

Budapesti Műszaki és Gazdaságtudományi Egyetem
Intelligens rendszerfelügyelet (VIMIA370)

Konfigurációkezelés (2B)

Házi feladat

Horányi Gergő (IMJ7FZ)

2011. április 9.

1 Bevezető

A házi feladat során egy olyan Powershell szkript készült, amely a hálózaton keresztül (WS-Management protokoll segítségével) lekérdezi Windows gépek merevlemez adatait.

2 A megírandó szkript

A megírandó Get-RemoteStorage.ps1 szkript tehát egy Powershell szkript. Egy nevesített paramétert vár: machineData <fájlnév>. Az szkript számára így átadott fájl egy egyszerű vesszővel tagolt (CSV) fájl, amely leírja a lekérdezendő gépek kapcsolódási adatait. Mintaként szolgáljon a következő fájl:

```
machineName,user,password,useSSL
192.168.255.128,administrator,password,false
testmachine,meres,password2,false
```

A szkript a futtatás során az eredményeket a képernyőre írja ki. Minden gép esetén kilistázza a benne található lemezeket, a rajta található partíciókat, valamint a partíciókon található köteteket. Mindegyik bejegyzés mellett feltüntet egy egyedi azonosítót, valamint néhány hasznos információt. A gépeket egy üres sor választja el egymástól. Ha egy géphez nem sikerül csatlakozni, akkor ezt szintén jelzi a felhasználó felé. Egy minta kimenet:

```
-- localhost -
Disk: \\.\PHYSICALDRIVE0, model: VMware, VMware Virtual S SCSI Disk Device, size:
40954.7680664063 MB
  Partition: Disk #0, Partition #0, boot: true, size: 40958 MB
    Volume: C:, free: 31797.99609375 MB, filesystem: NTFS

-- 192.168.1.82 -
Disk: \\.\PHYSICALDRIVE0, model: VMware, VMware Virtual S SCSI Disk Device, size:
40954.7680664063 MB
  Partition: Disk #0, Partition #0, boot: true, size: 40958 MB
    Volume: C:, free: 31508.07421875 MB, filesystem: NTFS

-- Could not connect to randomgep --
```

3 Megvalósítás

A következőekben részletesen leírom, hogy milyen tervezési döntéseket hoztam, valamint részletesen leírom, a szkript működését.

3.1 Paraméterellenőrzés, bemenet kezelése

A szkript elsőként a *param* utasítással ellenőrzi, hogy a megfelelő paramétert kapta-e. Mint korábban említettem, egyetlen, nevesített (*machineData*) paramétert vár. Hiba esetén tájékoztatja a felhasználót a program helyes paraméterezéséről.

```
param (
    [string] $machineData = $(throw "Usage: Get-RemoteStorage.ps1 -machineData
<filename>")
);
```

Ezután a szkript megpróbálja betölteni az átadott CSV fájlt. Sikertelen fájlbetöltés esetén egy hibaüzenettel tájékoztatja a felhasználót a problémáról. A betöltést valamint ellenőrzést az alábbi kódrészlet végzi:

```
try
    $machines = Import-Csv $machineData;
} catch {
    Write-Output "Could not load machineData file";
exit
}
```

Amennyiben az *Import-Csv* utasítás hibát észlel, egy kivételt dob. Ezt kapja el a *catch* ág és jelzi a felhasználó felé.

Ezt követően a szkript bejárja a teljes *\$machines* tömböt egy ciklussal. A ciklus éppen aktuális eleme a *\$machine* változóba kerül majd. Egy ilyen változó magában foglalja a kapcsolódáshoz szükséges összes adatot:

- *machineName*: a távoli gép elérési útja
- *user*: a csatlakozáshoz használandó felhasználói név
- *password*: a csatlakozáshoz szükséges jelszó
- *useSSL*: a kapcsolat titkosítva történjen-e

