendszeren keresztül.

4. A rendszer felépítése és részfeladatok

Az eszközök fejezetben megjelölt IP címen egy Linux alapú szerver fut. A szerveren egyes biztonsági funkciók lettek implementálva. Amennyiben a hallgató a rendszerek kliens oldali részeit implementálni tudja, úgy hozzáférést nyer az egyedi kódjához, a "flag"-hez, amit a házi feladat részeként be kell nyújtania.

A feladatok részletesebben kifejtve:

- A szerveren ún. port knocking megoldás működik. A hallgató csak akkor fér hozzá a rendszer többi részéhez, ha a folyamat kezdeteként kéréseket küld a 1337-es, 2674 és 4011-es TCP portokra rövid (de nem túl rövid, kb. 1 sec) időn belül, a fenti sorrendben.

- Amennyiben a port knocking sikeres, a felhasználó már el tudja érni a 8888-as porton futó szervert rövid ideig, ahol a feladat több részét tudja elérni.
- A szerver TCP szolgáltatásként fut. Új csatlakozás esetén a szerver a következő üzenetet adja ki:

  ```
  Give me your neptun code:
  ```

- A hallgató feladata ekkor a kliens programjával a NEPTUN kódjának elküldése a szerver felé a nyitott TCP porton keresztül.
- A következő lépésben a szerver egyszerű összeadások/kivonások eredményeit fogja kérni a következő formában:
  - A szerver üzenete:

    ```
    I will send you 6 equations!
    ```

    Ahol 6 helyett tetszőleges szám szerepelhet, majd a számmal megegyező mennyiségű egyszerű összeadás/kivonási feladat.
  - Például

    ```
    01. 56661 + 48079 + 32445 - 64972 + 66169 =
    ```

    A számok mennyisége is változhat.
    - A válaszban *138382*-t kell küldeni.
    - A szerverben ne bízzon meg teljesen, ne használjon *eval*t!
- A következő lépésben el kell küldeni a szerver részére a neptun kód és az utolsó feladvány eredményének az összefűzéséből keletkezett string SHA1 hashét. Pl.:

  ```
  Now give me the (lowercase) sha1 hash of your neptun
  concatenated with the last result!
  sha1('NEPTUN-10001'):
  ```

  - Válasz: 110e431da3aa52bbce0183be50961e024f061a32
- A következő lépésben a szerver egy kihívást küld a kliens felé ilyen formában:

```
Now extend 'NEPTUN-10001' so that sha1('NEPTUN-10001'+x)
begins with '0000'!
The data should contain only printable characters.
Data:
```

A fenti módon a hallgató programjának elő kell állítani egy olyan ASCII stringet, amelyik 4 db. 0 hexadecimális (0-9, A-F) karakterrel kezdődő SHA1 lenyomatot állít elő abban az esetben, ha a lenyomat forrásszövege a neki megküldött egyedi, Neptun kód alapján képzett egyszer használatos szöveg kiegészítve a hallgató programja által generált stringgel.

A rendszer ellenőrzi a hash lenyomatot és a további lépések csak helyes hash küldése esetén érhetőek el.

A fenti példára egy jó megoldás például: *NEPTUN-1000188446*
(SHA1 lenyomata: 0000fe86dbfccb4bb4e3709980aa3d298e2e8e02)

- Amennyiben a hash lenyomat helyes volt, úgy a szerver generál egy 30 másodpercig érvényes X.509 kliens tanúsítványt, valamint további instrukciókat küld vissza a megnyitott porton át:

```
Correct. Now we generate a client cert for you... Generated
You can download the cert from: http://152.66.249.144
  - Login with your neptun and this password: 'crysys'
  - After login GET: /getcert.php
  - After login GET: /getkey.php
  - Use these files to GET: https://152.66.249.144
```

A fenti instrukciók szerint a hallgatónak ezen a ponton egy web kliens segítségével le kell töltenie a részére generált X.509 tanúsítványt és a hozzá tartozó nyilvános-titkos kulcspárt.

Amennyiben a hashellenőrzés nélkül futtatják a getcert-et, akkor nem fog tudni érvényes tanúsítványt visszaadni!

A bejelentkezés neptun kóddal, valamint a statikus 'crysys' jelszóval történik, egy POST kérésben. Ennek a folyamata böngészőből megtekinthető. Bejelentkezés után a SESSION cookie használata szükséges a megjelölt fájlok letöltéséhez.

- A hallgató programjának az X.509 tanúsítvány lejárata előtt (30 másodperc a létrehozáshoz képest!) be kell lépnie a szerverre kliens oldali tanúsítással. Az instrukcióknak megfelelően a kérést a https://152.66.249.144/ felé kell küldeni.

- Amennyiben a szerver további nem dokumentált feltételeket is támaszt, akkor azoknak is meg kell felelni a flag kinyeréséhez.

- Helyes lekérdezés esetén a szerver visszaadja a felhasználó programjának az egyedi *flag*et, amit be kell nyújtania a házi feladat megoldása részeként. Figyelem! A benyújtás csak a programkód és a feladat megoldásáról szóló rövid jegyzőkönyv benyújtásával érvényes!