

## 2P-2 Házi feladat

### Konfigurációkezelés

**FIGYELEM:** A házi feladat megoldása előtt olvassa el a tárgy weblapján lévő HF tudnivalókat! A házi feladat leadása előtt nézze végig a HF tudnivalóknál szereplő ellenőrző listát!

### **Feladat**

Az általunk fejlesztett belső ügyviteli rendszerhez sorozatosan érkeznek vissza a hibajegyek, hogy a felhasználók rejtélyes hibákat tapasztalnak, de általában csak akkor, ha már viszonylag régen be vannak jelentkezve a számítógépükre. Hogy segítsük a hiba okának azonosítását, és az esetleges problémás felhasználók behatárolását, a főnökünk azt kérte tőlünk, hogy készítsünk egy olyan szkriptet, ami összegyűjti, hogy melyik felhasználók azok, akik épp a keresett programokat futtatják.

Készítsünk tehát egy olyan PowerShell szkriptet, ami lekérdezi **WS-Management** segítségével a megadott számítógépeket, és összegyűjti a kért adatokat. Figyeljünk arra, hogy egy távoli számítógépen több felhasználó is lehet egyszerre bejelentkezve (az ügyfeleinknél sok helyen használnak terminál szervereket ugyanis).

Feltételezhetjük, hogy a távoli számítógépek mind Windowst futtatnak. A kért adatokat kötelező a WMI adatbázisából összegyűjteni, és kötelező WS-Management protokollt használni a kommunikációhoz (a DCOM portok nincsenek nyitva a távoli gépeken).

### **A szkript elnevezése és paraméterezése**

```
Get-RemoteAppLogins -MachineData <string> -ProcessName <string[]>
```

A szkriptnek kötelező ezt az elnevezést és paraméterezést használnia. A szkript használjon nevesített paramétereket.

A szkript a paramétereket a következő formában fogadja:

- **MachineData:** a távoli számítógépek adatait tartalmazó CSV fájl elérési útja, kötelező megadni. Abszolút és relatív fájlnevvvel is megadható.
- **ProcessName:** a keresendő folyamatok nevét tartalmazó tömb, kötelező megadni.

Példa a szkript egy lehetséges használatára:

```
Get-RemoteAppLogins -ProcessName "a1.exe", "a2.exe" -MachineData m.csv
```

### **Bemeneti fájl**

A bemeneti fájl egy egyszerű CSV fájl:

```
machineName,port,protocol,user,password  
192.168.250.128,5985,http,administrator,password  
testmachine,5986,https,meres,password2
```

(A jelszó nyílt szövegben tárolása éles környezetben nem javasolt megoldás, ez most csak a házi feladat egyszerűsége miatt engedhető meg. Éles környezetben a jelszót érdemes ilyenkor például titkosítva tárolni vagy nyilvános kulcsú titkosításra alapuló módszereket alkalmazni.)

### Kimenet

A szkript az eredményt a stdout kimenetre írja a következő formában:

```
--192.168.250.128: OK
a1.exe
  joe; S-1-5-21-15536888-2748905542-3078846009-1001; 2012-05-12 18:30:23
  john; S-1-5-21-15536888-2748905542-3078846009-1005; 2012-05-14 19:54:32
a2.exe
--testmachine: Error
```

Tehát kiírja az egyes feldolgozandó gépeket, valamint azt, hogy sikerült-e csatlakozni hozzájuk. Mindegyik géphez kilistázza a kért programokat, majd alá azokat a felhasználókat, akik futtatják épp most az adott programot. A felhasználókhöz megadja felhasználói nevüket, a SID-üket és azt, hogy mikor léptek be az adott gépre. Ha egy keresett programot épp nem futtat senki, akkor is ki kell írni a nevét.

### További elvárások

- A szkript végezzen alapvető ellenőrzéseket a bemeneti paramétereken (pl. bemeneti fájl létezik-e, a kimeneti fájl nem létezik stb.).
- A szkript kezelje azt az esetet, ha a távoli géphez nem sikerül csatlakozni. Ilyenkor is kerüljön bele a gép a kimenetbe, a szkript írjon ki hibaüzenetet, és folytassa a feldolgozást.

### Megjegyzések

- Kliens Windowson a *Switch user* funkció segítségével tudunk egyszerűen több felhasználóval párhuzamosan bejelentkezni.

### Plusz feladat

- Alapesetben nem titkosított kommunikáció zajlik, módosítsuk a szkriptet és a tesztkörnyezetet úgy, hogy a kommunikáció SSL csatorna felett működjön.