A felhasználói azonosításhoz egy *PSCredential* típusú objektumra van szükség. Később a lekérdezéseknél ezt paraméterként lehet átadni. Egy ilyen objektum konstruktora paraméterül a felhasználónevet valamint a jelszót (*SecureString* formában) várja. A jelszót elsőként tehát át kell konvertálni a *SecureString*-be, majd a felhasználónévvel együtt létrehozható az azonosításért felelős *PSCredential* objektum:

```
$password = ConvertTo-SecureString $machine.password -AsPlainText -Force
$credential = new-object -typename System.Management.Automation.PSCredential -
argumentlist $machine.user,$password;
```

A konvertálásnál az eredeti szöveg formátuma *Plain* szöveg (*PlainText*). A *Force* paraméter szükséges, mert ezzel erősítem meg, hogy ténylegesen *plain* szöveg kódolású jelszót akarok feldolgozni. Ez nyilván valós környezetben nem lenne alkalmazható.

3.2 A kapcsolat tesztelése

A program ezután ellenőrzi, hogy tud-e kapcsolódni az adott kiszolgálóhoz. Erre a *Test-WSMan* utasítás használható. Amennyiben a kapcsolódás sikeres, az utasítás kiírja a képernyőre a kiszolgáló néhány adatát. Hiba esetén kivételt dob.

Minden egyes *WSMan* lekérés esetén elsőként megvizsgálom, hogy az adott kapcsolat igényel-e *SSL* titkosítást. Ha igen, akkor a lekérés mögé illesztem a *UseSSL* kapcsolót. Erre a továbbiakban nem térek ki külön, de ezt az ellenőrzést minden lekérés előtt megteszem.

A tesztelés az alábbi utasítással történik:

```
Test-WSMan -ComputerName $machineName -Authentication Basic -Credential
$credential
```

A *Test-WSMan* paramétereinek jelentése a következő:

- *ComputerName*: A csatlakozandó gép elérési címe.
- *Authentication*: *Basic* - *HTTP Basic* típusú azonosítás használata a lekéréshez. Ez nem túlzottan biztonságos, cserébe nagyon egyszerű beállítani, alkalmazni. A házi feladathoz megfelelő, de a valós életben valami komolyabb azonosítási algoritmust alkalmaznánk.

- Credential: Az azonosításhoz szükséges PSCredential objektum, amelyet korábban már bemutattam.

A program kimenetét egy csővezetékkel és az Out-Null utasítással elrejttem. Hiba esetén a catch ág elkapja a keletkezett kivételt, és kiírja, hogy nem tudott csatlakozni a kiszolgálóhoz. Ezután a következő gépre ugrik a végrehajtás. Egyéb esetben kiírja a gép nevét.

3.3 Gépszintű lekérések

A lekéréseket, mint korábban már többször leírtam a WS-Management protokoll segítségével valósítottam meg. A Powershell környezet egy Get-WSManInstance cmdlet-et nyújt a programozó felé, amely segítségével ilyen lekérések készíthetők.

Elsőként három lekérést hajt végre a program. Ebből az első, amely lekéri a gépben található fizikai lemezek azonosítóját, méretét és pontos modelljét. Ez, és a későbbi lekérések is a wmicimv2 névteret használják. Ez tulajdonképp egy alias a következő névtérre:

```
http://schemas.microsoft.com/wbem/wsmn/1/wmi/root/cimv2
```

A névteret azért alkalmazhatom, mert a feladatkiírás szerint feltételezhetjük, hogy a távoli gépen Windows van. A lemezek a Win32_DiskDrive osztályból származnak. Az osztály sok olyan adatot is tartalmaz, amelyre nincs szüksége a szkriptnek a működéshez. A szűrést egy WQL kifejezéssel hajtom végre. A lemezeket lekérő utasítás, valamint a paraméterek magyarázata:

```
$disks = Get-WSManInstance wmicimv2/* -ComputerName $machineName -Enumerate -
Filter "Select DeviceID, Size, Model FROM Win32_DiskDrive" -Authentication Basic -
Credential $credential -UseSSL
```

- wmicimv2/*: A lekérést ebben a névtérben szeretném elvégezni
- ComputerName: a távoli gép címe
- Enumerate: Több eredmény esetén egy listában kérem vissza az eredményt
- Filter: A paraméter tartalmazza a korábban már említett WQL lekérést, amely azt eredményezi, hogy a lehető legkevesebb adatot kelljen átmozgatni a szerverről. A lekérés az SQL szintaxisa alapján teljesen egyértelmű.
- Authentication: Basic típusú azonosítás (lásd korábban)
- Credential: Azonosításhoz szükséges adatok (lásd korábban)
- UseSSL: opcionálisan, titkosított adatkapcsolat (lásd korábban)

Az eredményeket egy tömbben eltárolom későbbi feldolgozásra.

Ezután két lekérdezést végzek még, amelyet minden gép esetén egyszer elegendő elvégezni. A két lekérés a DiskDriveToDiskPartition és a LogicalDiskToPartition objektumokat érinti. Ezek az objektumok kötik össze a lemezeket a partíciókkal, valamint a partíciókat a kötetekkel. A feldolgozásuk később történik majd csak meg. Az objektumok összes attribútumát le kell kérni, hiszen ezek mind szükségesek. Lehetséges lenne mindig csak az aktuális lemezhez tartozó táblarészletet lekérni, de végeredményben az összes bejegyzést le kellett volna kérni. Így a lekérdezések száma jóval kisebb lehet, így ezt találtam optimálisabb megoldásnak.

3.4 Feldolgozás

Ezután a program feldolgozza a már lekért lemezeket egy ciklusban. A lemezeke sorra véve, először kiírja a lekért adatokat:

```
$diskid = $disk.DeviceID;
$model = $disk.Model;
$size = $disk.Size/1MB;
Write-Output "Disk: $diskid, model: $model, size: $size MB";
```

Majd ezt követően a lemez azonosítója alapján kikeresi a hozzá tartozó partíciókat. Ehhez a korábban lekért `DiskDriveToDiskPartition` osztály példányai nyújtanak segítséget. A szkript végignézi az összes példány. Mindegyiknek két attribútuma van:

- `Antecedent`: A lemezre mutató hivatkozás
- `Dependent`: A partícióra mutató hivatkozás

Mindkét bejegyzés XML csomópont, amelyeket még fel kell dolgozni, ezt az alábbi utasítások végzik:

```
$antecedent =
$mapline.GetElementsByTagName("p:Antecedent").Item(0).GetElementsByTagName("w:Selector") | select -Property @{n="text"; e={$_. "#text"}};
$dependent =
$mapline.GetElementsByTagName("p:Dependent").Item(0).GetElementsByTagName("w:Selector") | select -Property @{n="text"; e={$_. "#text"}};
```

Jól látható, hogy a két bejegyzés feldolgozásában nincs számottevő különbség. Elsőként az XML csomópontból le kell kérni a megfelelő elemet. A lekérés egy listát ad vissza, azonban megállapítható, hogy a listának minden esetben egyetlen eleme lesz (a szabványnak megfelelően). Ebből tehát az első elemet lekérve szintén egy XML csomópontot kapunk. Ebből a `Selector` azonosítójú elemet lekérve már elérhető egy olyan objektum, amelynek a `#text` attribútumában a kívánt (lemez vagy partíció) azonosítója található. A későbbi könnyebb kezelés érdekében a `#` jelet eltávolítom az attribútum nevéből (ezt végzi a csővezeték után található `select` kifejezés).

Ha a feldolgozott lemez azonosítója megegyezik az éppen aktuális lemezhez tartozóval, akkor a szkript talált egy összetartozó lemez - partíció párt.

Ilyen esetekben a szkript lekéri a megtalált partícióhoz tartozó adatokat. A lekérést az alábbi utasítás végzi:

```
Get-WSManInstance wmicimv2/* -ComputerName $machineName -Enumerate -Filter "SELECT BootPartition, Size FROM Win32_DiskPartition WHERE DeviceID='$partitionid'" -
Authentication Basic -Credential $credential -UseSSL
```

A lényeges eltérés a korábbi lekérésektől a `Filter` WQL kifejezésben található. A lekérendő objektumok a `Win32_DiskPartition` típusúak. A lekérésben csak a `Boot` partíciót és a méretet leíró tulajdonságokra van szükség. A lekérés csak azokat az eredményeket adja vissza, amelyeknél az azonosító megegyezik a korábban megtalált partíció azonosítójával (`WHERE` klóz).

A lekérés eredményét ezután a képernyőre kiírja a program:

```
$boot = $partition.BootPartition;
$size = $partition.Size/1MB;
Write-Output " Partition: $partitionid, boot: $boot, size: $size MB";
```

A program ezt követő futása nagyon hasonlít a lemez-partíció összerendeléshez, csak itt a partícióhoz keres kötetet. Az összerendeléshez a korábban már lekért `LogicalDiskToPartition` bejegyzések adnak segítséget. Ezekben a bejegyzésekben ugyanúgy XML csomópont bejegyzések vannak, amelyből kinyerhető, hogy melyik kötet tartozik egy partícióhoz.

A kötet megfelelő adatait (Üres hely és fájlrendszer) ezután az alábbi lekérés gyűjti be:

```
$volume = Get-WSManInstance wmicimv2/* -ComputerName $machineName -Enumerate -
Filter "SELECT FreeSpace, FileSystem FROM Win32_LogicalDisk WHERE
DeviceID='$volumeid'" -Authentication Basic -Credential $credential -UseSSL
```

Jól látható, hogy a lekérés nagyon hasonlít a korábbiakhoz, a lényeges eltérést, hogy Win32_LogicalDisk példányokat kér le.

A kiírásért az alábbi utasítássorozat felel ezután:

```
$free = $volume.FreeSpace/1MB;  
$fs = $volume.FileSystem;  
Write-Output " Volume: $volumeid, free: $free MB, filesystem: $fs";
```

A ciklusok lezárultával egy üres sort illeszt még a program a kimenet végére, hogy elválassza a következő gép adataival.

4 Tesztelés

4.1 A tesztkörnyezet kialakítása

A szkript teszteléséhez egy megfelelő tesztkörnyezetet kellett létrehoznom. A tesztelést egy három gépből álló rendszerben végeztem. Mindegyik gép virtualizált, azonban a virtualizációs keretrendszert is két külön gépen futtatom.

A virtuális gépekre Microsoft Windows 7 Professional operációs rendszer 64 bites változata került. A gazdagépeken Apple Mac OS X 10.6.6 és a VMWare Fusion rendszerek futnak, de ezeknek a tesztelésben nincs különösebb szerepe. Mindegyik virtuális gép "Bridged" módban csatlakozik a hálózatra, így a hálózaton teljesen önálló gépként jelenhetnek meg.

A gépek az alábbi lemezeszközökkel rendelkeznek:

- 1. számú gép, "localhost"
 - 1 lemez, amelyen 1 partíció, amelyen 1 kötet (40 GB, NTFS)
- 2. számú gép, 192.168.1.82
 - 1 lemez, amelyen 1 partíció, amelyen 1 kötet (40 GB, NTFS)
 - 2 darab 200 MB-os RAID1-be kötött lemez
 - 1 darab 400 MB-os lemez
 - két kötet
 - 1 darab 150 MB-os lemez
 - két kötet, amelyből az egyiknek nincs betűjele
- 3. számú gép, 192.168.1.87
 - 1 lemez, amelyen 1 partíció, amelyen 1 kötet (40 GB, NTFS)
 - 1 darab 4 GB-os USB PenDrive

Kötet	Elrendezés	Típus	Fájlrendszer	Állapot
Új ...	Egyszerű	Alaplemez	NTFS	Kifogástalan (Elsődleges partíció)
Új ...	Tükrözött	Dinamikus	NTFS	Kifogástalan
Új ...	Egyszerű	Alaplemez	NTFS	Kifogástalan (Elsődleges partíció)
Új ...	Egyszerű	Alaplemez	NTFS	Kifogástalan (Elsődleges partíció)
Új ...	Egyszerű	Alaplemez	NTFS	Kifogástalan (Elsődleges partíció)

Lemez 0 Alaplemez 40,00 GB Online	(C:) 40,00 GB NTFS Kifogástalan (Rendszer, Rendszerindítás, Lapozófáj, Aktív, Összeomlási memóriakép, Elsődleges partíció)		
Lemez 1 Dinamikus 205 MB Online	Új kötet (E:) 202 MB NTFS Kifogástalan		
Lemez 2 Dinamikus 205 MB Online	Új kötet (E:) 202 MB NTFS Kifogástalan		
Lemez 3 Alaplemez 409 MB Online	<table border="1"> <tbody> <tr> <td>Új kötet (F:) 150 MB NTFS Kifogástalan (Elsődleges partíció)</td> <td>Új kötet (G:) 257 MB NTFS Kifogástalan (Elsődleges partíció)</td> </tr> </tbody> </table>	Új kötet (F:) 150 MB NTFS Kifogástalan (Elsődleges partíció)	Új kötet (G:) 257 MB NTFS Kifogástalan (Elsődleges partíció)
Új kötet (F:) 150 MB NTFS Kifogástalan (Elsődleges partíció)	Új kötet (G:) 257 MB NTFS Kifogástalan (Elsődleges partíció)		
Lemez 4 Alaplemez 153 MB Online	<table border="1"> <tbody> <tr> <td>Új kötet (H:) 60 MB NTFS Kifogástalan (Elsődleges partíció)</td> <td>Új kötet 91 MB NTFS Kifogástalan (Elsődleges partíció)</td> </tr> </tbody> </table>	Új kötet (H:) 60 MB NTFS Kifogástalan (Elsődleges partíció)	Új kötet 91 MB NTFS Kifogástalan (Elsődleges partíció)
Új kötet (H:) 60 MB NTFS Kifogástalan (Elsődleges partíció)	Új kötet 91 MB NTFS Kifogástalan (Elsődleges partíció)		

4.1.1 Konfigurálás

Bizonyos beállításokat el kellett végezni ahhoz, hogy a rendszer működőképes legyen. Ezeket minden gépen egységesen elvégeztem. Elsőként a tűzfalbeállításokat állítottam "Munkahely" típusú hálózatra.

A további beállításokat egy rendszergazda jogosultságú Powershell konzollal hajtottam végre. Elindítottam a WSMAN szolgáltatást:

```
Set-WSManQuickConfig
```

A WS-Management szolgáltatás beállításai a WSMAN "köteten" találhatóak

```
cd WSMAN:
cd localhost
cd Service
```

Itt beállítottam, hogy engedélyezzen titkosítatlan kapcsolódást, valamint alkalmazza a Basic hitelesítést.

```
Set-Item AllowUnencrypted -Value true
cd Auth
Set-Item Basic -Value true
cd..
cd..
```

A localhost-on, azaz a lekérdezést futtató gépen ezen felül még az alábbi beállításokat is megtettem:

```
cd Client
Set-Item AllowUnencrypted -Value true
Set-Item TrustedHosts -Value "192.168.1.82,localhost,192.168.1.87"
```

Ezek a beállítások azt eredményezik, hogy a kliens hajlandó lesz az adott gépekkel titkosítás nélkül kapcsolódni.

A tesztrendszer ezzel összeállt, működőképes.

4.2 Tesztelés

A teszteléshez igyekeztem minél több különböző esetet kipróbálni. Ezeknek a kimeneteit itt dokumentálom és ahol szükséges magyarázom. A tesztelést minden esetben a "localhost"-két bemutatott gépről végzem egy Powershell konzolból (amely nem rendelkezik adminisztrátori jogosultsággal)

4.2.1 Hibás paraméterezés

```
PS C:\Users\Gergő Horányi\Desktop> .\Get-RemoteStorage.ps1 -file .\machineData.txt
Usage: Get-RemoteStorage.ps1 -machineData <filename>
At C:\Users\Gergő Horányi\Desktop\Get-RemoteStorage.ps1:9 char:33
+ [string] $machineData = $(throw <<<< "Usage: Get-RemoteStorage.ps1 -
machineData <filename>")
+ CategoryInfo          : OperationStopped: (Usage: Get-Remo...Data
<filename>:String) [], RuntimeException
+ FullyQualifiedErrorId : Usage: Get-RemoteStorage.ps1 -machineData <filename>
```

Látható, hogy hibás paraméterezés esetén az elvárt hibaüzenet megjelenik.

4.2.2 Nem létező fájl esetén

```
PS C:\Users\Gergő Horányi\Desktop> .\Get-RemoteStorage.ps1 -machineData
nemletezo.txt
Could not load machineData file
```

4.2.3 Nem elérhető host esetén

Az átadott fájl tartalma:

```
machineName,user,password,useSSL
randomgep,user,pass,jelszo
```

A kimenet pedig:

```
PS C:\Users\Gergő Horányi\Desktop> .\Get-RemoteStorage.ps1 -machineData
.\machineData.txt
-- Could not connect to randomgep --
```

4.2.4 SSL tesztelése

Sajnos nem tudtam a gépemre SSL Certificate-t telepíteni, így ennek hiányában csak azt tudom ellenőrizni, hogy a program valóban figyelembe veszi az átadott fájlban az SSL oszlopot, és úgy próbálja elérni.

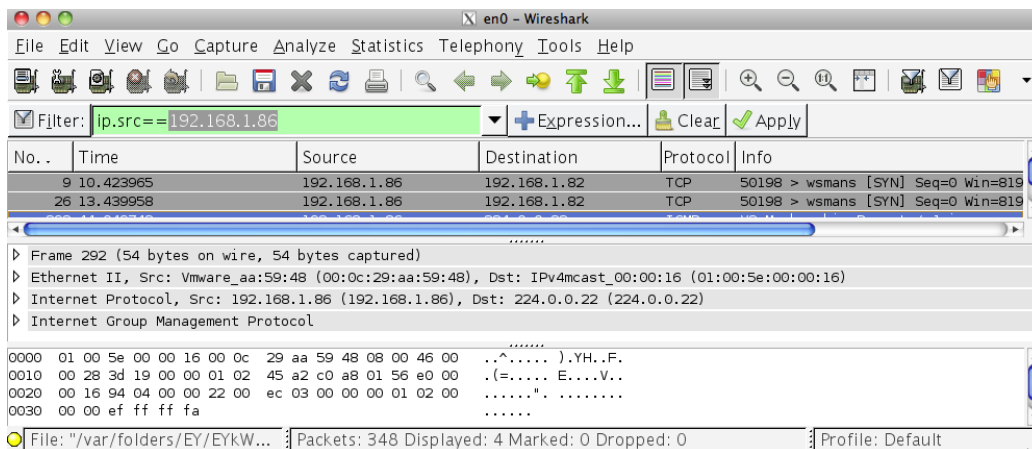
A bemeneti fájl tartalma:

```
machineName,user,password,useSSL
192.168.1.82,somi,jelszo,true
```

A kimenetre érezhetően többet kell várni, mint eddig (nyilvánvalóan valami timeout-ra vár a rendszer). A kimenet:

```
PS C:\Users\Gergő Horányi\Desktop> .\Get-RemoteStorage.ps1 -machineData
.\machineData.txt
-- Could not connect to 192.168.1.82 --
```


A hálózati forgalmat Wireshark-kal is megfigyeltem, és ez láthatóvá tette, hogy a program a wsmans nevű (5198-as számú) porton próbálkozik, tehát ténylegesen az SSL-t használná, ha tudná.



4.2.5 Teljes tesztkonfiguráció tesztelése

A bemeneti fájl tartalma:

```
machineName,user,password,useSSL
randomgep,user,pass,jelszo
localhost,pumafi,jelszo,false
192.168.1.82,somi,jelszo,false
192.168.1.87,irf,jelszo,false
```

Az így kapott kimenet a következő:

```
PS C:\Users\Gergő Horányi\Desktop> .\Get-RemoteStorage.ps1 -machineData
.\machineData.txt
-- Could not connect to randomgep --

-- localhost -
Disk: \\.\PHYSICALDRIVE0, model: VMware, VMware Virtual S SCSI Disk Device, size:
40954.7680664063 MB
  Partition: Disk #0, Partition #0, boot: true, size: 40958 MB
  Volume: C:, free: 31797.421875 MB, filesystem: NTFS

-- 192.168.1.82 -
Disk: \\.\PHYSICALDRIVE0, model: VMware, VMware Virtual S SCSI Disk Device, size:
40954.7680664063 MB
  Partition: Disk #0, Partition #0, boot: true, size: 40958 MB
  Volume: C:, free: 31512.28515625 MB, filesystem: NTFS
Disk: \\.\PHYSICALDRIVE1, model: VMware, VMware Virtual S SCSI Disk Device, size:
203.9501953125 MB
  Partition: Disk #1, Partition #0, boot: false, size: 203.76904296875 MB
  Volume: E:, free: 182.234375 MB, filesystem: NTFS
Disk: \\.\PHYSICALDRIVE2, model: VMware, VMware Virtual S SCSI Disk Device, size:
203.9501953125 MB
  Partition: Disk #2, Partition #0, boot: false, size: 203.76904296875 MB
  Volume: E:, free: 182.234375 MB, filesystem: NTFS
Disk: \\.\PHYSICALDRIVE3, model: VMware, VMware Virtual S SCSI Disk Device, size:
407.900390625 MB
  Partition: Disk #3, Partition #0, boot: false, size: 150 MB
```

```
Volume: F:, free: 134.265625 MB, filesystem: NTFS
Partition: Disk #3, Partition #1, boot: false, size: 257 MB
Volume: G:, free: 234.68359375 MB, filesystem: NTFS
Disk: \\.\PHYSICALDRIVE4, model: VMware, VMware Virtual S SCSI Disk Device, size:
149.04052734375 MB
Partition: Disk #4, Partition #0, boot: false, size: 60 MB
Volume: H:, free: 48.26953125 MB, filesystem: NTFS
Partition: Disk #4, Partition #1, boot: false, size: 91 MB

-- 192.168.1.87 -
Disk: \\.\PHYSICALDRIVE0, model: VMware, VMware Virtual S SCSI Disk Device, size:
40954.7680664063 MB
Partition: Disk #0, Partition #0, boot: true, size: 40958 MB
Volume: C:, free: 32204.6875 MB, filesystem: NTFS
Disk: \\.\PHYSICALDRIVE1, model: MEM Drive Mini Metal USB Device, size:
3851.52099609375 MB
Partition: Disk #1, Partition #0, boot: false, size: 3855.95068359375 MB
Volume: E:, free: 3800.5625 MB, filesystem: FAT32
```

4.2.6 Eredmények értékelése

Jól látható, hogy a program elvégezte feladatát, hiszen a szükséges adatokat begyűjtötte és kiírta az elvárt formátumban. A tesztkörnyezet tartalmaz néhány olyan konfigurációt, amelyre azonban a szkript nincs még felkészítve. Ilyen például a RAID rendszer kezelése. Az "E:" meghajtó tükrözve van egy másik lemezre is. Ilyen esetekben a kötet-partíció leképzésben a kötet két partícióra is le van képezve, így ennek megfelelően két partíció alatt is megjelenik. Ez nyilvánvalóan nem a legjobb megoldás, hiszen nem jelzi a felhasználónak, hogy ez RAID, de a valóságnak tulajdonképpen megfelel, hiszen mindkét partíció tartalma a logikai "E" kötet.

Egy másik probléma, hogy ha egy meghajtónak nincsen betűjele, akkor az nem jelenik meg itt. A Disk #4, Partition #1 partíción látható, hogy nem található kötet, annak ellenére, hogy azon a területen is van egy NTFS fájlrendszerű kötet. A hiba javításához le kellene kérni az összes kötetet, majd ezt kellene illeszteni a partíciókhoz.